---
title: "Account Management"
description: "Learn how to manage your Vercel account and team members."
last_updated: "2026-01-16T02:19:24.277Z"
source: "https://vercel.com/docs/accounts"

Account Management

When you first sign up for Vercel, you'll create an account. This account is used to manage your Vercel resources. Vercel has three types

- [Hobby](/docs/plans/hobby)
- [Pro](/docs/plans/pro-plan)
- [Enterprise](/docs/plans/enterprise)

Each plan offers different features and resources, allowing you to choose the right plan for your needs.

When signing up for Vercel, you can choose to sign up with an email address or a Git provider.

Sign up with email

To sign up with email:

1. Enter your email address to receive the six-digit one-time password (OTP)
2. Enter the OTP to proceed with logging in successfully.

When signing up with your email, no Git provider will be connected by default. See [login methods and connections](#login-methods-and-con

Sign up with a Git provider

You can sign up with any of the following supported Git providers:

- [**GitHub**](/docs/git/vercel-for-github)
- [**GitLab**](/docs/git/vercel-for-gitlab)
- [**Bitbucket**](/docs/git/vercel-for-bitbucket)

Authorize Vercel to access your Git provider account. **This will be the default login connection on your account**.

Once signed up you can manage your login connections in the [authentication section](/account/authentication) of your dashboard.

Login methods and connections

You can manage your login connections in the **Authentication** section of [your account settings](/account/authentication). To find this

1. Select your profile picture near the top-right of the dashboard
2. Select **Settings** in the dropdown that appears
3. Select **Authentication** in the list near the left side of the page

Login with passkeys

Passkeys allow you to log into your Vercel account using biometrics such as face or fingerprint recognition, PINs, hardware security keys

To add a new passkey:

1. From the dashboard, click your account avatar and select **Settings**. In your [account settings](/account/authentication), go to the
2. Under **Add New**, select the **Passkey** button and then click **Continue**
3. Select the authenticator of preference. This list depends on your browser and your eligible devices. By default, Vercel will default t
4. Follow the instructions on the device or with the account you've chosen as an authenticator

When you're done, the passkey will appear in a list of login methods on the **Authentication** page, alongside your other connections.

Logging in with SAML Single Sign-On

SAML Single Sign-On enables you to log into your Vercel team with your organization's identity provider which manages your credentials.

SAML Single Sign-On is available to Enterprise teams, or Pro teams can purchase it as a paid add-on from their [Billing settings](https:/

Choosing a connection when creating a project

When you create an account on Vercel, you will be prompted to create a project by either importing a Git repository or using a template.

Either way, you must connect a Git provider to your account, which you'll be able to use as a login method in the future.

Using an existing login connection

Your Hobby team on Vercel can have only one login connection per third-party service. For example, you can only log into your Hobby team \

For multiple logins from the same service, create a new Vercel Hobby team.

Teams

Teams on Vercel let you collaborate with other members on projects and access additional resources.

Creating a team

\['Dashboard'

1. Click on the scope selector at the top left of the nav bar
2. Choose to create a new team
3. Name your team
4. Depending on the types of team plans that you have already created, you'll be able to select a team plan option:

'cURL'

To create an Authorization Bearer token, see the [access token](/docs/rest-api/reference/welcome#creating-an-access-token) section of the

```bash filename="cURL"
curl --request POST \
  --url https://api.vercel.com/v1/teams \
  --header "Authorization: Bearer $VERCEL_TOKEN" \
```

```
  --header "Content-Type: application/json" \
  --data '{
  "slug": "<team-slug>",
  "name": "<team-name>"
}'
```

'SDK']

To create an Authorization Bearer token, see the [access token](/docs/rest-api/reference/welcome#creating-an-access-token) section of the

```ts filename="createTeam"
import { Vercel } from '@vercel/sdk';

const vercel = new Vercel({
  bearerToken: '<YOUR_BEARER_TOKEN_HERE>',
});

async function run() {
  const result = await vercel.teams.createTeam({
    slug: 'team-slug',
    name: 'team-name',
  });

  // Handle the result
  console.log(result);
}

run();
```

Collaborating with other members on projects is available on the [Pro](/docs/plans/pro-plan) and [Enterprise](/docs/plans/enterprise) pla

Upgrade from the [Hobby](/docs/plans/hobby) plan to [Pro](/docs/plans/hobby#upgrading-to-pro) to add team members.

After [creating a new trial](/docs/plans/pro-plan/trials), you'll have 14 days of Pro premium features and collaboration for free.

Team membership

You can join a Vercel team through an invitation from a [team owner](/docs/rbac/access-roles#owner-role), automatic addition by a team's

Leaving a team

> **💡 Note:** You can't leave a team if you are the last remaining
> [owner](/docs/rbac/access-roles#owner-role) or the last confirmed
> [member](/docs/rbac/access-roles#member-role).

To leave a team:

1. If there isn't another owner for your team, you must assign a different confirmed member as the team owner
2. Go to your team's dashboard and select the **Settings** tab
3. Scroll to the **Leave Team** section and select the **Leave Team** button
4. Click **Confirm**
5. If you are the only remaining member, you should delete the team instead

Deleting a team

To delete a team:

1. Remove all team domains
2. Go to your team's dashboard and select the **Settings** tab
3. Scroll to the **Delete Team** section and select the **Delete Team** button
4. Click **Confirm**

If you'd prefer to cease payment instead of deleting your team, you can [downgrade to Hobby](/docs/plans/pro-plan#downgrading-to-hobby).

Default team

Your default team will be used when you make a request through the [API](/docs/rest-api) or [CLI](/docs/cli) and don’t specify a specific

How to change your default team

If you delete, leave, or are removed from your default team, Vercel will automatically choose a new default team for you. However, you ma

1. Navigate to vercel.com/account/settings
2. Under **Default Team**, select your new default team from the dropdown
3. Press **Save**

Find your team ID

Your Team ID is a unique and unchangeable identifier that's automatically assigned when your team is created.

There are a couple of methods you can use to locate your Team ID:

- **Vercel API**: Use the [Vercel API](/docs/rest-api/reference/endpoints/teams/list-all-teams) to retrieve your Team ID
- **Dashboard**: Find your Team ID directly from your team's Dashboard on Vercel:
  - Navigate to the following URL, replacing `your_team_name_here` with your actual team's name: `https://vercel.com/teams/your_team_name
    If you're unable to locate your Team ID using the URL method, follow these steps:
  - Open your team's dashboard and head over to the **Settings** tab
  - Choose **General** from the left-hand navigation
  - Scroll down to the Team ID section and your Team ID will be there ready for you to copy

Managing emails

To access your email settings from the dashboard:

1. Select your avatar in the top right corner of the [dashboard](/dashboard).
2. Select **Account Settings** from the list.
3. Select the **Settings** tab and scroll down to the **Emails** section.
4. You can then [add](/docs/accounts#adding-a-new-email-address), [remove](/docs/accounts#removing-an-email-address), or [change](/docs/a

Adding a new email address

To add a new email address

1. Follow the steps above and select the **Add Another** button in the **Emails** section of your account settings.
2. Once you have added the new email address, Vercel will send an email with a verification link to the newly added email. Follow the lin
3. Once verified, all email addresses can be used to log in to your account, including your primary email address.

You can add up to three emails per account, with a single email domain shared by two emails at most.

Changing your primary email address

Your primary email address is the email address that will be used to send you notifications, such as when you receive a new [preview comm

Once you have added and verified a new email address, you can change your primary email address by selecting **Set as Primary** in the do

Removing an email address

To remove an email address select the **Delete** button in the dot menu.

If you wish to remove your primary email address, you will need to set a new primary email address first.

---
title: "Using the Activity Log"
description: "Learn how to use the Activity Log, which provides a list of all events on a Hobby team or team, chronologically organized s
last_updated: "2026-01-16T02:19:24.024Z"
source: "https://vercel.com/docs/activity-log"

Using the Activity Log

The [Activity Log](/dashboard/activity) provides a list of all events on a Hobby team or team, chronologically organized since its creati

- User(s) involved with the event
- Type of event performed
- Type of account
- Time of the event (hover over the time to reveal the exact timestamp)

> **💡 Note:** Vercel does not emit any logs to third-party services. The Activity Log is
> only available to the account owner and team members.

When to use the Activity log

Common use cases for viewing the Activity log include:

- If a user was removed or deleted by mistake, use the list to find when the event happened and who requested it
- A domain can be disconnected from your deployment. Use the list to see if a domain related event was recently triggered
- Check if a specific user was removed from a team

Events logged

The table below shows a list of events logged on the Activity page.

---
title: "Vercel Agent Investigation"
description: "Let AI investigate your error alerts to help you debug faster"
last_updated: "2026-01-16T02:19:24.043Z"
source: "https://vercel.com/docs/agent/investigation"

Vercel Agent Investigation

When you get an error alert, Vercel Agent can investigate what's happening in your logs and metrics to help you figure out the root cause

Investigations happen automatically when an error alert fires. The AI digs into patterns in your data, checks what changed, and gives you

Getting started with Agent Investigation

You'll need two things before you can use Agent Investigation:

1. An [Observability Plus](/docs/observability/observability-plus) subscription, which includes **10 investigations per billing cycle**
2. [Sufficient credits](/docs/agent/pricing) to cover the cost of additional investigations

To allow investigations to run **automatically for every error alert**, you should [enable Vercel Agent Investigations](#enable-agent-inv

You can [run an investigation manually](#run-an-investigation-manually) if you want to investigate an alert that has already fired.

> **💡 Note:** Agent Investigation will not automatically start running if you had previously only enabled Vercel Agent for code review.

Enable Agent Investigations

To run investigations automatically for every error alert, enable Vercel Agent Investigations in your team's settings:

1. Go to your team's [Settings](https://vercel.com/d?to=%2Fteams%2F%5Bteam%5D%2Fsettings\&title=Go+to+Settings\&personalTo=%2Faccount) pa
2. In the **General** section, find **Vercel Agent** and under **Investigations**, switch the toggle to **Enabled**.
3. Select **Save** to confirm your changes.

Once enabled, investigations will run automatically when an error alert fires. You'll need to make sure your team has [enough credits](/d

How to use Agent Investigation

When [Agent Investigations are enabled](#enable-agent-investigations), they run automatically when an error alert fires. The AI queries y

To view an investigation:

1. Go to your [Vercel dashboard](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fobservability%2Falerts\&title=Open+Alerts) and navigate to
2. Find the alert you want to review and click on it.

3. The investigation results will appear alongside your alert details. You'll see the analysis stream in real time if the investigation i

If you want to run the investigation again with fresh data, click the **Rerun** button.

Run an investigation manually

If you do not have Agent Investigations enabled and running automatically, you can run an investigation manually from the alert details p

1. Go to your [Vercel dashboard](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fobservability%2Falerts\&title=Open+Alerts) and navigate to
2. Find the alert you want to review and click on it.
3. Click the **Investigate** (or **Rerun**) button to run an investigation manually.

Pricing

Agent Investigation uses a credit-based system. All teams with Observability Plus have **10 investigations included in their subscription

Additional investigations cost a fixed $0.30 USD plus token costs billed at the Agent's underlying AI provider's rate, with no additional

Pro teams can redeem a $100 USD promotional credit when enabling Agent. You can [purchase credits and enable auto-reload](/docs/agent/pri

Disable Agent Investigation

To disable Agent Investigation:

1. Go to the your team's [Settings](https://vercel.com/d?to=%2Fteams%2F%5Bteam%5D%2Fsettings\&title=Go+to+Settings\&personalTo=%2Faccount
2. In the **General** section, find **Vercel Agent** and under **Investigations**, switch the toggle to **Disabled**.
3. Select **Save** to confirm your changes.

Once disabled, Agent Investigation won't run automatically on any new alerts. You can re-enable Agent Investigation at any time from the

---
title: "Vercel Agent"
description: "AI-powered development tools that speed up your workflow and help resolve issues faster"
last_updated: "2026-01-16T02:19:24.074Z"
source: "https://vercel.com/docs/agent"

Vercel Agent

Vercel Agent is a suite of AI-powered development tools built to speed up your workflow. Instead of spending hours debugging production i

Agent works because it already understands your application. Vercel builds your code, deploys your functions, and serves your traffic. Ag

Everything runs on [Vercel's AI Cloud](https://vercel.com/ai), infrastructure designed specifically for AI workloads. This means Agent ca

Features

Code Review

Get automatic code reviews on every pull request. Code Review analyzes your changes, identifies potential issues, and suggests fixes you

What it does:

- Performs multi-step reasoning to identify security vulnerabilities, logic errors, and performance issues
- Generates patches and runs them in secure sandboxes with your real builds, tests, and linters
- Only suggests fixes that pass validation checks, allowing you to apply specific code changes with one click

Learn more in the [Code Review docs](/docs/agent/pr-review).

Investigation

When error alerts fire, Vercel Agent Investigations can analyze what's happening to help you debug faster. Instead of manually digging th

What it does:

- Queries logs and metrics around the time of the alert
- Looks for patterns and correlations that might explain the problem
- Provides insights about potential root causes

Learn more in the [Agent Investigation docs](/docs/agent/investigation).

Getting started

You can enable Vercel Agent in the [Agent tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fvercel-agent\&title=Open+Vercel+Agent) of yo

- **Code Review**: You'll need to configure which repositories to review and whether to review draft PRs. See [Code Review setup](/docs/a
- **Agent Investigation**: This requires [Observability Plus](/docs/observability/observability-plus) and in order to run investigations

Pricing

Vercel Agent uses a credit-based system. Each review or investigation costs a fixed $0.30 USD plus token costs billed at the Agent's unde

You can [purchase credits and enable auto-reload](/docs/agent/pricing#adding-credits) in the Agent tab of your dashboard. For complete pr

Privacy

Vercel Agent doesn't store or train on your data. It only uses LLMs from providers on our [subprocessor list](https://security.vercel.com

---
title: "Vercel Agent Code Review"
description: "Get automatic AI-powered code reviews on your pull requests"
last_updated: "2026-01-16T02:19:23.965Z"
source: "https://vercel.com/docs/agent/pr-review"

Vercel Agent Code Review

AI Code Review is part of [Vercel Agent](/docs/agent), a suite of AI-powered development tools. When you open a pull request, it automati

It generates patches and runs them in [secure sandboxes](/docs/vercel-sandbox) with your real builds, tests, and linters to validate fixe

How to set up Code Review

To enable code reviews for your [repositories](/docs/git#supported-git-providers), navigate to the
[**Agent** tab](/d?to=%2F%5Bteam%5D%2F%7E%2Fvercel-agent\&title=Open+Vercel+Agent) of the dashboard.

1. Click **Enable** to turn on Vercel Agent.
2. Under **Repositories**, choose which repositories to review:
   - All repositories (default)
   - Public only
   - Private only
3. Under **Review Draft PRs**, select whether to:
   - Skip draft PRs (default)
   - Review draft PRs
4. Optionally, configure **Auto-Recharge** to keep your balance topped up automatically:
   - Set the threshold for **When Balance Falls Below**
   - Set the amount for **Recharge To Target Balance**
   - Optionally, add a **Monthly Spending Limit**
5. Click **Save** to confirm your settings.

Once you've set up Code Review, it will automatically review pull requests in repositories connected to your Vercel projects.

How it works

Code Review runs automatically when:

- A pull request is created
- A batch of commits is pushed to an open PR
- A draft PR is created, if you've enabled draft reviews in your settings

When triggered, Code Review analyzes all human-readable files in your codebase, including:

- Source code files (JavaScript, TypeScript, Python, etc.)
- Test files
- Configuration files (`package.json`, YAML files, etc.)
- Documentation (markdown files, README files)
- Comments within code

The AI uses your entire codebase as context to understand how your changes fit into the larger system.

Code Review then generates patches, runs them in [secure sandboxes](/docs/vercel-sandbox), and executes your real builds, tests, and lint

Code guidelines

Code Review automatically detects and applies coding guidelines from your repository. When guidelines are found, they're used during revi

Supported guideline files

Code Review looks for these files in priority order (highest to lowest):

File	Description
`AGENTS.md`	OpenAI Codex / universal standard
`CLAUDE.md`	Claude Code instructions
`.github/copilot-instructions.md`	GitHub Copilot
`.cursor/rules/*.mdc`	Cursor rules
`.cursorrules`	Cursor (legacy)
`.windsurfrules`	Windsurf
`.windsurf/rules/*.md`	Windsurf (directory)
`.clinerules`	Cline
`.github/instructions/*.instructions.md`	GitHub Copilot workspace
`.roo/rules/*.md`	Roo Code
`.aiassistant/rules/*.md`	JetBrains AI Assistant
`CONVENTIONS.md`	Aider
`.rules/*.md`	Generic rules
`agent.md`	Generic agent file

When multiple guideline files exist in the same directory, the highest-priority file is used.

How guidelines are applied

- **Hierarchical**: Guidelines from parent directories are inherited. A `CLAUDE.md` at the root applies to all files, while a `src/compon
- **Scoped**: Guidelines only affect files within their directory subtree. A guideline in `src/` won't apply to files in `lib/`.
- **Nested references**: Guidelines can reference other files using `@import "file.md"` or relative markdown links. Referenced files are
- **Size limit**: Guidelines are capped at 50 KB total.

Writing effective guidelines

Guidelines should focus on project-specific conventions that help the reviewer understand your codebase:

- Code style preferences not enforced by linters
- Architecture patterns and design decisions
- Common pitfalls specific to your project
- Testing requirements and patterns

Guidelines are treated as context, not instructions. The reviewer's core behavior (identifying bugs, security issues, and performance pro

Managing reviews

Check out [Managing Reviews](/docs/agent/pr-review/usage) for details on how to customize which repositories get reviewed and monitor you

Pricing

Code Review uses a credit-based system. Each review costs a fixed $0.30 USD plus token costs billed at the Agent's underlying AI provider

Pro teams can redeem a $100 USD promotional credit when enabling Agent. You can [purchase credits and enable auto-reload](/docs/agent/pri

Privacy

Code Review doesn't store or train on your data. It only uses LLMs from providers on our [subprocessor list](https://security.vercel.com/

---
title: "Managing Code Reviews"
description: "Customize which repositories get reviewed and track your review metrics and spending."
last_updated: "2026-01-16T02:19:24.182Z"
source: "https://vercel.com/docs/agent/pr-review/usage"

Managing Code Reviews

Once you've [set up Code Review](/docs/agent/pr-review#how-to-set-up-code-review), you can customize settings and monitor performance fro

Choose which repositories to review

You might want to control which repositories receive automatic reviews, especially when you're testing Code Review for the first time or

To choose which repositories get reviewed:

1. Go to the [Agent tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fvercel-agent\&title=Open+Vercel+Agent) in your dashboard.
2. Click the **…** button, and then select **Settings** to view the Vercel Agent settings.
3. Under **Repositories**, choose which repositories to review:
   - **All repositories** (default): Reviews every repository connected to your Vercel projects
   - **Public only**: Only reviews publicly accessible repositories
   - **Private only**: Only reviews private repositories
4. Click **Save** to apply your changes.

These settings help you start small with specific repos or focus on the repositories that matter most to your team.

Allow reviews on draft PRs

By default, Code Review skips draft pull requests since they're often work-in-progress. You can enable draft reviews if you want early fe

To enable reviews on draft PRs:

1. Go to the [Agent tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fvercel-agent\&title=Open+Vercel+Agent) in your dashboard.
2. Click the **…** button, and then select **Settings** to view the Vercel Agent settings.
3. Under **Review Draft PRs**, select **Review draft PRs**.
4. Click **Save** to apply your changes.

Enabling this setting means you'll use credits on drafts, but you'll get feedback earlier in your development process.

Track spending and costs

You can monitor your spending in real time to manage your budget. The Agent tab shows the cost of each review and your total spending ove

For detailed information about tracking costs, viewing your credit balance, and understanding cost breakdowns, see the [cost tracking sec

Track the suggestions

The Agent tab also shows you the total number of suggestions over a given period, as well as the number of suggestions for each individua

To view suggestions:

1. Go to the [Agent tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fvercel-agent\&title=Open+Vercel+Agent).
2. Check the **Suggestions** column for each review.

A high number of suggestions might indicate complex changes or code that needs more attention. A low number might mean your code is alrea

Review agent efficiency

Understanding how Code Review performs helps you optimize your setup and get the most value from your credits.

The Agent tab provides several metrics for each review:

- **Repository**: Which repository was reviewed
- **PR**: The pull request identifier (click to view the PR)
- **Suggestions**: Number of code changes recommended
- **Review time**: How long the review took to complete
- **Files read**: Number of files the AI analyzed
- **Spend**: Total cost for that review
- **Time**: When the review occurred

Use this data to identify patterns:

- **Expensive reviews**: If certain repositories consistently have high costs, consider whether they need special handling or different r
- **Long review times**: Reviews taking longer than expected might indicate complex codebases or large PRs that could benefit from smalle
- **High file counts**: Repositories with many files analyzed might benefit from more focused review scopes

Export review metrics

You can export all your review data to CSV for deeper analysis, reporting, or tracking trends over time.

To export your data:

1. Go to the [Agent tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fvercel-agent\&title=Open+Vercel+Agent).
2. Click the **Export** button.
3. Save the CSV file to your computer.

The exported data includes all metrics from the dashboard, letting you:

- Create custom reports for your team or stakeholders
- Analyze trends across multiple repositories
- Calculate ROI by comparing review costs to time saved
- Track adoption and usage patterns over time

Disable Vercel Agent

If you need to turn off Vercel Agent completely, you can disable it from the Agent tab. This stops all reviews across all repositories.

To disable Vercel Agent:

1. Go to the [Agent tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fvercel-agent\&title=Open+Vercel+Agent) in your dashboard.
2. Click the **…** button, and then select **Disable Vercel Agent**.
3. Confirm the action in the prompt that appears.

Once disabled, Code Review won't run on any new pull requests. You can re-enable Vercel Agent at any time from the same menu.

---
title: "Vercel Agent Pricing"
description: "Understand how Vercel Agent pricing works and how to manage your credits"
last_updated: "2026-01-16T02:19:24.221Z"
source: "https://vercel.com/docs/agent/pricing"

Vercel Agent Pricing

Vercel Agent uses a credit-based system and all agent features and tools will use the same credit pool.

All teams with Observability Plus have **10 investigations included in their subscription every billing cycle** at no extra cost.
Additional investigations cost both:

Cost component	Price	Details
Fixed cost	$0.30 USD	Charged for each Code Review or additional investigation
Token costs	Pass-through pricing	Billed at the Agent's underlying AI provider's rate, with no additional markup

Your total cost per action is the fixed cost plus the token costs.

The token cost varies based on the complexity and amount of data the AI needs to analyze. You can track your spending in real time in the

Promotional credit

When you enable Agent for the first time, Pro teams can redeem a $100 USD promotional credit. This credit can be used by any Vercel Agent

To redeem your promotional credit:

1. Go to the [Agent tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fvercel-agent\&title=Open+Vercel+Agent) in your dashboard.
2. If you haven't enabled Agent yet, you'll be prompted to **Enable with $100 free credits**.

Once your promotional credit is redeemed, you can track your remaining credits in the [Agent tab](https://vercel.com/d?to=%2F%5Bteam%5D%2

Track costs and spending

Each Code Review or additional investigation costs $0.30 USD plus token costs. You can monitor your spending in real time to manage your

To view costs:

1. Go to the [Agent tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fvercel-agent\&title=Open+Vercel+Agent).
2. Check your current credit balance at the top of the page. Click the **Credits** button to view more details and add credits.
3. View the **Cost** column in the reviews table to see the cost of each individual Code Review or investigation.

The Agent tab shows you the cost of all reviews and investigations over a given period, as well as the cost of each individual action. If

Adding credits

You can add credits to your account at any time through manual purchases or by enabling auto-reload to keep your balance topped up automa

Manual credit purchases

To manually add credits:

1. Go to the [Agent tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fvercel-agent\&title=Open+Vercel+Agent) in your dashboard.
2. Click the **Credits** button at the top of the page.
3. In the dialog that appears, enter the amount you want to add to your balance.
4. Click **Continue to Payment** to enter your card details and complete the purchase.

Your new credit balance will be available immediately and will be used for all Agent features.

Auto-reload

Auto-reload automatically adds credits when your balance falls below a threshold you set. This helps prevent the Vercel Agent tools from

To enable auto-reload:

1. Go to the [Agent tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fvercel-agent\&title=Open+Vercel+Agent) in your dashboard.
2. Click the **Credits** button at the top of the page and select **Enable** next to the auto-reload option.
3. On the next screen, toggle the switch to **Enabled**.
4. Then, configure your auto-reload preferences:
   - **When Balance Falls Below**: Set the threshold that triggers an automatic recharge (for example, $10 USD)
   - **Recharge To Target Balance**: Set the amount your balance will be recharged to (for example, $50 USD)
   - **Monthly Spending Limit** (optional): Set a maximum amount VercelAgent can spend per month to control costs
5. Click **Save** to enable auto-reload.

When your balance drops below the threshold, Vercel will automatically charge your payment method and add the specified amount to your cr

---
title: "Build with AI agents on Vercel"
description: "Install AI agents and services through the Vercel Marketplace to automate workflows and build custom AI systems."
last_updated: "2026-01-16T02:19:24.375Z"
source: "https://vercel.com/docs/agent-integrations"

Build with AI agents on Vercel

Integrating AI agents in your application often means working with separate dashboards, billing systems, and authentication flows for eac

With [AI agents](#ai-agents) and [AI agent services](#ai-agent-services) on the Vercel Marketplace, you can add AI-powered workflows to y

You have access to two types of AI building blocks:

- [**Agents**](#ai-agents): Pre-built systems that handle specialized workflows on your behalf
- [**Services**](#ai-agent-services): Infrastructure you use to build and run your own agents

Getting started

To add an agent or service to your project:

1. Go to the [AI agents and services section](https://vercel.com/marketplace/category/agents) of the Vercel Marketplace and select the ag

2. Review the details and click **Install**.

3. If you selected an agent that needs GitHub access for tasks like code reviews, you'll be prompted to select a Git namespace.

4. Choose an **Installation Plan** from the available options.

5. Click **Continue**.

6. On the configuration page, update the **Resource Name**, review your selections, and click **Create**.

7. Click **Done** once the installation is complete.

You'll be taken to the installation detail page where you can complete the onboarding process to connect your project with the agent or s

Providers

If you're building agents or AI infrastructure, check out [Integrate with Vercel](/docs/integrations/create-integration) to learn how to

AI agents

Agents are pre-built systems that reason, act, and adapt inside your existing workflows, like CodeRabbit, Corridor, and Sourcery. For exa

Each agent integrates with GitHub through a single onboarding flow. Once installed, the agent begins monitoring your repositories and act

AI agent services

Services give you the foundation to create, customize, monitor, and scale your own agents, including Braintrust, Kubiks, Autonoma, Chatba

These services plug into your Vercel workflows so you can build agents specific to your company, products, and customers. They'll integra

More resources

- [AI agents and services on the Vercel Marketplace](https://vercel.com/marketplace/category/agents)
- [Learn how to add and manage a native integration](/docs/integrations/install-an-integration/product-integration)
- [Learn how to create a native integration](/docs/integrations/create-integration/marketplace-product)

---
title: "Adding a Model"
description: "Learn how to add a new AI model to your Vercel projects"
last_updated: "2026-01-16T02:19:24.384Z"
source: "https://vercel.com/docs/ai/adding-a-model"

Adding a Model

If you have integrations installed, scroll to the bottom to access the models explorer.

Exploring models

To explore models:

1. Use the search bar, provider select, or type filter to find the model you want to add
2. Select the model you want to add by pressing the **Explore** button
3. The model playground will open, and you can test the model before adding it to your project

Using the model playground

The model playground lets you test the model you are interested in before adding it to your project. If you have not installed an AI prov

You can use the model playground to test the model's capabilities and see if it fits your projects needs.

The model playground differs depending on the model you are testing. For example, if you are testing a chat model, you can input a prompt

The playground also lets you also configure the model's settings, such as temperature, maximum output length, duration, continuation, top

Adding a model to your project

Once you have decided on the model you want to add to your project:

1. Select the **Add Model** button
2. If you have more than one provider that supports the model you are adding, you will be prompted to select the provider you want to use
3. Review the provider card which displays the models available, along with a description of the provider and links to their website, pri
4. You can now select which projects the provider will have access to. You can choose from **All Projects** or **Specific Projects**
   - If you select **Specific Projects**, you'll be prompted to select the projects you want to connect to the provider. The list will di
   - Multiple projects can be selected during this step
5. You'll be redirected to the provider's website to complete the connection process
6. Once the connection is complete, you'll be redirected back to the Vercel dashboard, and the provider integration dashboard page. From

Featured AI integrations

---
title: "Adding a Provider"
description: "Learn how to add a new AI provider to your Vercel projects."
last_updated: "2026-01-16T02:19:24.355Z"

source: "https://vercel.com/docs/ai/adding-a-provider"

Adding a Provider

When you navigate to the **AI** tab, you'll see a list of installed AI integrations. If you don't have installed integrations, you can br

Adding a native integration provider

1. Select the **Install AI Provider** button on the top right of the **AI** dashboard page.
2. From the list of Marketplace AI Providers, select the provider that you would like to install and click **Continue**.
3. Select a plan from the list of available plans that can include both prepaid and post-paid plans.
   - For prepaid plans, once you select your plan and click Continue:
     - You are taken to a **Manage Funds** screen where you can set up an initial balance for the prepayment.
     - You can also enable auto recharge with a maximum monthly spend. Auto recharge can also be configured at a later stage.
4. Click **Continue**, provide a name for your installation and click **Install**.
5. Once the installation is complete, you are taken to the installation's detail page where you can:
   - Link a project by clicking **Connect Project**
   - Follow a quickstart in different languages to test your installation
   - View the list of all connected projects
   - View the usage of the service

For more information on managing native integration providers, review [Manage native integrations](/docs/integrations/install-an-integrat

Adding a connectable account provider

If no integrations are installed, browse the list of available providers and click on the provider you would like to add.

1. Select the **Add** button next to the provider you want to integrate
2. Review the provider card which displays the models available, along with a description of the provider and links to their website, pri
3. Select the **Add Provider** button
4. You can now select which projects the provider will have access to. You can choose from **All Projects** or **Specific Projects**
   - If you select **Specific Projects**, you'll be prompted to select the projects you want to connect to the provider. The list will di
   - Multiple projects can be selected during this step
5. Select the **Connect to Project** button
6. You'll be redirected to the provider's website to complete the connection process
7. Once the connection is complete, you'll be redirected back to the Vercel dashboard, and the provider integration dashboard page. From

Once you add a provider, the **AI** tab will display a list of the providers you have installed or connected to. To add more providers:

1. Select the **Install AI Provider** button on the top right of the page.
2. Browse down to the list of connectable accounts.
3. Select the provider that you would like to connect to and click **Continue** and follow the instructions from step 4 above.

Featured AI integrations

---
title: "Vercel Deep Infra Integration"
description: "Learn how to add the Deep Infra native integration with Vercel."
last_updated: "2026-01-16T02:19:24.620Z"
source: "https://vercel.com/docs/ai/deepinfra"

Vercel Deep Infra Integration

 provides scalable and
cost-effective infrastructure for deploying and managing machine learning
models. It's optimized for reduced latency and low costs compared to traditional
cloud providers.

This integration gives you access to the large selection of available AI models and allows you to manage your tokens, billing and usage d

Use cases

You can use the [Vercel and Deep Infra integration](https://vercel.com/marketplace/deepinfra) to:

- Seamlessly connect AI models such as DeepSeek and Llama with your Vercel projects.
- Deploy and run inference with high-performance AI models optimized for speed and efficiency.

Available models

Deep Infra provides a diverse range of AI models designed for high-performance tasks for a variety of applications.

More resources

---
title: "Vercel ElevenLabs Integration"
description: "Learn how to add the ElevenLabs connectable account integration with Vercel."
last_updated: "2026-01-16T02:19:24.589Z"
source: "https://vercel.com/docs/ai/elevenlabs"

Vercel ElevenLabs Integration

 specializes in advanced voice
synthesis and audio processing technologies. Its integration with Vercel allows
you to incorporate realistic voice and audio enhancements into your
applications, ideal for creating interactive media experiences.

Use cases

You can use the Vercel and ElevenLabs integration to power a variety of AI applications, including:

- **Voice synthesis**: Use ElevenLabs for generating natural-sounding synthetic voices in applications such as virtual assistants or audi
- **Audio enhancement**: Use ElevenLabs to enhance audio quality in applications, including noise reduction and sound clarity improvement
- **Interactive media**: Use ElevenLabs to implement voice synthesis and audio processing in interactive media and gaming for realistic s

Available models

ElevenLabs offers models that specialize in advanced voice synthesis and audio processing, delivering natural-sounding speech and audio e

More resources

---
title: "Vercel fal Integration"
description: "Learn how to add the fal native integration with Vercel."
last_updated: "2026-01-16T02:19:24.625Z"
source: "https://vercel.com/docs/ai/fal"

Vercel fal Integration

 enables the
development of real-time AI applications with a focus on rapid inference speeds,
achieving response times under ~120ms. Specializing in diffusion models, fal has
no cold starts and a pay-for-what-you-use pricing model.

Use cases

You can use the [Vercel and fal integration](https://vercel.com/marketplace/fal) to power a variety of AI applications, including:

- **Text-to-image applications**: Use fal to integrate real-time text-to-image generation in applications, enabling users to create compl
- **Real-time image processing**: Use fal for applications requiring instantaneous image analysis and modification, such as real-time fil
- **Depth maps creation**: Use fal's AI models for generating depth maps from images, supporting applications in 3D modeling, augmented r

Available models

fal provides a diverse range of AI models designed for high-performance tasks in image and text processing.

More resources

---
title: "Vercel Groq Integration"
description: "Learn how to add the Groq native integration with Vercel."
last_updated: "2026-01-16T02:19:24.629Z"
source: "https://vercel.com/docs/ai/groq"

Vercel Groq Integration

 is a high-performance AI inference
service with an ultra-fast Language Processing Unit (LPU) architecture. It
enables fast response times for language model inference, making it ideal for
applications requiring low latency.

Use cases

You can use the [Vercel and Groq integration](https://vercel.com/marketplace/groq) to:

- Connect AI models such as Whisper-large-v3 for audio processing and Llama models for text generation to your Vercel projects.
- Deploy and run inference with optimized performance.

Available models

Groq provides a diverse range of AI models designed for high-performance tasks.

More resources

---
title: "Vercel LMNT Integration"
description: "Learn how to add LMNT connectable account integration with Vercel."
last_updated: "2026-01-16T02:19:24.600Z"
source: "https://vercel.com/docs/ai/lmnt"

Vercel LMNT Integration

 provides data processing and
predictive analytics models, known for their precision and efficiency.
Integrating LMNT with Vercel enables your applications to offer accurate
insights and forecasts, particularly useful in finance and healthcare sectors.

Use cases

You can use the Vercel and LMNT integration to power a variety of AI applications, including:

- **High quality text-to-speech**: Use LMNT to generate realistic speech that powers chatbots, AI-agents, games, and other digital media
- **Studio quality custom voices**: Use LMNT to clone voices that will faithfully reproduce the emotional richness and realism of actual
- **Reliably low latency, full duplex streaming**: Use LMNT to enable superior performance for conversational experiences, with consisten

More resources

---
title: "Vercel & OpenAI Integration"
description: "Integrate your Vercel project with OpenAI"
last_updated: "2026-01-16T02:19:24.478Z"
source: "https://vercel.com/docs/ai/openai"

Vercel & OpenAI Integration

Vercel integrates with [OpenAI](https://platform.openai.com/overview) to enable developers to build fast, scalable, and secure [AI applic

You can integrate with [any OpenAI model](https://platform.openai.com/docs/models/overview) using the [AI SDK](https://sdk.vercel.ai), in

- **GPT-4o**: Understand and generate natural language or code
- **GPT-4.5**: Latest language model with enhanced emotional intelligence
- **o3-mini**: Reasoning model specialized in code generation and complex tasks
- **DALL·E 3**: Generate and edit images from natural language
- **Embeddings**: Convert term into vectors

Getting started

To help you get started, we have built a [variety of AI templates](https://vercel.com/templates/ai) integrating OpenAI with Vercel.

Getting Your OpenAI API Key

Before you begin, ensure you have an [OpenAI account](https://platform.openai.com/signup). Once registered:

- ### Navigate to API Keys
  Log into your [OpenAI Dashboard](https://platform.openai.com/) and [view API keys](https://platform.openai.com/account/api-keys).

- ### Generate API Key
  Click on **Create new secret key**. Copy the generated API key securely.
  > **💡 Note:** Always keep your API keys confidential. Do not expose them in client-side code. Use [Vercel Environment Variables](/docs

- ### Set Environment Variable
  Finally, add the `OPENAI_API_KEY` environment variable in your project:
  ```shell filename=".env.local"
  OPENAI_API_KEY='sk-...3Yu5'
  ```

Building chat interfaces with the AI SDK

Integrating OpenAI into your Vercel project is seamless with the [AI SDK](https://sdk.vercel.ai/docs).

Install the AI SDK in your project with your favorite package manager:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i ai
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i ai
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i ai
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i ai
    ```
  </Code>
</CodeBlock>

You can use the SDK to build AI applications with [React (Next.js)](https://sdk.vercel.ai/docs/getting-started/nextjs-app-router), [Vue (

Using OpenAI Functions with Vercel

The AI SDK also has **full support** for [OpenAI Functions (tool calling)](https://openai.com/blog/function-calling-and-other-api-updates

Learn more about using [tools with the AI SDK](https://sdk.vercel.ai/docs/foundations/tools).

---
title: "Build with AI on Vercel"
description: "Integrate powerful AI services and models seamlessly into your Vercel projects."
last_updated: "2026-01-16T02:19:24.504Z"
source: "https://vercel.com/docs/ai"

Build with AI on Vercel

AI services and models help enhance and automate the building and deployment of applications for various use cases:

- Chatbots and virtual assistants improve customer interactions.
- AI-powered content generation automates and optimizes digital content.
- Recommendation systems deliver personalized experiences.
- Natural language processing (NLP) enables advanced text analysis and translation.
- Retrieval-augmented generation (RAG) enhances documentation with context-aware responses.
- AI-driven image and media services optimize visual content.

Integrating with AI providers

With Vercel AI integrations, you can build and deploy these AI-powered applications efficiently. Through the Vercel Marketplace, you can

- **Native integrations**: Built-in solutions that work seamlessly with Vercel and include resources with built-in billing and account pr
- **Connectable accounts**: Third-party services you can link to your projects.

Using AI integrations

You can view your installed AI integrations by navigating to the **AI** tab of your Vercel [dashboard](/dashboard). If you don't have ins

See the [adding a provider](/docs/ai/adding-a-provider) guide to learn how to add a provider to your Vercel project, or the [adding a mod

Featured AI integrations

More resources

- [AI Integrations for Vercel](https://www.youtube.com/watch?v=so4Jatc85Aw)

---
title: "Vercel Perplexity Integration"
description: "Learn how to add Perplexity connectable account integration with Vercel."
last_updated: "2026-01-16T02:19:24.614Z"
source: "https://vercel.com/docs/ai/perplexity"

Vercel Perplexity Integration

 specializes in providing
accurate, real-time answers to user questions by combining AI-powered search
with large language models, delivering concise, well-sourced, and conversational
responses. Integrating Perplexity via its [Sonar
API](https://sonar.perplexity.ai/) with Vercel allows your applications to
deliver real-time, web-wide research and question-answering
capabilities—complete with accurate citations, customizable sources, and
advanced reasoning—enabling users to access up-to-date, trustworthy information
directly within your product experience.

Use cases

You can use the Vercel and Perplexity integration to power a variety of AI applications, including:

- **Real-time, citation-backed answers:** Integrate Perplexity to provide users with up-to-date information grounded in live web data, co
- **Customizable search and data sourcing:** Tailor your application's responses by specifying which sources Perplexity should use, ensur
- **Complex, multi-step query handling:** Leverage advanced models like Sonar Pro to process nuanced, multi-part questions, deliver in-de
- **Optimized speed and efficiency:** Benefit from Perplexity's lightweight, fast models that deliver nearly instant answers at scale, ma
- **Fine-grained output control:** Adjust model parameters (e.g., creativity, repetition) and manage output quality to align with your ap

Available models

The Sonar models are each optimized for tasks such as real-time search, advanced reasoning, and in-depth research. Please refer to Perple

More resources

---
title: "Vercel Pinecone Integration"
description: "Learn how to add Pinecone connectable account integration with Vercel."
last_updated: "2026-01-16T02:19:24.609Z"
source: "https://vercel.com/docs/ai/pinecone"

Vercel Pinecone Integration

 is a [vector
database](/kb/guide/vector-databases) service that handles the storage and search
of complex data. With Pinecone, you can use machine-learning models for content
recommendation systems, personalized search, image recognition, and more. The
Vercel Pinecone integration allows you to deploy your models to Vercel and use
them in your applications.

Use cases

You can use the Vercel and Pinecone integration to power a variety of AI applications, including:

- **Personalized search**: Use Pinecone's vector database to provide personalized search results. By analyzing user behavior and preferen
- **Image and video retrieval**: Use Pinecone's vector database in image and video retrieval systems. They can quickly find images or vid
- **Recommendation systems**: Use Pinecone's vector database in e-commerce apps and streaming services to help power recommendation syste

Deploy a template

You can deploy a template to Vercel that includes a pre-trained model and a sample application that uses the model:

More resources

---
title: "Vercel Replicate Integration"
description: "Learn how to add Replicate connectable account integration with Vercel."
last_updated: "2026-01-16T02:19:24.605Z"
source: "https://vercel.com/docs/ai/replicate"

Vercel Replicate Integration

 provides a platform for
accessing and deploying a wide range of open-source artificial intelligence
models. These models span various AI applications such as image and video
processing, natural language processing, and audio synthesis. With the Vercel
Replicate integration, you can incorporate these AI capabilities into your
applications, enabling advanced functionalities and enhancing user experiences.

Use cases

You can use the Vercel and Replicate integration to power a variety of AI applications, including:

- **Content generation**: Use Replicate for generating text, images, and audio content in creative and marketing applications
- **Image and video processing**: Use Replicate in applications for image enhancement, style transfer, or object detection
- **NLP and chat-bots**: Use Replicate's language processing models in chat-bots and natural language interfaces

Available models

Replicate models cover a broad spectrum of AI applications ranging from image and video processing to natural language processing and aud

Deploy a template

You can deploy a template to Vercel that uses a pre-trained model from Replicate:

More resources

---
title: "Vercel Together AI Integration"
description: "Learn how to add Together AI connectable account integration with Vercel."
last_updated: "2026-01-16T02:19:24.862Z"
source: "https://vercel.com/docs/ai/togetherai"

Vercel Together AI Integration

 offers models for interactive
AI experiences, focusing on collaborative and real-time engagement. Integrating
Together AI with Vercel empowers your applications with enhanced user
interaction and co-creative functionalities.

Use cases

You can use the Vercel and Together AI integration to power a variety of AI applications, including:

- **Co-creative platforms**: Use Together AI in platforms that enable collaborative creative processes, such as design or writing
- **Interactive learning environments**: Use Together AI in educational tools for interactive and adaptive learning experiences
- **Real-time interaction tools**: Use Together AI for developing applications that require real-time user interaction and engagement

Available models

Together AI offers models that specialize in collaborative and interactive AI experiences. These models are adept at facilitating real-ti

More resources

---
title: "Vercel xAI Integration"
description: "Learn how to add the xAI native integration with Vercel."
last_updated: "2026-01-16T02:19:24.895Z"
source: "https://vercel.com/docs/ai/xai"

Vercel xAI Integration

 provides language,
chat and vision AI capabilities with integrated billing through Vercel.

Use cases

You can use the [Vercel and xAI integration](https://vercel.com/marketplace/xai) to:

- Perform text generation, translation and question answering in your Vercel projects.
- Use the language with vision model for advanced language understanding and visual processing.

Available models

xAI provides language and language with vision AI models.

More resources

---
title: "Advanced Features"
description: "Advanced Anthropic API features including extended thinking and web search."
last_updated: "2026-01-16T02:19:24.844Z"
source: "https://vercel.com/docs/ai-gateway/anthropic-compat/advanced"

Advanced Features

Extended thinking

Configure extended thinking for models that support chain-of-thought reasoning. The `thinking` parameter allows you to control how reason

Example request

TypeScript

```typescript filename="thinking.ts"
import Anthropic from '@anthropic-ai/sdk';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const anthropic = new Anthropic({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh',
});

const message = await anthropic.messages.create({
  model: 'anthropic/claude-sonnet-4.5',
  max_tokens: 2048,
  thinking: {
    type: 'enabled',
    budget_tokens: 5000,
  },
  messages: [
    {
      role: 'user',
      content: 'Explain quantum entanglement in simple terms.',
    },
  ],
```

```
});

for (const block of message.content) {
  if (block.type === 'thinking') {
    console.log('🧠 Thinking:', block.thinking);
  } else if (block.type === 'text') {
    console.log('💬 Response:', block.text);
  }
}
```

Python

```python filename="thinking.py"
import os
import anthropic

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = anthropic.Anthropic(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh'
)

message = client.messages.create(
    model='anthropic/claude-sonnet-4.5',
    max_tokens=2048,
    thinking={
        'type': 'enabled',
        'budget_tokens': 5000,
    },
    messages=[
        {
            'role': 'user',
            'content': 'Explain quantum entanglement in simple terms.'
        }
    ],
)

for block in message.content:
    if block.type == 'thinking':
        print('🧠 Thinking:', block.thinking)
    elif block.type == 'text':
        print('💬 Response:', block.text)
```

Thinking parameters

- **`type`**: Set to `'enabled'` to enable extended thinking
- **`budget_tokens`**: Maximum number of tokens to allocate for thinking

Response with thinking

When thinking is enabled, the response includes thinking blocks:

```json
{
  "id": "msg_123",
  "type": "message",
  "role": "assistant",
  "content": [
    {
      "type": "thinking",
      "thinking": "Let me think about how to explain quantum entanglement...",
      "signature": "anthropic-signature-xyz"
    },
    {
      "type": "text",
      "text": "Quantum entanglement is like having two magic coins..."
    }
  ],
  "model": "anthropic/claude-sonnet-4.5",
  "stop_reason": "end_turn",
  "usage": {
    "input_tokens": 15,
    "output_tokens": 150
  }
}
```

Web search

Use the built-in web search tool to give the model access to current information from the web.

Example request

TypeScript

```typescript filename="web-search.ts"
import Anthropic from '@anthropic-ai/sdk';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const anthropic = new Anthropic({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh',
});

const message = await anthropic.messages.create({
  model: 'anthropic/claude-sonnet-4.5',
```

```
  max_tokens: 2048,
  tools: [
    {
      type: 'web_search_20250305',
      name: 'web_search',
    },
  ],
  messages: [
    {
      role: 'user',
      content: 'What are the latest developments in quantum computing?',
    },
  ],
});

for (const block of message.content) {
  if (block.type === 'text') {
    console.log(block.text);
  } else if (block.type === 'web_search_tool_result') {
    console.log('Search results received');
  }
}
```

Python

```python filename="web-search.py"
import os
import anthropic

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = anthropic.Anthropic(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh'
)

message = client.messages.create(
    model='anthropic/claude-sonnet-4.5',
    max_tokens=2048,
    tools=[
        {
            'type': 'web_search_20250305',
            'name': 'web_search',
        }
    ],
    messages=[
        {
            'role': 'user',
            'content': 'What are the latest developments in quantum computing?'
        }
    ],
)

for block in message.content:
    if block.type == 'text':
        print(block.text)
    elif block.type == 'web_search_tool_result':
        print('Search results received')
```

---
title: "File Attachments"
description: "Send images and PDF documents as part of your Anthropic API message requests."
last_updated: "2026-01-16T02:19:24.858Z"
source: "https://vercel.com/docs/ai-gateway/anthropic-compat/file-attachments"

File Attachments

Send images and PDF documents as part of your message request.

Example request

TypeScript

```typescript filename="file-attachment.ts"
import Anthropic from '@anthropic-ai/sdk';
import fs from 'node:fs';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const anthropic = new Anthropic({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh',
});

// Read files as base64
const pdfData = fs.readFileSync('./document.pdf');
const imageData = fs.readFileSync('./image.png');

const pdfBase64 = pdfData.toString('base64');
const imageBase64 = imageData.toString('base64');

const message = await anthropic.messages.create({
  model: 'anthropic/claude-sonnet-4.5',
  max_tokens: 1024,
  messages: [
    {
```

```
      role: 'user',
      content: [
        {
          type: 'document',
          source: {
            type: 'base64',
            media_type: 'application/pdf',
            data: pdfBase64,
          },
        },
        {
          type: 'image',
          source: {
            type: 'base64',
            media_type: 'image/png',
            data: imageBase64,
          },
        },
        {
          type: 'text',
          text: 'Please summarize the PDF and describe the image.',
        },
      ],
    },
  ],
});

console.log('Response:', message.content[0].text);
```

Python

```python filename="file-attachment.py"
import os
import base64
import anthropic

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = anthropic.Anthropic(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh'
)

# Read files as base64
with open('./document.pdf', 'rb') as f:
    pdf_base64 = base64.b64encode(f.read()).decode('utf-8')

with open('./image.png', 'rb') as f:
    image_base64 = base64.b64encode(f.read()).decode('utf-8')

message = client.messages.create(
    model='anthropic/claude-sonnet-4.5',
    max_tokens=1024,
    messages=[
        {
            'role': 'user',
            'content': [
                {
                    'type': 'document',
                    'source': {
                        'type': 'base64',
                        'media_type': 'application/pdf',
                        'data': pdf_base64,
                    },
                },
                {
                    'type': 'image',
                    'source': {
                        'type': 'base64',
                        'media_type': 'image/png',
                        'data': image_base64,
                    },
                },
                {
                    'type': 'text',
                    'text': 'Please summarize the PDF and describe the image.',
                },
            ],
        }
    ],
)

print('Response:', message.content[0].text)
```

Supported file types

- **Images**: `image/jpeg`, `image/png`, `image/gif`, `image/webp`
- **Documents**: `application/pdf`

---
title: "Messages"
description: "Create messages using the Anthropic Messages API format with support for streaming."
last_updated: "2026-01-16T02:19:24.667Z"
source: "https://vercel.com/docs/ai-gateway/anthropic-compat/messages"

Messages

Create messages using the Anthropic Messages API format.

Endpoint

```
POST /v1/messages
```

Basic message

Create a non-streaming message.

Example request

TypeScript

```typescript filename="generate.ts"
import Anthropic from '@anthropic-ai/sdk';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const anthropic = new Anthropic({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh',
});

const message = await anthropic.messages.create({
  model: 'anthropic/claude-sonnet-4.5',
  max_tokens: 150,
  messages: [
    {
      role: 'user',
      content: 'Write a one-sentence bedtime story about a unicorn.',
    },
  ],
  temperature: 0.7,
});

console.log('Response:', message.content[0].text);
console.log('Usage:', message.usage);
```

Python

```python filename="generate.py"
import os
import anthropic

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = anthropic.Anthropic(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh'
)

message = client.messages.create(
    model='anthropic/claude-sonnet-4.5',
    max_tokens=150,
    messages=[
        {
            'role': 'user',
            'content': 'Write a one-sentence bedtime story about a unicorn.'
        }
    ],
    temperature=0.7,
)

print('Response:', message.content[0].text)
print('Usage:', message.usage)
```

Response format

```json
{
  "id": "msg_123",
  "type": "message",
  "role": "assistant",
  "content": [
    {
      "type": "text",
      "text": "Once upon a time, a gentle unicorn with a shimmering silver mane danced through moonlit clouds, sprinkling stardust dreams
    }
  ],
  "model": "anthropic/claude-sonnet-4.5",
  "stop_reason": "end_turn",
  "usage": {
    "input_tokens": 15,
    "output_tokens": 28
  }
}
```

Streaming messages

Create a streaming message that delivers tokens as they are generated.

Example request

TypeScript

```typescript filename="stream.ts"
import Anthropic from '@anthropic-ai/sdk';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const anthropic = new Anthropic({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh',
});

const stream = await anthropic.messages.create({
  model: 'anthropic/claude-sonnet-4.5',
  max_tokens: 150,
  messages: [
    {
      role: 'user',
      content: 'Write a one-sentence bedtime story about a unicorn.',
    },
  ],
  temperature: 0.7,
  stream: true,
});

for await (const event of stream) {
  if (event.type === 'content_block_delta') {
    if (event.delta.type === 'text_delta') {
      process.stdout.write(event.delta.text);
    }
  }
}
```

Python

```python filename="stream.py"
import os
import anthropic

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = anthropic.Anthropic(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh'
)

with client.messages.stream(
    model='anthropic/claude-sonnet-4.5',
    max_tokens=150,
    messages=[
        {
            'role': 'user',
            'content': 'Write a one-sentence bedtime story about a unicorn.'
        }
    ],
    temperature=0.7,
) as stream:
    for text in stream.text_stream:
        print(text, end='', flush=True)
```

Streaming event types

Streaming responses use [Server-Sent Events (SSE)](https://developer.mozilla.org/en-US/docs/Web/API/Server-sent_events). The key event ty

- `message_start` - Initial message metadata
- `content_block_start` - Start of a content block (text, tool use, etc.)
- `content_block_delta` - Incremental content updates
- `content_block_stop` - End of a content block
- `message_delta` - Final message metadata (stop reason, usage)
- `message_stop` - End of the message

---
title: "Anthropic-Compatible API"
description: "Use Anthropic-compatible API endpoints with the AI Gateway for seamless integration with Anthropic SDK tools."
last_updated: "2026-01-16T02:19:24.706Z"
source: "https://vercel.com/docs/ai-gateway/anthropic-compat"

Anthropic-Compatible API

AI Gateway provides Anthropic-compatible API endpoints, so you can use the Anthropic SDK and tools like [Claude Code](https://www.claude.

The Anthropic-compatible API implements the same specification as the [Anthropic Messages API](https://docs.anthropic.com/en/api/messages

For more on using AI Gateway with Claude Code, see the [Claude Code instructions](/docs/ai-gateway/claude-code).

Base URL

The Anthropic-compatible API is available at the following base URL:

```
https://ai-gateway.vercel.sh
```

Authentication

The Anthropic-compatible API supports the same authentication methods as the main AI Gateway:

- **API key**: Use your AI Gateway API key with the `x-api-key` header or `Authorization: Bearer <token>` header
- **OIDC token**: Use your Vercel OIDC token with the `Authorization: Bearer <token>` header

You only need to use one of these forms of authentication. If an API key is specified it will take precedence over any OIDC token, even i

Supported endpoints

The AI Gateway supports the following Anthropic-compatible endpoint:

- [`POST /v1/messages`](/docs/ai-gateway/anthropic-compat/messages) - Create messages with support for streaming, [tool calls](/docs/ai-g

For advanced features, see:

- [Advanced features](/docs/ai-gateway/anthropic-compat/advanced) - Extended thinking and web search

Configuring Claude Code

[Claude Code](https://code.claude.com/docs) is Anthropic's agentic coding tool. You can configure it to use Vercel AI Gateway, enabling y

- Route requests through multiple AI providers
- Monitor traffic and spend in your AI Gateway Overview
- View detailed traces in Vercel Observability under AI
- Use any model available through the gateway

- ### Configure environment variables
  Configure Claude Code to use the AI Gateway by setting these [environment variables](https://code.claude.com/docs/en/settings#environme

Variable	Value
`ANTHROPIC_BASE_URL`	`https://ai-gateway.vercel.sh`
`ANTHROPIC_AUTH_TOKEN`	Your AI Gateway API key
`ANTHROPIC_API_KEY`	`""` (empty string)
  > **💡 Note:** Setting `ANTHROPIC_API_KEY` to an empty string is important. Claude Code
  > checks this variable first, and if it's set to a non-empty value, it will use
  > that instead of `ANTHROPIC_AUTH_TOKEN`.
  #### Option 1: Shell alias (simplest)
  Add this alias to your `~/.zshrc` (or `~/.bashrc`):
  ```bash
  alias claude-vercel='ANTHROPIC_BASE_URL="https://ai-gateway.vercel.sh" ANTHROPIC_AUTH_TOKEN="your-api-key-here" ANTHROPIC_API_KEY="" cl
  ```
  Then reload your shell:
  ```bash
  source ~/.zshrc
  ```
  #### Option 2: Wrapper script
  For more flexibility (e.g., adding additional logic), create a wrapper script at `~/bin/claude-vercel`:
  ```bash filename="claude-vercel"
  #!/usr/bin/env bash
  # Routes Claude Code through Vercel AI Gateway

  ANTHROPIC_BASE_URL="https://ai-gateway.vercel.sh" \
  ANTHROPIC_AUTH_TOKEN="your-api-key-here" \
  ANTHROPIC_API_KEY="" \
  claude "$@"
  ```
  Make it executable and ensure `~/bin` is in your PATH:
  ```bash
  mkdir -p ~/bin
  chmod +x ~/bin/claude-vercel
  echo 'export PATH="$HOME/bin:$PATH"' >> ~/.zshrc
  source ~/.zshrc
  ```

- ### Run Claude Code
  Run `claude-vercel` to start Claude Code with AI Gateway:
  ```bash
  claude-vercel
  ```
  Your requests will now be routed through Vercel AI Gateway.

- ### (Optional) Use different models
  You can override the default models that Claude Code uses by setting additional environment variables:
  ```bash
  ANTHROPIC_DEFAULT_SONNET_MODEL="kwaipilot/kat-coder-pro-v1"
  ANTHROPIC_DEFAULT_OPUS_MODEL="zai/glm-4.7"
  ANTHROPIC_DEFAULT_HAIKU_MODEL="minimax/minimax-m2.1"
  ```
  This allows you to use any model available through the AI Gateway while still using Claude Code's familiar interface.
  > **💡 Note:** Models vary widely in their support for tools, extended thinking, and other
  > features that Claude Code relies on. Performance may differ significantly
  > depending on the model and provider you select.

Integration with Anthropic SDK

You can use the AI Gateway's Anthropic-compatible API with the official [Anthropic SDK](https://docs.anthropic.com/en/api/client-sdks). P

> **💡 Note:** The examples and content in this section are not comprehensive. For complete
> documentation on available parameters, response formats, and advanced
> features, refer to the [Anthropic Messages
> API](https://docs.anthropic.com/en/api/messages) documentation.

TypeScript

```typescript filename="client.ts"
import Anthropic from '@anthropic-ai/sdk';

const anthropic = new Anthropic({
  apiKey: process.env.AI_GATEWAY_API_KEY,
  baseURL: 'https://ai-gateway.vercel.sh',
```

````
});

const message = await anthropic.messages.create({
  model: 'anthropic/claude-sonnet-4.5',
  max_tokens: 1024,
  messages: [{ role: 'user', content: 'Hello, world!' }],
});
```

#### Python

```python filename="client.py"
import os
import anthropic

client = anthropic.Anthropic(
    api_key=os.getenv('AI_GATEWAY_API_KEY'),
    base_url='https://ai-gateway.vercel.sh'
)

message = client.messages.create(
    model='anthropic/claude-sonnet-4.5',
    max_tokens=1024,
    messages=[
        {'role': 'user', 'content': 'Hello, world!'}
    ]
)
```

## Parameters

The messages endpoint supports the following parameters:

### Required parameters

- `model` (string): The model to use (e.g., `anthropic/claude-sonnet-4.5`)
- `max_tokens` (integer): Maximum number of tokens to generate
- `messages` (array): Array of message objects with `role` and `content` fields

### Optional parameters

- `stream` (boolean): Whether to stream the response. Defaults to `false`
- `temperature` (number): Controls randomness in the output. Range: 0-1
- `top_p` (number): Nucleus sampling parameter. Range: 0-1
- `top_k` (integer): Top-k sampling parameter
- `stop_sequences` (array): Stop sequences for the generation
- `tools` (array): Array of tool definitions for function calling
- `tool_choice` (object): Controls which tools are called
- `thinking` (object): Extended thinking configuration
- `system` (string or array): System prompt

## Error handling

The API returns standard HTTP status codes and error responses:

### Common error codes

- `400 Bad Request`: Invalid request parameters
- `401 Unauthorized`: Invalid or missing authentication
- `403 Forbidden`: Insufficient permissions
- `404 Not Found`: Model or endpoint not found
- `429 Too Many Requests`: Rate limit exceeded
- `500 Internal Server Error`: Server error

### Error response format

```json
{
  "type": "error",
  "error": {
    "type": "invalid_request_error",
    "message": "Invalid request: missing required parameter 'max_tokens'"
  }
}
```



---
title: "Tool Calls"
description: "Use Anthropic-compatible function calling to allow models to call tools and functions."
last_updated: "2026-01-16T02:19:24.720Z"
source: "https://vercel.com/docs/ai-gateway/anthropic-compat/tool-calls"
---

# Tool Calls

The AI Gateway supports Anthropic-compatible function calling, allowing models to call tools and functions.

Example request

#### TypeScript

```typescript filename="tool-calls.ts"
import Anthropic from '@anthropic-ai/sdk';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const anthropic = new Anthropic({
  apiKey,
````

```
  baseURL: 'https://ai-gateway.vercel.sh',
});

const message = await anthropic.messages.create({
  model: 'anthropic/claude-sonnet-4.5',
  max_tokens: 1024,
  tools: [
    {
      name: 'get_weather',
      description: 'Get the current weather in a given location',
      input_schema: {
        type: 'object',
        properties: {
          location: {
            type: 'string',
            description: 'The city and state, e.g. San Francisco, CA',
          },
          unit: {
            type: 'string',
            enum: ['celsius', 'fahrenheit'],
            description: 'The unit for temperature',
          },
        },
        required: ['location'],
      },
    },
  ],
  messages: [
    {
      role: 'user',
      content: 'What is the weather like in San Francisco?',
    },
  ],
});

console.log('Response:', JSON.stringify(message.content, null, 2));
```

Python

```python filename="tool-calls.py"
import os
import anthropic

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = anthropic.Anthropic(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh'
)

message = client.messages.create(
    model='anthropic/claude-sonnet-4.5',
    max_tokens=1024,
    tools=[
        {
            'name': 'get_weather',
            'description': 'Get the current weather in a given location',
            'input_schema': {
                'type': 'object',
                'properties': {
                    'location': {
                        'type': 'string',
                        'description': 'The city and state, e.g. San Francisco, CA'
                    },
                    'unit': {
                        'type': 'string',
                        'enum': ['celsius', 'fahrenheit'],
                        'description': 'The unit for temperature'
                    }
                },
                'required': ['location']
            }
        }
    ],
    messages=[
        {
            'role': 'user',
            'content': 'What is the weather like in San Francisco?'
        }
    ],
)

print('Response:', message.content)
```

Tool call response format

When the model makes tool calls, the response includes tool use blocks:

```json
{
  "id": "msg_123",
  "type": "message",
  "role": "assistant",
  "content": [
    {
      "type": "tool_use",
      "id": "toolu_123",
      "name": "get_weather",
```

```
      "input": {
        "location": "San Francisco, CA",
        "unit": "fahrenheit"
      }
    }
  ],
  "model": "anthropic/claude-sonnet-4.5",
  "stop_reason": "tool_use",
  "usage": {
    "input_tokens": 82,
    "output_tokens": 45
  }
}
```

---
title: "App Attribution"
description: "Attribute your requests so Vercel can identify and feature your app on AI Gateway pages"
last_updated: "2026-01-16T02:19:24.744Z"
source: "https://vercel.com/docs/ai-gateway/app-attribution"

App Attribution

App attribution allows Vercel to identify the application making a request through AI Gateway. When provided, your app can be featured on AI Gateway pages, driving awareness.

> **💡 Note:** App Attribution is optional. If you do not send these headers, your requests will work normally.

How it works

AI Gateway reads two request headers when present:

- `http-referer`: The URL of the page or site making the request.
- `x-title`: A human-readable name for your app (for example, *"Acme Chat"*).

You can set these headers directly in your server-side requests to AI Gateway.

Examples

\[
    'TypeScript (AI SDK)'

```typescript filename="ai-sdk.ts"
import { streamText } from 'ai';

const result = streamText({
  headers: {
    'http-referer': 'https://myapp.vercel.app',
    'x-title': 'MyApp',
  },
  model: 'anthropic/claude-sonnet-4.5',
  prompt: 'Hello, world!',
});

for await (const part of result.textStream) {
  process.stdout.write(part);
}
```

'TypeScript (OpenAI)'

```typescript filename="openai.ts"
import OpenAI from 'openai';

const openai = new OpenAI({
  apiKey: process.env.AI_GATEWAY_API_KEY,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

const response = await openai.chat.completions.create(
  {
    model: 'anthropic/claude-sonnet-4.5',
    messages: [
      {
        role: 'user',
        content: 'Hello, world!',
      },
    ],
  },
  {
    headers: {
      'http-referer': 'https://myapp.vercel.app',
      'x-title': 'MyApp',
    },
  },
);

console.log(response.choices[0].message.content);
```

'Python (OpenAI)'

```python filename="openai.py"
import os
from openai import OpenAI

client = OpenAI(
```

```
  api_key=os.getenv('AI_GATEWAY_API_KEY'),
  base_url='https://ai-gateway.vercel.sh/v1'
)

response = client.chat.completions.create(
    model='anthropic/claude-sonnet-4.5',
    messages=[
        {
            'role': 'user',
            'content': 'Hello, world!',
        },
    ],
    extra_headers={
        'http-referer': 'https://myapp.vercel.app',
        'x-title': 'MyApp',
    },
)

print(response.choices[0].message.content)
```

Setting headers at the provider level

You can also configure attribution headers when you create the AI Gateway
provider instance. This way, the headers are automatically included in
all requests without needing to specify them for each function call.

```typescript filename="provider-level.ts"
import { streamText } from 'ai';
import { createGateway } from '@ai-sdk/gateway';

const gateway = createGateway({
  headers: {
    'http-referer': 'https://myapp.vercel.app',
    'x-title': 'MyApp',
  },
});

const result = streamText({
  model: gateway('anthropic/claude-sonnet-4.5'),
  prompt: 'Hello, world!',
});

for await (const part of result.textStream) {
  process.stdout.write(part);
}
```

Using the Global Default Provider

You can also use the AI SDK's [global provider configuration](https://ai-sdk.dev/docs/ai-sdk-core/provider-management#global-provider-con

```typescript filename="global-provider.ts"
import { streamText } from 'ai';
import { createGateway } from '@ai-sdk/gateway';

const gateway = createGateway({
  headers: {
    'http-referer': 'https://myapp.vercel.app',
    'x-title': 'MyApp',
  },
});

// Set your provider as the default to allow plain-string model id creation with this instance
globalThis.AI_SDK_DEFAULT_PROVIDER = gateway;

// Now you can use plain string model IDs and they'll use your custom provider
const result = streamText({
  model: 'anthropic/claude-sonnet-4.5', // Uses the gateway provider with headers
  prompt: 'Hello, world!',
});

for await (const part of result.textStream) {
  process.stdout.write(part);
}
```

---
title: "Authentication"
description: "Learn how to authenticate with the AI Gateway using API keys and OIDC tokens."
last_updated: "2026-01-16T02:19:24.766Z"
source: "https://vercel.com/docs/ai-gateway/authentication"

Authentication

To use the AI Gateway, you need to authenticate your requests. There are two authentication methods available:

1. **API Key Authentication**: Create and manage API keys through the Vercel Dashboard
2. **OIDC Token Authentication**: Use Vercel's automatically generated OIDC tokens

API key

API keys provide a secure way to authenticate your requests to the AI Gateway. You can create and manage multiple API keys through the Ve

Creating an API Key

- ### Navigate to the AI Gateway tab
  From the [Vercel dashboard](https://vercel.com/dashboard), click the **AI Gateway** tab to access the AI Gateway settings.

- ### Access API key management
  Click **API Keys** on the left sidebar to view and manage your API keys.

- ### Create a new API key
  Click **Create key** and proceed with **Create key** from the dialog to generate a new API key.

- ### Save your API key
  Once you have the API key, save it to `.env.local` at the root of your project (or in your preferred environment file):
  ```bash filename=".env.local"
  AI_GATEWAY_API_KEY=your_api_key_here
  ```

Using the API key

When you specify a model id as a plain string, the AI SDK will automatically use the Vercel AI Gateway provider to route the request. The

```typescript filename="app/api/chat/route.ts" {5}
import { generateText } from 'ai';

export async function GET() {
  const result = await generateText({
    model: 'xai/grok-4.1-fast-non-reasoning',
    prompt: 'Why is the sky blue?',
  });
  return Response.json(result);
}
```

OIDC token

The [Vercel OIDC token](/docs/oidc) is a way to authenticate your requests to the AI Gateway without needing to manage an API key. Vercel

> **💡 Note:** Vercel OIDC tokens are only valid for 12 hours, so you will need to refresh
> them periodically during local development. You can do this by running `vercel
>   env pull` again.

Setting up OIDC authentication

- ### Link to a Vercel project
  Before you can use the OIDC token during local development, ensure that you link your application to a Vercel project:
  ```bash filename="terminal"
  vercel link
  ```

- ### Pull environment variables
  Pull the environment variables from Vercel to get the OIDC token:
  ```bash filename="terminal"
  vercel env pull
  ```

- ### Use OIDC authentication in your code
  With OIDC authentication, you can directly use the gateway provider without needing to obtain an API key or set it in an environment va
  ```typescript filename="app/api/chat/route.ts" {5}
  import { generateText } from 'ai';

  export async function GET() {
    const result = await generateText({
      model: 'xai/grok-4.1-fast-non-reasoning',
      prompt: 'Why is the sky blue?',
    });
    return Response.json(result);
  }
  ```

---
title: "Bring Your Own Key (BYOK)"
description: "Learn how to configure your own provider keys with the AI Gateway."
last_updated: "2026-01-16T02:19:24.800Z"
source: "https://vercel.com/docs/ai-gateway/byok"

Bring Your Own Key (BYOK)

Using your own credentials with an external AI provider allows AI Gateway to authenticate requests on your behalf with [no added markup](
This approach is useful for utilizing credits provided by the AI provider or executing AI queries that access private cloud data.
If a query using your credentials fails, AI Gateway will retry the query with its system credentials to improve service availability.

Integrating credentials like this with AI Gateway is sometimes referred to as **Bring-Your-Own-Key**, or **BYOK**. In the Vercel dashboar

Provider credentials are scoped to be available throughout your Vercel team, so you can use the same credentials across multiple projects

Getting started

- ### Retrieve credentials from your AI provider
  First, retrieve credentials from your AI provider. These credentials will be used first to authenticate requests made to that provider

- ### Add the credentials to your Vercel team
  1. Go to the [AI Gateway](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fai%2F\&title=) tab in your [Vercel dashboard](https://vercel.co
  2. Click on the **Bring Your Own Key (BYOK)** section on the left sidebar.
  3. Find your provider from the list and click **Add**.
  4. In the dialog that appears, enter the credentials you retrieved from the provider.
  5. Ensure that the **Enabled** toggle is turned on so that the credentials are active.
  6. Click **Test Key** to validate and add your credentials.

- ### Use the credentials in your AI Gateway requests
  Once the credentials are added, it will automatically be included in your requests to the AI Gateway. You can now use these credentials

Request-scoped BYOK

In addition to configuring credentials in the dashboard, you can pass provider credentials on a per-request basis using the `byok` option

When request-scoped BYOK credentials are provided, any cached BYOK credentials configured in the dashboard are not considered for that re

AI SDK usage

```typescript
import type { GatewayProviderOptions } from '@ai-sdk/gateway';
import { generateText } from 'ai';

const { text } = await generateText({
  model: 'anthropic/claude-sonnet-4.5',
  prompt: 'Hello, world!',
  providerOptions: {
    gateway: {
      byok: {
        anthropic: [{ apiKey: process.env.ANTHROPIC_API_KEY }],
      },
    } satisfies GatewayProviderOptions,
  },
});
```

Credential structure by provider

Each provider has its own credential structure:

- **Anthropic**: `{ apiKey: string }`
- **OpenAI**: `{ apiKey: string }`
- **Google Vertex AI**: `{ project: string, location: string, googleCredentials: { privateKey: string, clientEmail: string } }`
- **Amazon Bedrock**: `{ accessKeyId: string, secretAccessKey: string, region?: string }`

For detailed credential parameters for each provider, see the [AI SDK providers documentation](https://ai-sdk.dev/providers/ai-sdk-provid

Multiple credentials

You can specify multiple credentials per provider (tried in order) and credentials for multiple providers:

```typescript
providerOptions: {
  gateway: {
    byok: {
      // Multiple credentials for the same provider (tried in order)
      vertex: [
        { project: 'proj-1', location: 'us-east5', googleCredentials: { privateKey: '...', clientEmail: '...' } },
        { project: 'proj-2', location: 'us-east5', googleCredentials: { privateKey: '...', clientEmail: '...' } },
      ],
      // Multiple providers
      anthropic: [{ apiKey: 'sk-ant-...' }],
      bedrock: [{ accessKeyId: '...', secretAccessKey: '...', region: 'us-east-1' }],
    },
  } satisfies GatewayProviderOptions,
},
```

> **💡 Note:** For OpenAI-compatible API usage with request-scoped BYOK, see the
> [OpenAI-Compatible API
> documentation](/docs/ai-gateway/openai-compat#request-scoped-byok-bring-your-own-key).

Testing your credentials

After successfully adding your credentials for a provider, you can verify that they're working directly from the **Bring Your Own Key (BY

1. In the [AI Gateway](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fai%2F\&title=) tab, navigate to the **Bring Your Own Key (BYOK)** se
2. Click the menu for your configured provider.
3. Select **Test Key** from the dropdown.

This will execute a small test query using a cheap and fast model from the selected provider to verify the health of your credentials. Th

Once the test completes, you can click on the test result badge to open a detailed test result modal. This modal includes:

- The code used to make the test request
- The raw JSON response returned by the AI Gateway

---
title: "Claude Code"
description: "Use Claude Code with the AI Gateway."
last_updated: "2026-01-16T02:19:24.927Z"
source: "https://vercel.com/docs/ai-gateway/claude-code"

Claude Code

AI Gateway provides [Anthropic-compatible API endpoints](/docs/ai-gateway/anthropic-compat) so you can use [Claude Code](https://www.clau

Configuring Claude Code

[Claude Code](https://code.claude.com/docs) is Anthropic's agentic coding tool. You can configure it to use Vercel AI Gateway, enabling y

- Route requests through multiple AI providers
- Monitor traffic and spend in your AI Gateway Overview
- View detailed traces in Vercel Observability under AI
- Use any model available through the gateway

- ### Configure environment variables
  First, log out if you're already logged in:
  ```bash
  claude /logout
  ```

  ```
  Next, ensure you have your AI Gateway API key handy, and configure Claude Code to use the AI Gateway by adding this to your shell confi
  ```bash
  export ANTHROPIC_BASE_URL="https://ai-gateway.vercel.sh"
  export ANTHROPIC_AUTH_TOKEN="your-ai-gateway-api-key"
  export ANTHROPIC_API_KEY=""
  ```
  > **💡 Note:** Setting `ANTHROPIC_API_KEY` to an empty string is important. Claude Code
  > checks this variable first, and if it's set to a non-empty value, it will use
  > that instead of `ANTHROPIC_AUTH_TOKEN`.

- ### Run Claude Code
  Run `claude` to start Claude Code with AI Gateway:
  ```bash
  claude
  ```
  Your requests will now be routed through Vercel AI Gateway.

- ### (Optional) Use different models
  You can override the default models that Claude Code uses by setting additional environment variables:
  ```bash
  export ANTHROPIC_DEFAULT_SONNET_MODEL="kwaipilot/kat-coder-pro-v1"
  export ANTHROPIC_DEFAULT_OPUS_MODEL="zai/glm-4.7"
  export ANTHROPIC_DEFAULT_HAIKU_MODEL="minimax/minimax-m2.1"
  ```
  This allows you to use any model available through the AI Gateway while still using Claude Code's familiar interface.
  > **💡 Note:** Models vary widely in their support for tools, extended thinking, and other
  > features that Claude Code relies on. Performance may differ significantly
  > depending on the model and provider you select.

- ### (Optional) macOS: Secure token storage with Keychain
  If you're on a Mac and would like to manage your API key through a keychain for improved security, set your API key in the keystore wit
  ```bash
  security add-generic-password -a "$USER" -s "ANTHROPIC_AUTH_TOKEN" \
    -w "your-ai-gateway-api-key"
  ```
  and edit the `ANTHROPIC_AUTH_TOKEN` line above to:
  ```bash
  export ANTHROPIC_AUTH_TOKEN=$(
    security find-generic-password -a "$USER" -s "ANTHROPIC_AUTH_TOKEN" -w
  )
  ```
  If you need to update the API key value later, you can do it with:
  ```bash
  security add-generic-password -U -a "$USER" -s "ANTHROPIC_AUTH_TOKEN" \
    -w "new-ai-gateway-api-key"
  ```


---
title: "LangChain"
description: "Learn how to integrate Vercel AI Gateway with LangChain to access multiple AI models through a unified interface"
last_updated: "2026-01-16T02:19:24.939Z"
source: "https://vercel.com/docs/ai-gateway/framework-integrations/langchain"

LangChain

[LangChain](https://js.langchain.com) gives you tools
for every step of the agent development lifecycle.
This guide demonstrates how to integrate [Vercel AI Gateway](/docs/ai-gateway)
with LangChain to access various AI models and providers.

Getting started

- ### Create a new project
  First, create a new directory for your project and initialize it:
  ```bash filename="terminal"
  mkdir langchain-ai-gateway
  cd langchain-ai-gateway
  pnpm dlx init -y
  ```


- ### Install dependencies
  Install the required LangChain packages along with the `dotenv` and `@types/node` packages:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i langchain @langchain/core @langchain/openai dotenv @types/node
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i langchain @langchain/core @langchain/openai dotenv @types/node
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i langchain @langchain/core @langchain/openai dotenv @types/node
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i langchain @langchain/core @langchain/openai dotenv @types/node
      ```
    </Code>
  </CodeBlock>

- ### Configure environment variables
  Create a `.env` file with your [Vercel AI Gateway API key](/docs/ai-gateway#using-the-ai-gateway-with-an-api-key):
      ```

  ```bash filename=".env"
  AI_GATEWAY_API_KEY=your-api-key-here
  ```
  > **💡 Note:** If you're using the [AI Gateway from within a Vercel
  > deployment](/docs/ai-gateway#using-the-ai-gateway-with-a-vercel-oidc-token),
  > you can also use the `VERCEL_OIDC_TOKEN` environment variable which will be
  > automatically provided.

- ### Create your LangChain application
  Create a new file called `index.ts` with the following code:
  ```typescript filename="index.ts" {9, 16}
  import 'dotenv/config';
  import { ChatOpenAI } from '@langchain/openai';
  import { HumanMessage } from '@langchain/core/messages';

  async function main() {
    console.log('=== LangChain Chat Completion with AI Gateway ===');

    const apiKey =
      process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

    const chat = new ChatOpenAI({
      apiKey: apiKey,
      modelName: 'openai/gpt-5.2',
      temperature: 0.7,
      configuration: {
        baseURL: 'https://ai-gateway.vercel.sh/v1',
      },
    });

    try {
      const response = await chat.invoke([
        new HumanMessage('Write a one-sentence bedtime story about a unicorn.'),
      ]);

      console.log('Response:', response.content);
    } catch (error) {
      console.error('Error:', error);
    }
  }

  main().catch(console.error);
  ```
  The following code:
  - Initializes a `ChatOpenAI` instance configured to use the AI Gateway
  - Sets the model `temperature` to `0.7`
  - Makes a chat completion request
  - Handles any potential errors

- ### Running the application
  Run your application using Node.js:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>
  You should see a response from the AI model in your console.

---
title: "LangFuse"
description: "Learn how to integrate Vercel AI Gateway with LangFuse to access multiple AI models through a unified interface"
last_updated: "2026-01-16T02:19:24.965Z"
source: "https://vercel.com/docs/ai-gateway/framework-integrations/langfuse"

LangFuse

[LangFuse](https://langfuse.com/) is an LLM engineering platform
that helps teams collaboratively develop, monitor, evaluate, and debug AI applications.
This guide demonstrates how to integrate [Vercel AI Gateway](/docs/ai-gateway)
with LangFuse to access various AI models and providers.

Getting started

- ### Create a new project
  First, create a new directory for your project and initialize it:
  ```bash filename="terminal"
  mkdir langfuse-ai-gateway
  cd langfuse-ai-gateway
  pnpm dlx init -y
  ```

- ### Install dependencies
  Install the required LangFuse packages along with the `dotenv` and `@types/node` packages:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i langfuse openai dotenv @types/node
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i langfuse openai dotenv @types/node
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i langfuse openai dotenv @types/node
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i langfuse openai dotenv @types/node
      ```
    </Code>
  </CodeBlock>

- ### Configure environment variables
  Create a `.env` file with your [Vercel AI Gateway API key](/docs/ai-gateway#using-the-ai-gateway-with-an-api-key)
  and LangFuse API keys:
  ```bash filename=".env"
  AI_GATEWAY_API_KEY=your-api-key-here

  LANGFUSE_PUBLIC_KEY=your_langfuse_public_key
  LANGFUSE_SECRET_KEY=your_langfuse_secret_key
  LANGFUSE_HOST=https://cloud.langfuse.com
  ```
  > **💡 Note:** If you're using the [AI Gateway from within a Vercel
  > deployment](/docs/ai-gateway#using-the-ai-gateway-with-a-vercel-oidc-token),
  > you can also use the `VERCEL_OIDC_TOKEN` environment variable which will be
  > automatically provided.

- ### Create your LangFuse application
  Create a new file called `index.ts` with the following code:
  ```typescript filename="index.ts" {6, 14}
  import { observeOpenAI } from 'langfuse';
  import OpenAI from 'openai';

  const openaiClient = new OpenAI({
    apiKey: process.env.AI_GATEWAY_API_KEY,
    baseURL: 'https://ai-gateway.vercel.sh/v1',
  });

  const client = observeOpenAI(openaiClient, {
    generationName: 'fun-fact-request', // Optional: Name of the generation in Langfuse
  });

  const response = await client.chat.completions.create({
    model: 'moonshotai/kimi-k2',
    messages: [
      { role: 'system', content: 'You are a helpful assistant.' },
      { role: 'user', content: 'Tell me about the food scene in San Francisco.' },
    ],
  });

  console.log(response.choices[0].message.content);
  ```
  The following code:
  - Creates an OpenAI client configured to use the Vercel AI Gateway
  - Uses `observeOpenAI` to wrap the client for automatic tracing and logging
  - Makes a chat completion request through the AI Gateway
  - Automatically captures request/response data, token usage, and metrics

- ### Running the application
  Run your application using Node.js:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>
  You should see a response from the AI model in your console.

---
title: "LiteLLM"

description: "Learn how to integrate Vercel AI Gateway with LiteLLM to access multiple AI models through a unified interface"
last_updated: "2026-01-16T02:19:24.972Z"
source: "https://vercel.com/docs/ai-gateway/framework-integrations/litellm"

LiteLLM

[LiteLLM](https://www.litellm.ai/) is an open-source library that provides a unified interface to call LLMs.
This guide demonstrates how to integrate [Vercel AI Gateway](/docs/ai-gateway)
with LiteLLM to access various AI models and providers.

Getting started

- ### Create a new project
  First, create a new directory for your project:
  ```bash filename="terminal"
  mkdir litellm-ai-gateway
  cd litellm-ai-gateway
  ```

- ### Install dependencies
  Install the required LiteLLM Python package:
  ```bash filename="terminal" package-manager="pip"
  pip install litellm python-dotenv
  ```

- ### Configure environment variables
  Create a `.env` file with your [Vercel AI Gateway API key](/docs/ai-gateway#using-the-ai-gateway-with-an-api-key):
  ```bash filename=".env"
  VERCEL_AI_GATEWAY_API_KEY=your-api-key-here
  ```
  > **💡 Note:** If you're using the [AI Gateway from within a Vercel
  > deployment](/docs/ai-gateway#using-the-ai-gateway-with-a-vercel-oidc-token),
  > you can also use the `VERCEL_OIDC_TOKEN` environment variable which will be
  > automatically provided.

- ### Create your LiteLLM application
  Create a new file called `main.py` with the following code:
  ```python filename="main.py" {16}
  import os
  import litellm
  from dotenv import load_dotenv

  load_dotenv()

  os.environ["VERCEL_AI_GATEWAY_API_KEY"] = os.getenv("VERCEL_AI_GATEWAY_API_KEY")

  # Define messages
  messages = [
      {"role": "system", "content": "You are a helpful assistant."},
      {"role": "user", "content": "Tell me about the food scene in San Francisco."}
  ]

  response = litellm.completion(
      model="vercel_ai_gateway/openai/gpt-4o",
      messages=messages
  )

  print(response.choices[0].message.content)
  ```
  The following code:
  - Uses LiteLLM's `completion` function to make requests through Vercel AI Gateway
  - Specifies the model using the `vercel_ai_gateway/` prefix
  - Makes a chat completion request and prints the response

- ### Running the application
  Run your Python application:
  ```bash filename="terminal"
  python main.py
  ```
  You should see a response from the AI model in your console.

---
title: "LlamaIndex"
description: "Learn how to integrate Vercel AI Gateway with LlamaIndex to access multiple AI models through a unified interface"
last_updated: "2026-01-16T02:19:24.979Z"
source: "https://vercel.com/docs/ai-gateway/framework-integrations/llamaindex"

LlamaIndex

[LlamaIndex](https://www.llamaindex.ai/) makes it simple to
build knowledge assistants using LLMs connected to your enterprise data.
This guide demonstrates how to integrate [Vercel AI Gateway](/docs/ai-gateway)
with LlamaIndex to access various AI models and providers.

Getting started

- ### Create a new project
  First, create a new directory for your project and initialize it:
  ```bash filename="terminal"
  mkdir llamaindex-ai-gateway
  cd llamaindex-ai-gateway
  ```

- ### Install dependencies
  Install the required LlamaIndex packages along with the `python-dotenv` package:
  ```bash filename="terminal"
  pip install llama-index-llms-vercel-ai-gateway llama-index python-dotenv
  ```

  ```

- ### Configure environment variables
  Create a `.env` file with your [Vercel AI Gateway API key](/docs/ai-gateway#using-the-ai-gateway-with-an-api-key):
  ```bash filename=".env"
  AI_GATEWAY_API_KEY=your-api-key-here
  ```
  > **💡 Note:** If you're using the [AI Gateway from within a Vercel
  > deployment](/docs/ai-gateway#using-the-ai-gateway-with-a-vercel-oidc-token),
  > you can also use the `VERCEL_OIDC_TOKEN` environment variable which will be
  > automatically provided.

- ### Create your LlamaIndex application
  Create a new file called `main.py` with the following code:
  ```python filename="main.py" {2, 8, 12}
  from dotenv import load_dotenv
  from llama_index.llms.vercel_ai_gateway import VercelAIGateway
  from llama_index.core.llms import ChatMessage
  import os

  load_dotenv()

  llm = VercelAIGateway(
      api_key=os.getenv("AI_GATEWAY_API_KEY"),
      max_tokens=200000,
      context_window=64000,
      model="anthropic/claude-4-sonnet",
  )

  message = ChatMessage(role="user", content="Tell me a story in 250 words")
  resp = llm.stream_chat([message])
  for r in resp:
      print(r.delta, end="")
  ```
  The following code:
  - Initializes a `VercelAIGateway` LLM instance with your API key
  - Configures the model to use Anthropic's Claude 4 Sonnet via the AI Gateway
  - Creates a chat message and streams the response

- ### Running the application
  Run your application using Python:
  ```bash filename="terminal"
  python main.py
  ```
  You should see a streaming response from the AI model.

---
title: "Mastra"
description: "Learn how to integrate Vercel AI Gateway with Mastra to access multiple AI models through a unified interface"
last_updated: "2026-01-16T02:19:24.986Z"
source: "https://vercel.com/docs/ai-gateway/framework-integrations/mastra"

Mastra

[Mastra](https://mastra.ai) is a framework for building and deploying AI-powered features
using a modern JavaScript stack powered by the [Vercel AI SDK](/docs/ai-sdk).
Integrating with AI Gateway provides unified model management and routing capabilities.

Getting started

- ### Create a new Mastra project
  First, create a new Mastra project using the CLI:
  ```bash filename="terminal"
  pnpm dlx create-mastra@latest
  ```
  During the setup, the system prompts you to name your project, choose a default provider, and more.
  and more. Feel free to use the default settings.

- ### Install dependencies
  To use the AI Gateway provider, install the `@ai-sdk/gateway` package along with Mastra:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i @ai-sdk/gateway mastra @mastra/core @mastra/memory
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i @ai-sdk/gateway mastra @mastra/core @mastra/memory
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i @ai-sdk/gateway mastra @mastra/core @mastra/memory
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i @ai-sdk/gateway mastra @mastra/core @mastra/memory
      ```
    </Code>
  </CodeBlock>

- ### Configure environment variables
  Create or update your `.env` file with
  your [Vercel AI Gateway API key](/docs/ai-gateway#using-the-ai-gateway-with-an-api-key):
  ```bash filename=".env"
  AI_GATEWAY_API_KEY=your-api-key-here

  ```

- ### Configure your agent to use AI Gateway
  Now, swap out the `@ai-sdk/openai` package (or your existing model provider)
  for the `@ai-sdk/gateway` package.

  Update your agent configuration file, typically `src/mastra/agents/weather-agent.ts` to the following code:
  ```typescript filename="src/mastra/agents/weather-agent.ts" {2, 24}
  import 'dotenv/config';
  import { gateway } from '@ai-sdk/gateway';
  import { Agent } from '@mastra/core/agent';
  import { Memory } from '@mastra/memory';
  import { LibSQLStore } from '@mastra/libsql';
  import { weatherTool } from '../tools/weather-tool';

  export const weatherAgent = new Agent({
    name: 'Weather Agent',
    instructions: `
        You are a helpful weather assistant that provides accurate weather information and can help planning activities based on the weat

        Your primary function is to help users get weather details for specific locations. When responding:
        - Always ask for a location if none is provided
        - If the location name isn't in English, please translate it
        - If giving a location with multiple parts (e.g. "New York, NY"), use the most relevant part (e.g. "New York")
        - Include relevant details like humidity, wind conditions, and precipitation
        - Keep responses concise but informative
        - If the user asks for activities and provides the weather forecast, suggest activities based on the weather forecast.
        - If the user asks for activities, respond in the format they request.

        Use the weatherTool to fetch current weather data.
  `,
    model: gateway('google/gemini-2.5-flash'),
    tools: { weatherTool },
    memory: new Memory({
      storage: new LibSQLStore({
        url: 'file:../mastra.db', // path is relative to the .mastra/output directory
      }),
    }),
  });

  (async () => {
    try {
      const response = await weatherAgent.generate(
        "What's the weather in San Francisco today?",
      );
      console.log('Weather Agent Response:', response.text);
    } catch (error) {
      console.error('Error invoking weather agent:', error);
    }
  })();
  ```

- ### Running the application
  Since your agent is now configured to use AI Gateway,
  run the Mastra development server:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>
  Open the [Mastra Playground and Mastra API](https://mastra.ai/en/docs/server-db/local-dev-playground) to test your agents, workflows, a

---
title: "Framework Integrations"
description: "Explore available community framework integrations with Vercel AI Gateway"
last_updated: "2026-01-16T02:19:25.000Z"
source: "https://vercel.com/docs/ai-gateway/framework-integrations"

Framework Integrations

The Vercel [AI Gateway](/docs/ai-gateway) integrates with popular community AI frameworks and tools,
enabling you to build powerful AI applications while
leveraging the Gateway's features like [cost tracking](/docs/ai-gateway/observability) and [unified API access](/docs/ai-gateway/models-a

Integration overview

You can integrate the AI Gateway with popular frameworks in several ways:

- **OpenAI Compatibility Layer**: Use the AI Gateway's [OpenAI-compatible endpoints](/docs/ai-gateway/openai-compat)
- **Native Support**: Direct integration through plugins or official support
- **AI SDK Integration**: Leverage the [AI SDK](/docs/ai-sdk) to access [AI Gateway](/docs/ai-gateway) capabilities directly
      ```

Supported frameworks

The following below list is a non-exhaustive list of frameworks that currently support AI Gateway integration:

- [LangChain](/docs/ai-gateway/framework-integrations/langchain)
- [LangFuse](/docs/ai-gateway/framework-integrations/langfuse)
- [LiteLLM](/docs/ai-gateway/framework-integrations/litellm)
- [LlamaIndex](/docs/ai-gateway/framework-integrations/llamaindex)
- [Mastra](/docs/ai-gateway/framework-integrations/mastra)
- [Pydantic AI](/docs/ai-gateway/framework-integrations/pydantic-ai)

---
title: "Pydantic AI"
description: "Learn how to integrate Vercel AI Gateway with Pydantic AI to access multiple AI models through a unified interface"
last_updated: "2026-01-16T02:19:25.016Z"
source: "https://vercel.com/docs/ai-gateway/framework-integrations/pydantic-ai"

Pydantic AI

[Pydantic AI](https://ai.pydantic.dev/) is a Python agent framework
designed to make it easy to build production grade applications with AI.
This guide demonstrates how to integrate [Vercel AI Gateway](/docs/ai-gateway)
with Pydantic AI to access various AI models and providers.

Getting started

- ### Create a new project
  First, create a new directory for your project and initialize it:
  ```bash filename="terminal"
  mkdir pydantic-ai-gateway
  cd pydantic-ai-gateway
  ```

- ### Install dependencies
  Install the required Pydantic AI packages along with the `python-dotenv` package:
  ```bash filename="terminal"
  pip install pydantic-ai python-dotenv
  ```

- ### Configure environment variables
  Create a `.env` file with your [Vercel AI Gateway API key](/docs/ai-gateway#using-the-ai-gateway-with-an-api-key):
  ```bash filename=".env"
  VERCEL_AI_GATEWAY_API_KEY=your-api-key-here
  ```
  > **💡 Note:** If you're using the [AI Gateway from within a Vercel
  > deployment](/docs/ai-gateway#using-the-ai-gateway-with-a-vercel-oidc-token),
  > you can also use the `VERCEL_OIDC_TOKEN` environment variable which will be
  > automatically provided.

- ### Create your Pydantic AI application
  Create a new file called `main.py` with the following code:
  ```python filename="main.py" {5, 16}
  from dotenv import load_dotenv
  from pydantic import BaseModel
  from pydantic_ai import Agent
  from pydantic_ai.models.openai import OpenAIModel
  from pydantic_ai.providers.vercel import VercelProvider

  load_dotenv()

  class CityInfo(BaseModel):
      city: str
      country: str
      population: int
      famous_for: str

  agent = Agent(
      OpenAIModel('anthropic/claude-4-sonnet', provider=VercelProvider()),
      output_type=CityInfo,
      system_prompt='Provide accurate city information.'
  )

  if __name__ == '__main__':
      cities = ["Tokyo", "Paris", "New York"]

      for city in cities:
          result = agent.run_sync(f'Tell me about {city}')
          info = result.output

          print(f"City: {info.city}")
          print(f"Country: {info.country}")
          print(f"Population: {info.population:,}")
          print(f"Famous for: {info.famous_for}")
          print("-" * 5)
  ```
  The following code:
  - Defines a `CityInfo` Pydantic model for structured output
  - Uses the `VercelProvider` to route requests through the AI Gateway
  - Handles the response data using Pydantic's type validation

- ### Running the application
  Run your application using Python:
  ```bash filename="terminal"
  python main.py
  ```
  You should see structured city information for Tokyo, Paris, and New York displayed in your console.

---
title: "Getting Started"
description: "Guide to getting started with AI Gateway"
last_updated: "2026-01-16T02:19:25.045Z"
source: "https://vercel.com/docs/ai-gateway/getting-started"

Getting Started

This quickstart will walk you through making an AI
model request with Vercel's [AI Gateway](https://vercel.com/ai-gateway).
While this guide uses the [AI SDK](https://ai-sdk.dev),
you can also integrate with the [OpenAI SDK](/docs/ai-gateway/openai-compat),
[Anthropic SDK](/docs/ai-gateway/anthropic-compat),
or other [community frameworks](/docs/ai-gateway/framework-integrations).

- ### Set up your application
  Start by creating a new directory using the `mkdir` command.
  Change into your new directory and then run the `pnpm init`
  command, which will create a `package.json`.
  ```bash filename="Terminal"
  mkdir demo
  cd demo
  pnpm init
  ```

- ### Install dependencies
  Install the AI SDK package, `ai`, along with other necessary dependencies.
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i ai dotenv @types/node tsx typescript
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i ai dotenv @types/node tsx typescript
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i ai dotenv @types/node tsx typescript
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i ai dotenv @types/node tsx typescript
      ```
    </Code>
  </CodeBlock>
  `dotenv` is used to access environment variables
  (your AI Gateway API key) within your application. The `tsx` package is a TypeScript runner
  that allows you to run your TypeScript code. The `typescript` package is the TypeScript compiler.
  The `@types/node` package is the TypeScript definitions for the Node.js API.

- ### Set up your API key
  To create an API key, go to the [**AI Gateway**](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fai\&title=Go+to+AI+Gateway) tab of the d
  1. Select **API Keys** on the left side bar
  2. Then select **Create key** and proceed with **Create key** from the dialog
  Once you have the API key, create a `.env.local` file and save your API key:
  ```bash filename=".env.local"
  AI_GATEWAY_API_KEY=your_ai_gateway_api_key
  ```
  > **💡 Note:** Instead of using an API key, you can use [OIDC
  > tokens](/docs/ai-gateway/authentication#oidc-token-authentication) to
  > authenticate your requests.
  The AI Gateway provider will default to using the `AI_GATEWAY_API_KEY` environment variable.

- ### Create and run your script
  Create an `index.ts` file in the root of your project and add the following code:
  ```typescript filename="index.ts" {6}
  import { streamText } from 'ai';
  import 'dotenv/config';

  async function main() {
    const result = streamText({
      model: 'openai/gpt-5.2',
      prompt: 'Invent a new holiday and describe its traditions.',
    });

    for await (const textPart of result.textStream) {
      process.stdout.write(textPart);
    }

    console.log();
    console.log('Token usage:', await result.usage);
    console.log('Finish reason:', await result.finishReason);
  }

  main().catch(console.error);
  ```
  Now, run your script:
  ```bash filename="Terminal"
  pnpm tsx index.ts
  ```
  You should see the AI model's response to your prompt.

- ### Next steps
  Continue with the [AI SDK documentation](https://ai-sdk.dev/getting-started) to learn advanced configuration, set up [provider and mode

Using OpenAI SDK

The AI Gateway provides OpenAI-compatible API endpoints that allow you to use existing OpenAI client libraries and tools with the AI Gate

The OpenAI-compatible API includes:

- **Model Management**: List and retrieve the available models
- **Chat Completions**: Create chat completions that support streaming, images, and file attachments
- **Tool Calls**: Call functions with automatic or explicit tool selection
- **Existing Tool Integration**: Use your existing OpenAI client libraries and tools without needing modifications
- **Multiple Languages**: Use the OpenAI SDK in TypeScript and Python, or any language via the REST API

Learn more about using the OpenAI SDK with the AI Gateway in the [OpenAI-Compatible API page](/docs/ai-gateway/openai-compat).

Using Anthropic SDK

The AI Gateway provides Anthropic-compatible API endpoints that allow you to use the Anthropic SDK and tools like Claude Code with the AI

The Anthropic-compatible API includes:

- **Messages API**: Create messages with support for streaming and multi-turn conversations
- **Tool Calls**: Call functions with automatic or explicit tool selection
- **Extended Thinking**: Enable extended thinking for complex reasoning tasks
- **File Attachments**: Attach files and images to your messages
- **Multiple Languages**: Use the Anthropic SDK in TypeScript and Python, or any language via the REST API

Learn more about using the Anthropic SDK with the AI Gateway in the [Anthropic-Compatible API page](/docs/ai-gateway/anthropic-compat).

Using other community frameworks

The AI Gateway is designed to work with any framework that supports the OpenAI API or AI SDK v5/v6, and can also be used with tools like

See the [framework integrations](/docs/ai-gateway/framework-integrations) section to learn more about using AI Gateway with community fra

---
title: "Image Generation with AI SDK"
description: "Generate and edit images using AI models through Vercel AI Gateway with the AI SDK."
last_updated: "2026-01-16T02:19:25.082Z"
source: "https://vercel.com/docs/ai-gateway/image-generation/ai-sdk"

Image Generation with AI SDK

AI Gateway supports image generation using the [AI SDK](https://ai-sdk.dev/docs/ai-sdk-core/image-generation) for the models listed under
page](https://vercel.com/ai-gateway/models?type=image), including multimodal LLMs and image-only models.

Multimodal LLMs

These models can generate both text and images in their responses. They use `generateText` or `streamText` functions with special configu

Nano Banana Pro (`google/gemini-3-pro-image`)

Google's Nano Banana Pro model offers state-of-the-art image generation and editing capabilities with higher quality outputs. Images are

generateText

```typescript filename="generate-nanobanana-pro.ts"
import { generateText } from 'ai';
import 'dotenv/config';

async function main() {
  const result = await generateText({
    model: 'google/gemini-3-pro-image',
    prompt: `Create a detailed illustration of a turquoise-throated puffleg hummingbird resting on a branch covered with dew at sunrise`,
  });

  // Print any text response from the model
  if (result.text) {
    console.log(result.text);
  }

  // Images are available in result.files
  console.log(`Generated ${result.files.length} image(s)`);
  console.log('Usage:', JSON.stringify(result.usage, null, 2));
}

main().catch(console.error);
```

streamText

```typescript filename="stream-nanobanana-pro.ts"
import { streamText } from 'ai';
import 'dotenv/config';

async function main() {
  const result = streamText({
    model: 'google/gemini-3-pro-image',
    prompt: `Generate an artistic rendering of a pond tortoise sleeping on a log in a misty lake at sunset`,
  });

  // Stream text output as it arrives
  for await (const delta of result.fullStream) {
    if (delta.type === 'text-delta') {
      process.stdout.write(delta.text);
    }
  }
```

```
  // Access generated images after streaming completes
  const finalResult = await result;
  console.log(`\nGenerated ${finalResult.files.length} image(s)`);
  console.log('Usage:', JSON.stringify(finalResult.usage, null, 2));
}

main().catch(console.error);
```

Nano Banana (`google/gemini-2.5-flash-image`)

Google's Nano Banana model offers fast, efficient image generation alongside text responses. Images are returned as content parts in `res

generateText

```typescript filename="generate-nanobanana.ts"
import { generateText } from 'ai';
import 'dotenv/config';

async function main() {
  const result = await generateText({
    model: 'google/gemini-2.5-flash-image',
    prompt: `Render two different images of a snowy plover at dusk looking out at San Francisco Bay`,
  });

  // Print any text response from the model
  if (result.text) {
    console.log(result.text);
  }

  // Images are available in result.files
  console.log(`Generated ${result.files.length} image(s)`);
  console.log('Usage:', JSON.stringify(result.usage, null, 2));
}

main().catch(console.error);
```

streamText

```typescript filename="stream-nanobanana.ts"
import { streamText } from 'ai';
import 'dotenv/config';

async function main() {
  const result = streamText({
    model: 'google/gemini-2.5-flash-image',
    prompt: `Render two images of a golden-crowned kinglet perched on a frost-covered pine branch`,
  });

  // Stream text output as it arrives
  for await (const delta of result.fullStream) {
    if (delta.type === 'text-delta') {
      process.stdout.write(delta.text);
    }
  }

  // Access generated images after streaming completes
  const finalResult = await result;
  console.log(`\nGenerated ${finalResult.files.length} image(s)`);
  console.log('Usage:', JSON.stringify(finalResult.usage, null, 2));
}

main().catch(console.error);
```

Save images from Nano Banana models

Nano Banana models (like `google/gemini-2.5-flash-image` and `google/gemini-3-pro-image`) return images as content parts in `result.files

```typescript filename="save-nanobanana-images.ts"
import fs from 'node:fs';
import path from 'node:path';

// Filter for image files from result.files
const imageFiles = result.files.filter((f) =>
  f.mediaType?.startsWith('image/'),
);

if (imageFiles.length > 0) {
  const outputDir = 'output';
  fs.mkdirSync(outputDir, { recursive: true });

  const timestamp = Date.now();

  for (const [index, file] of imageFiles.entries()) {
    const extension = file.mediaType?.split('/')[1] || 'png';
    const filename = `image-${timestamp}-${index}.${extension}`;
    const filepath = path.join(outputDir, filename);

    // Save to file (uint8Array can be written directly)
    await fs.promises.writeFile(filepath, file.uint8Array);
    console.log(`Saved image to ${filepath}`);
  }
}
```

OpenAI models with image generation tool

OpenAI's GPT-5 model variants and a few others support multi-modal image generation through a provider-defined tool. The image generation

Learn more about the [OpenAI Image Generation Tool](https://ai-sdk.dev/providers/ai-sdk-providers/openai#image-generation-tool) in the AI

generateText

```typescript filename="generate-openai-image.ts"
import { generateText } from 'ai';
import 'dotenv/config';
import { openai } from '@ai-sdk/openai';

async function main() {
  const result = await generateText({
    model: 'openai/gpt-5.1-instant',
    prompt: `Generate an image of a black shiba inu dog eating a cake in a green grass field`,
    tools: {
      image_generation: openai.tools.imageGeneration({
        outputFormat: 'webp',
        quality: 'high',
      }),
    },
  });

  // Extract generated images from tool results
  for (const toolResult of result.staticToolResults) {
    if (toolResult.toolName === 'image_generation') {
      const base64Image = toolResult.output.result;
      console.log(
        'Generated image (base64):',
        base64Image.substring(0, 50) + '...',
      );
    }
  }

  console.log('Usage:', JSON.stringify(result.usage, null, 2));
}

main().catch(console.error);
```

streamText

```typescript filename="stream-openai-image.ts"
import { streamText } from 'ai';
import 'dotenv/config';
import { openai } from '@ai-sdk/openai';

async function main() {
  const result = streamText({
    model: 'openai/gpt-5.1-instant',
    prompt: `Generate an image of a corgi puppy playing with colorful balloons in a sunny garden`,
    tools: {
      image_generation: openai.tools.imageGeneration({
        outputFormat: 'webp',
        quality: 'high',
      }),
    },
  });

  for await (const part of result.fullStream) {
    if (part.type === 'tool-result' && !part.dynamic) {
      if (part.toolName === 'image_generation') {
        const base64Image = part.output.result;
        console.log(
          'Generated image (base64):',
          base64Image.substring(0, 50) + '...',
        );
      }
    }
  }

  console.log('Usage:', JSON.stringify(await result.usage, null, 2));
}

main().catch(console.error);
```

Save images from OpenAI tool results

OpenAI models return images as base64-encoded strings in tool results. The approach differs depending on whether you use `generateText` o

generateText

With `generateText`, images are available in `result.staticToolResults` after the call completes:

```typescript filename="save-openai-images.ts"
import fs from 'node:fs';
import path from 'node:path';

const outputDir = 'output';
fs.mkdirSync(outputDir, { recursive: true });

const timestamp = Date.now();

// Extract images from staticToolResults and save to file
for (const [index, toolResult] of result.staticToolResults.entries()) {
  if (toolResult.toolName === 'image_generation') {
    // Decode base64 image from tool result
    const base64Image = toolResult.output.result;
    const buffer = Buffer.from(base64Image, 'base64');
```

```
    const filename = `image-${timestamp}-${index}.webp`;
    const filepath = path.join(outputDir, filename);

    // Save to file
    await fs.promises.writeFile(filepath, buffer);
    console.log(`Saved image to ${filepath}`);
  }
}
```

streamText

With `streamText`, images arrive as `tool-result` events in the stream. Save them as they come in:

```typescript filename="save-openai-images-stream.ts"
import fs from 'node:fs';
import path from 'node:path';

const outputDir = 'output';
fs.mkdirSync(outputDir, { recursive: true });

const timestamp = Date.now();
let imageIndex = 0;

// Extract images from tool-result events and save to file
for await (const part of result.fullStream) {
  if (part.type === 'tool-result' && !part.dynamic) {
    if (part.toolName === 'image_generation') {
      // Decode base64 image from tool result
      const base64Image = part.output.result;
      const buffer = Buffer.from(base64Image, 'base64');

      const filename = `image-${timestamp}-${imageIndex}.webp`;
      const filepath = path.join(outputDir, filename);

      // Save to file
      await fs.promises.writeFile(filepath, buffer);
      console.log(`Saved image to ${filepath}`);
      imageIndex++;
    }
  }
}
```

Image-only models

These models are specialized for image generation and use the `experimental_generateImage` function.

Google Vertex Imagen

Google's Imagen models provide high-quality image generation with fine-grained control over output parameters. Multiple Imagen models are

- `google/imagen-4.0-ultra-generate-001`
- `google/imagen-4.0-generate-001`

```typescript filename="generate-imagen.ts"
import { experimental_generateImage as generateImage } from 'ai';
import 'dotenv/config';

async function main() {
  const result = await generateImage({
    model: 'google/imagen-4.0-ultra-generate-001',
    prompt: `A majestic Bengal tiger drinking water from a crystal-clear mountain stream at golden hour`,
    n: 2,
    aspectRatio: '16:9',
  });

  console.log(`Generated ${result.images.length} image(s)`);
}

main().catch(console.error);
```

Black Forest Labs

Black Forest Labs' Flux models offer advanced image generation with support for various aspect ratios and capabilities. Multiple Flux mod

- `bfl/flux-2-pro`
- `bfl/flux-2-flex`
- `bfl/flux-kontext-max`
- `bfl/flux-kontext-pro`
- `bfl/flux-pro-1.0-fill`
- `bfl/flux-pro-1.1`

```typescript filename="generate-bfl.ts"
import { experimental_generateImage as generateImage } from 'ai';
import 'dotenv/config';

async function main() {
  const result = await generateImage({
    model: 'bfl/flux-2-pro',
    prompt: `A vibrant coral reef ecosystem with tropical fish swimming around colorful sea anemones`,
    aspectRatio: '4:3',
  });

  console.log(`Generated ${result.images.length} image(s)`);
}

main().catch(console.error);
```

Save generated images from image-only models

All generated images from image-only models are returned in `result.images` as objects containing:

- `base64`: The image as a base64-encoded string
- `mediaType`: The MIME type (e.g., `image/png`, `image/jpeg`, `image/webp`)

```typescript filename="save-image-only-models.ts"
import fs from 'node:fs';
import path from 'node:path';

const outputDir = 'output';
fs.mkdirSync(outputDir, { recursive: true });

const timestamp = Date.now();

// Extract images from result.images and save to file
for (const [index, image] of result.images.entries()) {
  // Decode base64 image
  const buffer = Buffer.from(image.base64, 'base64');

  const extension = image.mediaType?.split('/')[1] || 'png';
  const filename = `image-${timestamp}-${index}.${extension}`;
  const filepath = path.join(outputDir, filename);

  // Save to file
  await fs.promises.writeFile(filepath, buffer);
  console.log(`Saved image to ${filepath}`);
}
```

For more information on generating images with the AI SDK, see the [AI SDK documentation](https://ai-sdk.dev/docs/ai-sdk-core/image-gener

---
title: "Image Generation with OpenAI-Compatible API"
description: "Generate and edit images using AI models through Vercel AI Gateway with OpenAI-compatible API."
last_updated: "2026-01-16T02:19:25.181Z"
source: "https://vercel.com/docs/ai-gateway/image-generation/openai"

Image Generation with OpenAI-Compatible API

AI Gateway supports image generation using the OpenAI-compatible API for the models listed under the **Image Gen** filter at the [AI Gate
page](https://vercel.com/ai-gateway/models?type=image), including multimodal LLMs and image-only models.

Multimodal LLMs

Multimodal LLMs like Nano Banana, Nano Banana Pro, and GPT-5 variants can generate images alongside text using the `/v1/chat/completions`

Generate response format

```json
{
  "id": "chatcmpl-123",
  "object": "chat.completion",
  "created": 1677652288,
  "model": "google/gemini-3-pro-image",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "I've generated a beautiful sunset image for you.",
        "images": [
          {
            "type": "image_url",
            "image_url": {
              "url": "..."
            }
          }
        ]
      },
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 15,
    "completion_tokens": 28,
    "total_tokens": 43
  }
}
```

Streaming response format

For streaming requests, images are delivered in delta chunks:

```json
{
  "id": "chatcmpl-123",
  "object": "chat.completion.chunk",
  "created": 1677652288,
  "model": "google/gemini-3-pro-image",
  "choices": [
    {
      "index": 0,
      "delta": {
        "images": [
```

```
          {
            "type": "image_url",
            "image_url": {
              "url": "..."
            }
          }
        ]
      },
      "finish_reason": null
    }
  ]
}
```

Image-only models

Image-only models use the OpenAI Images API (`/v1/images/generations`) for specialized image creation.

Google Vertex Imagen

Google's Imagen models provide high-quality image generation with fine-grained control. Multiple models are available including `google/i

View available [Imagen provider options](https://ai-sdk.dev/providers/ai-sdk-providers/google-vertex#image-models) for configuration deta

TypeScript (Basic)

```typescript filename="generate-imagen-simple.ts"
import OpenAI from 'openai';
import 'dotenv/config';

async function main() {
  const openai = new OpenAI({
    apiKey: process.env.AI_GATEWAY_API_KEY,
    baseURL: 'https://ai-gateway.vercel.sh/v1',
  });

  const result = await openai.images.generate({
    model: 'google/imagen-4.0-ultra-generate-001',
    prompt: `A snow leopard prowling through a rocky mountain landscape during a light snowfall`,
    n: 2,
  });

  // Process the generated images
  for (const image of result.data) {
    if (image.b64_json) {
      console.log(
        'Generated image (base64):',
        image.b64_json.substring(0, 50) + '...',
      );
    }
  }
}

main().catch(console.error);
```

TypeScript (With Options)

```typescript filename="generate-imagen-options.ts"
import OpenAI from 'openai';
import 'dotenv/config';

async function main() {
  const openai = new OpenAI({
    apiKey: process.env.AI_GATEWAY_API_KEY,
    baseURL: 'https://ai-gateway.vercel.sh/v1',
  });

  const result = await openai.images.generate({
    model: 'google/imagen-4.0-ultra-generate-001',
    prompt: `A cascading waterfall in a lush rainforest with mist rising and exotic birds flying`,
    n: 2,
    // @ts-expect-error - Provider options are not in OpenAI types
    providerOptions: {
      googleVertex: {
        aspectRatio: '1:1',
        safetyFilterLevel: 'block_some',
      },
    },
  });

  // Process the generated images
  for (const image of result.data) {
    if (image.b64_json) {
      console.log(
        'Generated image (base64):',
        image.b64_json.substring(0, 50) + '...',
      );
    }
  }
}

main().catch(console.error);
```

Python

```python filename="generate-imagen.py"
import base64
import json
```

```
import os
from datetime import datetime

from dotenv import load_dotenv
from openai import OpenAI

load_dotenv()

def main():
    api_key = os.getenv("AI_GATEWAY_API_KEY") or os.getenv("VERCEL_OIDC_TOKEN")
    base_url = (
        os.getenv("AI_GATEWAY_BASE_OPENAI_COMPAT_URL")
        or "https://ai-gateway.vercel.sh/v1"
    )

    client = OpenAI(
        api_key=api_key,
        base_url=base_url,
    )

    result = client.images.generate(
        model="google/imagen-4.0-ultra-generate-001",
        prompt=(
            "A red fox walking through a snowy forest clearing "
            "with pine trees in the background"
        ),
        n=2,
        response_format="b64_json",
        extra_body={
            "providerOptions": {
                "googleVertex": {
                    "aspectRatio": "1:1",
                    "safetyFilterLevel": "block_some",
                }
            }
        },
    )

    if not result or not result.data or len(result.data) == 0:
        raise Exception("No image data received from OpenAI-compatible endpoint")

    print(f"Generated {len(result.data)} image(s)")

    for i, image in enumerate(result.data):
        if hasattr(image, "b64_json") and image.b64_json:
            # Decode base64 to get image size
            image_bytes = base64.b64decode(image.b64_json)
            print(f"Image {i+1}:")
            print(f"  Size: {len(image_bytes)} bytes")
            print(f"  Base64 preview: {image.b64_json[:50]}...")

            # Save image to file with timestamp
            timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
            output_file = f"output/output_image_{timestamp}_{i+1}.png"
            print(f"  Saving image to {output_file}")
            with open(output_file, "wb") as f:
                f.write(image_bytes)

    if hasattr(result, "provider_metadata"):
        print("\nProvider metadata:")
        print(json.dumps(result.provider_metadata, indent=2))

if __name__ == "__main__":
    main()
```

Black Forest Labs

Black Forest Labs' Flux models offer advanced image generation with various capabilities. Multiple models are available including but not

- `bfl/flux-2-pro`
- `bfl/flux-2-flex`
- `bfl/flux-kontext-max`
- `bfl/flux-kontext-pro`
- `bfl/flux-pro-1.0-fill`
- `bfl/flux-pro-1.1`

View available [Black Forest Labs provider options](https://ai-sdk.dev/providers/ai-sdk-providers/black-forest-labs#provider-options) for

TypeScript (Basic)

```typescript filename="generate-bfl-simple.ts"
import OpenAI from 'openai';
import 'dotenv/config';

async function main() {
  const openai = new OpenAI({
    apiKey: process.env.AI_GATEWAY_API_KEY,
    baseURL: 'https://ai-gateway.vercel.sh/v1',
  });

  const result = await openai.images.generate({
    model: 'bfl/flux-2-pro',
    prompt: `Render an echidna swimming across the Mozambique channel at sunset with phosphorescent jellyfish`,
  });

  // Process the generated images
  for (const image of result.data) {
    if (image.b64_json) {
      console.log(
```

```
        'Generated image (base64):',
        image.b64_json.substring(0, 50) + '...',
      );
    }
  }
}

main().catch(console.error);
```

TypeScript (With Options)

```typescript filename="generate-bfl-options.ts"
import OpenAI from 'openai';
import 'dotenv/config';

async function main() {
  const openai = new OpenAI({
    apiKey: process.env.AI_GATEWAY_API_KEY,
    baseURL: 'https://ai-gateway.vercel.sh/v1',
  });

  const result = await openai.images.generate({
    model: 'bfl/flux-2-pro',
    prompt: `Draw a gorgeous image of a river made of white owl feathers snaking through a serene winter landscape`,
    // @ts-expect-error - Provider options are not in OpenAI types
    providerOptions: {
      blackForestLabs: {
        outputFormat: 'jpeg',
        safetyTolerance: 2,
      },
    },
  });

  // Process the generated images
  for (const image of result.data) {
    if (image.b64_json) {
      console.log(
        'Generated image (base64):',
        image.b64_json.substring(0, 50) + '...',
      );
    }
  }
}

main().catch(console.error);
```

Python

```python filename="generate-bfl.py"
import base64
import json
import os
from datetime import datetime

from dotenv import load_dotenv
from openai import OpenAI

load_dotenv()

def main():
    api_key = os.getenv("AI_GATEWAY_API_KEY") or os.getenv("VERCEL_OIDC_TOKEN")
    base_url = (
        os.getenv("AI_GATEWAY_BASE_OPENAI_COMPAT_URL")
        or "https://ai-gateway.vercel.sh/v1"
    )

    client = OpenAI(
        api_key=api_key,
        base_url=base_url,
    )

    result = client.images.generate(
        model="bfl/flux-2-pro",
        prompt=(
            "A mystical aurora borealis dancing over a frozen lake "
            "with snow-covered mountains reflected in the ice"
        ),
        n=1,
        response_format="b64_json",
        extra_body={
            "providerOptions": {
                "blackForestLabs": {
                    "outputFormat": "jpeg",
                    "safetyTolerance": 2,
                }
            }
        },
    )

    if not result or not result.data or len(result.data) == 0:
        raise Exception("No image data received from OpenAI-compatible endpoint")

    print(f"Generated {len(result.data)} image(s)")

    for i, image in enumerate(result.data):
        if hasattr(image, "b64_json") and image.b64_json:
            # Decode base64 to get image size
            image_bytes = base64.b64decode(image.b64_json)
```

```
            print(f"Image {i+1}:")
            print(f"  Size: {len(image_bytes)} bytes")
            print(f"  Base64 preview: {image.b64_json[:50]}...")

            # Save image to file with timestamp
            timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
            output_file = f"output/output_image_{timestamp}_{i+1}.png"
            print(f"  Saving image to {output_file}")
            with open(output_file, "wb") as f:
                f.write(image_bytes)

    if hasattr(result, "provider_metadata"):
        print("\nProvider metadata:")
        print(json.dumps(result.provider_metadata, indent=2))

if __name__ == "__main__":
    main()
```

Python

You can use the OpenAI Python client to generate images with the AI Gateway:

```python filename="generate-image.py"
import base64
import os
from datetime import datetime

from dotenv import load_dotenv
from openai import OpenAI

load_dotenv()

def main():
    # Initialize the OpenAI client with AI Gateway
    client = OpenAI(
        api_key=os.getenv("AI_GATEWAY_API_KEY"),
        base_url="https://ai-gateway.vercel.sh/v1",
    )

    # Generate an image
    result = client.images.generate(
        model="bfl/flux-2-pro",
        prompt="A majestic blue whale breaching the ocean surface at sunset",
        n=1,
        response_format="b64_json",
    )

    if not result.data:
        raise Exception("No image data received")

    print(f"Generated {len(result.data)} image(s)")

    # Save images to disk
    for i, image in enumerate(result.data):
        if image.b64_json:
            image_bytes = base64.b64decode(image.b64_json)
            timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
            output_file = f"output/image_{timestamp}_{i+1}.png"

            with open(output_file, "wb") as f:
                f.write(image_bytes)
            print(f"Saved image to {output_file}")

if __name__ == "__main__":
    main()
```

REST API

You can use the OpenAI Images API directly via REST without a client library:

```typescript filename="generate-image-rest.ts"
import 'dotenv/config';

async function main() {
  const apiKey = process.env.AI_GATEWAY_API_KEY;
  const baseURL = 'https://ai-gateway.vercel.sh/v1';

  // Send POST request to images/generations endpoint
  const response = await fetch(`${baseURL}/images/generations`, {
    method: 'POST',
    headers: {
      Authorization: `Bearer ${apiKey}`,
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      model: 'bfl/flux-2-pro',
      prompt: `A playful dolphin pod jumping through ocean waves at sunrise with seabirds flying overhead`,
      providerOptions: {
        blackForestLabs: { outputFormat: 'jpeg' },
      },
      n: 3,
    }),
  });

  if (!response.ok) {
    throw new Error(`Image generation failed: ${response.status}`);
  }
```

```
  const json = await response.json();

  // Images are returned as base64 strings in json.data
  for (const image of json.data) {
    if (image.b64_json) {
      console.log(
        'Generated image (base64):',
        image.b64_json.substring(0, 50) + '...',
      );
    }
  }

  console.log('Generated', json.data.length, 'image(s)');
}

main().catch(console.error);
```

---
title: "Image Generation"
description: "Generate and edit images using AI models through Vercel AI Gateway with support for multiple providers and modalities."
last_updated: "2026-01-16T02:19:25.185Z"
source: "https://vercel.com/docs/ai-gateway/image-generation"

Image Generation

The Vercel [AI Gateway](/docs/ai-gateway) supports image generation and editing capabilities. You can generate new images from text promp

To see which models AI Gateway supports for image generation, use the **Image Gen** filter at the [AI Gateway Models
page](https://vercel.com/ai-gateway/models?type=image).

Integration methods

To implement image generation with AI Gateway, use one of the following methods:

- **[AI SDK](/docs/ai-gateway/image-generation/ai-sdk)**: Use the AI SDK for TypeScript/JavaScript applications with native support for s
- **[OpenAI-Compatible API](/docs/ai-gateway/image-generation/openai)**: Use the OpenAI-compatible endpoints for compatibility with exist

---
title: "Model Variants"
description: "Enable provider-specific capabilities (like Anthropic 1M context) via headers when calling models through AI Gateway."
last_updated: "2026-01-16T02:19:25.200Z"
source: "https://vercel.com/docs/ai-gateway/model-variants"

Model Variants

Some AI inference providers offer special variants of models. These models can
have different features such as a larger context size. They may incur different
costs associated with requests as well.

When AI Gateway makes these models available they will be highlighted on the
model detail page with a **Model Variants** section in the relevant provider
card providing an overview of the feature set and linking to more detail.

Model variants sometimes rely on preview or beta features offered by the
inference provider. Their ongoing availability can therefore be less predictable
than that of a stable model feature. Check the provider's site for the latest
information.

Anthropic Claude Sonnet 4: 1M token context (beta)

Enable with header `anthropic-beta: context-1m-2025-08-07`.

- **Learn more**: [Announcement](https://www.anthropic.com/news/1m-context), [Context windows docs](https://docs.anthropic.com/en/build-w
- **Pricing (summary)**: If total input tokens (prompt + cache reads/writes) exceed 200K, input is charged 2× and output 1.5×; otherwise

TypeScript (AI SDK)

```typescript filename="ai-sdk.ts"
import { streamText } from 'ai';
import { largePrompt } from './largePrompt.ts';

const result = streamText({
  headers: {
    'anthropic-beta': 'context-1m-2025-08-07',
  },
  model: 'anthropic/claude-sonnet-4.5',
  prompt: `You have a big brain. Summarize into 3 sentences: ${largePrompt}`,
  providerOptions: {
    gateway: { only: ['anthropic'] },
  },
});

for await (const part of result.textStream) {
  process.stdout.write(part);
}
// Log final chunk with provider metadata detail.
console.log(JSON.stringify(await result.providerMetadata, null, 2));
```

TypeScript (OpenAI)

```typescript filename="openai.ts"
import OpenAI from 'openai';
import { largePrompt } from './largePrompt.ts';
```

```
const openai = new OpenAI({
  apiKey: process.env.AI_GATEWAY_API_KEY,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

// @ts-expect-error
const stream = await openai.chat.completions.create(
  {
    model: 'anthropic/claude-sonnet-4.5',
    messages: [
      {
        role: 'user',
        content: `You have a big brain. Summarize into 3 sentences: ${largePrompt}`,
      },
    ],
    stream: true,
    providerOptions: {
      gateway: { only: ['anthropic'] },
    },
  },
  {
    headers: {
      'anthropic-beta': 'context-1m-2025-08-07',
    },
  },
);

for await (const chunk of stream) {
  const content = chunk.choices[0]?.delta?.content;
  if (content) {
    process.stdout.write(content);
  } else {
    // Log final chunk with provider metadata detail.
    console.log(JSON.stringify(chunk, null, 2));
  }
}
```

Python (OpenAI)

```python filename="openai.py"
import json
import os
from openai import OpenAI

client = OpenAI(
  api_key=os.getenv('AI_GATEWAY_API_KEY'),
  base_url='https://ai-gateway.vercel.sh/v1'
)
large_prompt = 'your-large-prompt'

stream = client.chat.completions.create(
    model='anthropic/claude-sonnet-4.5',
    messages=[
        {
            'role': 'user',
            'content': f'You have a big brain. Summarize into 3 sentences: {large_prompt}',
        },
    ],
    extra_headers={
        'anthropic-beta': 'context-1m-2025-08-07',
    },
    stream=True
)

for chunk in stream:
    if chunk.choices[0].delta.content:
        print(chunk.choices[0].delta.content, end='', flush=True)
    # Log final chunk with provider metadata detail.
    if chunk.choices[0].finish_reason and hasattr(chunk.choices[0].delta, 'provider_metadata') and chunk.choices[0].delta.provider_metada
        print('\nProvider metadata:')
        print(json.dumps(
            chunk.choices[0].delta.provider_metadata, indent=2))
```


---
title: "Models & Providers"
description: "Learn about models and providers for the AI Gateway."
last_updated: "2026-01-16T02:19:25.239Z"
source: "https://vercel.com/docs/ai-gateway/models-and-providers"

Models & Providers

The AI Gateway's unified API is built to be flexible, allowing you to switch between [different AI models](https://vercel.com/ai-gateway/m

> **💡 Note:** To view the list of supported models and providers, check out the [AI Gateway
> models page](https://vercel.com/ai-gateway/models).

What are models and providers?

Models are AI algorithms that process your input data to generate responses, such as [Grok 4.1](/ai-gateway/models/grok-4.1-fast-reasonin

In some cases, multiple providers, including the model creator, host the same model. For example, you can use the `xai/grok-code-fast-1` m

Different providers may have different specifications for the same model such as different pricing and performance. You can choose the on

You can view the list of supported models and providers by following these steps:

1. Go to the [**AI Gateway** tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fai\&title=Go+to+AI+Gateway) in your Vercel dashboard.
2. Click on [**Models**](/ai-gateway/models) at the top of the AI Gateway page.

Specifying the model

There are two ways to specify the model and provider to use for an AI Gateway request:

- [As part of an AI SDK function call](#as-part-of-an-ai-sdk-function-call)
- [Globally for all requests in your application](#globally-for-all-requests-in-your-application)

As part of an AI SDK function call

In the AI SDK, you can specify the model and provider directly in your API calls using either plain strings or the AI Gateway provider. T

To use AI Gateway, specify a model and provider via a plain string, for example:

```typescript filename="app/api/chat/route.ts" {6}
import { generateText } from 'ai';
import { NextRequest } from 'next/server';

export async function GET() {
  const result = await generateText({
    model: 'xai/grok-4.1-fast-non-reasoning',
    prompt: 'Tell me the history of the San Francisco Mission-style burrito.',
  });
  return Response.json(result);
}
```

You can test different models by changing the `model` parameter and opening your browser to `http://localhost:3000/api/chat`.

You can also use a provider instance. This can be useful if you'd like to create models to use with a [custom provider](https://ai-sdk.de

Install the `@ai-sdk/gateway` package directly as a dependency in your project.

```bash filename="terminal"
pnpm install @ai-sdk/gateway
```

You can change the model by changing the string passed to `gateway()`.

```typescript filename="app/api/chat/route.ts" {2, 7}
import { generateText } from 'ai';
import { gateway } from '@ai-sdk/gateway';
import { NextRequest } from 'next/server';

export async function GET() {
  const result = await generateText({
    model: gateway('anthropic/claude-opus-4.5'),
    prompt: 'Tell me the history of the San Francisco Mission-style burrito.',
  });
  return Response.json(result);
}
```

The example above uses the default `gateway` provider instance. You can also create a custom provider instance to use in your application

```typescript filename="app/api/chat/route.ts" {4-7, 11}
import { generateText } from 'ai';
import { createGateway } from '@ai-sdk/gateway';

const gateway = createGateway({
  apiKey: process.env.AI_GATEWAY_API_KEY, // the default environment variable for the API key
  baseURL: 'https://ai-gateway.vercel.sh/v1/ai', // the default base URL
});

export async function GET() {
  const result = await generateText({
    model: gateway('anthropic/claude-opus-4.5'),
    prompt: 'Why is the sky blue?',
  });
  return Response.json(result);
}
```

Globally for all requests in your application

The Vercel AI Gateway is the default provider for the AI SDK when a model is specified as a string. You can set a different provider as t

This is intended to be done in a file that runs before any other AI SDK calls. In the case of a Next.js application, you can do this in [

```typescript filename="instrumentation.ts" {1, 5}
import { openai } from '@ai-sdk/openai';

export async function register() {
  // This runs once when the Node.js runtime starts
  globalThis.AI_SDK_DEFAULT_PROVIDER = openai;

  // You can also do other initialization here
  console.log('App initialization complete');
}
```

Then, you can use the `generateText` function without specifying the provider in each call.

```typescript filename="app/api/chat/route.ts" {13}
import { generateText } from 'ai';
import { NextRequest } from 'next/server';

export async function GET(request: NextRequest) {
```

```
  const { searchParams } = new URL(request.url);
  const prompt = searchParams.get('prompt');

  if (!prompt) {
    return Response.json({ error: 'Prompt is required' }, { status: 400 });
  }

  const result = await generateText({
    model: 'openai/gpt-5.2',
    prompt,
  });

  return Response.json(result);
}
```

Embedding models

Generate vector embeddings for semantic search, similarity matching, and retrieval-augmented generation (RAG).

Single value

```typescript filename="app/api/embed/route.ts" {5-7}
import { embed } from 'ai';

export async function GET() {
  const result = await embed({
    model: 'openai/text-embedding-3-small',
    value: 'Sunny day at the beach',
  });

  return Response.json(result);
}
```

Multiple values

```typescript filename="app/api/embed/route.ts" {5-7}
import { embedMany } from 'ai';

export async function GET() {
  const result = await embedMany({
    model: 'openai/text-embedding-3-small',
    values: ['Sunny day at the beach', 'Cloudy city skyline'],
  });

  return Response.json(result);
}
```

Gateway provider instance

Alternatively, if you're using the Gateway provider instance, specify embedding models with `gateway.textEmbeddingModel(...)`.

```typescript filename="app/api/embed/route.ts" {2,6}
import { embed } from 'ai';
import { gateway } from '@ai-sdk/gateway';

export async function GET() {
  const result = await embed({
    model: gateway.textEmbeddingModel('openai/text-embedding-3-small'),
    value: 'Sunny day at the beach',
  });

  return Response.json(result);
}
```

Dynamic model discovery

You can programmatically discover all available models and their pricing through the AI SDK or REST API.

Using AI SDK

The `getAvailableModels` function retrieves detailed information about
all models configured for the `gateway` provider, including each model's `id`, `name`, `description`, and `pricing` details.

```typescript filename="app/api/chat/route.ts" {4}
import { gateway } from '@ai-sdk/gateway';
import { generateText } from 'ai';

const availableModels = await gateway.getAvailableModels();

availableModels.models.forEach((model) => {
  console.log(`${model.id}: ${model.name}`);
  if (model.description) {
    console.log(`  Description: ${model.description}`);
  }
  if (model.pricing) {
    console.log(`  Input: $${model.pricing.input}/token`);
    console.log(`  Output: $${model.pricing.output}/token`);

    // Some models have tiered pricing based on context size
    if (model.pricing.inputTiers) {
      console.log('  Input tiers:');
      model.pricing.inputTiers.forEach((tier) => {
        const range =
          tier.max !== undefined ? `${tier.min}-${tier.max}` : `${tier.min}+`;
        console.log(`    ${range} tokens: $${tier.cost}/token`);
      });
```

```
    }

    if (model.pricing.cachedInputTokens) {
      console.log(
        `  Cached input (read): $${model.pricing.cachedInputTokens}/token`,
      );
    }
    if (model.pricing.cacheCreationInputTokens) {
      console.log(
        `  Cache creation (write): $${model.pricing.cacheCreationInputTokens}/token`,
      );
    }
  }
});

const { text } = await generateText({
  model: availableModels.models[0].id, // e.g., 'openai/gpt-5.2'
  prompt: 'Hello world',
});
```

Using REST API

You can also query the models endpoint directly via REST. This endpoint follows the OpenAI models API format and requires no authenticati

```
GET /v1/models
```

```typescript filename="discover-models.ts"
const response = await fetch('https://ai-gateway.vercel.sh/v1/models');
const { data: models } = await response.json();

models.forEach((model) => {
  console.log(`${model.id}: ${model.name}`);
  console.log(`  Type: ${model.type}`);
  console.log(`  Context window: ${model.context_window} tokens`);
  console.log(`  Max output: ${model.max_tokens} tokens`);
  if (model.pricing) {
    if (model.pricing.input) {
      console.log(`  Input: $${model.pricing.input}/token`);
    }
    if (model.pricing.output) {
      console.log(`  Output: $${model.pricing.output}/token`);
    }

    // Some models have tiered pricing based on context size
    if (model.pricing.input_tiers) {
      console.log('  Input tiers:');
      model.pricing.input_tiers.forEach((tier) => {
        const range =
          tier.max !== undefined ? `${tier.min}-${tier.max}` : `${tier.min}+`;
        console.log(`    ${range} tokens: $${tier.cost}/token`);
      });
    }

    if (model.pricing.image) {
      console.log(`  Per image: $${model.pricing.image}`);
    }
  }
});
```

Response format

```json
{
  "object": "list",
  "data": [
    {
      "id": "google/gemini-3-pro",
      "object": "model",
      "created": 1755815280,
      "owned_by": "google",
      "name": "Gemini 3 Pro",
      "description": "This model improves upon Gemini 2.5 Pro and is catered towards challenging tasks, especially those involving comple
      "context_window": 1000000,
      "max_tokens": 64000,
      "type": "language",
      "tags": ["file-input", "tool-use", "reasoning", "vision"],
      "pricing": {
        "input": "0.000002",
        "input_tiers": [
          { "cost": "0.000002", "min": 0, "max": 200001 },
          { "cost": "0.000004", "min": 200001 }
        ],
        "output": "0.000012",
        "output_tiers": [
          { "cost": "0.000012", "min": 0, "max": 200001 },
          { "cost": "0.000018", "min": 200001 }
        ],
        "input_cache_read": "0.0000002",
        "input_cache_read_tiers": [
          { "cost": "0.0000002", "min": 0, "max": 200001 },
          { "cost": "0.0000004", "min": 200001 }
        ],
        "input_cache_write": "0.000002",
        "input_cache_write_tiers": [
          { "cost": "0.000002", "min": 0, "max": 200001 },
          { "cost": "0.000004", "min": 200001 }
```

```
        ]
      }
    }
  ]
}
```

Response fields

Field	Type	Description
`object`	string	Always `"list"`
`data`	array	Array of available models
`data[].id`	string	Model identifier (e.g., `openai/gpt-5.2`)
`data[].object`	string	Always `"model"`
`data[].created`	integer	Unix timestamp when the model was added
`data[].owned_by`	string	Model provider/owner
`data[].name`	string	Human-readable model name
`data[].description`	string	Model description
`data[].context_window`	integer	Maximum context length in tokens
`data[].max_tokens`	integer	Maximum output tokens
`data[].type`	string	Model type: `language`, `embedding`, or `image`
`data[].tags`	string\[]	Capability tags (e.g., `reasoning`, `tool-use`, `vision`)
`data[].pricing`	object	Pricing information (structure varies by model type)
`data[].pricing.input`	string	Base cost per input token (language and embedding models)
`data[].pricing.input_tiers`	array	Tiered pricing for input tokens based on token count
`data[].pricing.input_tiers[].cost`	string	Cost per token for this tier
`data[].pricing.input_tiers[].min`	integer	Minimum token count for this tier (inclusive)
`data[].pricing.input_tiers[].max`	integer	Maximum token count for this tier (exclusive, omitted if none)
`data[].pricing.output`	string	Base cost per output token (language models only)
`data[].pricing.output_tiers`	array	Tiered pricing for output tokens based on token count
`data[].pricing.output_tiers[].cost`	string	Cost per token for this tier
`data[].pricing.output_tiers[].min`	integer	Minimum token count for this tier (inclusive)
`data[].pricing.output_tiers[].max`	integer	Maximum token count for this tier (exclusive, omitted if none)
`data[].pricing.input_cache_read`	string	Base cost per cached input token when reading from cache
`data[].pricing.input_cache_read_tiers`	array	Tiered pricing for cache reads based on token count
`data[].pricing.input_cache_write`	string	Base cost per input token when writing to cache
`data[].pricing.input_cache_write_tiers`	array	Tiered pricing for cache writes based on token count
`data[].pricing.image`	string	Cost per generated image (image models only)
`data[].pricing.web_search`	string	Cost per web search request

Get provider endpoints for a model

For models available through multiple providers, you can query for all available provider endpoints. This returns detailed pricing and ca

```
GET /v1/models/{creator}/{model}/endpoints
```

```typescript filename="endpoints.ts"
const response = await fetch(
  'https://ai-gateway.vercel.sh/v1/models/google/gemini-3-pro/endpoints',
);
const { data } = await response.json();

console.log(`Model: ${data.name}`);
console.log(`Modality: ${data.architecture.modality}`);
console.log(`Input Modalities: ${data.architecture.input_modalities.join(', ')}`);
console.log(`Output Modalities: ${data.architecture.output_modalities.join(', ')}`);
console.log(`\nAvailable from ${data.endpoints.length} provider(s):`);

data.endpoints.forEach((endpoint) => {
  console.log(`\n  ${endpoint.provider_name}:`);
  console.log(`    Context: ${endpoint.context_length} tokens`);
  console.log(`    Prompt: $${endpoint.pricing.prompt}/token`);
  console.log(`    Completion: $${endpoint.pricing.completion}/token`);
  console.log(`    Parameters: ${endpoint.supported_parameters.join(', ')}`);

  if (endpoint.pricing.prompt_tiers) {
    console.log('    Prompt tiers:');
    endpoint.pricing.prompt_tiers.forEach((tier) => {
      const range =
        tier.max !== undefined ? `${tier.min}-${tier.max}` : `${tier.min}+`;
      console.log(`      ${range} tokens: $${tier.cost}/token`);
    });
  }
});
```

Response format

```json
{
  "data": {
    "id": "google/gemini-3-pro",
    "name": "Gemini 3 Pro",
    "created": 1755815280,
    "description": "This model improves upon Gemini 2.5 Pro and is catered towards challenging tasks, especially those involving complex
    "architecture": {
      "tokenizer": null,
      "instruct_type": null,
      "modality": "text+image+file→text",
      "input_modalities": ["text", "image", "file"],
      "output_modalities": ["text"]
    },
    "endpoints": [
      {
        "name": "google | google/gemini-3-pro",
        "model_name": "Gemini 3 Pro",
        "context_length": 1000000,
```

```
        "pricing": {
          "prompt": "0.000002",
          "prompt_tiers": [
            { "cost": "0.000002", "min": 0, "max": 200001 },
            { "cost": "0.000004", "min": 200001 }
          ],
          "completion": "0.000012",
          "completion_tiers": [
            { "cost": "0.000012", "min": 0, "max": 200001 },
            { "cost": "0.000018", "min": 200001 }
          ],
          "request": "0",
          "image": "0",
          "image_output": "0",
          "web_search": "0",
          "internal_reasoning": "0",
          "input_cache_read": "0.0000002",
          "input_cache_read_tiers": [
            { "cost": "0.0000002", "min": 0, "max": 200001 },
            { "cost": "0.0000004", "min": 200001 }
          ],
          "input_cache_write": "0.000002",
          "input_cache_write_tiers": [
            { "cost": "0.000002", "min": 0, "max": 200001 },
            { "cost": "0.000004", "min": 200001 }
          ],
          "discount": 0
        },
        "provider_name": "google",
        "tag": "google",
        "quantization": null,
        "max_completion_tokens": 64000,
        "max_prompt_tokens": null,
        "supported_parameters": ["max_tokens", "temperature", "stop", "tools", "tool_choice", "reasoning", "include_reasoning"],
        "status": 0,
        "uptime_last_30m": null,
        "supports_implicit_caching": false
      }
    ]
  }
}
```

Response fields

Field	Type	Description
`data.id`	string	Model identifier (e.g., `google/gemini-3-pro`)
`data.name`	string	Human-readable model name
`data.created`	integer	Unix timestamp when the model was added
`data.description`	string	Model description
`data.architecture`	object	Model architecture details
`data.architecture.modality`	string	Input/output modality string (e.g., `text+image→text`)
`data.architecture.input_modalities`	string\[]	Supported input types (`text`, `image`, `file`)
`data.architecture.output_modalities`	string\[]	Supported output types (`text`, `image`)
`data.endpoints`	array	Array of provider endpoints
`data.endpoints[].name`	string	Endpoint name (e.g., `google | google/gemini-3-pro`)
`data.endpoints[].provider_name`	string	Provider name (e.g., `google`, `anthropic`)
`data.endpoints[].context_length`	integer	Maximum context window in tokens
`data.endpoints[].max_completion_tokens`	integer	Maximum output tokens
`data.endpoints[].pricing.prompt`	string	Cost per prompt token
`data.endpoints[].pricing.prompt_tiers`	array	Tiered pricing for prompt tokens (if applicable)
`data.endpoints[].pricing.completion`	string	Cost per completion token
`data.endpoints[].pricing.completion_tiers`	array	Tiered pricing for completion tokens (if applicable)
`data.endpoints[].pricing.input_cache_read`	string	Cost per cached input token (read)
`data.endpoints[].pricing.input_cache_read_tiers`	array	Tiered pricing for cache reads (if applicable)
`data.endpoints[].pricing.input_cache_write`	string	Cost per input token (cache write)
`data.endpoints[].pricing.input_cache_write_tiers`	array	Tiered pricing for cache writes (if applicable)
`data.endpoints[].supported_parameters`	string\[]	API parameters supported by this endpoint
`data.endpoints[].supports_implicit_caching`	boolean	Whether provider supports automatic caching
`data.endpoints[].status`	integer	Endpoint status: `0` = active

Tiered pricing

Some models have tiered pricing based on context size. When tiered pricing is available, the `*_tiers` arrays contain pricing tiers with:

Field	Type	Description
`cost`	string	Cost per token for this tier
`min`	number	Minimum token count (inclusive)
`max`	number	Maximum token count (exclusive), omitted for the highest tier

For example, a model with tiered prompt pricing might charge `$0.000002/token` for prompts up to 200K tokens, and `$0.000004/token` for p

Filtering models by type

You can filter the available models by their type to separate language models, embedding models, and image models:

```typescript filename="app/api/models/route.ts"
// Using AI SDK
import { gateway } from '@ai-sdk/gateway';

const { models } = await gateway.getAvailableModels();
const textModels = models.filter((m) => m.modelType === 'language');
const embeddingModels = models.filter((m) => m.modelType === 'embedding');
const imageModels = models.filter((m) => m.modelType === 'image');
```

```typescript filename="filter-models-rest.ts"
// Using REST API
```

```
const response = await fetch('https://ai-gateway.vercel.sh/v1/models');
const { data: models } = await response.json();

const textModels = models.filter((m) => m.type === 'language');
const embeddingModels = models.filter((m) => m.type === 'embedding');
const imageModels = models.filter((m) => m.type === 'image');
```

---
title: "Observability"
description: "Learn how to monitor and debug your AI Gateway requests."
last_updated: "2026-01-16T02:19:25.296Z"
source: "https://vercel.com/docs/ai-gateway/observability"

Observability

The AI Gateway logs observability metrics related to your requests, which you can use to monitor and debug.

You can view these [metrics](#metrics):

- [The **Observability** tab in your Vercel dashboard](#observability-tab)
- [The **AI Gateway** tab in your Vercel dashboard](#ai-gateway-tab)

Observability tab

You can access these metrics from the **Observability** tab of your Vercel dashboard by clicking **AI Gateway** on the left side of the *

Team scope

When you access the **AI Gateway** section of the **Observability** tab under the [team scope](/docs/dashboard-features#scope-selector),

Project scope

When you access the **AI Gateway** section of the **Observability** tab for a specific project, you can view metrics for all requests to

AI Gateway tab

You can also access these metrics by clicking the **AI Gateway** tab of your Vercel dashboard under the team scope. You can see a recent

Metrics

Requests by Model

The **Requests by Model** chart shows the number of requests made to each model over time. This can help you identify which models are be

Time to First Token (TTFT)

The **Time to First Token** chart shows the average time it takes for the AI Gateway to return the first token of a response. This can he

Input/output Token Counts

The **Input/output Token Counts** chart shows the number of input and output tokens for each request. This can help you understand the si

Spend

The **Spend** chart shows the total amount spent on AI Gateway requests over time. This can help you monitor your spending and identify a

---
title: "Advanced Configuration"
description: "Configure reasoning, provider options, model fallbacks, BYOK credentials, prompt caching, and extended context windows."
last_updated: "2026-01-16T02:19:25.475Z"
source: "https://vercel.com/docs/ai-gateway/openai-compat/advanced"

Advanced Configuration

Reasoning configuration

Configure reasoning behavior for models that support extended thinking or chain-of-thought reasoning. The `reasoning` parameter allows yo

Example request

TypeScript

```typescript filename="reasoning-openai-sdk.ts"
import OpenAI from 'openai';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const openai = new OpenAI({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

// @ts-expect-error - reasoning parameter not yet in OpenAI types
const completion = await openai.chat.completions.create({
  model: 'anthropic/claude-sonnet-4.5',
  messages: [
    {
      role: 'user',
      content: 'What is the meaning of life? Think before answering.',
    },
  ],
  stream: false,
  reasoning: {
    max_tokens: 2000, // Limit reasoning tokens
    enabled: true, // Enable reasoning
```

```
  },
});

console.log('Reasoning:', completion.choices[0].message.reasoning);
console.log('Answer:', completion.choices[0].message.content);
console.log(
  'Reasoning tokens:',
  completion.usage.completion_tokens_details?.reasoning_tokens,
);
```

Python

```python filename="reasoning.py"
import os
from openai import OpenAI

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = OpenAI(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh/v1'
)

completion = client.chat.completions.create(
    model='anthropic/claude-sonnet-4.5',
    messages=[
        {
            'role': 'user',
            'content': 'What is the meaning of life? Think before answering.'
        }
    ],
    stream=False,
    extra_body={
        'reasoning': {
            'max_tokens': 2000,
            'enabled': True
        }
    }
)

print('Reasoning:', completion.choices[0].message.reasoning)
print('Answer:', completion.choices[0].message.content)
print('Reasoning tokens:', completion.usage.completion_tokens_details.reasoning_tokens)
```

Reasoning parameters

The `reasoning` object supports the following parameters:

- **`enabled`** (boolean, optional): Enable reasoning output. When `true`, the model will provide its reasoning process.

- **`max_tokens`** (number, optional): Maximum number of tokens to allocate for reasoning. This helps control costs and response times. C

- **`effort`** (string, optional): Control reasoning effort level. Accepts:

  - `'none'` - Disables reasoning
  - `'minimal'` - ~10% of max_tokens
  - `'low'` - ~20% of max_tokens
  - `'medium'` - ~50% of max_tokens
  - `'high'` - ~80% of max_tokens
  - `'xhigh'` - ~95% of max_tokens

  Cannot be used with `max_tokens`.

- **`exclude`** (boolean, optional): When `true`, excludes reasoning content from the response but still generates it internally. Useful

> **💡 Note:** **Mutually exclusive parameters:** You cannot specify both `effort` and
> `max_tokens` in the same request. Choose one based on your use case.

Response format with reasoning

When reasoning is enabled, the response includes reasoning content:

```json
{
  "id": "chatcmpl-123",
  "object": "chat.completion",
  "created": 1677652288,
  "model": "anthropic/claude-sonnet-4.5",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "The meaning of life is a deeply personal question...",
        "reasoning": "Let me think about this carefully. The question asks about..."
      },
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 15,
    "completion_tokens": 150,
    "total_tokens": 165,
    "completion_tokens_details": {
      "reasoning_tokens": 50
    }
  }
}
```

```

#### Streaming with reasoning

Reasoning content is streamed incrementally in the `delta.reasoning` field:

#### TypeScript

```typescript filename="reasoning-streaming.ts"
import OpenAI from 'openai';

const openai = new OpenAI({
  apiKey: process.env.AI_GATEWAY_API_KEY,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

// @ts-expect-error - reasoning parameter not yet in OpenAI types
const stream = await openai.chat.completions.create({
  model: 'anthropic/claude-sonnet-4.5',
  messages: [
    {
      role: 'user',
      content: 'What is the meaning of life? Think before answering.',
    },
  ],
  stream: true,
  reasoning: {
    enabled: true,
  },
});

for await (const chunk of stream) {
  const delta = chunk.choices[0]?.delta;

  // Handle reasoning content
  if (delta?.reasoning) {
    process.stdout.write(`[Reasoning] ${delta.reasoning}`);
  }

  // Handle regular content
  if (delta?.content) {
    process.stdout.write(delta.content);
  }
}
```

Python

```python filename="reasoning-streaming.py"
import os
from openai import OpenAI

client = OpenAI(
    api_key=os.getenv('AI_GATEWAY_API_KEY'),
    base_url='https://ai-gateway.vercel.sh/v1'
)

stream = client.chat.completions.create(
    model='anthropic/claude-sonnet-4.5',
    messages=[
        {
            'role': 'user',
            'content': 'What is the meaning of life? Think before answering.'
        }
    ],
    stream=True,
    extra_body={
        'reasoning': {
            'enabled': True
        }
    }
)

for chunk in stream:
    if chunk.choices and chunk.choices[0].delta:
        delta = chunk.choices[0].delta

        # Handle reasoning content
        if hasattr(delta, 'reasoning') and delta.reasoning:
            print(f"[Reasoning] {delta.reasoning}", end='', flush=True)

        # Handle regular content
        if hasattr(delta, 'content') and delta.content:
            print(delta.content, end='', flush=True)
```

Preserving reasoning details across providers

The AI Gateway preserves reasoning details from models across interactions,
normalizing the different formats used by OpenAI, Anthropic, and other providers into a consistent structure.
This allows you to switch between models without rewriting your conversation management logic.

This is particularly useful during tool calling workflows where the model needs to
resume its thought process after receiving tool results.

Controlling reasoning details

When `reasoning.enabled` is `true` (or when `reasoning.exclude` is not set),
responses include a `reasoning_details` array alongside the standard `reasoning` text field.
This structured field captures cryptographic signatures, encrypted content, and other verification
```

data that providers include with their reasoning output.

Each detail object contains:

- **`type`**: one or more of the below, depending on the provider and model
  - `'reasoning.text'`: Contains the actual reasoning content as plain text in the `text` field. May include a `signature` field (Anthrop
  - `'reasoning.encrypted'`: Contains encrypted or redacted reasoning content in the `data` field. Used by OpenAI models when reasoning i
  - `'reasoning.summary'`: Contains a condensed version of the reasoning process in the `summary` field. Used by OpenAI models to provide
- **`id`** (optional): Unique identifier for the reasoning block, used for tracking and correlation
- **`format`**: Provider format identifier - `'openai-responses-v1'`, `'anthropic-claude-v1'`, or `'unknown'`
- **`index`** (optional): Position in the reasoning sequence (for responses with multiple reasoning blocks)

Example response with reasoning details

For Anthropic models:

```json
{
  "id": "chatcmpl-123",
  "object": "chat.completion",
  "created": 1677652288,
  "model": "anthropic/claude-sonnet-4.5",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "The meaning of life is a deeply personal question...",
        "reasoning": "Let me think about this carefully. The question asks about...",
        "reasoning_details": [
          {
            "type": "reasoning.text",
            "text": "Let me think about this carefully. The question asks about...",
            "signature": "anthropic-signature-xyz",
            "format": "anthropic-claude-v1",
            "index": 0
          }
        ]
      },
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 15,
    "completion_tokens": 150,
    "total_tokens": 165,
    "completion_tokens_details": {
      "reasoning_tokens": 50
    }
  }
}
```

For OpenAI models (returns both summary and encrypted):

```json
{
  "id": "chatcmpl-456",
  "object": "chat.completion",
  "created": 1677652288,
  "model": "openai/o3-mini",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "The answer is 42.",
        "reasoning": "Let me calculate this step by step...",
        "reasoning_details": [
          {
            "type": "reasoning.summary",
            "summary": "Let me calculate this step by step...",
            "format": "openai-responses-v1",
            "index": 0
          },
          {
            "type": "reasoning.encrypted",
            "data": "encrypted_reasoning_content_xyz",
            "format": "openai-responses-v1",
            "index": 1
          }
        ]
      },
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 15,
    "completion_tokens": 150,
    "total_tokens": 165,
    "completion_tokens_details": {
      "reasoning_tokens": 50
    }
  }
}
```

Streaming reasoning details

When streaming, reasoning details are delivered incrementally in `delta.reasoning_details`:

For Anthropic models:

```json
{
  "id": "chatcmpl-123",
  "object": "chat.completion.chunk",
  "created": 1677652288,
  "model": "anthropic/claude-sonnet-4.5",
  "choices": [
    {
      "index": 0,
      "delta": {
        "reasoning": "Let me think.",
        "reasoning_details": [
          {
            "type": "reasoning.text",
            "text": "Let me think.",
            "signature": "anthropic-signature-xyz",
            "format": "anthropic-claude-v1",
            "index": 0
          }
        ]
      },
      "finish_reason": null
    }
  ]
}
```

For OpenAI models (summary chunks during reasoning, then encrypted at end):

```json
{
  "id": "chatcmpl-456",
  "object": "chat.completion.chunk",
  "created": 1677652288,
  "model": "openai/o3-mini",
  "choices": [
    {
      "index": 0,
      "delta": {
        "reasoning": "Step 1:",
        "reasoning_details": [
          {
            "type": "reasoning.summary",
            "summary": "Step 1:",
            "format": "openai-responses-v1",
            "index": 0
          }
        ]
      },
      "finish_reason": null
    }
  ]
}
```

Provider-specific behavior

The AI Gateway automatically maps reasoning parameters to each provider's native format:

- **OpenAI**: Maps `effort` to `reasoningEffort` and controls summary detail
- **Anthropic**: Maps `max_tokens` to thinking budget tokens
- **Google**: Maps to `thinkingConfig` with budget and visibility settings
- **Groq**: Maps `exclude` to control reasoning format (hidden/parsed)
- **xAI**: Maps `effort` to reasoning effort levels
- **Other providers**: Generic mapping applied for compatibility

> **💡 Note:** **Automatic extraction:** For models that don't natively support reasoning
> output, the gateway automatically extracts reasoning
> from `<think>` tags in the response.

Provider options

The AI Gateway can route your requests across multiple AI providers for better reliability and performance. You can control which provide

Example request

TypeScript

```typescript filename="provider-options.ts"
import OpenAI from 'openai';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const openai = new OpenAI({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

// @ts-expect-error
const completion = await openai.chat.completions.create({
  model: 'anthropic/claude-sonnet-4.5',
  messages: [
    {
      role: 'user',
      content:
        'Tell me the history of the San Francisco Mission-style burrito in two paragraphs.',
    },
```

```
  ],
  stream: false,
  // Provider options for gateway routing preferences
  providerOptions: {
    gateway: {
      order: ['vertex', 'anthropic'], // Try Vertex AI first, then Anthropic
    },
  },
});

console.log('Assistant:', completion.choices[0].message.content);
console.log('Tokens used:', completion.usage);
```

Python

```python filename="provider-options.py"
import os
from openai import OpenAI

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = OpenAI(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh/v1'
)

completion = client.chat.completions.create(
    model='anthropic/claude-sonnet-4.5',
    messages=[
        {
            'role': 'user',
            'content': 'Tell me the history of the San Francisco Mission-style burrito in two paragraphs.'
        }
    ],
    stream=False,
    # Provider options for gateway routing preferences
    extra_body={
        'providerOptions': {
            'gateway': {
                'order': ['vertex', 'anthropic']  # Try Vertex AI first, then Anthropic
            }
        }
    }
)

print('Assistant:', completion.choices[0].message.content)
print('Tokens used:', completion.usage)
```

> **💡 Note:** **Provider routing:** In this example, the gateway will first attempt to use
> Vertex AI to serve the Claude model. If Vertex AI is unavailable or fails, it
> will fall back to Anthropic. Other providers are still available but will only
> be used after the specified providers.

Model fallbacks

You can specify fallback models that will be tried in order if the primary model fails. There are two ways to do this:

Option 1: Direct `models` field

The simplest way is to use the `models` field directly at the top level of your request:

TypeScript

```typescript filename="model-fallbacks.ts"
import OpenAI from 'openai';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const openai = new OpenAI({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

const completion = await openai.chat.completions.create({
  model: 'openai/gpt-5.2', // Primary model
  // @ts-ignore - models is a gateway extension
  models: ['anthropic/claude-sonnet-4.5', 'google/gemini-3-pro'], // Fallback models
  messages: [
    {
      role: 'user',
      content: 'Write a haiku about TypeScript.',
    },
  ],
  stream: false,
});

console.log('Assistant:', completion.choices[0].message.content);

// Check which model was actually used
console.log('Model used:', completion.model);
```

Python

```python filename="model-fallbacks.py"
import os
from openai import OpenAI
```

```
api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = OpenAI(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh/v1'
)

completion = client.chat.completions.create(
    model='openai/gpt-5.2',  # Primary model
    messages=[
        {
            'role': 'user',
            'content': 'Write a haiku about TypeScript.'
        }
    ],
    stream=False,
    # models is a gateway extension for fallback models
    extra_body={
        'models': ['anthropic/claude-sonnet-4.5', 'google/gemini-3-pro']  # Fallback models
    }
)

print('Assistant:', completion.choices[0].message.content)

# Check which model was actually used
print('Model used:', completion.model)
```

Option 2: Via provider options

Alternatively, you can specify model fallbacks through the `providerOptions.gateway.models` field:

TypeScript

```typescript filename="model-fallbacks-provider-options.ts"
import OpenAI from 'openai';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const openai = new OpenAI({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

// @ts-expect-error
const completion = await openai.chat.completions.create({
  model: 'openai/gpt-5.2', // Primary model
  messages: [
    {
      role: 'user',
      content: 'Write a haiku about TypeScript.',
    },
  ],
  stream: false,
  // Model fallbacks via provider options
  providerOptions: {
    gateway: {
      models: ['anthropic/claude-sonnet-4.5', 'google/gemini-3-pro'], // Fallback models
    },
  },
});

console.log('Assistant:', completion.choices[0].message.content);
console.log('Model used:', completion.model);
```

Python

```python filename="model-fallbacks-provider-options.py"
import os
from openai import OpenAI

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = OpenAI(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh/v1'
)

completion = client.chat.completions.create(
    model='openai/gpt-5.2',  # Primary model
    messages=[
        {
            'role': 'user',
            'content': 'Write a haiku about TypeScript.'
        }
    ],
    stream=False,
    # Model fallbacks via provider options
    extra_body={
        'providerOptions': {
            'gateway': {
                'models': ['anthropic/claude-sonnet-4.5', 'google/gemini-3-pro']  # Fallback models
            }
        }
    }
)

print('Assistant:', completion.choices[0].message.content)
print('Model used:', completion.model)
```

```

> **💡 Note:** **Which approach to use:** Both methods achieve the same result. Use the
> direct `models` field (Option 1) for simplicity, or use `providerOptions`
> (Option 2) if you're already using provider options for other configurations.

Both configurations will:

1. Try the primary model (`openai/gpt-4o`) first
2. If it fails, try `openai/gpt-5-nano`
3. If that also fails, try `gemini-2.0-flash`
4. Return the result from the first model that succeeds

#### Streaming with provider options

Provider options work with streaming requests as well:

#### TypeScript

```typescript filename="streaming-provider-options.ts"
import OpenAI from 'openai';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const openai = new OpenAI({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

// @ts-expect-error
const stream = await openai.chat.completions.create({
  model: 'anthropic/claude-sonnet-4.5',
  messages: [
    {
      role: 'user',
      content:
        'Tell me the history of the San Francisco Mission-style burrito in two paragraphs.',
    },
  ],
  stream: true,
  providerOptions: {
    gateway: {
      order: ['vertex', 'anthropic'],
    },
  },
});

for await (const chunk of stream) {
  const content = chunk.choices[0]?.delta?.content;
  if (content) {
    process.stdout.write(content);
  }
}
```

Python

```python filename="streaming-provider-options.py"
import os
from openai import OpenAI

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = OpenAI(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh/v1'
)

stream = client.chat.completions.create(
    model='anthropic/claude-sonnet-4.5',
    messages=[
        {
            'role': 'user',
            'content': 'Tell me the history of the San Francisco Mission-style burrito in two paragraphs.'
        }
    ],
    stream=True,
    extra_body={
        'providerOptions': {
            'gateway': {
                'order': ['vertex', 'anthropic']
            }
        }
    }
)

for chunk in stream:
    content = chunk.choices[0].delta.content
    if content:
        print(content, end='', flush=True)
```

For more details about available providers and advanced provider configuration, see the [Provider Options documentation](/docs/ai-gateway

Request-scoped BYOK (Bring Your Own Key)

You can pass your own provider credentials on a per-request basis using the `byok` option in `providerOptions.gateway`. This allows you t

Example request
```

TypeScript

```typescript filename="byok.ts"
import OpenAI from 'openai';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const openai = new OpenAI({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

// @ts-expect-error - byok is a gateway extension
const completion = await openai.chat.completions.create({
  model: 'anthropic/claude-sonnet-4.5',
  messages: [
    {
      role: 'user',
      content: 'Hello, world!',
    },
  ],
  providerOptions: {
    gateway: {
      byok: {
        anthropic: [{ apiKey: process.env.ANTHROPIC_API_KEY }],
      },
    },
  },
});

console.log(completion.choices[0].message.content);
```

Python

```python filename="byok.py"
import os
from openai import OpenAI

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = OpenAI(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh/v1'
)

completion = client.chat.completions.create(
    model='anthropic/claude-sonnet-4.5',
    messages=[
        {
            'role': 'user',
            'content': 'Hello, world!'
        }
    ],
    extra_body={
        'providerOptions': {
            'gateway': {
                'byok': {
                    'anthropic': [{'apiKey': os.getenv('ANTHROPIC_API_KEY')}]
                }
            }
        }
    }
)

print(completion.choices[0].message.content)
```

The `byok` option is a record where keys are provider slugs and values are arrays of credential objects. Each provider can have multiple

Credential structure by provider:

- **Anthropic**: `{ apiKey: string }`
- **OpenAI**: `{ apiKey: string }`
- **Google Vertex AI**: `{ project: string, location: string, googleCredentials: { privateKey: string, clientEmail: string } }`
- **Amazon Bedrock**: `{ accessKeyId: string, secretAccessKey: string, region?: string }`

For detailed credential parameters for each provider, see the [AI SDK providers documentation](https://ai-sdk.dev/providers/ai-sdk-provid

Multiple credentials example:

```typescript
providerOptions: {
  gateway: {
    byok: {
      // Multiple credentials for the same provider (tried in order)
      vertex: [
        { project: 'proj-1', location: 'us-east5', googleCredentials: { privateKey: '...', clientEmail: '...' } },
        { project: 'proj-2', location: 'us-east5', googleCredentials: { privateKey: '...', clientEmail: '...' } },
      ],
      // Multiple providers
      anthropic: [{ apiKey: 'sk-ant-...' }],
    },
  },
},
```

> **💡 Note:** **Credential precedence:** When request-scoped BYOK credentials are provided,
> any cached BYOK credentials configured in the gateway settings are not
> considered. Requests may still fall back to system credentials if the provided

> credentials fail. For persistent BYOK configuration, see the [BYOK
> documentation](/docs/ai-gateway/byok).

Prompt caching

Anthropic Claude models support prompt caching, which can significantly reduce costs and latency for repeated prompts. When you mark cont

Example request

TypeScript

```typescript filename="prompt-caching.ts"
import OpenAI from 'openai';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const openai = new OpenAI({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

const response = await openai.chat.completions.create({
  model: 'anthropic/claude-sonnet-4.5',
  messages: [
    {
      role: 'user',
      content: 'Analyze this document and summarize the key points.',
      cache_control: {
        type: 'ephemeral',
      },
    },
  ],
});

console.log(response.choices[0].message.content);
```

Python

```python filename="prompt-caching.py"
import os
from openai import OpenAI

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = OpenAI(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh/v1'
)

response = client.chat.completions.create(
    model='anthropic/claude-sonnet-4.5',
    messages=[
        {
            'role': 'user',
            'content': 'Analyze this document and summarize the key points.',
            'cache_control': {
                'type': 'ephemeral'
            }
        }
    ]
)

print(response.choices[0].message.content)
```

> **💡 Note:** **Cache control types:** The `ephemeral` cache type stores content for the
> duration of the session. This is useful for large system prompts, documents,
> or context that you want to reuse across multiple requests. Prompt caching
> works with Anthropic models across all supported providers (Anthropic, Vertex
> AI, and Bedrock). For more details, see [Anthropic's prompt caching
> documentation](https://platform.claude.com/docs/en/build-with-claude/prompt-caching).

Extended context window (1M tokens)

Anthropic Claude models support an extended context window of up to 1 million tokens for processing very large documents or conversations

Example request

TypeScript

```typescript filename="extended-context.ts"
import OpenAI from 'openai';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const openai = new OpenAI({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

const response = await openai.chat.completions.create(
  {
    model: 'anthropic/claude-sonnet-4.5',
    messages: [
      {
        role: 'user',
        content: 'Your very long prompt here...',
      },
    ],
```

```
  },
  {
    headers: {
      'anthropic-beta': 'context-1m-2025-08-07',
    },
  },
);

console.log(response.choices[0].message.content);
```

Python

```python filename="extended-context.py"
import os
from openai import OpenAI

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = OpenAI(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh/v1'
)

response = client.chat.completions.create(
    model='anthropic/claude-sonnet-4.5',
    messages=[
        {
            'role': 'user',
            'content': 'Your very long prompt here...'
        }
    ],
    extra_headers={
        'anthropic-beta': 'context-1m-2025-08-07'
    }
)

print(response.choices[0].message.content)
```

> **💡 Note:** **When to use extended context:** The 1M context window is useful when working
> with very large documents, extensive codebases, or long conversation histories
> that exceed the standard 200K token limit. Note that longer contexts may
> increase latency and costs. For more details, see [Anthropic's context window
> documentation](https://platform.claude.com/docs/en/build-with-claude/context-windows#1-m-token-context-window).

---
title: "Chat Completions"
description: "Create chat completions using the OpenAI-compatible API with support for streaming, image attachments, and PDF documents."
last_updated: "2026-01-16T02:19:25.341Z"
source: "https://vercel.com/docs/ai-gateway/openai-compat/chat-completions"

Chat Completions

Create chat completions using various AI models available through the AI Gateway.

Endpoint

```
POST /chat/completions
```

Basic chat completion

Create a non-streaming chat completion.

Example request

TypeScript

```typescript filename="chat-completion.ts"
import OpenAI from 'openai';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const openai = new OpenAI({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

const completion = await openai.chat.completions.create({
  model: 'anthropic/claude-sonnet-4.5',
  messages: [
    {
      role: 'user',
      content: 'Write a one-sentence bedtime story about a unicorn.',
    },
  ],
  stream: false,
});

console.log('Assistant:', completion.choices[0].message.content);
console.log('Tokens used:', completion.usage);
```

Python

```python filename="chat-completion.py"
```

```
import os
from openai import OpenAI

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = OpenAI(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh/v1'
)

completion = client.chat.completions.create(
    model='anthropic/claude-sonnet-4.5',
    messages=[
        {
            'role': 'user',
            'content': 'Write a one-sentence bedtime story about a unicorn.'
        }
    ],
    stream=False,
)

print('Assistant:', completion.choices[0].message.content)
print('Tokens used:', completion.usage)
```

Response format

```json
{
  "id": "chatcmpl-123",
  "object": "chat.completion",
  "created": 1677652288,
  "model": "anthropic/claude-sonnet-4.5",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "Once upon a time, a gentle unicorn with a shimmering silver mane danced through moonlit clouds, sprinkling stardust d
      },
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 15,
    "completion_tokens": 28,
    "total_tokens": 43
  }
}
```

Streaming chat completion

Create a streaming chat completion that streams tokens as they are generated.

Example request

TypeScript

```typescript filename="streaming-chat.ts"
import OpenAI from 'openai';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const openai = new OpenAI({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

const stream = await openai.chat.completions.create({
  model: 'anthropic/claude-sonnet-4.5',
  messages: [
    {
      role: 'user',
      content: 'Write a one-sentence bedtime story about a unicorn.',
    },
  ],
  stream: true,
});

for await (const chunk of stream) {
  const content = chunk.choices[0]?.delta?.content;
  if (content) {
    process.stdout.write(content);
  }
}
```

Python

```python filename="streaming-chat.py"
import os
from openai import OpenAI

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = OpenAI(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh/v1'
)
```

```
stream = client.chat.completions.create(
    model='anthropic/claude-sonnet-4.5',
    messages=[
        {
            'role': 'user',
            'content': 'Write a one-sentence bedtime story about a unicorn.'
        }
    ],
    stream=True,
)

for chunk in stream:
    content = chunk.choices[0].delta.content
    if content:
        print(content, end='', flush=True)
```

Streaming response format

Streaming responses are sent as [Server-Sent Events (SSE)](https://developer.mozilla.org/en-US/docs/Web/API/Server-sent_events), a web st

The response format follows the OpenAI streaming specification:

```http
data: {"id":"chatcmpl-123","object":"chat.completion.chunk","created":1677652288,"model":"anthropic/claude-sonnet-4.5","choices":[{"index

data: {"id":"chatcmpl-123","object":"chat.completion.chunk","created":1677652288,"model":"anthropic/claude-sonnet-4.5","choices":[{"index

data: [DONE]
```

Key characteristics:

- Each line starts with `data:` followed by JSON
- Content is delivered incrementally in the `delta.content` field
- The stream ends with `data: [DONE]`
- Empty lines separate events

SSE Parsing Libraries:

If you're building custom SSE parsing (instead of using the OpenAI SDK), these libraries can help:

- **JavaScript/TypeScript**: [`eventsource-parser`](https://www.npmjs.com/package/eventsource-parser) - Robust SSE parsing with support f
- **Python**: [`httpx-sse`](https://pypi.org/project/httpx-sse/) - SSE support for HTTPX, or [`sseclient-py`](https://pypi.org/project/ss

For more details about the SSE specification, see the [W3C specification](https://html.spec.whatwg.org/multipage/server-sent-events.html)

Image attachments

Send images as part of your chat completion request.

Example request

TypeScript

```typescript filename="image-analysis.ts"
import fs from 'node:fs';
import OpenAI from 'openai';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const openai = new OpenAI({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

// Read the image file as base64
const imageBuffer = fs.readFileSync('./path/to/image.png');
const imageBase64 = imageBuffer.toString('base64');

const completion = await openai.chat.completions.create({
  model: 'anthropic/claude-sonnet-4.5',
  messages: [
    {
      role: 'user',
      content: [
        { type: 'text', text: 'Describe this image in detail.' },
        {
          type: 'image_url',
          image_url: {
            url: `${imageBase64}`,
            detail: 'auto',
          },
        },
      ],
    },
  ],
  stream: false,
});

console.log('Assistant:', completion.choices[0].message.content);
console.log('Tokens used:', completion.usage);
```

Python

```python filename="image-analysis.py"
import os
import base64
```

```
from openai import OpenAI

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = OpenAI(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh/v1'
)

# Read the image file as base64
with open('./path/to/image.png', 'rb') as image_file:
    image_base64 = base64.b64encode(image_file.read()).decode('utf-8')

completion = client.chat.completions.create(
    model='anthropic/claude-sonnet-4.5',
    messages=[
        {
            'role': 'user',
            'content': [
                {'type': 'text', 'text': 'Describe this image in detail.'},
                {
                    'type': 'image_url',
                    'image_url': {
                        'url': f'{image_base64}',
                        'detail': 'auto'
                    }
                }
            ]
        }
    ],
    stream=False,
)

print('Assistant:', completion.choices[0].message.content)
print('Tokens used:', completion.usage)
```

PDF attachments

Send PDF documents as part of your chat completion request.

Example request

TypeScript

```typescript filename="pdf-analysis.ts"
import fs from 'node:fs';
import OpenAI from 'openai';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const openai = new OpenAI({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

// Read the PDF file as base64
const pdfBuffer = fs.readFileSync('./path/to/document.pdf');
const pdfBase64 = pdfBuffer.toString('base64');

const completion = await openai.chat.completions.create({
  model: 'anthropic/claude-sonnet-4.5',
  messages: [
    {
      role: 'user',
      content: [
        {
          type: 'text',
          text: 'What is the main topic of this document? Please summarize the key points.',
        },
        {
          type: 'file',
          file: {
            data: pdfBase64,
            media_type: 'application/pdf',
            filename: 'document.pdf',
          },
        },
      ],
    },
  ],
  stream: false,
});

console.log('Assistant:', completion.choices[0].message.content);
console.log('Tokens used:', completion.usage);
```

Python

```python filename="pdf-analysis.py"
import os
import base64
from openai import OpenAI

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = OpenAI(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh/v1'
```

```
)

# Read the PDF file as base64
with open('./path/to/document.pdf', 'rb') as pdf_file:
    pdf_base64 = base64.b64encode(pdf_file.read()).decode('utf-8')

completion = client.chat.completions.create(
    model='anthropic/claude-sonnet-4.5',
    messages=[
        {
            'role': 'user',
            'content': [
                {
                    'type': 'text',
                    'text': 'What is the main topic of this document? Please summarize the key points.'
                },
                {
                    'type': 'file',
                    'file': {
                        'data': pdf_base64,
                        'media_type': 'application/pdf',
                        'filename': 'document.pdf'
                    }
                }
            ]
        }
    ],
    stream=False,
)

print('Assistant:', completion.choices[0].message.content)
print('Tokens used:', completion.usage)
```

Parameters

The chat completions endpoint supports the following parameters:

Required parameters

- `model` (string): The model to use for the completion (e.g., `anthropic/claude-sonnet-4`)
- `messages` (array): Array of message objects with `role` and `content` fields

Optional parameters

- `stream` (boolean): Whether to stream the response. Defaults to `false`
- `temperature` (number): Controls randomness in the output. Range: 0-2
- `max_tokens` (integer): Maximum number of tokens to generate
- `top_p` (number): Nucleus sampling parameter. Range: 0-1
- `frequency_penalty` (number): Penalty for frequent tokens. Range: -2 to 2
- `presence_penalty` (number): Penalty for present tokens. Range: -2 to 2
- `stop` (string or array): Stop sequences for the generation
- `tools` (array): Array of tool definitions for function calling
- `tool_choice` (string or object): Controls which tools are called (`auto`, `none`, or specific function)
- `providerOptions` (object): [Provider routing and configuration options](/docs/ai-gateway/openai-compat/advanced#provider-options)
- `response_format` (object): Controls the format of the model's response
  - For OpenAI standard format: `{ type: "json_schema", json_schema: { name, schema, strict?, description? } }`
  - For legacy format: `{ type: "json", schema?, name?, description? }`
  - For plain text: `{ type: "text" }`
  - See [Structured outputs](/docs/ai-gateway/openai-compat/structured-outputs) for detailed examples

Message format

Messages support different content types:

Text messages

```json
{
  "role": "user",
  "content": "Hello, how are you?"
}
```

Multimodal messages

```json
{
  "role": "user",
  "content": [
    { "type": "text", "text": "What's in this image?" },
    {
      "type": "image_url",
      "image_url": {
        "url": "..."
      }
    }
  ]
}
```

File messages

```json
{
  "role": "user",
  "content": [
    { "type": "text", "text": "Summarize this document" },
    {
      "type": "file",
```

```
      "file": {
        "data": "JVBERi0xLjQKJcfsj6IKNSAwIG9iago8PAovVHlwZSAvUGFnZQo...",
        "media_type": "application/pdf",
        "filename": "document.pdf"
      }
    }
  ]
}
```

---
title: "Embeddings"
description: "Generate vector embeddings from input text for semantic search, similarity matching, and RAG applications."
last_updated: "2026-01-16T02:19:25.245Z"
source: "https://vercel.com/docs/ai-gateway/openai-compat/embeddings"

Embeddings

Generate vector embeddings from input text for semantic search, similarity matching, and retrieval-augmented generation (RAG).

Endpoint

```
POST /embeddings
```

Example request

TypeScript

```typescript filename="embeddings.ts"
import OpenAI from 'openai';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const openai = new OpenAI({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

const response = await openai.embeddings.create({
  model: 'openai/text-embedding-3-small',
  input: 'Sunny day at the beach',
});

console.log(response.data[0].embedding);
```

Python

```python filename="embeddings.py"
import os
from openai import OpenAI

api_key = os.getenv("AI_GATEWAY_API_KEY") or os.getenv("VERCEL_OIDC_TOKEN")

client = OpenAI(
    api_key=api_key,
    base_url="https://ai-gateway.vercel.sh/v1",
)

response = client.embeddings.create(
    model="openai/text-embedding-3-small",
    input="Sunny day at the beach",
)

print(response.data[0].embedding)
```

Response format

```json
{
  "object": "list",
  "data": [
    {
      "object": "embedding",
      "index": 0,
      "embedding": [-0.0038, 0.021, ...]
    },
  ],
  "model": "openai/text-embedding-3-small",
  "usage": {
    "prompt_tokens": 6,
    "total_tokens": 6
  },
  "providerMetadata": {
    "gateway": {
      "routing": { ... }, // Detailed routing info
      "cost": "0.00000012"
    }
  }
}
```

Dimensions parameter

You can set the root-level `dimensions` field (from the [OpenAI Embeddings API spec](https://platform.openai.com/docs/api-reference/embed

TypeScript

```typescript filename="embeddings-dimensions.ts"
const response = await openai.embeddings.create({
  model: 'openai/text-embedding-3-small',
  input: 'Sunny day at the beach',
  dimensions: 768,
});
```

Python

```python filename="embeddings-dimensions.py"
response = client.embeddings.create(
    model='openai/text-embedding-3-small',
    input='Sunny day at the beach',
    dimensions=768,
)
```

---
title: "Image Generation"
description: "Generate images using AI models that support multimodal output through the OpenAI-compatible API."
last_updated: "2026-01-16T02:19:25.261Z"
source: "https://vercel.com/docs/ai-gateway/openai-compat/image-generation"

Image Generation

Generate images using AI models that support multimodal output through the OpenAI-compatible API. This feature allows you to create image

Endpoint

```
POST /chat/completions
```

Parameters

To enable image generation, include the `modalities` parameter in your request:

- `modalities` (array): Array of strings specifying the desired output modalities. Use `['text', 'image']` for both text and image genera

Example requests

TypeScript

```typescript filename="image-generation.ts"
import OpenAI from 'openai';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const openai = new OpenAI({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

const completion = await openai.chat.completions.create({
  model: 'google/gemini-2.5-flash-image-preview',
  messages: [
    {
      role: 'user',
      content:
        'Generate a beautiful sunset over mountains and describe the scene.',
    },
  ],
  // @ts-expect-error - modalities not yet in OpenAI types but supported by gateway
  modalities: ['text', 'image'],
  stream: false,
});

const message = completion.choices[0].message;

// Text content is always a string
console.log('Text:', message.content);

// Images are in a separate array
if (message.images && Array.isArray(message.images)) {
  console.log(`Generated ${message.images.length} images:`);
  for (const [index, img] of message.images.entries()) {
    if (img.type === 'image_url' && img.image_url) {
      console.log(`Image ${index + 1}:`, {
        size: img.image_url.url?.length || 0,
        preview: `${img.image_url.url?.substring(0, 50)}...`,
      });
    }
  }
}
```

Python

```python filename="image-generation.py"
import os
from openai import OpenAI

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')
```

```
client = OpenAI(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh/v1'
)

completion = client.chat.completions.create(
    model='google/gemini-2.5-flash-image-preview',
    messages=[
        {
            'role': 'user',
            'content': 'Generate a beautiful sunset over mountains and describe the scene.'
        }
    ],
    # Note: modalities parameter is not yet in OpenAI Python types but supported by our gateway
    extra_body={'modalities': ['text', 'image']},
    stream=False,
)

message = completion.choices[0].message

# Text content is always a string
print(f"Text: {message.content}")

# Images are in a separate array
if hasattr(message, 'images') and message.images:
    print(f"Generated {len(message.images)} images:")
    for i, img in enumerate(message.images):
        if img.get('type') == 'image_url' and img.get('image_url'):
            image_url = img['image_url']['url']
            data_size = len(image_url) if image_url else 0
            print(f"Image {i+1}: size: {data_size} chars")
            print(f"Preview: {image_url[:50]}...")

print(f'Tokens used: {completion.usage}')
```

Response format

When image generation is enabled, the response separates text content from generated images:

```json
{
  "id": "chatcmpl-123",
  "object": "chat.completion",
  "created": 1677652288,
  "model": "google/gemini-2.5-flash-image-preview",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "Here's a beautiful sunset scene over the mountains...",
        "images": [
          {
            "type": "image_url",
            "image_url": {
              "url": "
            }
          }
        ]
      },
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 15,
    "completion_tokens": 28,
    "total_tokens": 43
  }
}
```

Response structure details

- **`content`**: Contains the text description as a string
- **`images`**: Array of generated images, each with:
  - `type`: Always `"image_url"`
  - `image_url.url`: Base64-encoded data URI of the generated image

Streaming responses

For streaming requests, images are delivered in delta chunks:

```json
{
  "id": "chatcmpl-123",
  "object": "chat.completion.chunk",
  "created": 1677652288,
  "model": "google/gemini-2.5-flash-image-preview",
  "choices": [
    {
      "index": 0,
      "delta": {
        "images": [
          {
            "type": "image_url",
            "image_url": {
              "url": "
            }
          }
```

```
        ]
      },
      "finish_reason": null
    }
  ]
}
```

Handling streaming image responses

When processing streaming responses, check for both text content and images in each delta:

TypeScript

```typescript filename="streaming-images.ts"
import OpenAI from 'openai';

const openai = new OpenAI({
  apiKey: process.env.AI_GATEWAY_API_KEY,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

const stream = await openai.chat.completions.create({
  model: 'google/gemini-2.5-flash-image-preview',
  messages: [{ role: 'user', content: 'Generate a sunset image' }],
  // @ts-expect-error - modalities not yet in OpenAI types
  modalities: ['text', 'image'],
  stream: true,
});

for await (const chunk of stream) {
  const delta = chunk.choices[0]?.delta;

  // Handle text content
  if (delta?.content) {
    process.stdout.write(delta.content);
  }

  // Handle images
  if (delta?.images) {
    for (const img of delta.images) {
      if (img.type === 'image_url' && img.image_url) {
        console.log(`\n[Image received: ${img.image_url.url.length} chars]`);
      }
    }
  }
}
```

Python

```python filename="streaming-images.py"
import os
from openai import OpenAI

client = OpenAI(
    api_key=os.getenv('AI_GATEWAY_API_KEY'),
    base_url='https://ai-gateway.vercel.sh/v1'
)

stream = client.chat.completions.create(
    model='google/gemini-2.5-flash-image-preview',
    messages=[{'role': 'user', 'content': 'Generate a sunset image'}],
    extra_body={'modalities': ['text', 'image']},
    stream=True,
)

for chunk in stream:
    if chunk.choices and chunk.choices[0].delta:
        delta = chunk.choices[0].delta

        # Handle text content
        if hasattr(delta, 'content') and delta.content:
            print(delta.content, end='', flush=True)

        # Handle images
        if hasattr(delta, 'images') and delta.images:
            for img in delta.images:
                if img.get('type') == 'image_url' and img.get('image_url'):
                    image_url = img['image_url']['url']
                    print(f"\n[Image received: {len(image_url)} chars]")
```

> **💡 Note:** **Image generation support:** Currently, image generation is supported by
> Google's Gemini 2.5 Flash Image model. The generated images are returned as
> base64-encoded data URIs in the response. For more detailed information about
> image generation capabilities, see the [Image Generation
> documentation](/docs/ai-gateway/image-generation).

---
title: "OpenAI-Compatible API"
description: "Use OpenAI-compatible API endpoints with the AI Gateway for seamless integration with existing tools and libraries."
last_updated: "2026-01-16T02:19:25.285Z"
source: "https://vercel.com/docs/ai-gateway/openai-compat"

OpenAI-Compatible API

AI Gateway provides OpenAI-compatible API endpoints, letting you use multiple AI providers through a familiar interface. You can use exis

The OpenAI-compatible API implements the same specification as the [OpenAI API](https://platform.openai.com/docs/api-reference/chat).

Base URL

The OpenAI-compatible API is available at the following base URL:

```
https://ai-gateway.vercel.sh/v1
```

Authentication

The OpenAI-compatible API supports the same authentication methods as the main AI Gateway:

- **API key**: Use your AI Gateway API key with the `Authorization: Bearer <token>` header
- **OIDC token**: Use your Vercel OIDC token with the `Authorization: Bearer <token>` header

You only need to use one of these forms of authentication. If an API key is specified it will take precedence over any OIDC token, even i

Supported endpoints

The AI Gateway supports the following OpenAI-compatible endpoints:

- [`GET /models`](#list-models) - List available models
- [`GET /models/{model}`](#retrieve-model) - Retrieve a specific model
- [`POST /chat/completions`](/docs/ai-gateway/openai-compat/chat-completions) - Create chat completions with support for streaming, attac
- [`POST /embeddings`](/docs/ai-gateway/openai-compat/embeddings) - Generate vector embeddings

For advanced features, see:

- [Advanced configuration](/docs/ai-gateway/openai-compat/advanced) - Reasoning, provider options, model fallbacks, BYOK, prompt caching,
- [Image generation](/docs/ai-gateway/openai-compat/image-generation) - Generate images using multimodal models
- [Direct REST API usage](/docs/ai-gateway/openai-compat/rest-api) - Use the API without client libraries

Integration with existing tools

You can use the AI Gateway's OpenAI-compatible API with existing tools and
libraries like the [OpenAI client libraries](https://platform.openai.com/docs/libraries) and [AI SDK](https://ai-sdk.dev/). Point your ex
client to the AI Gateway's base URL and use your AI Gateway [API key](/docs/ai-gateway/authentication#api-key) or [OIDC token](/docs/ai-g

OpenAI client libraries

TypeScript

```typescript filename="client.ts"
import OpenAI from 'openai';

const openai = new OpenAI({
  apiKey: process.env.AI_GATEWAY_API_KEY,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

const response = await openai.chat.completions.create({
  model: 'anthropic/claude-sonnet-4.5',
  messages: [{ role: 'user', content: 'Hello, world!' }],
});
```

Python

```python filename="client.py"
import os
from openai import OpenAI

client = OpenAI(
    api_key=os.getenv('AI_GATEWAY_API_KEY'),
    base_url='https://ai-gateway.vercel.sh/v1'
)

response = client.chat.completions.create(
    model='anthropic/claude-sonnet-4.5',
    messages=[
        {'role': 'user', 'content': 'Hello, world!'}
    ]
)
```

AI SDK

For compatibility with [AI SDK](https://ai-sdk.dev/) and AI Gateway, install the [@ai-sdk/openai-compatible](https://ai-sdk.dev/providers

```typescript filename="client.ts"
import { createOpenAICompatible } from '@ai-sdk/openai-compatible';
import { generateText } from 'ai';

const gateway = createOpenAICompatible({
  name: 'openai',
  apiKey: process.env.AI_GATEWAY_API_KEY,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

const response = await generateText({
  model: gateway('anthropic/claude-sonnet-4.5'),
  prompt: 'Hello, world!',
});
```

List models

Retrieve a list of all available models that can be used with the AI Gateway.

Endpoint

```
GET /models
```

Example request

TypeScript

```typescript filename="list-models.ts"
import OpenAI from 'openai';

const openai = new OpenAI({
  apiKey: process.env.AI_GATEWAY_API_KEY,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

const models = await openai.models.list();
console.log(models);
```

Python

```python filename="list-models.py"
import os
from openai import OpenAI

client = OpenAI(
    api_key=os.getenv('AI_GATEWAY_API_KEY'),
    base_url='https://ai-gateway.vercel.sh/v1'
)

models = client.models.list()
print(models)
```

Response format

The response follows the OpenAI API format:

```json
{
  "object": "list",
  "data": [
    {
      "id": "anthropic/claude-sonnet-4.5",
      "object": "model",
      "created": 1677610602,
      "owned_by": "anthropic"
    },
    {
      "id": "openai/gpt-5.2",
      "object": "model",
      "created": 1677610602,
      "owned_by": "openai"
    }
  ]
}
```

Retrieve model

Retrieve details about a specific model.

Endpoint

```
GET /models/{model}
```

Parameters

- `model` (required): The model ID to retrieve (e.g., `anthropic/claude-sonnet-4`)

Example request

TypeScript

```typescript filename="retrieve-model.ts"
import OpenAI from 'openai';

const openai = new OpenAI({
  apiKey: process.env.AI_GATEWAY_API_KEY,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

const model = await openai.models.retrieve('anthropic/claude-sonnet-4.5');
console.log(model);
```

Python

```python filename="retrieve-model.py"
import os
from openai import OpenAI

client = OpenAI(
```

```
    api_key=os.getenv('AI_GATEWAY_API_KEY'),
    base_url='https://ai-gateway.vercel.sh/v1'
)

model = client.models.retrieve('anthropic/claude-sonnet-4.5')
print(model)
```

Response format

```json
{
  "id": "anthropic/claude-sonnet-4.5",
  "object": "model",
  "created": 1677610602,
  "owned_by": "anthropic"
}
```

Error handling

The API returns standard HTTP status codes and error responses:

Common error codes

- `400 Bad Request`: Invalid request parameters
- `401 Unauthorized`: Invalid or missing authentication
- `403 Forbidden`: Insufficient permissions
- `404 Not Found`: Model or endpoint not found
- `429 Too Many Requests`: Rate limit exceeded
- `500 Internal Server Error`: Server error

Error response format

```json
{
  "error": {
    "message": "Invalid request: missing required parameter 'model'",
    "type": "invalid_request_error",
    "param": "model",
    "code": "missing_parameter"
  }
}
```

---
title: "Direct REST API Usage"
description: "Use the AI Gateway API directly without client libraries using curl and fetch."
last_updated: "2026-01-16T02:19:25.522Z"
source: "https://vercel.com/docs/ai-gateway/openai-compat/rest-api"

Direct REST API Usage

If you prefer to use the AI Gateway API directly without the OpenAI client libraries, you can make HTTP requests using any HTTP client. H

List models

cURL

```bash filename="list-models.sh"
curl -X GET "https://ai-gateway.vercel.sh/v1/models" \
  -H "Authorization: Bearer $AI_GATEWAY_API_KEY" \
  -H "Content-Type: application/json"
```

JavaScript

```javascript filename="list-models.js"
const response = await fetch('https://ai-gateway.vercel.sh/v1/models', {
  method: 'GET',
  headers: {
    Authorization: `Bearer ${process.env.AI_GATEWAY_API_KEY}`,
    'Content-Type': 'application/json',
  },
});

const models = await response.json();
console.log(models);
```

Basic chat completion

cURL

```bash filename="chat-completion.sh"
curl -X POST "https://ai-gateway.vercel.sh/v1/chat/completions" \
  -H "Authorization: Bearer $AI_GATEWAY_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "model": "anthropic/claude-sonnet-4.5",
    "messages": [
      {
        "role": "user",
        "content": "Write a one-sentence bedtime story about a unicorn."
      }
    ],
    "stream": false
  }'
```

```

#### JavaScript

```javascript filename="chat-completion.js"
const response = await fetch(
  'https://ai-gateway.vercel.sh/v1/chat/completions',
  {
    method: 'POST',
    headers: {
      Authorization: `Bearer ${process.env.AI_GATEWAY_API_KEY}`,
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      model: 'anthropic/claude-sonnet-4.5',
      messages: [
        {
          role: 'user',
          content: 'Write a one-sentence bedtime story about a unicorn.',
        },
      ],
      stream: false,
    }),
  },
);

const result = await response.json();
console.log(result);
```

Streaming chat completion

cURL

```bash filename="streaming-chat.sh"
curl -X POST "https://ai-gateway.vercel.sh/v1/chat/completions" \
  -H "Authorization: Bearer $AI_GATEWAY_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "model": "anthropic/claude-sonnet-4.5",
    "messages": [
      {
        "role": "user",
        "content": "Write a one-sentence bedtime story about a unicorn."
      }
    ],
    "stream": true
  }' \
  --no-buffer
```

JavaScript

```javascript filename="streaming-chat.js"
const response = await fetch(
  'https://ai-gateway.vercel.sh/v1/chat/completions',
  {
    method: 'POST',
    headers: {
      Authorization: `Bearer ${process.env.AI_GATEWAY_API_KEY}`,
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      model: 'anthropic/claude-sonnet-4.5',
      messages: [
        {
          role: 'user',
          content: 'Write a one-sentence bedtime story about a unicorn.',
        },
      ],
      stream: true,
    }),
  },
);

const reader = response.body.getReader();
const decoder = new TextDecoder();

while (true) {
  const { done, value } = await reader.read();
  if (done) break;

  const chunk = decoder.decode(value);
  const lines = chunk.split('\n');

  for (const line of lines) {
    if (line.startsWith('data: ')) {
      const data = line.slice(6);
      if (data === '[DONE]') {
        console.log('Stream complete');
        break;
      } else if (data.trim()) {
        const parsed = JSON.parse(data);
        const content = parsed.choices?.[0]?.delta?.content;
        if (content) {
          process.stdout.write(content);
        }
      }
    }
  }
```
```

```
}
```

Image analysis

cURL

```bash filename="image-analysis.sh"
# First, convert your image to base64
IMAGE_BASE64=$(base64 -i ./path/to/image.png)

curl -X POST "https://ai-gateway.vercel.sh/v1/chat/completions" \
  -H "Authorization: Bearer $AI_GATEWAY_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "model": "anthropic/claude-sonnet-4.5",
    "messages": [
      {
        "role": "user",
        "content": [
          {
            "type": "text",
            "text": "Describe this image in detail."
          },
          {
            "type": "image_url",
            "image_url": {
              "url": "'"$IMAGE_BASE64"'",
              "detail": "auto"
            }
          }
        ]
      }
    ],
    "stream": false
  }'
```

JavaScript

```javascript filename="image-analysis.js"
import fs from 'node:fs';

// Read the image file as base64
const imageBuffer = fs.readFileSync('./path/to/image.png');
const imageBase64 = imageBuffer.toString('base64');

const response = await fetch(
  'https://ai-gateway.vercel.sh/v1/chat/completions',
  {
    method: 'POST',
    headers: {
      Authorization: `Bearer ${process.env.AI_GATEWAY_API_KEY}`,
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      model: 'anthropic/claude-sonnet-4.5',
      messages: [
        {
          role: 'user',
          content: [
            { type: 'text', text: 'Describe this image in detail.' },
            {
              type: 'image_url',
              image_url: {
                url: `${imageBase64}`,
                detail: 'auto',
              },
            },
          ],
        },
      ],
      stream: false,
    }),
  },
);

const result = await response.json();
console.log(result);
```

Tool calls

cURL

```bash filename="tool-calls.sh"
curl -X POST "https://ai-gateway.vercel.sh/v1/chat/completions" \
  -H "Authorization: Bearer $AI_GATEWAY_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "model": "anthropic/claude-sonnet-4.5",
    "messages": [
      {
        "role": "user",
        "content": "What is the weather like in San Francisco?"
      }
    ],
    "tools": [
      {
        "type": "function",
```

```
        "function": {
          "name": "get_weather",
          "description": "Get the current weather in a given location",
          "parameters": {
            "type": "object",
            "properties": {
              "location": {
                "type": "string",
                "description": "The city and state, e.g. San Francisco, CA"
              },
              "unit": {
                "type": "string",
                "enum": ["celsius", "fahrenheit"],
                "description": "The unit for temperature"
              }
            },
            "required": ["location"]
          }
        }
      }
    ],
    "tool_choice": "auto",
    "stream": false
  }'
```

JavaScript

```javascript filename="tool-calls.js"
const response = await fetch(
  'https://ai-gateway.vercel.sh/v1/chat/completions',
  {
    method: 'POST',
    headers: {
      Authorization: `Bearer ${process.env.AI_GATEWAY_API_KEY}`,
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      model: 'anthropic/claude-sonnet-4.5',
      messages: [
        {
          role: 'user',
          content: 'What is the weather like in San Francisco?',
        },
      ],
      tools: [
        {
          type: 'function',
          function: {
            name: 'get_weather',
            description: 'Get the current weather in a given location',
            parameters: {
              type: 'object',
              properties: {
                location: {
                  type: 'string',
                  description: 'The city and state, e.g. San Francisco, CA',
                },
                unit: {
                  type: 'string',
                  enum: ['celsius', 'fahrenheit'],
                  description: 'The unit for temperature',
                },
              },
              required: ['location'],
            },
          },
        },
      ],
      tool_choice: 'auto',
      stream: false,
    }),
  },
);

const result = await response.json();
console.log(result);
```

Provider options

cURL

```bash filename="provider-options.sh"
curl -X POST "https://ai-gateway.vercel.sh/v1/chat/completions" \
  -H "Authorization: Bearer $AI_GATEWAY_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "model": "anthropic/claude-sonnet-4.5",
    "messages": [
      {
        "role": "user",
        "content": "Tell me the history of the San Francisco Mission-style burrito in two paragraphs."
      }
    ],
    "stream": false,
    "providerOptions": {
      "gateway": {
        "order": ["vertex", "anthropic"]
      }
```

```
    }
  }'
```

JavaScript

```javascript filename="provider-options.js"
const response = await fetch(
  'https://ai-gateway.vercel.sh/v1/chat/completions',
  {
    method: 'POST',
    headers: {
      Authorization: `Bearer ${process.env.AI_GATEWAY_API_KEY}`,
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      model: 'anthropic/claude-sonnet-4.5',
      messages: [
        {
          role: 'user',
          content:
            'Tell me the history of the San Francisco Mission-style burrito in two paragraphs.',
        },
      ],
      stream: false,
      providerOptions: {
        gateway: {
          order: ['vertex', 'anthropic'], // Try Vertex AI first, then Anthropic
        },
      },
    }),
  },
);

const result = await response.json();
console.log(result);
```

---
title: "Structured Outputs"
description: "Generate structured JSON responses that conform to a specific schema using the OpenAI-compatible API."
last_updated: "2026-01-16T02:19:25.539Z"
source: "https://vercel.com/docs/ai-gateway/openai-compat/structured-outputs"

Structured Outputs

Generate structured JSON responses that conform to a specific schema, ensuring predictable and reliable data formats for your application

JSON Schema format

Use the OpenAI standard `json_schema` response format for the most robust structured output experience. This follows the official [OpenAI

Example request

TypeScript

```typescript filename="structured-output-json-schema.ts"
import OpenAI from 'openai';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const openai = new OpenAI({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

const completion = await openai.chat.completions.create({
  model: 'openai/gpt-5.2',
  messages: [
    {
      role: 'user',
      content: 'Create a product listing for a wireless gaming headset.',
    },
  ],
  stream: false,
  response_format: {
    type: 'json_schema',
    json_schema: {
      name: 'product_listing',
      description: 'A product listing with details and pricing',
      schema: {
        type: 'object',
        properties: {
          name: {
            type: 'string',
            description: 'Product name',
          },
          brand: {
            type: 'string',
            description: 'Brand name',
          },
          price: {
            type: 'number',
            description: 'Price in USD',
          },
          category: {
            type: 'string',
            description: 'Product category',
```

```
          },
          description: {
            type: 'string',
            description: 'Product description',
          },
          features: {
            type: 'array',
            items: { type: 'string' },
            description: 'Key product features',
          },
        },
        required: ['name', 'brand', 'price', 'category', 'description'],
        additionalProperties: false,
      },
    },
  },
});

console.log('Assistant:', completion.choices[0].message.content);

// Parse the structured response
const structuredData = JSON.parse(completion.choices[0].message.content);
console.log('Structured Data:', structuredData);
```

Python

```python filename="structured-output-json-schema.py"
import os
import json
from openai import OpenAI

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = OpenAI(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh/v1'
)

completion = client.chat.completions.create(
    model='openai/gpt-5.2',
    messages=[
        {
            'role': 'user',
            'content': 'Create a product listing for a wireless gaming headset.'
        }
    ],
    stream=False,
    response_format={
        'type': 'json_schema',
        'json_schema': {
            'name': 'product_listing',
            'description': 'A product listing with details and pricing',
            'schema': {
                'type': 'object',
                'properties': {
                    'name': {
                        'type': 'string',
                        'description': 'Product name'
                    },
                    'brand': {
                        'type': 'string',
                        'description': 'Brand name'
                    },
                    'price': {
                        'type': 'number',
                        'description': 'Price in USD'
                    },
                    'category': {
                        'type': 'string',
                        'description': 'Product category'
                    },
                    'description': {
                        'type': 'string',
                        'description': 'Product description'
                    },
                    'features': {
                        'type': 'array',
                        'items': {'type': 'string'},
                        'description': 'Key product features'
                    }
                },
                'required': ['name', 'brand', 'price', 'category', 'description'],
                'additionalProperties': False
            },
        }
    }
)

print('Assistant:', completion.choices[0].message.content)

# Parse the structured response
structured_data = json.loads(completion.choices[0].message.content)
print('Structured Data:', json.dumps(structured_data, indent=2))
```

Response format

The response contains structured JSON that conforms to your specified schema:

```json
{
  "id": "chatcmpl-123",
  "object": "chat.completion",
  "created": 1677652288,
  "model": "openai/gpt-5.2",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "{\"name\":\"SteelSeries Arctis 7P\",\"brand\":\"SteelSeries\",\"price\":149.99,\"category\":\"Gaming Headsets\",\"des
      },
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 25,
    "completion_tokens": 45,
    "total_tokens": 70
  }
}
```

JSON Schema parameters

- **`type`**: Must be `"json_schema"`
- **`json_schema`**: Object containing schema definition
  - **`name`** (required): Name of the response schema
  - **`description`** (optional): Human-readable description of the expected output
  - **`schema`** (required): Valid JSON Schema object defining the structure

Legacy JSON format (alternative)

> **💡 Note:** **Legacy format:** The following format is supported for backward
> compatibility. For new implementations, use the `json_schema` format above.

TypeScript

```typescript filename="structured-output-legacy.ts"
import OpenAI from 'openai';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const openai = new OpenAI({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

const completion = await openai.chat.completions.create({
  model: 'openai/gpt-5.2',
  messages: [
    {
      role: 'user',
      content: 'Create a product listing for a wireless gaming headset.',
    },
  ],
  stream: false,
  // @ts-expect-error - Legacy format not in OpenAI types
  response_format: {
    type: 'json',
    name: 'product_listing',
    description: 'A product listing with details and pricing',
    schema: {
      type: 'object',
      properties: {
        name: { type: 'string', description: 'Product name' },
        brand: { type: 'string', description: 'Brand name' },
        price: { type: 'number', description: 'Price in USD' },
        category: { type: 'string', description: 'Product category' },
        description: { type: 'string', description: 'Product description' },
        features: {
          type: 'array',
          items: { type: 'string' },
          description: 'Key product features',
        },
      },
      required: ['name', 'brand', 'price', 'category', 'description'],
    },
  },
});

console.log('Assistant:', completion.choices[0].message.content);
```

Python

```python filename="structured-output-legacy.py"
import os
import json
from openai import OpenAI

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = OpenAI(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh/v1'
)

completion = client.chat.completions.create(
```

```
    model='openai/gpt-5.2',
    messages=[
        {
            'role': 'user',
            'content': 'Create a product listing for a wireless gaming headset.'
        }
    ],
    stream=False,
    response_format={
        'type': 'json',
        'name': 'product_listing',
        'description': 'A product listing with details and pricing',
        'schema': {
            'type': 'object',
            'properties': {
                'name': {'type': 'string', 'description': 'Product name'},
                'brand': {'type': 'string', 'description': 'Brand name'},
                'price': {'type': 'number', 'description': 'Price in USD'},
                'category': {'type': 'string', 'description': 'Product category'},
                'description': {'type': 'string', 'description': 'Product description'},
                'features': {
                    'type': 'array',
                    'items': {'type': 'string'},
                    'description': 'Key product features'
                }
            },
            'required': ['name', 'brand', 'price', 'category', 'description']
        }
    }
)

print('Assistant:', completion.choices[0].message.content)

# Parse the structured response
structured_data = json.loads(completion.choices[0].message.content)
print('Structured Data:', json.dumps(structured_data, indent=2))
```

Streaming with structured outputs

Both `json_schema` and legacy `json` formats work with streaming responses:

TypeScript

```typescript filename="structured-streaming.ts"
import OpenAI from 'openai';

const openai = new OpenAI({
  apiKey: process.env.AI_GATEWAY_API_KEY,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

const stream = await openai.chat.completions.create({
  model: 'openai/gpt-5.2',
  messages: [
    {
      role: 'user',
      content: 'Create a product listing for a wireless gaming headset.',
    },
  ],
  stream: true,
  response_format: {
    type: 'json_schema',
    json_schema: {
      name: 'product_listing',
      description: 'A product listing with details and pricing',
      schema: {
        type: 'object',
        properties: {
          name: { type: 'string', description: 'Product name' },
          brand: { type: 'string', description: 'Brand name' },
          price: { type: 'number', description: 'Price in USD' },
          category: { type: 'string', description: 'Product category' },
          description: { type: 'string', description: 'Product description' },
          features: {
            type: 'array',
            items: { type: 'string' },
            description: 'Key product features',
          },
        },
        required: ['name', 'brand', 'price', 'category', 'description'],
        additionalProperties: false,
      },
    },
  },
});

let completeResponse = '';
for await (const chunk of stream) {
  const content = chunk.choices[0]?.delta?.content;
  if (content) {
    process.stdout.write(content);
    completeResponse += content;
  }
}

// Parse the complete structured response
const structuredData = JSON.parse(completeResponse);
console.log('\nParsed Product:', structuredData);
```

Python

```python filename="structured-streaming.py"
import os
import json
from openai import OpenAI

client = OpenAI(
    api_key=os.getenv('AI_GATEWAY_API_KEY'),
    base_url='https://ai-gateway.vercel.sh/v1'
)

stream = client.chat.completions.create(
    model='openai/gpt-5.2',
    messages=[
        {
            'role': 'user',
            'content': 'Create a product listing for a wireless gaming headset.'
        }
    ],
    stream=True,
    response_format={
        'type': 'json_schema',
        'json_schema': {
            'name': 'product_listing',
            'description': 'A product listing with details and pricing',
            'schema': {
                'type': 'object',
                'properties': {
                    'name': {'type': 'string', 'description': 'Product name'},
                    'brand': {'type': 'string', 'description': 'Brand name'},
                    'price': {'type': 'number', 'description': 'Price in USD'},
                    'category': {'type': 'string', 'description': 'Product category'},
                    'description': {'type': 'string', 'description': 'Product description'},
                    'features': {
                        'type': 'array',
                        'items': {'type': 'string'},
                        'description': 'Key product features'
                    }
                },
                'required': ['name', 'brand', 'price', 'category', 'description'],
                'additionalProperties': False
            },
        }
    }
)

complete_response = ''
for chunk in stream:
    if chunk.choices and chunk.choices[0].delta.content:
        content = chunk.choices[0].delta.content
        print(content, end='', flush=True)
        complete_response += content

# Parse the complete structured response
structured_data = json.loads(complete_response)
print('\nParsed Product:', json.dumps(structured_data, indent=2))
```

> **💡 Note:** **Streaming assembly:** When using structured outputs with streaming, you'll
> need to collect all the content chunks and parse the complete JSON response
> once the stream is finished.

---
title: "Tool Calls"
description: "Use OpenAI-compatible function calling to enable models to call tools and functions through the AI Gateway."
last_updated: "2026-01-16T02:19:25.546Z"
source: "https://vercel.com/docs/ai-gateway/openai-compat/tool-calls"

Tool Calls

The AI Gateway supports OpenAI-compatible function calling, allowing models to call tools and functions. This follows the same specificat

Basic tool calls

TypeScript

```typescript filename="tool-calls.ts"
import OpenAI from 'openai';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const openai = new OpenAI({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

const tools: OpenAI.Chat.Completions.ChatCompletionTool[] = [
  {
    type: 'function',
    function: {
      name: 'get_weather',
      description: 'Get the current weather in a given location',
      parameters: {
        type: 'object',
        properties: {
          location: {
```

```
            type: 'string',
            description: 'The city and state, e.g. San Francisco, CA',
          },
          unit: {
            type: 'string',
            enum: ['celsius', 'fahrenheit'],
            description: 'The unit for temperature',
          },
        },
        required: ['location'],
      },
    },
  },
];

const completion = await openai.chat.completions.create({
  model: 'anthropic/claude-sonnet-4.5',
  messages: [
    {
      role: 'user',
      content: 'What is the weather like in San Francisco?',
    },
  ],
  tools: tools,
  tool_choice: 'auto',
  stream: false,
});

console.log('Assistant:', completion.choices[0].message.content);
console.log('Tool calls:', completion.choices[0].message.tool_calls);
```

Python

```python filename="tool-calls.py"
import os
from openai import OpenAI

api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')

client = OpenAI(
    api_key=api_key,
    base_url='https://ai-gateway.vercel.sh/v1'
)

tools = [
    {
        'type': 'function',
        'function': {
            'name': 'get_weather',
            'description': 'Get the current weather in a given location',
            'parameters': {
                'type': 'object',
                'properties': {
                    'location': {
                        'type': 'string',
                        'description': 'The city and state, e.g. San Francisco, CA'
                    },
                    'unit': {
                        'type': 'string',
                        'enum': ['celsius', 'fahrenheit'],
                        'description': 'The unit for temperature'
                    }
                },
                'required': ['location']
            }
        }
    }
]

completion = client.chat.completions.create(
    model='anthropic/claude-sonnet-4.5',
    messages=[
        {
            'role': 'user',
            'content': 'What is the weather like in San Francisco?'
        }
    ],
    tools=tools,
    tool_choice='auto',
    stream=False,
)

print('Assistant:', completion.choices[0].message.content)
print('Tool calls:', completion.choices[0].message.tool_calls)
```

> **💡 Note:** **Controlling tool selection:** By default, `tool_choice` is set to ``auto``, allowing the model to decide when to use too
> * Force a specific tool with: `tool_choice: { type: 'function', function: { name: 'your_function_name' } }`

Tool call response format

When the model makes tool calls, the response includes tool call information:

```json
{
  "id": "chatcmpl-123",
  "object": "chat.completion",
  "created": 1677652288,
  "model": "anthropic/claude-sonnet-4.5",
```

```
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": null,
        "tool_calls": [
          {
            "id": "call_123",
            "type": "function",
            "function": {
              "name": "get_weather",
              "arguments": "{\"location\": \"San Francisco, CA\", \"unit\": \"celsius\"}"
            }
          }
        ]
      },
      "finish_reason": "tool_calls"
    }
  ],
  "usage": {
    "prompt_tokens": 82,
    "completion_tokens": 18,
    "total_tokens": 100
  }
}
```

---
title: "Image Input"
description: "Send images for analysis using the OpenResponses API."
last_updated: "2026-01-16T02:19:25.550Z"
source: "https://vercel.com/docs/ai-gateway/openresponses/image-input"

Image Input

The [OpenResponses API](/docs/ai-gateway/openresponses) supports sending images alongside text for vision-capable models to analyze. Incl

```typescript filename="image-input.ts"
const apiKey = process.env.AI_GATEWAY_API_KEY;

const response = await fetch('https://ai-gateway.vercel.sh/v1/responses', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    Authorization: `Bearer ${apiKey}`,
  },
  body: JSON.stringify({
    model: 'zai/glm-4.7',
    input: [
      {
        type: 'message',
        role: 'user',
        content: [
          { type: 'text', text: 'Describe this image in detail.' },
          {
            type: 'image_url',
            image_url: { url: 'https://example.com/image.jpg', detail: 'auto' },
          },
        ],
      },
    ],
  }),
});
```

Base64-encoded images

You can also use base64-encoded images:

```typescript
{
  type: 'image_url',
  image_url: {
    url: `${imageBase64}`,
    detail: 'high',
  },
}
```

Detail parameter

The `detail` parameter controls image resolution:

- `auto` - Let the model decide the appropriate resolution
- `low` - Use lower resolution for faster processing
- `high` - Use higher resolution for more detailed analysis

---
title: "OpenResponses API"
description: "Use the OpenResponses API specification with AI Gateway for a unified, provider-agnostic interface."
last_updated: "2026-01-16T02:19:25.558Z"
source: "https://vercel.com/docs/ai-gateway/openresponses"

OpenResponses API

AI Gateway supports the [OpenResponses API](https://openresponses.org) specification, an open standard for AI model interactions. OpenRes

Base URL

The OpenResponses-compatible API is available at:

```
https://ai-gateway.vercel.sh/v1
```

Authentication

The OpenResponses API supports the same [authentication methods](docs/ai-gateway/authentication) as the main AI Gateway:

- **API key**: Use your AI Gateway API key with the `Authorization: Bearer <token>` header
- **OIDC token**: Use your Vercel OIDC token with the `Authorization: Bearer <token>` header

You only need to use one of these forms of authentication. If an API key is specified it will take precedence over any OIDC token, even i

Supported features

The OpenResponses API supports the following features:

- [Text generation](/docs/ai-gateway/openresponses/text-generation) - Generate text responses from prompts
- [Streaming](/docs/ai-gateway/openresponses/streaming) - Stream tokens as they're generated
- [Image input](/docs/ai-gateway/openresponses/image-input) - Send images for analysis
- [Tool calling](/docs/ai-gateway/openresponses/tool-calling) - Define tools the model can call
- [Provider options](/docs/ai-gateway/openresponses/provider-options) - Configure model fallbacks and provider-specific settings

Getting started

Here's a simple example to generate a text response:

```typescript filename="quickstart.ts"
const apiKey = process.env.AI_GATEWAY_API_KEY;

const response = await fetch('https://ai-gateway.vercel.sh/v1/responses', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    Authorization: `Bearer ${apiKey}`,
  },
  body: JSON.stringify({
    model: 'anthropic/claude-sonnet-4.5',
    input: [
      {
        type: 'message',
        role: 'user',
        content: 'What is the capital of France?',
      },
    ],
  }),
});

const result = await response.json();
console.log(result.output[0].content[0].text);
```

Parameters

Required parameters

- `model` (string): The model ID in `provider/model` format (e.g., `openai/gpt-5.2`, `anthropic/claude-sonnet-4.5`)
- `input` (array): Array of message objects containing `type`, `role`, and `content` fields

Optional parameters

- `stream` (boolean): Stream the response. Defaults to `false`
- `temperature` (number): Controls randomness. Range: 0-2
- `top_p` (number): Nucleus sampling. Range: 0-1
- `max_output_tokens` (integer): Maximum tokens to generate
- `tools` (array): Tool definitions for function calling
- `tool_choice` (string): Tool selection mode: `auto`, `required`, or `none`
- `reasoning` (object): Reasoning configuration with `effort` level
- `providerOptions` (object): Provider-specific options for gateway configuration

Example with parameters

This example shows how to combine multiple parameters to control the model's behavior, set up fallbacks, and enable reasoning.

```typescript filename="parameters-example.ts"
const response = await fetch('https://ai-gateway.vercel.sh/v1/responses', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    Authorization: `Bearer ${process.env.AI_GATEWAY_API_KEY}`,
  },
  body: JSON.stringify({
    model: 'anthropic/claude-sonnet-4.5', // provider/model format
    input: [
      {
        type: 'message',
        role: 'user',
        content: 'Explain neural networks.',
      },
    ],
    stream: true, // stream tokens as generated
    max_output_tokens: 500, // limit response length
    reasoning: {
      effort: 'medium', // reasoning depth
```

```
    },
    providerOptions: {
      gateway: {
        models: ['anthropic/claude-sonnet-4.5', 'openai/gpt-5.2'], // fallbacks
      },
    },
  }),
});
```

Error handling

The API returns standard HTTP status codes and error responses.

Common error codes

- `400 Bad Request` - Invalid request parameters
- `401 Unauthorized` - Invalid or missing authentication
- `403 Forbidden` - Insufficient permissions
- `404 Not Found` - Model or endpoint not found
- `429 Too Many Requests` - Rate limit exceeded
- `500 Internal Server Error` - Server error

Error response format

When an error occurs, the API returns a JSON object with details about what went wrong.

```json
{
  "error": {
    "message": "Invalid request: missing required parameter 'model'",
    "type": "invalid_request_error",
    "param": "model",
    "code": "missing_parameter"
  }
}
```

---
title: "Provider Options"
description: "Configure provider routing, fallbacks, and restrictions using the OpenResponses API."
last_updated: "2026-01-16T02:19:25.563Z"
source: "https://vercel.com/docs/ai-gateway/openresponses/provider-options"

Provider Options

The [OpenResponses API](/docs/ai-gateway/openresponses) lets you configure AI Gateway behavior using `providerOptions`. The `gateway` nam

Model fallbacks

Set up automatic fallbacks so if your primary model is unavailable, requests route to backup models in order. Use the `models` array to s

```typescript filename="fallbacks.ts"
const apiKey = process.env.AI_GATEWAY_API_KEY;

const response = await fetch('https://ai-gateway.vercel.sh/v1/responses', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    Authorization: `Bearer ${apiKey}`,
  },
  body: JSON.stringify({
    model: 'anthropic/claude-sonnet-4.5',
    input: [{ type: 'message', role: 'user', content: 'Tell me a fun fact about octopuses.' }],
    providerOptions: {
      gateway: {
        models: ['anthropic/claude-sonnet-4.5', 'openai/gpt-5.2', 'google/gemini-3-flash'],
      },
    },
  }),
});
```

Provider routing

Control the order in which providers are tried using the `order` array. AI Gateway will attempt providers in the specified order until on

```typescript filename="routing.ts"
const response = await fetch('https://ai-gateway.vercel.sh/v1/responses', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    Authorization: `Bearer ${apiKey}`,
  },
  body: JSON.stringify({
    model: 'google/gemini-3-flash',
    input: [{ type: 'message', role: 'user', content: 'Explain quantum computing in one sentence.' }],
    providerOptions: {
      gateway: {
        order: ['google', 'openai', 'anthropic'],
      },
    },
  }),
});
```

Provider restriction

Restrict requests to specific providers using the `only` array. This ensures your requests only go to approved providers, which can be us

```typescript filename="restriction.ts"
const response = await fetch('https://ai-gateway.vercel.sh/v1/responses', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    Authorization: `Bearer ${apiKey}`,
  },
  body: JSON.stringify({
    model: 'zai/glm-4.7',
    input: [{ type: 'message', role: 'user', content: 'What makes a great cup of coffee?' }],
    providerOptions: {
      gateway: {
        only: ['zai', 'deepseek'],
      },
    },
  }),
});
```

---
title: "Streaming"
description: "Stream responses token by token using the OpenResponses API."
last_updated: "2026-01-16T02:19:25.568Z"
source: "https://vercel.com/docs/ai-gateway/openresponses/streaming"

Streaming

The [OpenResponses API](/docs/ai-gateway/openresponses) supports streaming to receive tokens as they're generated instead of waiting for

```typescript filename="stream.ts"
const apiKey = process.env.AI_GATEWAY_API_KEY;

const response = await fetch('https://ai-gateway.vercel.sh/v1/responses', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    Authorization: `Bearer ${apiKey}`,
  },
  body: JSON.stringify({
    model: 'google/gemini-3-flash',
    input: [
      {
        type: 'message',
        role: 'user',
        content: 'Write a haiku about debugging code.',
      },
    ],
    stream: true,
  }),
});

const reader = response.body.getReader();
const decoder = new TextDecoder();

while (true) {
  const { done, value } = await reader.read();
  if (done) break;

  const chunk = decoder.decode(value);
  const lines = chunk.split('\n');

  for (const line of lines) {
    if (line.startsWith('data:')) {
      const data = line.substring(6).trim();
      if (data) {
        const event = JSON.parse(data);
        if (event.type === 'response.output_text.delta') {
          process.stdout.write(event.delta);
        }
      }
    }
  }
}
```

Streaming events

- `response.created` - Response initialized
- `response.output_text.delta` - Text chunk received
- `response.output_text.done` - Text generation complete
- `response.completed` - Full response complete with usage stats

---
title: "Text Generation"
description: "Generate text responses using the OpenResponses API."
last_updated: "2026-01-16T02:19:25.577Z"
source: "https://vercel.com/docs/ai-gateway/openresponses/text-generation"

Text Generation

Use the [OpenResponses API](/docs/ai-gateway/openresponses) to generate text responses from AI models. The `input` array contains message

```typescript filename="generate.ts"
const apiKey = process.env.AI_GATEWAY_API_KEY;
```

```
const response = await fetch('https://ai-gateway.vercel.sh/v1/responses', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    Authorization: `Bearer ${apiKey}`,
  },
  body: JSON.stringify({
    model: 'openai/gpt-5.2',
    input: [
      {
        type: 'message',
        role: 'user',
        content: 'Why do developers prefer dark mode?',
      },
    ],
  }),
});

const result = await response.json();
```

Response format

The response includes the generated text in the `output` array, along with token usage information.

```json
{
  "id": "resp_abc123",
  "object": "response",
  "model": "openai/gpt-5.2",
  "output": [
    {
      "type": "message",
      "role": "assistant",
      "content": [
        {
          "type": "output_text",
          "text": "Habit and aesthetics reinforce the preference, but ergonomics and contrast are the primary drivers."
        }
      ]
    }
  ],
  "usage": {
    "input_tokens": 14,
    "output_tokens": 18
  }
}
```

---
title: "Tool Calling"
description: "Define tools the model can call using the OpenResponses API."
last_updated: "2026-01-16T02:19:25.583Z"
source: "https://vercel.com/docs/ai-gateway/openresponses/tool-calling"

Tool Calling

The [OpenResponses API](/docs/ai-gateway/openresponses) supports tool calling to give models access to external functions. Define tools i

```typescript filename="tool-calls.ts"
const apiKey = process.env.AI_GATEWAY_API_KEY;

const response = await fetch('https://ai-gateway.vercel.sh/v1/responses', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    Authorization: `Bearer ${apiKey}`,
  },
  body: JSON.stringify({
    model: 'deepseek/deepseek-v3.2-thinking',
    input: [
      {
        type: 'message',
        role: 'user',
        content: 'What is the weather like in New York?',
      },
    ],
    tools: [
      {
        type: 'function',
        function: {
          name: 'get_weather',
          description: 'Get the current weather in a location',
          parameters: {
            type: 'object',
            properties: {
              location: {
                type: 'string',
                description: 'The city and state, e.g. San Francisco, CA',
              },
            },
            required: ['location'],
          },
        },
      },
    ],
    tool_choice: 'auto',
```

```
  }),
});
```

Tool call response

When the model decides to call a tool, the response includes a `function_call` output:

```json
{
  "output": [
    {
      "type": "function_call",
      "name": "get_weather",
      "arguments": "{\"location\": \"New York, NY\"}",
      "call_id": "call_abc123"
    }
  ]
}
```

Tool choice options

- `auto` - The model decides whether to call a tool
- `required` - The model must call at least one tool
- `none` - The model cannot call any tools

---
title: "AI Gateway"
description: "TypeScript toolkit for building AI-powered applications with React, Next.js, Vue, Svelte and Node.js"
last_updated: "2026-01-16T02:19:25.600Z"
source: "https://vercel.com/docs/ai-gateway"

AI Gateway

The [AI Gateway](https://vercel.com/ai-gateway) provides a unified API to access [hundreds of models](https://vercel.com/ai-gateway/model
It gives you the ability to set budgets, monitor usage, load-balance requests, and manage fallbacks.

The design allows it to work seamlessly with [AI SDK v5 and v6](/docs/ai-gateway/getting-started), [OpenAI SDK](/docs/ai-gateway/openai-c

Key features

- **Unified API**: helps you switch between providers and models with minimal code changes
- **High reliability**: automatically retries requests to other providers if one fails
- **Embeddings support**: generate vector embeddings for search, retrieval, and other tasks
- **Spend monitoring**: monitor your spending across different providers
- **No markup on tokens**: tokens cost the same as they would from the provider directly, with 0% markup, including with [Bring Your Own

TypeScript

```typescript filename="index.ts" {4}
import { generateText } from 'ai';

const { text } = await generateText({
  model: 'anthropic/claude-sonnet-4.5',
  prompt: 'What is the capital of France?',
});

```

Python

```python filename="index.py" {10}
import os
from openai import OpenAI

client = OpenAI(
  api_key=os.getenv('AI_GATEWAY_API_KEY'),
  base_url='https://ai-gateway.vercel.sh/v1'
)

response = client.chat.completions.create(
  model='xai/grok-4',
  messages=[
    {
      'role': 'user',
      'content': 'Why is the sky blue?'
    }
  ]
)
```

cURL

```bash filename="index.sh" {5}
curl -X POST "https://ai-gateway.vercel.sh/v1/chat/completions" \
-H "Authorization: Bearer $AI_GATEWAY_API_KEY" \
-H "Content-Type: application/json" \
-d '{
  "model": "openai/gpt-5.2",
  "messages": [
    {
      "role": "user",
      "content": "Why is the sky blue?"
    }
  ],
  "stream": false
}'
```

```

## More resources

- [Getting started with AI Gateway](/docs/ai-gateway/getting-started)
- [Models and providers](/docs/ai-gateway/models-and-providers)
- [Provider options (routing & fallbacks)](/docs/ai-gateway/provider-options)
- [Web search](/docs/ai-gateway/web-search)
- [Observability](/docs/ai-gateway/observability)
- [Claude Code](/docs/ai-gateway/claude-code)
- [Anthropic compatibility](/docs/ai-gateway/anthropic-compat)
- [OpenAI compatibility](/docs/ai-gateway/openai-compat)
- [Usage and billing](/docs/ai-gateway/usage)
- [Authentication](/docs/ai-gateway/authentication)
- [Bring your own key](/docs/ai-gateway/byok)
- [Framework integrations](/docs/ai-gateway/framework-integrations)
- [App attribution](/docs/ai-gateway/app-attribution)

---
title: "Pricing"
description: "Learn about pricing for the AI Gateway."
last_updated: "2026-01-16T02:19:25.616Z"
source: "https://vercel.com/docs/ai-gateway/pricing"
---

# Pricing

You only pay for what you use on the AI Gateway by purchasing [AI Gateway Credits through the Vercel dashboard](#view-your-ai-gateway-cre

Charges are automatically deducted from your AI Gateway Credits balance and you can [top up the balance](#top-up-your-ai-gateway-credits)

## Free and paid tiers

The AI Gateway offers both a free tier and a paid tier for AI Gateway Credits. **For the paid tier, tokens are provided with zero markup,

### Free tier

Every Vercel team account includes $5 of free usage per month, giving you the opportunity to explore the AI Gateway without any upfront c

How it works:

- **$5 monthly credit**: you'll receive $5 AI Gateway Credits every 30 days after you make your first AI Gateway request.
- **Model flexibility**: choose from any available models, your free credits work across our entire model catalog.
- **No commitment**: you can stay on the free tier as long as you do not purchase AI Gateway Credits through the AI Gateway.

### Moving to paid tier

You can purchase AI Gateway Credits and move to a paid account on the AI Gateway, enabling you to run larger workloads.

Once you purchase AI Gateway Credits, your account transitions to our pay-as-you-go model:

- **No lock-in**: purchase AI Gateway Credits as you use them, with no obligation to renew your commitment.
- **No free tier**: once you create a paid account, you will not receive $5 of AI Gateway Credits per month.

## AI Gateway Rates

No matter whether you access the AI Gateway through a free or paid account, you'll pay the AI Gateway rates listed in the Models section

The charge for each request depends on the AI provider and model you select, and the number of input and output tokens processed. **You'r

## Using a custom API key

The AI Gateway also supports [using a custom API key](/docs/ai-gateway/byok) for any provider listed in our catalog.
If you use a custom API key, there is no markup or fee from AI Gateway.

## View your AI Gateway Credits balance

To view your balance:

1. Go to the **AI Gateway** tab of your Vercel dashboard.
2. On the upper right corner, you will see your AI Gateway Credits balance displayed.

## Top up your AI Gateway Credits

To add AI Gateway Credits:

1. Go to the **AI Gateway** tab of your Vercel dashboard.
2. In the upper right corner, click on the button that shows your AI Gateway Credits balance.
3. In the dialog that appears, you can select the amount of AI Gateway Credits you want to add.
4. Click on **Continue to Payment**.
5. Choose your payment method and click on **Confirm and Pay** to complete your purchase.

---
title: "Provider Options"
description: "Configure provider routing, ordering, and fallback behavior in Vercel AI Gateway"
last_updated: "2026-01-16T02:19:25.673Z"
source: "https://vercel.com/docs/ai-gateway/provider-options"
---

# Provider Options

AI Gateway can route your AI model requests across multiple AI providers. Each provider offers different models, pricing, and performance

With the Gateway Provider Options however, you have control over the routing order and fallback behavior of the models.

> **💡 Note:** If you want to customize individual AI model provider settings rather than
> general AI Gateway behavior, please refer to the model-specific provider
> options in the [AI SDK
```

> documentation](https://ai-sdk.dev/docs/foundations/prompts#provider-options).

Basic provider ordering

You can use the `order` array to specify the sequence in which providers should be attempted. Providers are specified using their `slug`

You can also copy the provider slug using the copy button next to a provider's name on a model's detail page. In the Vercel Dashboard:

1. Click the **AI Gateway** tab,
2. Then, click the **Model List** sub-tab on the left
3. Click a model entry in the list.

The bottom section of the page lists the available providers for that model. The copy button next to a provider's name will copy their sl

Getting started with adding a provider option

- ### Install the AI SDK package
  First, ensure you have the necessary package installed:
  ```bash filename="Terminal"
  pnpm install ai
  ```

- ### Configure the provider order in your request
  Use the `providerOptions.gateway.order` configuration:
  ```typescript filename="app/api/chat/route.ts" {7-11}
  import { streamText } from 'ai';

  export async function POST(request: Request) {
    const { prompt } = await request.json();

    const result = streamText({
      model: 'anthropic/claude-sonnet-4.5',
      prompt,
      providerOptions: {
        gateway: {
          order: ['bedrock', 'anthropic'], // Try Amazon Bedrock first, then Anthropic
        },
      },
    });

    return result.toUIMessageStreamResponse();
  }
  ```
  In this example:
  - The gateway will first attempt to use Amazon Bedrock to serve the Claude 4 Sonnet model
  - If Amazon Bedrock is unavailable or fails, it will fall back to Anthropic
  - Other providers (like Vertex AI) are still available but will only be used after the specified providers

- ### Test the routing behavior
  You can monitor which provider you used by checking the provider metadata in the response.
  ```typescript filename="app/api/chat/route.ts" {16-17}
  import { streamText } from 'ai';

  export async function POST(request: Request) {
    const { prompt } = await request.json();

    const result = streamText({
      model: 'anthropic/claude-sonnet-4.5',
      prompt,
      providerOptions: {
        gateway: {
          order: ['bedrock', 'anthropic'],
        },
      },
    });

    // Log which provider was actually used
    console.log(JSON.stringify(await result.providerMetadata, null, 2));

    return result.toUIMessageStreamResponse();
  }
  ```

Example provider metadata output

```json
{
  "zai": {},
  "gateway": {
    "routing": {
      "originalModelId": "zai/glm-4.6",
      "resolvedProvider": "zai",
      "resolvedProviderApiModelId": "glm-4.6",
      "internalResolvedModelId": "zai:glm-4.6",
      "fallbacksAvailable": [],
      "internalReasoning": "Selected zai as preferred provider for glm-4.6. 0 fallback(s) available: ",
      "planningReasoning": "System credentials planned for: zai. Total execution order: zai(system)",
      "canonicalSlug": "zai/glm-4.6",
      "finalProvider": "zai",
      "attempts": [
        {
          "provider": "zai",
          "internalModelId": "zai:glm-4.6",
          "providerApiModelId": "glm-4.6",
          "credentialType": "system",
          "success": true,
          "startTime": 458753.407267,
          "endTime": 459891.705775
        }
      ],
```

```
      "modelAttemptCount": 1,
      "modelAttempts": [
        {
          "modelId": "zai/glm-4.6",
          "canonicalSlug": "zai/glm-4.6",
          "success": true,
          "providerAttemptCount": 1,
          "providerAttempts": [
            {
              "provider": "zai",
              "internalModelId": "zai:glm-4.6",
              "providerApiModelId": "glm-4.6",
              "credentialType": "system",
              "success": true,
              "startTime": 458753.407267,
              "endTime": 459891.705775
            }
          ]
        }
      ],
      "totalProviderAttemptCount": 1
    },
    "cost": "0.0045405",
    "marketCost": "0.0045405",
    "generationId": "gen_01K8KPJ0FZA7172X6CSGNZGDWY"
  }
}
```

The `gateway.cost` value is the amount debited from your AI Gateway Credits balance for this request. It is returned as a decimal string.

In cases where your request encounters issues with one or more providers or if your BYOK credentials fail, you'll find error detail in th

```json
"attempts": [
  {
    "provider": "novita",
    "internalModelId": "novita:zai-org/glm-4.5",
    "providerApiModelId": "zai-org/glm-4.5",
    "credentialType": "byok",
    "success": false,
    "error": "Unauthorized",
    "startTime": 1754639042520,
    "endTime": 1754639042710
  },
  {
    "provider": "novita",
    "internalModelId": "novita:zai-org/glm-4.5",
    "providerApiModelId": "zai-org/glm-4.5",
    "credentialType": "system",
    "success": true,
    "startTime": 1754639042710,
    "endTime": 1754639043353
  }
]
```

Restrict providers with the `only` filter

Use the `only` array to restrict routing to a specific subset of providers. Providers are specified by their slug and are matched against

```typescript filename="app/api/chat/route.ts" {9-12}
import { streamText } from 'ai';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const result = streamText({
    model: 'anthropic/claude-sonnet-4.5',
    prompt,
    providerOptions: {
      gateway: {
        only: ['bedrock', 'anthropic'], // Only consider these providers.
        // This model is also available via 'vertex', but it won't be considered.
      },
    },
  });

  return result.toUIMessageStreamResponse();
}
```

In this example:

- **Restriction**: Only `bedrock` and `anthropic` will be considered for routing and fallbacks.
- **Error on mismatch**: If none of the specified providers are available for the model, the request fails with an error indicating the a

Using `only` together with `order`

When both `only` and `order` are provided, the `only` filter is applied first to define the allowed set, and then `order` defines the pri

```typescript filename="app/api/chat/route.ts" {9-12}
import { streamText } from 'ai';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const result = streamText({
    model: 'anthropic/claude-sonnet-4.5',
    prompt,
```

```
    providerOptions: {
      gateway: {
        only: ['anthropic', 'vertex'],
        order: ['vertex', 'bedrock', 'anthropic'],
      },
    },
  });

  return result.toUIMessageStreamResponse();
}
```

The final order will be `vertex → anthropic` (providers listed in `order` but not in `only` are ignored).

Model fallbacks with the `models` option

You can specify fallback models that will be tried in order if the primary model fails or is unavailable. This provides model-level fallb

```typescript filename="app/api/chat/route.ts" {8-11}
import { streamText } from 'ai';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const result = streamText({
    model: 'openai/gpt-4o', // Primary model
    prompt,
    providerOptions: {
      gateway: {
        models: ['openai/gpt-5-nano', 'gemini-2.0-flash'], // Fallback models
      },
    },
  });

  return result.toUIMessageStreamResponse();
}
```

In this example:

- The gateway will first attempt to use the primary model (`openai/gpt-4o`)
- If the primary model fails or is unavailable, it will try `openai/gpt-5-nano`
- If that also fails, it will try `gemini-2.0-flash`
- The response will come from the first model that succeeds

Combining `models` with provider options

You can combine model fallbacks with provider routing options for comprehensive failover strategies:

```typescript filename="app/api/chat/route.ts" {9-13}
import { streamText } from 'ai';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const result = streamText({
    model: 'openai/gpt-4o',
    prompt,
    providerOptions: {
      gateway: {
        models: ['openai/gpt-5-nano', 'anthropic/claude-sonnet-4.5'],
        order: ['azure', 'openai'], // Provider preference for each model
      },
    },
  });

  return result.toUIMessageStreamResponse();
}
```

This configuration will:

1. Try `openai/gpt-4o` via Azure first, then OpenAI
2. If both fail, try `openai/gpt-5-nano` via Azure first, then OpenAI
3. If those fail, try `anthropic/claude-sonnet-4` via available providers

Combining AI Gateway provider options with provider-specific options

You can combine AI Gateway provider options with provider-specific options. This allows you to control both the routing behavior and prov

```typescript filename="app/api/chat/route.ts"
import { streamText } from 'ai';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const result = streamText({
    model: 'anthropic/claude-sonnet-4.5',
    prompt,
    providerOptions: {
      anthropic: {
        thinkingBudget: 0.001,
      },
      gateway: {
        order: ['vertex'],
      },
    },
  });

  return result.toUIMessageStreamResponse();
```

}
```

In this example:

- We're using an Anthropic model (e.g. Claude 4 Sonnet) but accessing it through Vertex AI
- The Anthropic-specific options still apply to the model:
  - `thinkingBudget` sets a cost limit of $0.001 per request for the Claude model
- You can read more about provider-specific options in the [AI SDK documentation](https://ai-sdk.dev/docs/foundations/prompts#provider-op

## Request-scoped BYOK

You can pass your own provider credentials on a per-request basis using the `byok` option in `providerOptions.gateway`. This allows you t

```typescript filename="app/api/chat/route.ts" {9-13}
import { streamText } from 'ai';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const result = streamText({
    model: 'anthropic/claude-sonnet-4.5',
    prompt,
    providerOptions: {
      gateway: {
        byok: {
          anthropic: [{ apiKey: process.env.ANTHROPIC_API_KEY }],
        },
      },
    },
  });

  return result.toUIMessageStreamResponse();
}
```

For detailed information about credential structures, multiple credentials, and usage with the OpenAI-compatible API, see the [BYOK docum

Reasoning

For models that support reasoning (also known as "thinking"), you can use
`providerOptions` to configure reasoning behavior. The example below shows
how to control the computational effort and summary detail level when using OpenAI's `gpt-oss-120b` model.

For more details on reasoning support across different models and providers, see the [AI SDK providers documentation](https://ai-sdk.dev/

```typescript filename="app/api/chat/route.ts" {9-12}
import { streamText } from 'ai';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const result = streamText({
    model: 'openai/gpt-oss-120b',
    prompt,
    providerOptions: {
      openai: {
        reasoningEffort: 'high',
        reasoningSummary: 'detailed',
      },
    },
  });

  return result.toUIMessageStreamResponse();
}
```

Note: For `openai/gpt-5` and `openai/gpt-5.1` models, you must set both `reasoningEffort` and `reasoningSummary` in `providerOptions`

```typescript
providerOptions: {
  openai: {
    reasoningEffort: 'high', // or 'minimal', 'low', 'medium', 'none'
    reasoningSummary: 'detailed', // or 'auto', 'concise'
  },
}
```

Available providers

You can view the available models for a provider
in the **Model List** section under
the **[AI Gateway](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fai\&title=Go+to+AI+Gateway)** tab in your Vercel dashboard
or in the public [models page](https://vercel.com/ai-gateway/models).

Slug	**Name**	**Website**
`alibaba`	Alibaba Cloud	[alibabacloud.com](https://www.al
`anthropic`	[Anthropic](https://ai-sdk.dev/providers/ai-sdk-providers/anthropic)	[anthropic.com](https://anthropic
`arcee-ai`	Arcee AI	arcee.ai
`azure`	[Azure](https://ai-sdk.dev/providers/ai-sdk-providers/azure)	[ai.azure.com](https://ai.azure.c
`baseten`	[Baseten](https://ai-sdk.dev/providers/openai-compatible-providers/baseten)	[baseten.co](https://www.baseten.
`bedrock`	[Amazon Bedrock](https://ai-sdk.dev/providers/ai-sdk-providers/amazon-bedrock)	[aws.amazon.com/bedrock](https://
`bfl`	[Black Forest Labs](https://ai-sdk.dev/providers/ai-sdk-providers/black-forest-labs)	bfl.ai
`bytedance`	ByteDance	[byteplus.com](https://www.bytepl
`cerebras`	[Cerebras](https://ai-sdk.dev/providers/ai-sdk-providers/cerebras)	[cerebras.net](https://www.cerebr
`cohere`	[Cohere](https://ai-sdk.dev/providers/ai-sdk-providers/cohere)	cohere.com
`crusoe`	Crusoe	crusoe.ai
`deepinfra`	[DeepInfra](https://ai-sdk.dev/providers/ai-sdk-providers/deepinfra)	[deepinfra.com](https://deepinfra
`deepseek`	[DeepSeek](https://ai-sdk.dev/providers/ai-sdk-providers/deepseek)	[deepseek.ai](https://deepseek.ai
```

| `fireworks` | [Fireworks](https://ai-sdk.dev/providers/ai-sdk-providers/fireworks) | [fireworks.ai](https://fireworks.
| `google` | [Google](https://ai-sdk.dev/providers/ai-sdk-providers/google-generative-ai) | [ai.google.dev](https://ai.google
| `groq` | [Groq](https://ai-sdk.dev/providers/ai-sdk-providers/groq) | groq.com
| `inception` | Inception | [inceptionlabs.ai](https://incept
| `meituan` | Meituan | longcat.ai
| `minimax` | MiniMax | [minimax.io](https://www.minimax.
| `mistral` | [Mistral](https://ai-sdk.dev/providers/ai-sdk-providers/mistral) | mistral.ai
| `moonshotai` | Moonshot AI | [moonshot.ai](https://www.moonsho
| `morph` | Morph | [morphllm.com](https://morphllm.c
| `nebius` | Nebius | nebius.com
| `novita` | Novita | novita.ai
| `openai` | [OpenAI](https://ai-sdk.dev/providers/ai-sdk-providers/openai) | openai.com
| `parasail` | Parasail | [parasail.com](https://www.parasa
| `perplexity` | [Perplexity](https://ai-sdk.dev/providers/ai-sdk-providers/perplexity) | [perplexity.ai](https://www.perpl
| `prodia` | Prodia | [prodia.com](https://www.prodia.c
| `sambanova` | SambaNova | [sambanova.ai](https://sambanova.
| `streamlake` | StreamLake | [streamlake.ai](https://streamlak
| `togetherai` | [Together AI](https://ai-sdk.dev/providers/ai-sdk-providers/togetherai) | [together.ai](https://together.ai
| `vercel` | [Vercel](https://ai-sdk.dev/providers/ai-sdk-providers/vercel) | [v0.app](https://v0.app/docs/api/
| `vertex` | [Vertex AI](https://ai-sdk.dev/providers/ai-sdk-providers/google-vertex) | [cloud.google.com/vertex-ai](http
| `voyage` | [Voyage AI](https://ai-sdk.dev/providers/community-providers/voyage-ai) | [voyageai.com](https://www.voyage
| `xai` | [xAI](https://ai-sdk.dev/providers/ai-sdk-providers/xai) | x.ai
| `zai` | Z.ai | [z.ai](https://z.ai/model-api)

> **💡 Note:** Provider availability may vary by model. Some models may only be available
> through specific providers or may have different capabilities depending on the
> provider used.

---
title: "Python"
description: "Use the AI Gateway with Python through OpenAI or Anthropic SDKs."
last_updated: "2026-01-16T02:19:25.682Z"
source: "https://vercel.com/docs/ai-gateway/python"

Python

The AI Gateway supports Python through two official SDKs.

[OpenAI-compatible API](/docs/ai-gateway/openai-compat) - Use the [OpenAI Python SDK](https://github.com/openai/openai-python) with A

- [Chat completions and streaming](/docs/ai-gateway/openai-compat/chat-completions)
- [Tool calls](/docs/ai-gateway/openai-compat/tool-calls) and [structured outputs](/docs/ai-gateway/openai-compat/structured-outputs)
- [Image attachments](/docs/ai-gateway/openai-compat/advanced#image-inputs) and [PDFs](/docs/ai-gateway/openai-compat/advanced#pdf-inputs
- [Embeddings](/docs/ai-gateway/openai-compat/embeddings) and [prompt caching](/docs/ai-gateway/openai-compat/advanced#prompt-caching)
- [Reasoning](/docs/ai-gateway/openai-compat/advanced#reasoning) and [provider routing](/docs/ai-gateway/openai-compat/advanced#provider-

[Anthropic-compatible API](/docs/ai-gateway/anthropic-compat) - Use the [Anthropic Python SDK](https://github.com/anthropics/anthropi

- [Messages and streaming](/docs/ai-gateway/anthropic-compat/messages)
- [Tool calls](/docs/ai-gateway/anthropic-compat/tool-calls) and [extended thinking](/docs/ai-gateway/anthropic-compat/advanced#extended-
- [Web search](/docs/ai-gateway/anthropic-compat/advanced#web-search) and [file attachments](/docs/ai-gateway/anthropic-compat/file-attac
- [Claude Code integration](/docs/ai-gateway/claude-code)

Installation

Install your preferred SDK:

```bash
# OpenAI SDK (for OpenAI-compatible API)
pip install openai

# Anthropic SDK (for Anthropic-compatible API)
pip install anthropic
```

Quick start

OpenAI SDK

```python filename="openai-quickstart.py"
import os
from openai import OpenAI

client = OpenAI(
    api_key=os.getenv('AI_GATEWAY_API_KEY'),
    base_url='https://ai-gateway.vercel.sh/v1'
)

response = client.chat.completions.create(
    model='anthropic/claude-sonnet-4.5',
    messages=[
        {'role': 'user', 'content': 'Hello, world!'}
    ]
)

print(response.choices[0].message.content)
```

Anthropic SDK

```python filename="anthropic-quickstart.py"
import os
import anthropic

client = anthropic.Anthropic(
    api_key=os.getenv('AI_GATEWAY_API_KEY'),
    base_url='https://ai-gateway.vercel.sh'
)
```

```
message = client.messages.create(
    model='anthropic/claude-sonnet-4.5',
    max_tokens=1024,
    messages=[
        {'role': 'user', 'content': 'Hello, world!'}
    ]
)

print(message.content[0].text)
```

Authentication

Both SDKs support the same authentication methods:

- **API key**: Use your AI Gateway API key
- **OIDC token**: Use your Vercel OIDC token (for deployments on Vercel)

```python filename="authentication.py"
import os

# Option 1: API key (recommended for local development)
api_key = os.getenv('AI_GATEWAY_API_KEY')

# Option 2: OIDC token (for Vercel deployments)
# api_key = os.getenv('VERCEL_OIDC_TOKEN')

# You can also use both with a fallback
api_key = os.getenv('AI_GATEWAY_API_KEY') or os.getenv('VERCEL_OIDC_TOKEN')
```

Framework integrations

Several Python frameworks have dedicated AI Gateway integrations:

- **[Pydantic AI](/docs/ai-gateway/framework-integrations/pydantic-ai)** - Agent framework with native `VercelProvider` for structured ou
- **[LlamaIndex](/docs/ai-gateway/framework-integrations/llamaindex)** - Build knowledge assistants with the `llama-index-llms-vercel-ai-
- **[LiteLLM](/docs/ai-gateway/framework-integrations/litellm)** - Unified LLM interface with native `vercel_ai_gateway/` model prefix

See [Framework Integrations](/docs/ai-gateway/framework-integrations) for the full list.

---
title: "Usage & Billing"
description: "Monitor your AI Gateway credit balance, usage, and generation details."
last_updated: "2026-01-16T02:19:25.699Z"
source: "https://vercel.com/docs/ai-gateway/usage"

Usage & Billing

The AI Gateway provides endpoints to monitor your credit balance, track usage,
and retrieve detailed information about specific generations.

Base URL

The Usage & Billing API is available at the following base URL:

```
https://ai-gateway.vercel.sh/v1
```

Supported endpoints

The AI Gateway supports the following Usage & Billing endpoints:

- [`GET /credits`](#credits) - Check your credit balance and usage information
- [`GET /generation`](#generation-lookup) - Retrieve detailed information about a specific generation

Credits

Check your AI Gateway credit balance and usage information.

Endpoint

```
GET /credits
```

Example request

TypeScript

```typescript filename="credits.ts"
const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const response = await fetch('https://ai-gateway.vercel.sh/v1/credits', {
  method: 'GET',
  headers: {
    Authorization: `Bearer ${apiKey}`,
    'Content-Type': 'application/json',
  },
});

const credits = await response.json();
console.log(credits);
```

Python

```python filename="credits.py"
import os
import requests

api_key = os.getenv("AI_GATEWAY_API_KEY") or os.getenv("VERCEL_OIDC_TOKEN")

response = requests.get(
    "https://ai-gateway.vercel.sh/v1/credits",
    headers={
        "Authorization": f"Bearer {api_key}",
        "Content-Type": "application/json",
    },
)

credits = response.json()
print(credits)
```

Sample response

```json
{
  "balance": "95.50",
  "total_used": "4.50"
}
```

Response fields

- `balance`: The remaining credit balance
- `total_used`: The total amount of credits used

Generation lookup

Retrieve detailed information about a specific generation by its ID. This endpoint allows you to look up usage data, costs, and metadata

Endpoint

```
GET /generation?id={generation_id}
```

Parameters

- `id` (required): The generation ID to look up (format: `gen_<ulid>`)

Example request

TypeScript

```typescript filename="generation-lookup.ts"
const generationId = 'gen_01ARZ3NDEKTSV4RRFFQ69G5FAV';

const response = await fetch(
  `https://ai-gateway.vercel.sh/v1/generation?id=${generationId}`,
  {
    method: 'GET',
    headers: {
      Authorization: `Bearer ${process.env.AI_GATEWAY_API_KEY}`,
      'Content-Type': 'application/json',
    },
  },
);

const generation = await response.json();
console.log(generation);
```

Python

```python filename="generation-lookup.py"
import os
import requests

generation_id = 'gen_01ARZ3NDEKTSV4RRFFQ69G5FAV'

response = requests.get(
    f"https://ai-gateway.vercel.sh/v1/generation?id={generation_id}",
    headers={
        "Authorization": f"Bearer {os.getenv('AI_GATEWAY_API_KEY')}",
        "Content-Type": "application/json",
    },
)

generation = response.json()
print(generation)
```

Sample response

```json
{
  "data": {
    "id": "gen_01ARZ3NDEKTSV4RRFFQ69G5FAV",
    "total_cost": 0.00123,
    "usage": 0.00123,
    "created_at": "2024-01-01T00:00:00.000Z",
    "model": "gpt-4",
    "is_byok": false,
```

```
    "provider_name": "openai",
    "streamed": true,
    "latency": 200,
    "generation_time": 1500,
    "tokens_prompt": 100,
    "tokens_completion": 50,
    "native_tokens_prompt": 100,
    "native_tokens_completion": 50,
    "native_tokens_reasoning": 0,
    "native_tokens_cached": 0
  }
}
```

Response fields

- `id`: The generation ID
- `total_cost`: Total cost in USD for this generation
- `usage`: Usage cost (same as total_cost)
- `created_at`: ISO 8601 timestamp when the generation was created
- `model`: Model identifier used for this generation
- `is_byok`: Whether this generation used Bring Your Own Key credentials
- `provider_name`: The provider that served this generation
- `streamed`: Whether this generation used streaming (`true` for streamed responses, `false` otherwise)
- `latency`: Time to first token in milliseconds
- `generation_time`: Total generation time in milliseconds
- `tokens_prompt`: Number of prompt tokens
- `tokens_completion`: Number of completion tokens
- `native_tokens_prompt`: Native prompt tokens (provider-specific)
- `native_tokens_completion`: Native completion tokens (provider-specific)
- `native_tokens_reasoning`: Reasoning tokens used (if applicable)
- `native_tokens_cached`: Cached tokens used (if applicable)

> **💡 Note:** **Generation IDs:** Generation IDs are included in chat completion responses
> as the
> [`id`](https://platform.openai.com/docs/api-reference/chat/object#chat/object-id)
> field as well as in the provider metadata returned in the response.

---
title: "Web Search"
description: "Enable AI models to search the web for current information using built-in tools through AI Gateway."
last_updated: "2026-01-16T02:19:25.733Z"
source: "https://vercel.com/docs/ai-gateway/web-search"

Web Search

AI Gateway provides built-in web search capabilities that allow AI models to access current information from the web. This is useful when

AI Gateway supports two types of web search:

- **Search for all providers**: Use [Perplexity Search](#using-perplexity-search) with any model regardless of provider. This gives you c
- **Provider-specific search**: Use native web search tools from [Anthropic](#anthropic-web-search), [OpenAI](#openai-web-search), or [Go

Using Perplexity Search

The `perplexitySearch` tool can be used with any model regardless of the model provider or creator. This makes it a flexible option when

To use Perplexity Search, import `gateway` from `@ai-sdk/gateway` and pass `gateway.tools.perplexitySearch()` to the `tools` parameter. W

> **💡 Note:** Perplexity web search requests are charged at $5 per 1,000 requests. See
> [Perplexity's pricing](https://docs.perplexity.ai/getting-started/pricing) for
> more details.

streamText

```typescript filename="perplexity-web-search.ts" {10-12}
import { streamText } from 'ai';
import { gateway } from '@ai-sdk/gateway';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const result = streamText({
    model: 'openai/gpt-5.2', // Works with any model, not just Perplexity
    prompt,
    tools: {
      perplexity_search: gateway.tools.perplexitySearch(),
    },
  });

  for await (const part of result.fullStream) {
    if (part.type === 'text-delta') {
      process.stdout.write(part.text);
    } else if (part.type === 'tool-call') {
      console.log('Tool call:', part.toolName);
    } else if (part.type === 'tool-result') {
      console.log('Search results received');
    }
  }

  return result.toDataStreamResponse();
}
```

generateText

```typescript filename="perplexity-web-search.ts" {10-12}
import { generateText } from 'ai';
```

```
import { gateway } from '@ai-sdk/gateway';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const { text } = await generateText({
    model: 'openai/gpt-5.2', // Works with any model, not just Perplexity
    prompt,
    tools: {
      perplexity_search: gateway.tools.perplexitySearch(),
    },
  });

  return Response.json({ text });
}
```

Perplexity parameters

You can configure the `perplexitySearch` tool with these parameters:

- `maxResults`: Number of results to return (1-20). Defaults to 10.
- `maxTokens`: Total token budget across all results. Defaults to 25,000, max 1,000,000.
- `maxTokensPerPage`: Tokens extracted per webpage. Defaults to 2,048.
- `country`: ISO 3166-1 alpha-2 country code (e.g., ``'US'`, `'GB'`) for regional results.
- `searchLanguageFilter`: ISO 639-1 language codes (e.g., `['en', 'fr']`). Max 10 codes.
- `searchDomainFilter`: Domains to include (e.g., `['reuters.com']`) or exclude with `-` prefix (e.g., `['-reddit.com']`). Max 20 domains
- `searchRecencyFilter`: Filter by content recency. Values: ``day'`, `'week'`, `'month'`, or `'year'`.

streamText

```typescript filename="perplexity-web-search-params.ts" {10-20}
import { streamText } from 'ai';
import { gateway } from '@ai-sdk/gateway';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const result = streamText({
    model: 'openai/gpt-5.2',
    prompt,
    tools: {
      perplexity_search: gateway.tools.perplexitySearch({
        maxResults: 5,
        maxTokens: 50000,
        maxTokensPerPage: 2048,
        country: 'US',
        searchLanguageFilter: ['en'],
        searchDomainFilter: ['reuters.com', 'bbc.com', 'nytimes.com'],
        searchRecencyFilter: 'week',
      }),
    },
  });

  return result.toDataStreamResponse();
}
```

generateText

```typescript filename="perplexity-web-search-params.ts" {10-20}
import { generateText } from 'ai';
import { gateway } from '@ai-sdk/gateway';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const { text } = await generateText({
    model: 'openai/gpt-5.2',
    prompt,
    tools: {
      perplexity_search: gateway.tools.perplexitySearch({
        maxResults: 5,
        maxTokens: 50000,
        maxTokensPerPage: 2048,
        country: 'US',
        searchLanguageFilter: ['en'],
        searchDomainFilter: ['reuters.com', 'bbc.com', 'nytimes.com'],
        searchRecencyFilter: 'week',
      }),
    },
  });

  return Response.json({ text });
}
```

Provider-specific search

Use native web search tools from Anthropic, OpenAI, or Google. These tools are optimized for their respective providers and may offer add

> **💡 Note:** Pricing for provider-specific web search tools depends on the model you use.
> See the Web Search price column on the [model detail
> pages](https://vercel.com/ai-gateway/models) for exact pricing.

Anthropic web search

For Anthropic models, you can use the native [web search tool](https://platform.claude.com/docs/en/agents-and-tools/tool-use/web-search-t

streamText

```typescript filename="anthropic-web-search.ts" {10-12}
import { streamText } from 'ai';
import { anthropic } from '@ai-sdk/anthropic';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const result = streamText({
    model: 'anthropic/claude-opus-4.5',
    prompt,
    tools: {
      web_search: anthropic.tools.webSearch_20250305(),
    },
  });

  return result.toDataStreamResponse();
}
```

generateText

```typescript filename="anthropic-web-search.ts" {10-12}
import { generateText } from 'ai';
import { anthropic } from '@ai-sdk/anthropic';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const { text } = await generateText({
    model: 'anthropic/claude-opus-4.5',
    prompt,
    tools: {
      web_search: anthropic.tools.webSearch_20250305(),
    },
  });

  return Response.json({ text });
}
```

Anthropic parameters

The following parameters are supported:

- `maxUses`: Maximum number of web searches Claude can perform during the conversation.
- `allowedDomains`: Optional list of domains Claude is allowed to search. If provided, searches will be restricted to these domains.
- `blockedDomains`: Optional list of domains Claude should avoid when searching.
- `userLocation`: Optional user location information to provide geographically relevant search results.

streamText

```typescript filename="anthropic-web-search-params.ts" {10-23}
import { streamText } from 'ai';
import { anthropic } from '@ai-sdk/anthropic';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const result = streamText({
    model: 'anthropic/claude-opus-4.5',
    prompt,
    tools: {
      web_search: anthropic.tools.webSearch_20250305({
        maxUses: 3,
        allowedDomains: ['techcrunch.com', 'wired.com'],
        blockedDomains: ['example-spam-site.com'],
        userLocation: {
          type: 'approximate',
          country: 'US',
          region: 'California',
          city: 'San Francisco',
          timezone: 'America/Los_Angeles',
        },
      }),
    },
  });

  return result.toDataStreamResponse();
}
```

generateText

```typescript filename="anthropic-web-search-params.ts" {10-23}
import { generateText } from 'ai';
import { anthropic } from '@ai-sdk/anthropic';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const { text } = await generateText({
    model: 'anthropic/claude-opus-4.5',
    prompt,
    tools: {
      web_search: anthropic.tools.webSearch_20250305({
        maxUses: 3,
        allowedDomains: ['techcrunch.com', 'wired.com'],
        blockedDomains: ['example-spam-site.com'],
        userLocation: {
```

```
          type: 'approximate',
          country: 'US',
          region: 'California',
          city: 'San Francisco',
          timezone: 'America/Los_Angeles',
        },
      }),
    },
  });

  return Response.json({ text });
}
```

For more details on using the Anthropic-compatible API directly, see the [Anthropic advanced features](/docs/ai-gateway/anthropic-compat/

OpenAI web search

For OpenAI models, you can use the native [web search tool](https://platform.openai.com/docs/guides/tools-web-search) provided by the `@a

streamText

```typescript filename="openai-web-search.ts" {10-12}
import { streamText } from 'ai';
import { openai } from '@ai-sdk/openai';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const result = streamText({
    model: 'openai/gpt-5.2',
    prompt,
    tools: {
      web_search: openai.tools.webSearch({}),
    },
  });

  return result.toDataStreamResponse();
}
```

generateText

```typescript filename="openai-web-search.ts" {10-12}
import { generateText } from 'ai';
import { openai } from '@ai-sdk/openai';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const { text } = await generateText({
    model: 'openai/gpt-5.2',
    prompt,
    tools: {
      web_search: openai.tools.webSearch({}),
    },
  });

  return Response.json({ text });
}
```

Google web search

For Google Gemini models, you can use [Grounding with Google Search](https://docs.cloud.google.com/vertex-ai/generative-ai/docs/grounding

Google Vertex

Import `vertex` from `@ai-sdk/google-vertex` and pass `vertex.tools.googleSearch({})` to the `tools` parameter. For users who need zero d

streamText

```typescript filename="google-vertex-web-search.ts" {10-12}
import { streamText } from 'ai';
import { vertex } from '@ai-sdk/google-vertex';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const result = streamText({
    model: 'google/gemini-3-flash',
    prompt,
    tools: {
      google_search: vertex.tools.googleSearch({}),
    },
  });

  return result.toDataStreamResponse();
}
```

generateText

```typescript filename="google-vertex-web-search.ts" {10-12}
import { generateText } from 'ai';
import { vertex } from '@ai-sdk/google-vertex';

export async function POST(request: Request) {
  const { prompt } = await request.json();
```

```
  const { text } = await generateText({
    model: 'google/gemini-3-flash',
    prompt,
    tools: {
      google_search: vertex.tools.googleSearch({}),
    },
  });

  return Response.json({ text });
}
```

Enterprise web search

For users who need zero data retention, you can use [Enterprise Web Grounding](https://docs.cloud.google.com/vertex-ai/generative-ai/docs

> **💡 Note:** Enterprise web search uses indexed content that is a subset of the full web.
> Use Google search for more up-to-date and comprehensive results.

streamText

```typescript filename="enterprise-web-grounding.ts" {10-12}
import { streamText } from 'ai';
import { vertex } from '@ai-sdk/google-vertex';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const result = streamText({
    model: 'google/gemini-3-flash',
    prompt,
    tools: {
      enterprise_web_search: vertex.tools.enterpriseWebSearch({}),
    },
  });

  return result.toDataStreamResponse();
}
```

generateText

```typescript filename="enterprise-web-grounding.ts" {10-12}
import { generateText } from 'ai';
import { vertex } from '@ai-sdk/google-vertex';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const { text } = await generateText({
    model: 'google/gemini-3-flash',
    prompt,
    tools: {
      enterprise_web_search: vertex.tools.enterpriseWebSearch({}),
    },
  });

  return Response.json({ text });
}
```

Google AI Studio

Import `google` from `@ai-sdk/google` and pass `google.tools.googleSearch({})` to the `tools` parameter.

streamText

```typescript filename="google-ai-studio-web-search.ts" {10-12}
import { streamText } from 'ai';
import { google } from '@ai-sdk/google';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const result = streamText({
    model: 'google/gemini-3-flash',
    prompt,
    tools: {
      google_search: google.tools.googleSearch({}),
    },
  });

  return result.toDataStreamResponse();
}
```

generateText

```typescript filename="google-ai-studio-web-search.ts" {10-12}
import { generateText } from 'ai';
import { google } from '@ai-sdk/google';

export async function POST(request: Request) {
  const { prompt } = await request.json();

  const { text } = await generateText({
    model: 'google/gemini-3-flash',
    prompt,
    tools: {
      google_search: google.tools.googleSearch({}),
```

```
    },
  });

  return Response.json({ text });
}
```

---
title: "Zero Data Retention (ZDR)"
description: "Learn about zero data retention policies and how to enforce ZDR on a per-request basis with Vercel AI Gateway."
last_updated: "2026-01-16T02:19:25.742Z"
source: "https://vercel.com/docs/ai-gateway/zdr"

Zero Data Retention (ZDR)

Zero data retention (ZDR) is available for Vercel AI Gateway. There is an option to enforce zero data retention on a per request level on

Vercel

Vercel AI Gateway has a ZDR policy and does not retain prompts or sensitive data. User data is immediately and permanently deleted after

Providers

Vercel AI Gateway has agreements in place with specific providers for ZDR. A provider's default policy may not match with the status that

By default, Vercel AI Gateway does not route based on the data retention policy of providers.

Per request zero data retention (ZDR) enforcement

To restrict requests to only go through providers that state they provide zero data retention, use the `zeroDataRetention` parameter in `

If Vercel AI Gateway does not have a clear policy or agreement in place for a provider, we assume that the provider does not have a zero

If there are no providers available that have zero data retention agreements with Vercel AI Gateway, the request will fail with an error

This per request ZDR enforcement only applies for requests routed directly through Vercel AI Gateway (not BYOK). Since BYOK requests will

Using AI SDK

Set `zeroDataRetention` to `true` in `providerOptions`:

```typescript filename="zdr.ts" {9-13}
import type { GatewayProviderOptions } from '@ai-sdk/gateway';
import { streamText } from 'ai';
import 'dotenv/config';

async function main() {
  const result = streamText({
    model: 'zai/glm-4.6',
    prompt: 'Analyze this sensitive business data and provide insights.',
    providerOptions: {
      gateway: {
        zeroDataRetention: true, // For this request, use only ZDR compliant providers
      } satisfies GatewayProviderOptions,
    },
  });

  for await (const textPart of result.textStream) {
    process.stdout.write(textPart);
  }

  console.log();
  console.log(
    'Provider metadata:',
    JSON.stringify(await result.providerMetadata, null, 2),
  );
  console.log('Token usage:', await result.usage);
  console.log('Finish reason:', await result.finishReason);
}

main().catch(console.error);
```

Using OpenAI-compatible API

Set `zeroDataRetention` to `true` in `providerOptions`:

```typescript filename="zdr.ts" {19-23}
import OpenAI from 'openai';

const apiKey = process.env.AI_GATEWAY_API_KEY || process.env.VERCEL_OIDC_TOKEN;

const openai = new OpenAI({
  apiKey,
  baseURL: 'https://ai-gateway.vercel.sh/v1',
});

const completion = await openai.chat.completions.create({
  model: 'zai/glm-4.6',
  messages: [
    {
      role: 'user',
      content:
        'Tell me the history of the San Francisco Mission-style burrito in two paragraphs.',
    },
  ],
  providerOptions: {
    gateway: {
```

```
      zeroDataRetention: true, // Request only ZDR compliant providers
    },
  },
});
```

ZDR providers and policies

Only the following providers offer ZDR on Vercel AI Gateway. Please review each provider's ZDR policy carefully. A provider's default pol

- Amazon Bedrock
- Anthropic
- Baseten
- Cerebras
- DeepInfra
- Google Vertex

---
title: "AI SDK"
description: "TypeScript toolkit for building AI-powered applications with React, Next.js, Vue, Svelte and Node.js"
last_updated: "2026-01-16T02:19:25.756Z"
source: "https://vercel.com/docs/ai-sdk"

AI SDK

The [AI SDK](https://sdk.vercel.ai) is the TypeScript toolkit designed to help developers build AI-powered applications with [Next.js](ht

The AI SDK abstracts away the differences between model providers, eliminates boilerplate code for building chatbots, and allows you to g

Generating text

At the center of the AI SDK is [AI SDK Core](https://sdk.vercel.ai/docs/ai-sdk-core/overview), which provides a unified API to call any L

The following example shows how to generate text with the AI SDK using OpenAI's GPT-5:

```typescript
import { generateText } from 'ai';

const { text } = await generateText({
  model: 'openai/gpt-5.2',
  prompt: 'Explain the concept of quantum entanglement.',
});
```

The unified interface means that you can easily switch between providers by changing just two lines of code. For example, to use Anthropi

```typescript {2,5}
import { generateText } from 'ai';

const { text } = await generateText({
  model: 'anthropic/claude-opus-4.5',
  prompt: 'How many people will live in the world in 2040?',
});
```

Generating structured data

While text generation can be useful, you might want to generate structured JSON data. For example, you might want to extract information

The following example shows how to generate a type-safe recipe that conforms to a zod schema:

```ts
import { generateObject } from 'ai';
import { z } from 'zod';

const { object } = await generateObject({
  model: 'openai/gpt-5.2',
  schema: z.object({
    recipe: z.object({
      name: z.string(),
      ingredients: z.array(z.object({ name: z.string(), amount: z.string() })),
      steps: z.array(z.string()),
    }),
  }),
  prompt: 'Generate a lasagna recipe.',
});
```

Using tools with the AI SDK

The AI SDK supports tool calling out of the box, allowing it to interact with external systems and perform discrete tasks. The following

```ts
import { generateText, tool } from 'ai';

const { text } = await generateText({
  model: 'openai/gpt-5.2',
  prompt: 'What is the weather like today in San Francisco?',
  tools: {
    getWeather: tool({
      description: 'Get the weather in a location',
      inputSchema: z.object({
        location: z.string().describe('The location to get the weather for'),
      }),
      execute: async ({ location }) => ({
        location,
        temperature: 72 + Math.floor(Math.random() * 21) - 10,
      }),
```

```
    }),
  },
});
```

Getting started with the AI SDK

The AI SDK is available as a package. To install it, run the following command:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i ai
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i ai
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i ai
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i ai
    ```
  </Code>
</CodeBlock>

See the [AI SDK Getting Started](https://sdk.vercel.ai/docs/getting-started) guide for more information on how to get started with the AI

More resources

- [AI SDK documentation](https://ai-sdk.dev/docs)
- [AI SDK examples](https://ai-sdk.dev/cookbook)
- [AI SDK guides](https://ai-sdk.dev/cookbook/guides)
- [AI SDK templates](https://vercel.com/templates?type=ai)

---
title: "Alerts"
description: "Get notified when something"
last_updated: "2026-01-16T02:19:25.779Z"
source: "https://vercel.com/docs/alerts"

Alerts

Alerts let you know when something's wrong with your Vercel projects, like a spike in failed function invocations or unusual usage patter

By default, you'll be notified about:

- **Usage anomaly**: When your project's usage exceeds abnormal levels.
- **Error anomaly**: When your project's error rate of function invocations (those with a status code of 5xx) exceeds abnormal levels.

Alert types

Alert Type	Triggered when
Error Anomaly	Fires when your 5-minute error rate (5xx) is more than 4 standard deviations above your 24-hour average and exceeds
Usage Anomaly	Fires when your 5-minute usage is more than 4 standard deviations above your 24-hour average and exceeds the minimu

Configure alerts

Here's how to configure alerts for your projects:

1. First, head to your [Vercel dashboard](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fobservability%2Falerts).
2. Go to the **Observability** tab, find the **Alerts** tab, and click **Subscribe to Alerts**.
3. Then, pick how you'd like to be notified: [Email](#vercel-notifications), [Slack](#slack-integration), or [Webhook](#webhook).

Vercel Notifications

You can subscribe to alerts about anomalies through the standard [Vercel notifications](/docs/notifications), which will notify you throu

By default, users with team owner roles will receive notifications.

To enable notifications:

1. Go to your [Vercel dashboard](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fobservability%2Falerts), head to **Observability**, then *
2. Click **Subscribe to Alerts**.
3. Click **Manage** next to **Vercel Notifications**.
4. Select which alert you'd like to receive to each of the notification channels.

You can configure **your own** notification preferences in your [Vercel dashboard](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fsettings%

Slack integration

You'll need the correct permissions in your Slack workspace to install the Slack integration.

1. Install the Vercel [Slack integration](https://vercel.com/integrations/slack) if you haven't already.
2. Go to the Slack channel where you want alerts and run this command for alerts about usage and error anomalies:

   ```bash
   /vercel subscribe [team/project] alerts
   ```

   The dashboard will show you the exact command for your team or project.

Webhook

With webhooks, you can send alerts to any destination.

1. Go to your [Vercel dashboard](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fobservability%2Falerts), head to **Observability**, then *
2. Click **Subscribe to Alerts**.
3. Choose **Webhook**.
4. Fill out the webhook details:
   - Choose which projects to monitor
   - Add your endpoint URL

You can also set this up through [account webhooks](/docs/webhooks#account-webhooks), just pick the events you want under **Observability

Webhooks payload

To learn more about the webhook payload, see the [Webhooks API Reference](/docs/webhooks/webhooks-api):

- [Alerts triggered](/docs/webhooks/webhooks-api#alerts.triggered)

---
title: "Tracking custom events"
description: "Learn how to send custom analytics events from your application."
last_updated: "2026-01-16T02:19:25.878Z"
source: "https://vercel.com/docs/analytics/custom-events"

Tracking custom events

Vercel Web Analytics allows you to track custom events in your application using the `track()` function.
This is useful for tracking user interactions, such as button clicks, form submissions, or purchases.

> **💡 Note:** Make sure you have `@vercel/analytics` version 1.1.0 or later
> [installed](/docs/analytics/quickstart#add-@vercel/analytics-to-your-project).

Tracking a client-side event

> For \['nextjs', 'nextjs-app', 'sveltekit', 'nuxt', 'remix', 'other']:

To track an event:

1. Make sure you have `@vercel/analytics` version 1.1.0 or later [installed](/docs/analytics/quickstart#add-@vercel/analytics-to-your-pro
2. Import `{ track }` from `@vercel/analytics`.
3. In most cases you will want to track an event when a user performs an action, such as clicking a button or submitting a form, so you s
4. Call `track` and pass in a string representing the event name as the first argument. You can also pass [custom data](#tracking-an-even

   ```ts filename="component.ts"
   import { track } from '@vercel/analytics';

   // Call this function when a user clicks a button or performs an action you want to track
   track('Signup');
   ```

> For \['html']:

1. Add following snippet before the script tag in your HTML file:

   ```html filename="index.html"
   <script>
     window.va =
       window.va ||
       function () {
         (window.vaq = window.vaq || []).push(arguments);
       };
   </script>
   {/* Place it above this script tag when already added */}
   <script defer src="/_vercel/insights/script.js"></script>
   ```

2. In most cases you will want to track an event when a user performs an action, such as clicking a button or submitting a form, so you s
   ```html filename="index.html"
   va('event', { name: 'Signup' });
   ```
   For example, if you have a button that says **Sign Up**, you can track an event when the user clicks the button:

```html filename="index.html"
<div>
  <button onclick="va('event', { name: 'Signup' })">Sign Up</button>
</div>
```

> For \['nextjs', 'nextjs-app', 'sveltekit', 'nuxt', 'remix']:

For example, if you have a button that says **Sign Up**, you can track an event when the user clicks the button:

```ts filename="components/button.tsx" {6,7} framework=nextjs
import { track } from '@vercel/analytics';

function SignupButton() {
  return (
    <button
      onClick={() => {
        track('Signup');
        // ... other logic
      }}
    >
      Sign Up
    </button>
  );
```

```
}
```

```js filename="components/button.jsx" {6,7}framework=nextjs
import { track } from '@vercel/analytics';

function SignupButton() {
  return (
    <button
      onClick={() => {
        track('Signup');
        // ... other logic
      }}
    >
      Sign Up
    </button>
  );
}
```

```ts filename="components/button.tsx" {6,7}framework=nextjs-app
import { track } from '@vercel/analytics';

function SignupButton() {
  return (
    <button
      onClick={() => {
        track('Signup');
        // ... other logic
      }}
    >
      Sign Up
    </button>
  );
}
```

```js filename="components/button.jsx" {6,7}framework=nextjs-app
import { track } from '@vercel/analytics';

function SignupButton() {
  return (
    <button
      onClick={() => {
        track('Signup');
        // ... other logic
      }}
    >
      Sign Up
    </button>
  );
}
```

```ts filename="components/button.tsx" {6,7} framework=remix
import { track } from '@vercel/analytics';

function SignupButton() {
  return (
    <button
      onClick={() => {
        track('Signup');
        // ... other logic
      }}
    >
      Sign Up
    </button>
  );
}
```

```js filename="components/button.jsx" {6,7} framework=remix
import { track } from '@vercel/analytics';

function SignupButton() {
  return (
    <button
      onClick={() => {
        track('Signup');
        // ... other logic
      }}
    >
      Sign Up
    </button>
  );
}
```

```ts filename="App.svelte" {2,3}framework=sveltekit
<script>
  function handleClick() {
    track('Signup');
    // ... other logic
  }
</script>

<button on:click|once="{handleClick}">Signup</button>
```

```js filename="App.svelte" {2,3} framework=sveltekit
```

```
<script>
  function handleClick() {
    track('Signup');
    // ... other logic
  }
</script>

<button on:click|once="{handleClick}">Signup</button>
```

```ts filename="App.vue" {5} framework=nuxt
<script>
  export default {
    methods: {
      signup(event) {
        track('Signup');
        // ... other logic
      },
    },
  };
</script>

<template>
  <button @click="signup">Sign up</button>
</template>
```

```js filename="App.vue" {5} framework=nuxt
<script>
  export default {
    methods: {
      signup(event) {
        track('Signup');
        // ... other logic
      },
    },
  };
</script>

<template>
  <button @click="signup">Sign up</button>
</template>
```

Tracking an event with custom data

> For \['nextjs', 'nextjs-app', 'sveltekit', 'nuxt', 'remix', 'other']:

You can also pass custom data along with an event. To do so, pass an object
with key-value pairs as the second argument to `track()`:

> For \['html']:

You can also pass custom data along with an event. To do so, pass a `data`
property with key-value pairs as the second argument to `va()`:

```ts filename="component.ts" framework=nextjs
track('Signup', { location: 'footer' });
track('Purchase', { productName: 'Shoes', price: 49.99 });
```

```js filename="component.js"  framework=nextjs
track('Signup', { location: 'footer' });
track('Purchase', { productName: 'Shoes', price: 49.99 });
```

```ts filename="component.ts" framework=nextjs-app
track('Signup', { location: 'footer' });
track('Purchase', { productName: 'Shoes', price: 49.99 });
```

```js filename="component.js"  framework=nextjs-app
track('Signup', { location: 'footer' });
track('Purchase', { productName: 'Shoes', price: 49.99 });
```

```ts filename="component.ts"  framework=remix
track('Signup', { location: 'footer' });
track('Purchase', { productName: 'Shoes', price: 49.99 });
```

```js filename="component.js"  framework=remix
track('Signup', { location: 'footer' });
track('Purchase', { productName: 'Shoes', price: 49.99 });
```

```ts filename="component.ts"  framework=sveltekit
track('Signup', { location: 'footer' });
track('Purchase', { productName: 'Shoes', price: 49.99 });
```

```js filename="component.js"  framework=sveltekit
track('Signup', { location: 'footer' });
track('Purchase', { productName: 'Shoes', price: 49.99 });
```

```ts filename="component.ts"  framework=nuxt
track('Signup', { location: 'footer' });
track('Purchase', { productName: 'Shoes', price: 49.99 });
```

```js filename="component.js"  framework=nuxt
track('Signup', { location: 'footer' });
track('Purchase', { productName: 'Shoes', price: 49.99 });
```

> For \['html']:

```html filename="index.html"
<script>
  va('event', { name: 'Signup', data: { location: 'footer' } });
  va('event', {
    name: 'Purchase',
    data: { productName: 'Shoes', price: 49.99 },
  });
</script>
```

```ts filename="component.ts"  framework=other
import { track } from '@vercel/analytics';

track('Signup', { location: 'footer' });
track('Purchase', { productName: 'Shoes', price: 49.99 });
```

```js filename="component.js"  framework=other
import { track } from '@vercel/analytics';

track('Signup', { location: 'footer' });
track('Purchase', { productName: 'Shoes', price: 49.99 });
```

> For \['nextjs', 'nextjs-app', 'sveltekit', 'nuxt', 'remix']:

Tracking a server-side event

In scenarios such as when a user signs up or makes a purchase, it's more useful to track an event on the server-side. For this, you can u

To set up server-side events:

1. Make sure you have `@vercel/analytics` version 1.1.0 or later [installed](/docs/analytics/quickstart#add-@vercel/analytics-to-your-pro
2. Import `{ track }` from `@vercel/analytics/server`.
3. Use the `track` function in your API routes or server actions.
4. Pass in a string representing the event name as the first argument to the `track` function. You can also pass [custom data](#tracking-

For example, if you want to track a purchase event:

```ts filename="pages/api/purchase.ts" {8} framework=nextjs
import type { NextApiRequest, NextApiResponse } from 'next';
import { track } from '@vercel/analytics/server';

export default async function handler(
  req: NextApiRequest,
  res: NextApiResponse,
) {
  await track('Item purchased', {
    quantity: 1,
  });
}
```

```js filename="pages/api/purchase.js" {4} framework=nextjs
import { track } from '@vercel/analytics/server';

export default async function handler(req, res) {
  await track('Item purchased', {
    quantity: 1,
  });
}
```

```ts filename="app/actions.ts" {5}framework=nextjs-app
'use server';
import { track } from '@vercel/analytics/server';

export async function purchase() {
  await track('Item purchased', {
    quantity: 1,
  });
}
```

```js filename="app/actions.js" {5} framework=nextjs-app
'use server';
import { track } from '@vercel/analytics/server';

export async function purchase() {
  await track('Item purchased', {
    quantity: 1,
  });
}
```

```ts filename="app/routes/purchase.tsx" {4-6} framework=remix
import { track } from '@vercel/analytics/server';

export async function action() {
  await track('Item purchased', {
    quantity: 1,
  });
```

}
```

```js filename="app/routes/purchase.jsx" {4-6} framework=remix
import { track } from '@vercel/analytics/server';

export async function action() {
  await track('Item purchased', {
    quantity: 1,
  });
}
```

```ts filename="routes/+page.server.js" {6-8} framework=sveltekit
import { track } from '@vercel/analytics/server';

/** @type {import('./$types').Actions} */
export const actions = {
  default: async () => {
    await track('Item purchased', {
      quantity: 1,
    });
  },
};
```

```js filename="routes/+page.server.js" {6-8} framework=sveltekit
import { track } from '@vercel/analytics/server';

/** @type {import('./$types').Actions} */
export const actions = {
  default: async () => {
    await track('Item purchased', {
      quantity: 1,
    });
  },
};
```

```ts filename="server/api/event.ts" {4-6} framework=nuxt
import { track } from '@vercel/analytics/server';

export default defineEventHandler(async () => {
  await track('Item purchased', {
    quantity: 1,
  });
});
```

```js filename="server/api/event.js" {4-6} framework=nuxt
import { track } from '@vercel/analytics/server';

export default defineEventHandler(async () => {
  await track('Item purchased', {
    quantity: 1,
  });
});
```

Limitations

The following limitations apply to custom data:

- The number of custom data properties you can pass is limited based on your [plan](/docs/analytics/limits-and-pricing).
- Nested objects are not supported.
- Allowed values are `strings`, `numbers`, `booleans`, and `null`.
- You cannot set event name, key, or values to longer than 255 characters each.

Tracking custom events in the dashboard

Once you have tracked an event, you can view and filter for it in the dashboard. To view your events:

1. Go to your [dashboard](/dashboard), select your project, and click the **Analytics** tab.
2. From the **Web Analytics** page, scroll to the **Events** panel.
3. The events panel displays a list of all the event names that you have created in your project. Select the event name to drill down int
4. The event details page displays a list, organized by custom data properties, of all the events that have been tracked.

---
title: "Filtering Analytics"
description: "Learn how filters allow you to explore insights about your website"
last_updated: "2026-01-16T02:19:25.804Z"
source: "https://vercel.com/docs/analytics/filtering"

Filtering Analytics

Web Analytics provides you with a way to filter your data in order to gain a deeper understanding of your website traffic. This guide will show you how to use the filtering feature and provide examples of how to use it to answer specific questions.

Using filters

To filter the Web Analytics view:

1. Select a project from the dashboard and then click the **Analytics** tab.
2. Click on any row within a data panel you want to filter by. You can use multiple filters simultaneously. The following filters are ava

- Routes (if your application is based on a [supported framework](/docs/analytics/quickstart#add-the-analytics-component-to-your-app))
- Pages
```

- Hostname
- Referrers
- UTM Parameters (available with [Web Analytics Plus](/docs/analytics/limits-and-pricing) and Enterprise)
- Country
- Browsers
- Devices
- Operating System
- If configured: [Custom Events](/docs/analytics/custom-events) and [Feature Flags](/docs/feature-flags)

3. All panels on the Web Analytics page will then update to show data filtered to your selection.

For example, if you want to see data for
visitors from the United States:

1. Search for "United States" within the **Country** panel.
2. Click on the row:

Examples of using filters

By using the filtering feature in Web Analytics, you can gain a deeper understanding of your website traffic and make
data-driven decisions.

Find where visitors of a specific page came from

Let's say you want to find out where people came from that viewed your "About Us" page. To do this:

1. First, apply a filter in the **Pages** panel and click on the `/about-us` page. This will show you all of the data for visitors
   who viewed that page.
2. In the **Referrer** panel you can view all external pages that link directly to the filtered page.

Understand content popularity in a specific country

You can use the Web Analytics dashboard to find out what content people from a specific country viewed. For example, to see
what pages visitors from Canada viewed:

1. Go to the **Countries** panel, select **View All** to bring up the filter box.
2. Search for "Canada" and click on the row labeled "Canada". This will show you all of the data for visitors from Canada.
3. Go to the **Pages** panel to see what specific pages they viewed.

Discover route popularity from a specific referrer

To find out viewed pages from a specific referrer, such as Google:

1. From the **Analytics** tab, go to the **Referrers** panel.
2. Locate the row for "google.com" and click on it. This will show you all of the data for visitors who came from google.com.
3. Go to the **Routes** panel to see what specific pages they viewed.

Drill-downs

You can user certain panels to drill down into more specific information:

- The **Referrers** panel lets you drill-down into your referral data to identify the sources of referral traffic, and find out which spe
- The **Flags** panel lets you drill down into your feature flag data to find out which flag options are causing certain events to occur
- The **Custom Events** panel lets you drill down into your custom event data to find out which events are occurring and how many times t

Find Tweets from t.co referrer

Web Analytics allows you to track the origin of traffic from Twitter by using the Twitter Resolver feature. This feature can be especiall
referral traffic and finding out the origin of a specific link.

To use it:

1. From the **Referrers** panel, click **View All** and search for `t.co`
2. Click on the `t.co` row to filter for it. This performs a drill-down, which
   reveals all `t.co` links that refer to your page.
3. Clicking on any of these links a new tab will open and
   and redirect you to the Twitter search page with the URL as the search parameter. From there, you can find the original
   post of the link and gain insights into the traffic coming from Twitter.

Twitter search might not always be able to resolve to the original post of that link, and it may appear multiple times.

---
title: "Pricing for Web Analytics"
description: "Learn about pricing for Vercel Web Analytics."
last_updated: "2026-01-16T02:19:26.078Z"
source: "https://vercel.com/docs/analytics/limits-and-pricing"

Pricing for Web Analytics

Pricing

The Web Analytics pricing model is based on the number of [collected events](#what-is-an-event-in-vercel-web-analytics) across all projec
Once you've enabled Vercel Web Analytics, you will have access to various features depending on your plan.

	Hobby	Pro	[Pro with Web Analytics Pl
Included Events	50,000 Events	N/A	N/A
Additional Events	-	$3 / 100,000 Events (prorated)	$3 / 100,000 Events (prora
Included Projects	Unlimited	Unlimited	Unlimited
Reporting Window	1 Month	12 Months	24 Months
[Custom Events](/docs/analytics/custom-events)	-	Included	Included
Properties on Custom Events	-	2	8
[UTM Parameters](/docs/analytics/filtering#using-filters)	-	-	Included

On every billing cycle (every month for Hobby teams), you will be granted a certain number of events based on your plan.

Once you exceed your included limit, you will be charged for additional events.
If your team is on the Hobby plan, we will [pause](#hobby) the collection, as you cannot be charged for extra events.

Pro teams can also purchase the [Web Analytics Plus add-on](#pro-with-web-analytics-plus) for an additional $10/month per team, which gra

Usage

The table below shows the metrics for the [**Observability**](/docs/pricing/observability) section of the **Usage** dashboard where you c

To view information on managing each resource, select the resource link in the **Metric** column.
To jump straight to guidance on optimization, select the corresponding resource link in the **Optimize** column.

See the [manage and optimize Observability usage](/docs/pricing/observability) section for more information on how to optimize your usage

> **💡 Note:** Speed Insights and Web Analytics require scripts to do collection of [data
> points](/docs/speed-insights/metrics#understanding-data-points). These scripts
> are loaded on the client-side and therefore may incur additional usage and
> costs for [Data Transfer](/docs/manage-cdn-usage#fast-data-transfer) and [Edge
> Requests](/docs/manage-cdn-usage#edge-requests).

Billing information

Hobby

Web Analytics are free for Hobby users within the usage limits detailed above.

Vercel will [send you notifications](/docs/notifications#on-demand-usage-notifications) as you are nearing your usage limits.
You **will not pay for any additional usage**.
However, once you exceed the limits, a three day grace period will start before Vercel will stop capturing events.
In this scenario, you have two options to move forward:

- Wait 7 days before Vercel will start collecting events again
- Upgrade to Pro to capture more events, send custom events, and access an extended reporting window.

You can sign up for Pro and start a trial using the button below.

If you're expecting large number of page views, make sure to deploy your project to a Vercel [Team](/docs/accounts/create-a-team) on the

Pro

For Teams on a Pro trial, the [trial will end](/docs/plans/pro-plan/trials#post-trial-decision) after 14 days.

> **💡 Note:** Note that while you will not be charged during the time of the trial, once the
> trial ends, you will be charged for the events collected during the trial

You will be charged $0.00003 per event. These numbers are based on a per-billing cycle basis. Vercel will [send you notifications](/docs/

Pro teams can [set up Spend Management](/docs/spend-management#managing-your-spend-amount) to get notified or to automatically take actio

Analytics data is not collected while your project is paused, but becomes accessible again once you upgrade to Pro.

Pro with Web Analytics Plus

Teams on the Pro plan can optionally extend usage and capabilities through the Web Analytics Plus [add-on](/docs/pricing#pro-plan-add-ons

When enabled, all projects within the team have access to additional features.

To upgrade to Web Analytics Plus:

1. Visit the Vercel [dashboard](/dashboard) and select the **Settings** tab
2. From the left-nav, go to **Billing** and scroll to the Add-ons section
3. Under **Web Analytics Plus**, toggle to **Enable** the switch

FAQ

What is an event in Vercel Web Analytics?

An event in Vercel Web Analytics is either an automatically tracked page view or a [custom event](/docs/analytics/custom-events).
A page view is a default event that is automatically tracked by our script when a user visits a page on your website.
A custom event is any other action that you want to track on your website, such as a button click or form submission.

What happens when you reach the maximum number of events?

- Hobby teams won't be billed beyond their allocation. Instead, collection will be paused after the 3 days grace period.
- Pro and Enterprise teams will be billed per collected event.

Is usage shared across projects?

Yes, events are shared across all projects under the same Vercel account in Web Analytics.
This means that the events collected by each project count towards the total event limit for your account.
Keep in mind that if you have high-traffic websites or multiple projects with heavy event usage, you may need to upgrade to a higher-tier

What is the reporting window?

The reporting window in Vercel Web Analytics is the length of time that your analytics data is guaranteed to be stored and viewable for a
While only the reporting window is guaranteed to be stored, Vercel may store your data for longer periods to give you the option to upgra

---
title: "Advanced Web Analytics Config with @vercel/analytics"
description: "With the @vercel/analytics npm package, you are able to configure your application to send analytics data to Vercel."
last_updated: "2026-01-16T02:19:26.103Z"
source: "https://vercel.com/docs/analytics/package"

Advanced Web Analytics Config with @vercel/analytics

Getting started

To get started with analytics, follow our [Quickstart](/docs/analytics/quickstart) guide which will walk you through the process of setti

`mode`

Override the automatic environment detection.

> For \[
>   'nextjs',
>   'nextjs-app',
>   'sveltekit',
>   'remix',
>   'create-react-app',
>   'nuxt',
>   'vue',
>   'other',
>   'astro',
>   ]:

This option allows you to force a specific environment for the package.
If not defined, it will use `auto` which tries to set the `development` or `production` mode based on available environment variables suc

If your used framework does not expose these environment variables, the automatic detection won't work correctly.
In this case, you're able to provide the correct `mode` manually or by other helpers that your framework exposes.

If you're using the `<Analytics />` component, you can pass the `mode` prop to force a specific environment:

> For \['html']:

With plain HTML, you can not configure this option.

```tsx {8} filename="pages/_app.tsx" framework=nextjs
import type { AppProps } from 'next/app';
import { Analytics } from '@vercel/analytics/next';

function MyApp({ Component, pageProps }: AppProps) {
  return (
    <>
      <Component {...pageProps} />
      <Analytics mode="production" />;
    </>
  );
}

export default MyApp;
```

```jsx {7} filename="pages/_app.jsx" framework=nextjs
import { Analytics } from '@vercel/analytics/next';

function MyApp({ Component, pageProps }) {
  return (
    <>
      <Component {...pageProps} />
      <Analytics mode="production" />;
    </>
  );
}

export default MyApp;
```

```tsx {15} filename="app/layout.tsx" framework=nextjs-app
import { Analytics } from '@vercel/analytics/next';

export default function RootLayout({
  children,
}: {
  children: React.ReactNode;
}) {
  return (
    <html lang="en">
      <head>
        <title>Next.js</title>
      </head>
      <body>
        {children}
        <Analytics mode="production" />;
      </body>
    </html>
  );
}
```

```jsx {11} filename="app/layout.jsx" framework=nextjs-app
import { Analytics } from '@vercel/analytics/next';

export default function RootLayout({ children }) {
  return (
    <html lang="en">
      <head>
        <title>Next.js</title>
      </head>
      <body>
        {children}
        <Analytics mode="production" />;
      </body>
    </html>
  );
}
```

```tsx {7} filename="App.tsx" framework=create-react-app
import { Analytics } from '@vercel/analytics/react';
```

```jsx
export default function App() {
  return (
    <div>
      {/* ... */}
      <Analytics mode="production" />
    </div>
  );
}
```

```jsx {7} filename="App.jsx" framework=create-react-app
import { Analytics } from '@vercel/analytics/react';

export default function App() {
  return (
    <div>
      {/* ... */}
      <Analytics mode="production" />
    </div>
  );
}
```

```tsx {21} filename="app/root.tsx" framework=remix
import {
  Links,
  LiveReload,
  Meta,
  Outlet,
  Scripts,
  ScrollRestoration,
} from '@remix-run/react';
import { Analytics } from '@vercel/analytics/remix';

export default function App() {
  return (
    <html lang="en">
      <head>
        <meta charSet="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <Meta />
        <Links />
      </head>
      <body>
        <Analytics mode="production" />
        <Outlet />
        <ScrollRestoration />
        <Scripts />
        <LiveReload />
      </body>
    </html>
  );
}
```

```jsx {21} filename="app/root.jsx" framework=remix
import {
  Links,
  LiveReload,
  Meta,
  Outlet,
  Scripts,
  ScrollRestoration,
} from '@remix-run/react';
import { Analytics } from '@vercel/analytics/remix';

export default function App() {
  return (
    <html lang="en">
      <head>
        <meta charSet="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <Meta />
        <Links />
      </head>
      <body>
        <Analytics mode="production" />
        <Outlet />
        <ScrollRestoration />
        <Scripts />
        <LiveReload />
      </body>
    </html>
  );
}
```

```tsx {10} filename="src/layouts/Base.astro" framework=astro
---
import Analytics from '@vercel/analytics/astro';
{/* ... */}
---

<html lang="en">
        <head>
    <meta charset="utf-8" />
    <!-- ... -->
    <Analytics mode="production"/>
        </head>
        <body>
```

```
                <slot />
  </body>
</html>
```

```jsx {10}  filename="src/layouts/Base.astro" framework=astro
---
import Analytics from '@vercel/analytics/astro';
{/* ... */}
---

<html lang="en">
        <head>
    <meta charset="utf-8" />
    <!-- ... -->
    <Analytics mode="production"/>
        </head>
        <body>
                <slot />
  </body>
</html>
```

```tsx {6} filename="app.vue" framework=nuxt
<script setup lang="ts">
import { Analytics } from '@vercel/analytics/nuxt';
</script>

<template>
  <Analytics mode="production"/>
  <NuxtPage />
</template>
```

```jsx {6} filename="app.vue" framework=nuxt
<script setup>
import { Analytics } from '@vercel/analytics/nuxt';
</script>

<template>
  <Analytics mode="production"/>
  <NuxtPage />
</template>
```

```tsx {6} filename="src/App.vue" framework=vue
<script setup lang="ts">
import { Analytics } from '@vercel/analytics/vue';
</script>

<template>
  <Analytics mode="production" />
  <!-- your content -->
</template>
```

```jsx {6} filename="src/App.vue" framework=vue
<script setup>
import { Analytics } from '@vercel/analytics/vue';
</script>

<template>
  <Analytics mode="production" />
  <!-- your content -->
</template>
```

```ts {1, 4} filename="src/routes/+layout.ts" framework=sveltekit
import { dev } from '$app/environment';
import { injectAnalytics } from '@vercel/analytics/sveltekit';

injectAnalytics({ mode: dev ? 'development' : 'production' });
```

```js {1, 4} filename="src/routes/+layout.js" framework=sveltekit
import { dev } from '$app/environment';
import { injectAnalytics } from '@vercel/analytics/sveltekit';

injectAnalytics({ mode: dev ? 'development' : 'production' });
```

```ts {3, 6} filename="main.ts" framework=other
import { inject } from '@vercel/analytics';
// import some helper that is exposed by your current framework to determine the right mode manually
import { dev } from '$app/environment';

inject({
  mode: dev ? 'development' : 'production',
});
```

```js {3, 6} filename="main.js" framework=other
import { inject } from '@vercel/analytics';
// import some helper that is exposed by your current framework to determine the right mode manually
import { dev } from '$app/environment';

inject({
  mode: dev ? 'development' : 'production',
});
```

`debug`

> For \[
>   'nextjs',
>   'nextjs-app',
>   'sveltekit',
>   'remix',
>   'create-react-app',
>   'nuxt',
>   'vue',
>   'other',
>   'astro',
>   ]:

You'll see all analytics events in the browser's console with the debug mode.
This option is **automatically enabled** if the `NODE_ENV` environment
variable is available and either `development` or `test`.

You can manually disable it to prevent debug messages in your browsers console.

> For \[
>   'nextjs',
>   'nextjs-app',
>   'sveltekit',
>   'remix',
>   'create-react-app',
>   'nuxt',
>   'vue',
>   'other',
>   'astro',
>   ]:

To disable the debug mode for server-side events, you need to set the
`VERCEL_WEB_ANALYTICS_DISABLE_LOGS` environment variable to `true`.

```tsx {8} filename="pages/_app.tsx" framework=nextjs
import type { AppProps } from 'next/app';
import { Analytics } from '@vercel/analytics/next';

function MyApp({ Component, pageProps }: AppProps) {
  return (
    <>
      <Component {...pageProps} />
      <Analytics debug />
    </>
  );
}

export default MyApp;
```

```jsx {7} filename="pages/_app.jsx" framework=nextjs
import { Analytics } from '@vercel/analytics/next';

function MyApp({ Component, pageProps }) {
  return (
    <>
      <Component {...pageProps} />
      <Analytics debug />
    </>
  );
}

export default MyApp;
```

```tsx {15} filename="app/layout.tsx" framework=nextjs-app
import { Analytics } from '@vercel/analytics/next';

export default function RootLayout({
  children,
}: {
  children: React.ReactNode;
}) {
  return (
    <html lang="en">
      <head>
        <title>Next.js</title>
      </head>
      <body>
        {children}
        <Analytics debug />
      </body>
    </html>
  );
}
```

```jsx {11} filename="app/layout.jsx" framework=nextjs-app
import { Analytics } from '@vercel/analytics/next';

export default function RootLayout({ children }) {
  return (
    <html lang="en">
      <head>
        <title>Next.js</title>
      </head>
      <body>
        {children}
```

```
        <Analytics debug />
      </body>
    </html>
  );
}
```

```tsx {7} filename="App.tsx" framework=create-react-app
import { Analytics } from '@vercel/analytics/react';

export default function App() {
  return (
    <div>
      {/* ... */}
      <Analytics debug={true} />
    </div>
  );
}
```

```jsx {7} filename="App.jsx" framework=create-react-app
import { Analytics } from '@vercel/analytics/react';

export default function App() {
  return (
    <div>
      {/* ... */}
      <Analytics debug={true} />
    </div>
  );
}
```

```tsx {21} filename="app/root.tsx" framework=remix
import {
  Links,
  LiveReload,
  Meta,
  Outlet,
  Scripts,
  ScrollRestoration,
} from '@remix-run/react';
import { Analytics } from '@vercel/analytics/remix';

export default function App() {
  return (
    <html lang="en">
      <head>
        <meta charSet="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <Meta />
        <Links />
      </head>
      <body>
        <Analytics debug={true} />
        <Outlet />
        <ScrollRestoration />
        <Scripts />
        <LiveReload />
      </body>
    </html>
  );
}
```

```jsx {21} filename="app/root.jsx" framework=remix
import {
  Links,
  LiveReload,
  Meta,
  Outlet,
  Scripts,
  ScrollRestoration,
} from '@remix-run/react';
import { Analytics } from '@vercel/analytics/remix';

export default function App() {
  return (
    <html lang="en">
      <head>
        <meta charSet="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <Meta />
        <Links />
      </head>
      <body>
        <Analytics debug={true} />
        <Outlet />
        <ScrollRestoration />
        <Scripts />
        <LiveReload />
      </body>
    </html>
  );
}
```

```tsx {10} filename="src/layouts/Base.astro" framework=astro
---
import Analytics from '@vercel/analytics/astro';
```

```
{/* ... */}
---

<html lang="en">
        <head>
    <meta charset="utf-8" />
    <!-- ... -->
    <Analytics debug="true"/>
        </head>
        <body>
                <slot />
  </body>
</html>
```

```jsx {10}  filename="src/layouts/Base.astro" framework=astro
---
import Analytics from '@vercel/analytics/astro';
{/* ... */}
---

<html lang="en">
        <head>
    <meta charset="utf-8" />
    <!-- ... -->
    <Analytics debug={true}/>
        </head>
        <body>
                <slot />
  </body>
</html>
```

```tsx {6} filename="app.vue" framework=nuxt
<script setup lang="ts">
import { Analytics } from '@vercel/analytics/nuxt';
</script>

<template>
  <Analytics debug="true"/>
  <NuxtPage />
</template>
```

```jsx {6} filename="app.vue" framework=nuxt
<script setup>
import { Analytics } from '@vercel/analytics/nuxt';
</script>

<template>
  <Analytics debug="true"/>
  <NuxtPage />
</template>
```

```tsx {6} filename="src/App.vue" framework=vue
<script setup lang="ts">
import { Analytics } from '@vercel/analytics/vue';
</script>

<template>
  <Analytics debug="true" />
  <!-- your content -->
</template>
```

```jsx {6} filename="src/App.vue" framework=vue
<script setup>
import { Analytics } from '@vercel/analytics/vue';
</script>

<template>
  <Analytics debug="true" />
  <!-- your content -->
</template>
```

```ts {3} filename="src/routes/+layout.ts" framework=sveltekit
import { injectAnalytics } from '@vercel/analytics/sveltekit';

injectAnalytics({ debug: true });
```

```js {3} filename="src/routes/+layout.js" framework=sveltekit
import { dev } from '$app/environment';

injectAnalytics({ debug: true });
```

```ts {4} filename="main.ts" framework=other
import { inject } from '@vercel/analytics';

inject({
  debug: true,
});
```

```js {4} filename="main.js" framework=other
import { inject } from '@vercel/analytics';
```

```
inject({
  debug: true,
});
```

> For \['html']:

You have to change the script URL on your `.html` files:

```ts filename="index.html" framework=html
<script defer src="https://cdn.vercel-insights.com/v1/script.debug.js"></script>
```

```js filename="index.html" framework=html
<script defer src="https://cdn.vercel-insights.com/v1/script.debug.js"></script>
```

> For \['html']:

`beforeSend`

With the `beforeSend` option, you can modify the event data before it's sent to Vercel.
Below, you will see an example that ignores all events that have a `/private` inside the URL.

Returning `null` will ignore the event and no data will be sent.
You can also modify the URL and check our docs about [redacting sensitive data](/docs/analytics/redacting-sensitive-data).

```tsx {2, 9-14} filename="pages/_app.tsx" framework=nextjs
import type { AppProps } from 'next/app';
import { Analytics, type BeforeSendEvent } from '@vercel/analytics/next';

function MyApp({ Component, pageProps }: AppProps) {
  return (
    <>
      <Component {...pageProps} />
      <Analytics
        beforeSend={(event: BeforeSendEvent) => {
          if (event.url.includes('/private')) {
            return null;
          }
          return event;
        }}
      />
      ;
    </>
  );
}

export default MyApp;
```

```jsx {8-13} filename="pages/_app.jsx" framework=nextjs
import { Analytics } from '@vercel/analytics/next';

function MyApp({ Component, pageProps }) {
  return (
    <>
      <Component {...pageProps} />
      <Analytics
        beforeSend={(event) => {
          if (event.url.includes('/private')) {
            return null;
          }
          return event;
        }}
      />
      ;
    </>
  );
}

export default MyApp;
```

```tsx {1, 16-21} filename="app/layout.tsx" framework=nextjs-app
import { Analytics, type BeforeSendEvent } from '@vercel/analytics/next';

export default function RootLayout({
  children,
}: {
  children: React.ReactNode;
}) {
  return (
    <html lang="en">
      <head>
        <title>Next.js</title>
      </head>
      <body>
        {children}
        <Analytics
          beforeSend={(event: BeforeSendEvent) => {
            if (event.url.includes('/private')) {
              return null;
            }
            return event;
          }}
        />
      </body>
    </html>
  );
```

```
}
```

```jsx {12-17} filename="app/layout.jsx" framework=nextjs-app
import { Analytics } from '@vercel/analytics/next';

export default function RootLayout({ children }) {
  return (
    <html lang="en">
      <head>
        <title>Next.js</title>
      </head>
      <body>
        {children}
        <Analytics
          beforeSend={(event) => {
            if (event.url.includes('/private')) {
              return null;
            }
            return event;
          }}
        />
      </body>
    </html>
  );
}
```

```tsx {1, 8-13} filename="App.tsx" framework=create-react-app
import { Analytics, type BeforeSendEvent } from '@vercel/analytics/react';

export default function App() {
  return (
    <div>
      {/* ... */}
      <Analytics
        beforeSend={(event: BeforeSendEvent) => {
          if (event.url.includes('/private')) {
            return null;
          }
          return event;
        }}
      />
    </div>
  );
}
```

```jsx {8-13} filename="App.jsx" framework=create-react-app
import { Analytics } from '@vercel/analytics/react';

export default function App() {
  return (
    <div>
      {/* ... */}
      <Analytics
        beforeSend={(event) => {
          if (event.url.includes('/private')) {
            return null;
          }
          return event;
        }}
      />
    </div>
  );
}
```

```tsx {9, 22-27} filename="app/root.tsx" framework=remix
import {
  Links,
  LiveReload,
  Meta,
  Outlet,
  Scripts,
  ScrollRestoration,
} from '@remix-run/react';
import { Analytics, type BeforeSendEvent } from '@vercel/analytics/remix';

export default function App() {
  return (
    <html lang="en">
      <head>
        <meta charSet="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <Meta />
        <Links />
      </head>
      <body>
        <Analytics
          beforeSend={(event: BeforeSendEvent) => {
            if (event.url.includes('/private')) {
              return null;
            }
            return event;
          }}
        />
        <Outlet />
        <ScrollRestoration />
        <Scripts />
```

```
        <LiveReload />
      </body>
    </html>
  );
}
```

```jsx {22-27} filename="app/root.jsx" framework=remix
import {
  Links,
  LiveReload,
  Meta,
  Outlet,
  Scripts,
  ScrollRestoration,
} from '@remix-run/react';
import { Analytics } from '@vercel/analytics/remix';

export default function App() {
  return (
    <html lang="en">
      <head>
        <meta charSet="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <Meta />
        <Links />
      </head>
      <body>
        <Analytics
          beforeSend={(event: BeforeSendEvent) => {
            if (event.url.includes('/private')) {
              return null;
            }
            return event;
          }}
        />
        <Outlet />
        <ScrollRestoration />
        <Scripts />
        <LiveReload />
      </body>
    </html>
  );
}
```

```tsx {6-13} filename="src/layouts/Base.astro" framework=astro
---
import Analytics from '@vercel/analytics/astro';
{/* ... */}
---

<script is:inline>
  function webAnalyticsBeforeSend(event){
    if (event.url.includes('/private')) {
      return null;
    }
    return event;
  }
</script>

<html lang="en">
        <head>
    <meta charset="utf-8" />
    <!-- ... -->
    <Analytics />
        </head>
        <body>
                <slot />
  </body>
</html>
```

```jsx {6-13} filename="src/layouts/Base.astro" framework=astro
---
import Analytics from '@vercel/analytics/astro';
{/* ... */}
---

<script is:inline>
  function webAnalyticsBeforeSend(event){
    if (event.url.includes('/private')) {
      return null;
    }
    return event;
  }
</script>

<html lang="en">
        <head>
    <meta charset="utf-8" />
    <!-- ... -->
    <Analytics />
        </head>
        <body>
                <slot />
  </body>
</html>
```

```ts {2, 4-9, 13} filename="app.vue" framework=nuxt
<script setup lang="ts">
import { Analytics, type BeforeSendEvent } from '@vercel/analytics/nuxt';

const beforeSend = (event: BeforeSendEvent) => {
  if (event.url.includes('/private')) {
    return null;
  }
  return event
};
</script>

<template>
  <Analytics :before-send="beforeSend"/>
  <NuxtLayout>
    <NuxtPage />
  </NuxtLayout>
</template>
```

```js {4-9, 13} filename="app.vue" framework=nuxt
<script setup>
import { Analytics } from '@vercel/analytics/nuxt';

const beforeSend = (event) => {
  if (event.url.includes('/private')) {
    return null;
  }
  return event
};
</script>

<template>
  <Analytics :before-send="beforeSend"/>
  <NuxtPage />
</template>
```

```tsx {2, 4-9, 13} filename="src/App.vue" framework=vue
<script setup lang="ts">
import { Analytics, type BeforeSendEvent } from '@vercel/analytics/nuxt';

const beforeSend = (event: BeforeSendEvent) => {
  if (event.url.includes('/private')) {
    return null;
  }
  return event
};
</script>

<template>
  <Analytics :before-send="beforeSend"/>
  <!-- your content -->
</template>
```

```jsx {4-9, 13} filename="src/App.vue" framework=vue
<script setup>
import { Analytics } from '@vercel/analytics/nuxt';

const beforeSend = (event) => {
  if (event.url.includes('/private')) {
    return null;
  }
  return event
};
</script>

<template>
  <Analytics :before-send="beforeSend"/>
  <!-- your content -->
</template>
```

```ts {3, 7-12} filename="src/routes/+layout.ts" framework=sveltekit
import {
  injectAnalytics,
  type BeforeSendEvent,
} from '@vercel/analytics/sveltekit';

injectAnalytics({
  beforeSend(event: BeforeSendEvent) {
    if (event.url.includes('/private')) {
      return null;
    }
    return event;
  },
});
```

```js {4-9} filename="src/routes/+layout.js" framework=sveltekit
import { injectAnalytics } from '@vercel/analytics/sveltekit';

injectAnalytics({
  beforeSend(event) {
    if (event.url.includes('/private')) {
      return null;
    }
    return event;
  },
```

});
```

```ts {1, 4-9} filename="main.ts" framework=other
import { inject, type BeforeSendEvent } from '@vercel/analytics';

inject({
  beforeSend: (event: BeforeSendEvent) => {
    if (event.url.includes('/private')) {
      return null;
    }
    return event;
  },
});
```

```js {4-9} filename="main.js" framework=other
import { inject } from '@vercel/analytics';

inject({
  beforeSend: (event) => {
    if (event.url.includes('/private')) {
      return null;
    }
    return event;
  },
});
```

```ts {5-10} filename="index.html" framework=html
<script>
  window.va = function () {
    (window.vaq = window.vaq || []).push(arguments);
  };
  window.va('beforeSend', (event) => {
    if (event.url.includes('/private')) {
      return null;
    }
    return event;
  });
</script>
```

```js {5-10} filename="index.html" framework=html
<script>
  window.va = function () {
    (window.vaq = window.vaq || []).push(arguments);
  };
  window.va('beforeSend', (event) => {
    if (event.url.includes('/private')) {
      return null;
    }
    return event;
  });
</script>
```

`endpoint`

The `endpoint` option allows you to report the collected analytics to a different url than the default: `https://yourdomain.com/_vercel/i

This is useful when deploying several projects under the same domain, as it allows you to keep each application isolated.

For example, when `yourdomain.com` is managed outside of Vercel:

1. "alice-app" is deployed under `yourdomain.com/alice/*`, vercel alias is `alice-app.vercel.sh`
2. "bob-app" is deployed under `yourdomain.com/bob/*`, vercel alias is `bob-app.vercel.sh`
3. `yourdomain.com/_vercel/*` is routed to `alice-app.vercel.sh`

Both applications are sending their analytics to `alice-app.vercel.sh`. To restore the isolation, "bob-app" should use:

```tsx
<Analytics endpoint="https://bob-app.vercel.sh/_vercel/insights" />
```

`scriptSrc`

The `scriptSrc` option allows you to load the Web Analytics script from a different URL than the default one.

```tsx
<Analytics scriptSrc="https://bob-app.vercel.sh/_vercel/insights/script.js" />
```

---
title: "Vercel Web Analytics"
description: "With Web Analytics, you can get detailed insights into your website"
last_updated: "2026-01-16T02:19:25.907Z"
source: "https://vercel.com/docs/analytics"

Vercel Web Analytics

Web Analytics provides comprehensive insights into your website's visitors, allowing you to track the top visited pages, referrers for a

- **Privacy**: Web Analytics only stores anonymized data and [does not use cookies](#how-visitors-are-determined), providing data for you
- **Integrated Infrastructure**: Web Analytics is built into the Vercel platform and accessible from your project's dashboard so there's
- **Customizable**: You can configure Web Analytics to track custom events and feature flag usage to get a better understanding of how yo
```

To set up Web Analytics for your project, see the [Quickstart](/docs/analytics/quickstart).

If you're interested in learning more about how your site is performing, use [Speed Insights](/docs/speed-insights).

Visitors

The **Visitors** tab displays all your website's unique visitors within a selected timeframe. You can adjust the timeframe by selecting a value from the dropdown in the top right hand corner.

You can use the [panels](#panels) section to view a breakdown of specific information, organized by the total number of visitors.

How visitors are determined

Instead of relying on cookies like many analytics products, visitors are identified by a hash created from the incoming request. Using a

The generated hash is valid for a single day, at which point it is automatically reset.

If a visitor loads your website for the first time, we immediately track this visit as a page view. Subsequent page views are tracked thr

Page views

The **Page Views** tab, like the **Visitors** tab, shows a breakdown of every page loaded on your website during a certain time period. Page views are counted by the **total number of views** on a page. For page views, the same visitor can view the same page multiple times

You can use the [panels](#panels) section to view a breakdown of specific information, organized by the total number of page views.

Bounce rate

The **Bounce rate** is the percentage of visitors who land on a page and leave without taking any further action.

The higher the bounce rate, the less engaging the page is.

How bounce rate is calculated

> **💡 Note:** Bounce Rate (%) = (Single-Page Sessions / Total Sessions) × 100

Web Analytics defines a session as a group or page views by the same visitor. Custom event do not count towards the bounce rate.

For that reason, when filtering the dashboard for a given custom event, the bounce rate will always be 0%.

Panels

Panels provide a way to view detailed analytics for Visitors and Page Views, such as top pages and referrers. They'll also show additiona

By default, panels provide you with a list of top entries, categorized by the number of visitors. Depending on the panel, the information

You can export the up to 250 entries from the panel as a CSV file. See [Exporting data as CSV](/docs/analytics/using-web-analytics#export

Bots

Web Analytics does not count traffic that comes from automated processes or accounts. This is determined by inspecting the [User Agent](h

---
title: "Privacy and Compliance"
description: "Learn how Vercel supports privacy and data compliance standards with Vercel Web Analytics."
last_updated: "2026-01-16T02:19:25.918Z"
source: "https://vercel.com/docs/analytics/privacy-policy"

Privacy and Compliance

Vercel takes a privacy-focused approach to our products and strive to enable our customers to use Vercel with confidence. The company aim

Data collected

Vercel Web Analytics can be used globally and Vercel have designed it to align with leading data protection authority guidance. When usin

The recording of data points (for example, page views or custom events) is anonymous, so you have insight into your data without it being

Vercel Web Analytics does not collect or store any information that would enable you to reconstruct an end user's browsing session across

Visitor identification and data storage

Vercel Web Analytics allows you to track your website traffic and gather valuable insights without using any third-party cookies, instead

The lifespan of a visitor session is not stored permanently, it is automatically discarded after 24 hours.

After following the dashboard instructions to enable Vercel Web Analytics, see our [Quickstart](/docs/analytics/quickstart) for a step-by

All page views will automatically be tracked by Vercel Web Analytics, including both fresh page loads and client-side page transitions.

Data point information

The following information may be stored with every data point:

Collected Value	Example Value
Event Timestamp	2020-10-29 09:06:30
URL	`/blog/nextjs-10`
Dynamic Path	`/blog/[slug]`
Referrer	https://news.ycombinator.com/
Query Params (Filtered)	`?ref=hackernews`
Geolocation	US, California, San Francisco
Device OS & Version	Android 10
Browser & Version	Chrome 86 (Blink)
Device Type	Mobile (or Desktop/Tablet)
Web Analytics Script Version	1.0.0

Configuring Vercel Web Analytics

Some URLs and query parameters can include sensitive data and personal information (i.e. user ID, token, order ID or any other informatio

For example, automatic page view tracking may track personal information `https://acme.com/[name of individual]/invoice/[12345]`. You can

For [custom events](/docs/analytics/custom-events), you may want to prevent sending sensitive or personal information, such as email addr

---
title: "Getting started with Vercel Web Analytics"
description: "Vercel Web Analytics provides you detailed insights into your website"
last_updated: "2026-01-16T02:19:26.147Z"
source: "https://vercel.com/docs/analytics/quickstart"

Getting started with Vercel Web Analytics

This guide will help you get started with using Vercel Web Analytics on your project, showing you how to enable it, add the package to yo

Select your framework to view instructions on using the Vercel Web Analytics in your project.

Prerequisites

- A Vercel account. If you don't have one, you can [sign up for free](https://vercel.com/signup).
- A Vercel project. If you don't have one, you can [create a new project](https://vercel.com/new).
- The Vercel CLI installed. If you don't have it, you can install it using the following command:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i vercel
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i vercel
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i vercel
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i vercel
      ```
    </Code>
  </CodeBlock>

- ### Enable Web Analytics in Vercel
  On the [Vercel dashboard](/dashboard), select your Project and then click the **Analytics** tab and click **Enable** from the dialog.
  > **💡 Note:** Enabling Web Analytics will add new routes (scoped at `/_vercel/insights/*`)
  > after your next deployment.

> For \['nextjs', 'nextjs-app', 'sveltekit', 'remix', 'create-react-app', 'nuxt', 'vue', 'other', 'astro']:

- > For \[
  >   'nextjs',
  >   'nextjs-app',
  >   'remix',
  >   'create-react-app',
  >   'nuxt',
  >   'vue',
  >   'astro',
  >   ]:
  ### Add the `Analytics` component to your app
  > For \['sveltekit']:
  ### Call the `injectAnalytics` function in your app
  > For \['other']:
  ### Call the `inject` function in your app
  > For \['html']:
  ### Add the `script` tag to your site
  > For \['nextjs']:
  The `Analytics` component is a wrapper around the tracking script, offering more seamless integration with Next.js, including route sup

  If you are using the `pages` directory, add the following code to your main app file:
  ```tsx {2, 8} filename="pages/_app.tsx" framework=nextjs
  import type { AppProps } from 'next/app';
  import { Analytics } from '@vercel/analytics/next';

  function MyApp({ Component, pageProps }: AppProps) {
    return (
      <>
        <Component {...pageProps} />
        <Analytics />
      </>
    );
  }

  export default MyApp;
  ```
  ```jsx {1, 7} filename="pages/_app.js" framework=nextjs
  import { Analytics } from '@vercel/analytics/next';

  function MyApp({ Component, pageProps }) {
    return (
      <>
        <Component {...pageProps} />
  ```

```
      <Analytics />
    </>
  );
}

export default MyApp;
```
> For \['nextjs-app']:
The `Analytics` component is a wrapper around the tracking script, offering more seamless integration with Next.js, including route sup

Add the following code to the root layout:
```tsx {1, 15} filename="app/layout.tsx" framework=nextjs-app
import { Analytics } from '@vercel/analytics/next';

export default function RootLayout({
  children,
}: {
  children: React.ReactNode;
}) {
  return (
    <html lang="en">
      <head>
        <title>Next.js</title>
      </head>
      <body>
        {children}
        <Analytics />
      </body>
    </html>
  );
}
```
```jsx {1, 11} filename="app/layout.jsx" framework=nextjs-app
import { Analytics } from '@vercel/analytics/next';

export default function RootLayout({ children }) {
  return (
    <html lang="en">
      <head>
        <title>Next.js</title>
      </head>
      <body>
        {children}
        <Analytics />
      </body>
    </html>
  );
}
```
> For \['remix']:
The `Analytics` component is a wrapper around the tracking script, offering a seamless integration with Remix, including route detectio

Add the following code to your root file:
```tsx {9, 21} filename="app/root.tsx" framework=remix
import {
  Links,
  LiveReload,
  Meta,
  Outlet,
  Scripts,
  ScrollRestoration,
} from '@remix-run/react';
import { Analytics } from '@vercel/analytics/remix';

export default function App() {
  return (
    <html lang="en">
      <head>
        <meta charSet="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <Meta />
        <Links />
      </head>
      <body>
        <Analytics />
        <Outlet />
        <ScrollRestoration />
        <Scripts />
        <LiveReload />
      </body>
    </html>
  );
}
```
```jsx {9, 21} filename="app/root.jsx" framework=remix
import {
  Links,
  LiveReload,
  Meta,
  Outlet,
  Scripts,
  ScrollRestoration,
} from '@remix-run/react';
import { Analytics } from '@vercel/analytics/remix';

export default function App() {
  return (
    <html lang="en">
      <head>
        <meta charSet="utf-8" />
```

```
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <Meta />
        <Links />
      </head>
      <body>
        <Analytics />
        <Outlet />
        <ScrollRestoration />
        <Scripts />
        <LiveReload />
      </body>
    </html>
  );
}
```
> For \['nuxt']:
The `Analytics` component is a wrapper around the tracking script, offering more seamless integration with Nuxt, including route suppor

Add the following code to your main component.
```tsx {2,6} filename="app.vue" framework=nuxt
<script setup lang="ts">
import { Analytics } from '@vercel/analytics/nuxt';
</script>

<template>
  <Analytics />
  <NuxtPage />
</template>
```
```jsx {2,6} filename="app.vue" framework=nuxt
<script setup>
import { Analytics } from '@vercel/analytics/nuxt';
</script>

<template>
  <Analytics />
  <NuxtPage />
</template>
```
> For \['sveltekit']:
The `injectAnalytics` function is a wrapper around the tracking script, offering more seamless integration with SvelteKit.js, including

Add the following code to the main layout:
```ts filename="src/routes/+layout.ts" framework=sveltekit
import { dev } from '$app/environment';
import { injectAnalytics } from '@vercel/analytics/sveltekit';

injectAnalytics({ mode: dev ? 'development' : 'production' });
```
```js filename="src/routes/+layout.js" framework=sveltekit
import { dev } from '$app/environment';
import { injectAnalytics } from '@vercel/analytics/sveltekit';

injectAnalytics({ mode: dev ? 'development' : 'production' });
```
> For \['astro']:
The `Analytics` component is a wrapper around the tracking script, offering more seamless integration with Astro, including route suppo

Add the following code to your base layout:
```tsx {2, 10} filename="src/layouts/Base.astro" framework=astro
---
import Analytics from '@vercel/analytics/astro';
{/* ... */}
---

<html lang="en">
      <head>
    <meta charset="utf-8" />
    <!-- ... -->
    <Analytics />
      </head>
      <body>
              <slot />
  </body>
</html>
```
```jsx {2, 10}  filename="src/layouts/Base.astro" framework=astro
---
import Analytics from '@vercel/analytics/astro';
{/* ... */}
---

<html lang="en">
      <head>
    <meta charset="utf-8" />
    <!-- ... -->
    <Analytics />
      </head>
      <body>
              <slot />
  </body>
</html>
```
> For \['astro']:
The `Analytics` component is available in version `@vercel/analytics@1.4.0` and later.
If you are using an earlier version, you must configure the `webAnalytics` property of the Vercel adapter in your `astro.config.mjs` fi
For further information, see the [Astro adapter documentation](https://docs.astro.build/en/guides/integrations-guide/vercel/#webanalyti
```ts {7-9} filename="astro.config.mjs" framework=astro
import { defineConfig } from 'astro/config';
import vercel from '@astrojs/vercel/serverless';
```

```
export default defineConfig({
  output: 'server',
  adapter: vercel({
    webAnalytics: {
      enabled: true, // set to false when using @vercel/analytics@1.4.0
    },
  }),
});
```
```js {7-9} filename="astro.config.mjs" framework=astro
import { defineConfig } from 'astro/config';
import vercel from '@astrojs/vercel/serverless';

export default defineConfig({
  output: 'server',
  adapter: vercel({
    webAnalytics: {
      enabled: true, // set to false when using @vercel/analytics@1.4.0
    },
  }),
});
```
> For \['html']:
For plain HTML sites, you can add the following script to your `.html` files:
```ts filename="index.html" framework=html
<script>
  window.va = window.va || function () { (window.vaq = window.vaq || []).push(arguments); };
</script>
<script defer src="/_vercel/insights/script.js"></script>
```
```js filename="index.html" framework=html
<script>
  window.va = window.va || function () { (window.vaq = window.vaq || []).push(arguments); };
</script>
<script defer src="/_vercel/insights/script.js"></script>
```
> For \['html']:
> For \['other']:
Import the `inject` function from the package, which will add the tracking script to your app. **This should only be called once in you

> **💡 Note:** There is no route support with the `inject` function.

Add the following code to your main app file:
```ts filename="main.ts" framework=other
import { inject } from '@vercel/analytics';

inject();
```
```js filename="main.js" framework=other
import { inject } from '@vercel/analytics';

inject();
```
> For \['create-react-app']:
The `Analytics` component is a wrapper around the tracking script, offering more seamless integration with React.

Add the following code to the main app file:
```tsx {1, 7} filename="App.tsx" framework=create-react-app
import { Analytics } from '@vercel/analytics/react';

export default function App() {
  return (
    <div>
      {/* ... */}
      <Analytics />
    </div>
  );
}
```
```jsx {1, 7} filename="App.jsx" framework=create-react-app
import { Analytics } from '@vercel/analytics/react';

export default function App() {
  return (
    <div>
      {/* ... */}
      <Analytics />
    </div>
  );
}
```
> For \['vue']:
The `Analytics` component is a wrapper around the tracking script, offering more seamless integration with Vue.

Add the following code to your main component:
```tsx {2,6} filename="src/App.vue" framework=vue
<script setup lang="ts">
import { Analytics } from '@vercel/analytics/vue';
</script>

<template>
  <Analytics />
  <!-- your content -->
</template>
```
```jsx {2,6} filename="src/App.vue" framework=vue
<script setup>
import { Analytics } from '@vercel/analytics/vue';
</script>
```

  <template>
    <Analytics />
    <!-- your content -->
  </template>
  ```

- ### Deploy your app to Vercel
  Deploy your app using the following command:
  ```bash filename="terminal"
  vercel deploy
  ```
  If you haven't already, we also recommend [connecting your project's Git repository](/docs/git#deploying-a-git-repository),
  which will enable Vercel to deploy your latest commits to main without terminal commands.

  Once your app is deployed, it will start tracking visitors and page views.
  > **💡 Note:** If everything is set up properly, you should be able to see a Fetch/XHR
  > request in your browser's Network tab from `/_vercel/insights/view` when you
  > visit any page.

- ### View your data in the dashboard
  Once your app is deployed, and users have visited your site, you can view your data in the dashboard.

  To do so, go to your [dashboard](/dashboard), select your project, and click the **Analytics** tab.

  After a few days of visitors, you'll be able to start exploring your data by viewing and [filtering](/docs/analytics/filtering) the pan

  Users on Pro and Enterprise plans can also add [custom events](/docs/analytics/custom-events) to their data to track user interactions

Learn more about how Vercel supports [privacy and data compliance standards](/docs/analytics/privacy-policy) with Vercel Web Analytics.

Next steps

Now that you have Vercel Web Analytics set up, you can explore the following topics to learn more:

- [Learn how to use the `@vercel/analytics` package](/docs/analytics/package)
- [Learn how to set update custom events](/docs/analytics/custom-events)
- [Learn about filtering data](/docs/analytics/filtering)
- [Read about privacy and compliance](/docs/analytics/privacy-policy)
- [Explore pricing](/docs/analytics/limits-and-pricing)
- [Troubleshooting](/docs/analytics/troubleshooting)

---
title: "Redacting Sensitive Data from Web Analytics Events"
description: "Learn how to redact sensitive data from your Web Analytics events."
last_updated: "2026-01-16T02:19:26.110Z"
source: "https://vercel.com/docs/analytics/redacting-sensitive-data"

Redacting Sensitive Data from Web Analytics Events

Sometimes, URLs and query parameters may contain sensitive data. This could be a user ID, a token, an order ID, or any other data that yo

To prevent sensitive data from being sent to Vercel, you can pass in the `beforeSend` function that modifies the event before it is sent.

Ignoring events or routes

To ignore an event or route, you can return `null` from the `beforeSend` function. Returning the event or a modified version of it will t

```tsx {2, 9-14} filename="pages/_app.tsx" framework=nextjs
import type { AppProps } from 'next/app';
import { Analytics, type BeforeSendEvent } from '@vercel/analytics/next';

function MyApp({ Component, pageProps }: AppProps) {
  return (
    <>
      <Component {...pageProps} />
      <Analytics
        beforeSend={(event: BeforeSendEvent) => {
          if (event.url.includes('/private')) {
            return null;
          }
          return event;
        }}
      />
      ;
    </>
  );
}

export default MyApp;
```

```jsx {8-13} filename="pages/_app.jsx" framework=nextjs
import { Analytics } from '@vercel/analytics/next';

function MyApp({ Component, pageProps }) {
  return (
    <>
      <Component {...pageProps} />
      <Analytics
        beforeSend={(event) => {
          if (event.url.includes('/private')) {
            return null;
          }
          return event;
        }}
      />
      ;
```
```

```
    </>
  );
}

export default MyApp;
```

```tsx {1, 16-21} filename="app/layout.tsx" framework=nextjs-app
import { Analytics, type BeforeSendEvent } from '@vercel/analytics/next';

export default function RootLayout({
  children,
}: {
  children: React.ReactNode;
}) {
  return (
    <html lang="en">
      <head>
        <title>Next.js</title>
      </head>
      <body>
        {children}
        <Analytics
          beforeSend={(event: BeforeSendEvent) => {
            if (event.url.includes('/private')) {
              return null;
            }
            return event;
          }}
        />
      </body>
    </html>
  );
}
```

```jsx {12-17} filename="app/layout.jsx" framework=nextjs-app
import { Analytics } from '@vercel/analytics/next';

export default function RootLayout({ children }) {
  return (
    <html lang="en">
      <head>
        <title>Next.js</title>
      </head>
      <body>
        {children}
        <Analytics
          beforeSend={(event) => {
            if (event.url.includes('/private')) {
              return null;
            }
            return event;
          }}
        />
      </body>
    </html>
  );
}
```

```tsx {1, 8-13} filename="App.tsx" framework=create-react-app
import { Analytics, type BeforeSendEvent } from '@vercel/analytics/react';

export default function App() {
  return (
    <div>
      {/* ... */}
      <Analytics
        beforeSend={(event: BeforeSendEvent) => {
          if (event.url.includes('/private')) {
            return null;
          }
          return event;
        }}
      />
    </div>
  );
}
```

```jsx {8-13} filename="App.jsx" framework=create-react-app
import { Analytics } from '@vercel/analytics/react';

export default function App() {
  return (
    <div>
      {/* ... */}
      <Analytics
        beforeSend={(event) => {
          if (event.url.includes('/private')) {
            return null;
          }
          return event;
        }}
      />
    </div>
  );
}
```

```tsx {9, 22-27} filename="app/root.tsx" framework=remix
import {
  Links,
  LiveReload,
  Meta,
  Outlet,
  Scripts,
  ScrollRestoration,
} from '@remix-run/react';
import { Analytics, type BeforeSendEvent } from '@vercel/analytics/remix';

export default function App() {
  return (
    <html lang="en">
      <head>
        <meta charSet="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <Meta />
        <Links />
      </head>
      <body>
        <Analytics
          beforeSend={(event: BeforeSendEvent) => {
            if (event.url.includes('/private')) {
              return null;
            }
            return event;
          }}
        />
        <Outlet />
        <ScrollRestoration />
        <Scripts />
        <LiveReload />
      </body>
    </html>
  );
}
```

```jsx {22-27} filename="app/root.jsx" framework=remix
import {
  Links,
  LiveReload,
  Meta,
  Outlet,
  Scripts,
  ScrollRestoration,
} from '@remix-run/react';
import { Analytics } from '@vercel/analytics/remix';

export default function App() {
  return (
    <html lang="en">
      <head>
        <meta charSet="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <Meta />
        <Links />
      </head>
      <body>
        <Analytics
          beforeSend={(event: BeforeSendEvent) => {
            if (event.url.includes('/private')) {
              return null;
            }
            return event;
          }}
        />
        <Outlet />
        <ScrollRestoration />
        <Scripts />
        <LiveReload />
      </body>
    </html>
  );
}
```

```tsx {6-13} filename="src/layouts/Base.astro" framework=astro
---
import Analytics from '@vercel/analytics/astro';
{/* ... */}
---

<script is:inline>
  function webAnalyticsBeforeSend(event){
    if (event.url.includes('/private')) {
      return null;
    }
    return event;
  }
</script>

<html lang="en">
        <head>
    <meta charset="utf-8" />
    <!-- ... -->
    <Analytics />
        </head>
```

```
        <body>
                <slot />
  </body>
</html>
```

```jsx {6-13} filename="src/layouts/Base.astro" framework=astro
---
import Analytics from '@vercel/analytics/astro';
{/* ... */}
---

<script is:inline>
  function webAnalyticsBeforeSend(event){
    if (event.url.includes('/private')) {
      return null;
    }
    return event;
  }
</script>

<html lang="en">
        <head>
    <meta charset="utf-8" />
    <!-- ... -->
    <Analytics />
        </head>
        <body>
                <slot />
  </body>
</html>
```

```ts {2, 4-9, 13} filename="app.vue" framework=nuxt
<script setup lang="ts">
import { Analytics, type BeforeSendEvent } from '@vercel/analytics/nuxt';

const beforeSend = (event: BeforeSendEvent) => {
  if (event.url.includes('/private')) {
    return null;
  }
  return event
};
</script>

<template>
  <Analytics :before-send="beforeSend"/>
  <NuxtLayout>
    <NuxtPage />
  </NuxtLayout>
</template>
```

```js {4-9, 13} filename="app.vue" framework=nuxt
<script setup>
import { Analytics } from '@vercel/analytics/nuxt';

const beforeSend = (event) => {
  if (event.url.includes('/private')) {
    return null;
  }
  return event
};
</script>

<template>
  <Analytics :before-send="beforeSend"/>
  <NuxtPage />
</template>
```

```tsx {2, 4-9, 13} filename="src/App.vue" framework=vue
<script setup lang="ts">
import { Analytics, type BeforeSendEvent } from '@vercel/analytics/nuxt';

const beforeSend = (event: BeforeSendEvent) => {
  if (event.url.includes('/private')) {
    return null;
  }
  return event
};
</script>

<template>
  <Analytics :before-send="beforeSend"/>
  <!-- your content -->
</template>
```

```jsx {4-9, 13} filename="src/App.vue" framework=vue
<script setup>
import { Analytics } from '@vercel/analytics/nuxt';

const beforeSend = (event) => {
  if (event.url.includes('/private')) {
    return null;
  }
  return event
};
</script>
```

```
<template>
  <Analytics :before-send="beforeSend"/>
  <!-- your content -->
</template>
```

```ts {3, 7-12} filename="src/routes/+layout.ts" framework=sveltekit
import {
  injectAnalytics,
  type BeforeSendEvent,
} from '@vercel/analytics/sveltekit';

injectAnalytics({
  beforeSend(event: BeforeSendEvent) {
    if (event.url.includes('/private')) {
      return null;
    }
    return event;
  },
});
```

```js {4-9} filename="src/routes/+layout.js" framework=sveltekit
import { injectAnalytics } from '@vercel/analytics/sveltekit';

injectAnalytics({
  beforeSend(event) {
    if (event.url.includes('/private')) {
      return null;
    }
    return event;
  },
});
```

```ts {1, 4-9} filename="main.ts" framework=other
import { inject, type BeforeSendEvent } from '@vercel/analytics';

inject({
  beforeSend: (event: BeforeSendEvent) => {
    if (event.url.includes('/private')) {
      return null;
    }
    return event;
  },
});
```

```js {4-9} filename="main.js" framework=other
import { inject } from '@vercel/analytics';

inject({
  beforeSend: (event) => {
    if (event.url.includes('/private')) {
      return null;
    }
    return event;
  },
});
```

```ts {5-10} filename="index.html" framework=html
<script>
  window.va = function () {
    (window.vaq = window.vaq || []).push(arguments);
  };
  window.va('beforeSend', (event) => {
    if (event.url.includes('/private')) {
      return null;
    }
    return event;
  });
</script>
```

```js {5-10} filename="index.html" framework=html
<script>
  window.va = function () {
    (window.vaq = window.vaq || []).push(arguments);
  };
  window.va('beforeSend', (event) => {
    if (event.url.includes('/private')) {
      return null;
    }
    return event;
  });
</script>
```

Removing query parameters

To apply changes to the event, you can parse the URL and adjust it to your needs before you return the modified event.

In this example the query parameter `secret` is removed on all events.

```js filename="pages/_app.jsx" framework=nextjs
import { Analytics } from '@vercel/analytics/react';

function MyApp({ Component, pageProps }) {
```

```
  return (
    <>
      <Component {...pageProps} />
      <Analytics
        beforeSend={(event) => {
          const url = new URL(event.url);
          url.searchParams.delete('secret');
          return {
            ...event,
            url: url.toString(),
          };
        }}
      />
    </>
  );
}

export default MyApp;
```

```ts filename="pages/_app.tsx" framework=nextjs
import type { AppProps } from 'next/app';
import { Analytics } from '@vercel/analytics/react';

function MyApp({ Component, pageProps }: AppProps) {
  return (
    <>
      <Component {...pageProps} />
      <Analytics
        beforeSend={(event) => {
          const url = new URL(event.url);
          url.searchParams.delete('secret');
          return {
            ...event,
            url: url.toString(),
          };
        }}
      />
    </>
  );
}

export default MyApp;
```

```js filename="app/layout.jsx" framework=nextjs-app
'use client';
import { Analytics } from '@vercel/analytics/react';

export default function RootLayout({ children }) {
  return (
    <html lang="en">
      <head>
        <title>Next.js</title>
      </head>
      <body>
        {children}
        <Analytics
          beforeSend={(event) => {
            const url = new URL(event.url);
            url.searchParams.delete('secret');
            return {
              ...event,
              url: url.toString(),
            };
          }}
        />
      </body>
    </html>
  );
}
```

```ts filename="app/layout.tsx" framework=nextjs-app
'use client';
import { Analytics } from '@vercel/analytics/react';

export default function RootLayout({
  children,
}: {
  children: React.ReactNode;
}) {
  return (
    <html lang="en">
      <head>
        <title>Next.js</title>
      </head>
      <body>
        {children}
        <Analytics
          beforeSend={(event) => {
            const url = new URL(event.url);
            url.searchParams.delete('secret');
            return {
              ...event,
              url: url.toString(),
            };
          }}
        />
      </body>
```

```
    </html>
  );
}
```

```js filename="main.js" framework=other
import { inject } from '@vercel/analytics';

inject({
  beforeSend: (event) => {
    const url = new URL(event.url);
    url.searchParams.delete('secret');
    return {
      ...event,
      url: url.toString(),
    };
  },
});
```

```ts filename="main.ts" framework=other
import { inject } from '@vercel/analytics';

inject({
  beforeSend: (event) => {
    const url = new URL(event.url);
    url.searchParams.delete('secret');
    return {
      ...event,
      url: url.toString(),
    };
  },
});
```

```js filename="index.html" framework=html
<script>
  window.va = function () {
    (window.vaq = window.vaq || []).push(arguments);
  };
  window.va('beforeSend', (event) => {
    const url = new URL(event.url);
    url.searchParams.delete('secret');
    return {
      ...event,
      url: url.toString(),
    }
  });
</script>
```

```ts filename="index.html" framework=html
<script>
  window.va = function () {
    (window.vaq = window.vaq || []).push(arguments);
  };
  window.va('beforeSend', (event) => {
    const url = new URL(event.url);
    url.searchParams.delete('secret');
    return {
      ...event,
      url: url.toString(),
    }
  });
</script>
```

Allowing users to opt-out of tracking

You can also use `beforeSend` to allow users to opt-out of all tracking by setting a `localStorage` value (for example `va-disable`).

```js filename="pages/_app.jsx" framework=nextjs
import { Analytics } from '@vercel/analytics/react';

function MyApp({ Component, pageProps }) {
  return (
    <>
      <Component {...pageProps} />
      <Analytics
        beforeSend={(event) => {
          if (localStorage.getItem('va-disable')) {
            return null;
          }
          return event;
        }}
      />
    </>
  );
}

export default MyApp;
```

```ts filename="pages/_app.tsx" framework=nextjs
import type { AppProps } from 'next/app';
import { Analytics } from '@vercel/analytics/react';

function MyApp({ Component, pageProps }: AppProps) {
  return (
    <>
```

```
      <Component {...pageProps} />
      <Analytics
        beforeSend={(event) => {
          if (localStorage.getItem('va-disable')) {
            return null;
          }
          return event;
        }}
      />
    </>
  );
}

export default MyApp;
```

```js filename="app/layout.jsx" framework=nextjs-app
'use client';
import { Analytics } from '@vercel/analytics/react';

export default function RootLayout({ children }) {
  return (
    <html lang="en">
      <head>
        <title>Next.js</title>
      </head>
      <body>
        {children}
        <Analytics
          beforeSend={(event) => {
            if (localStorage.getItem('va-disable')) {
              return null;
            }
            return event;
          }}
        />
      </body>
    </html>
  );
}
```

```ts filename="app/layout.tsx" framework=nextjs-app
'use client';
import { Analytics } from '@vercel/analytics/react';

export default function RootLayout({
  children,
}: {
  children: React.ReactNode;
}) {
  return (
    <html lang="en">
      <head>
        <title>Next.js</title>
      </head>
      <body>
        {children}
        <Analytics
          beforeSend={(event) => {
            if (localStorage.getItem('va-disable')) {
              return null;
            }
            return event;
          }}
        />
      </body>
    </html>
  );
}
```

```js filename="main.js" framework=other
import { inject } from '@vercel/analytics';

inject({
  beforeSend: (event) => {
    if (localStorage.getItem('va-disable')) {
      return null;
    }
    return event;
  },
});
```

```ts filename="main.ts" framework=other
import { inject } from '@vercel/analytics';

inject({
  beforeSend: (event) => {
    if (localStorage.getItem('va-disable')) {
      return null;
    }
    return event;
  },
});
```

```js filename="index.html" framework=html
<script>
```

```
  window.va = function () {
    (window.vaq = window.vaq || []).push(arguments);
  };
  window.va('beforeSend', (event) => {
    if (localStorage.getItem('va-disable')) {
      return null;
    }
    return event;
  });
</script>
```

```ts filename="index.html" framework=html
<script>
  window.va = function () {
    (window.vaq = window.vaq || []).push(arguments);
  };
  window.va('beforeSend', (event) => {
    if (localStorage.getItem('va-disable')) {
      return null;
    }
    return event;
  });
</script>
```

---
title: "Vercel Web Analytics Troubleshooting"
description: "Learn how to troubleshoot common issues with Vercel Web Analytics."
last_updated: "2026-01-16T02:19:25.944Z"
source: "https://vercel.com/docs/analytics/troubleshooting"

Vercel Web Analytics Troubleshooting

No data visible in Web Analytics dashboard

Issue: If you are experiencing a situation where data is not visible in the analytics dashboard or a 404 error occurs while loading `

How to fix:

1. Make sure that you have [enabled Analytics](/docs/analytics/quickstart#enable-web-analytics-in-vercel) in the dashboard.
2. Re-deploy your app to Vercel.
3. Promote your latest deployment to production. To do so, visit the project in your dashboard, and select the **Deployments** tab. From

Web Analytics is not working with a proxy (e.g., Cloudflare)

Issue: Web Analytics may not function when using a proxy, such as Cloudflare.

How to fix:

1. Check your proxy configuration to make sure that all desired pages are correctly proxied to the deployment.
2. Additionally, forward all requests to `/_vercel/insights/*` to the deployments to ensure proper functioning of Web Analytics through t

Routes are not visible in Web Analytics dashboard

Issue: Not all data is visible in the Web Analytics dashboard

How to fix:

1. Verify that you are using the latest version of the `@vercel/analytics` package.
2. Make sure you are using the correct import statement.

```tsx
import { Analytics } from '@vercel/analytics/next'; // Next.js import
```

```tsx
import { Analytics } from '@vercel/analytics/react'; // Generic React import
```

---
title: "Using Web Analytics"
description: "Learn how to use Vercel"
last_updated: "2026-01-16T02:19:25.962Z"
source: "https://vercel.com/docs/analytics/using-web-analytics"

Using Web Analytics

Accessing Web Analytics

To access Web Analytics:

1. Select a project from your dashboard and navigate to the **Analytics** tab.
2. Select the [timeframe](/docs/analytics/using-web-analytics#specifying-a-timeframe) and [environment](/docs/analytics/using-web-analyti
3. Use the panels to [filter](/docs/analytics/filtering) the page or event data you want to view.

Viewing data for a specific dimension

1. Select a project from your dashboard and navigate to the **Analytics** tab.
2. Using panels you can choose whether to view data by:
   - **Pages**: The page url (without query parameters) that the visitor viewed.
   - **Route**: The route, as defined by your application's framework.
   - **Hostname**: Use this to analyze traffic by specific domains. This is beneficial for per-country domains, or for building multi-ten
   - **Referrers**: The URL of the page that referred the visitor to your site. Referrer data is tracked for custom events and for initia
   - **UTM Parameters** (available with [Web Analytics Plus](/docs/analytics/limits-and-pricing) and Enterprise): the forwarded UTM param
   - **Country**: Your visitors location.

   - **Browsers**: Your visitors browsers.
   - **Devices**: Distinction between mobile, tablet, and desktop devices.
   - **Operating System**: Your visitors operating systems.

Specifying a timeframe

1. Select a project from your dashboard and navigate to the **Analytics** tab.
2. Select the timeframe dropdown in the top-right of the page to choose a predefined timeframe. Alternatively, select the Calendar icon t

Viewing environment-specific data

1. Select a project from your dashboard and navigate to the **Analytics** tab.
2. Select the environments dropdown in the top-right of the page to choose **Production**, **Preview**, or **All Environments**. Productio

Exporting data as CSV

To export the data from a panel as a CSV file:

1. Select the **Analytics** tab from your project's [dashboard](/dashboard)
2. From the bottom of the panel you want to export data from, click the three-dot menu
3. Select the **Export as CSV** button

The export will include up to 250 entries from the panel, not just the top entries.

Disabling Web Analytics

1. Select a project from your dashboard and navigate to the **Analytics** tab.
2. Remove the `@vercel/analytics` package from your codebase and dependencies in order to prevent your app from sending analytics events
3. If events have been collected, click on the ellipsis on the top-right of the **Web Analytics** page and select **Disable Web Analytics

---
title: "Audit Logs"
description: "Learn how to track and analyze your team members"
last_updated: "2026-01-16T02:19:26.017Z"
source: "https://vercel.com/docs/audit-log"

Audit Logs

Audit logs help you track and analyze your [team members'](/docs/rbac/managing-team-members) activity. They can be accessed by team membe

Export audit logs

To export and download audit logs:

- Go to **Team Settings > Security > Audit Log**
- Select a timeframe to export a Comma Separated Value ([CSV](#audit-logs-csv-file-structure)) file containing all events occurred during
- Click the **Export CSV** button to download the file

The team owner requesting an export will then receive an email with a link containing the report. This link is used to access the report

Reports generated for the last 90 days (three months) will not impact your billing.

Custom SIEM Log Streaming

In addition to the standard audit log functionalities, Vercel supports custom log streaming to your Security Information and Event Manage

We support the following SIEM options out of the box:

- AWS S3
- Splunk
- Datadog
- Google Cloud Storage

We also support streaming logs to any HTTP endpoint, secured with a custom header.

Allowlisting IP Addresses

If your SIEM requires IP allowlisting, please use the following IP addresses:

```3.217.146.166
23.21.184.92
34.204.154.149
44.213.245.178
44.215.236.82
50.16.203.9
52.1.251.34
52.21.49.187
174.129.36.47
```

Setup Process

To set up custom log streaming to your SIEM:

- From your [dashboard](/dashboard) go to **Team Settings**, select the **Security & Privacy** tab, and scroll to **Audit Log**
- Click the **Configure** button
- Select one of the supported SIEM providers and follow the step-by-step guide

The HTTP POST provider is generic solution to stream audit logs to any configured endpoint. To set this up, you need to provide:

- **URL:** The endpoint that will accept HTTP POST requests
- **HTTP Header Name:** The name of the header, such as `Authorization`
- **HTTP Header Value:** The corresponding value, e.g. `Bearer <token>`

For the request body format, you can choose between:

- **JSON:** Sends a JSON array containing event objects
- **NDJSON:** Sends events as newline-delimited JSON objects, enabling individual processing

Audit Logs CSV file structure

The CSV file can be opened using any spreadsheet-compatible software, and includes the following fields:

Property	**Description**
timestamp	Time and date at which the event occurred
action	Name for the specific event. E.g, `project.created`, `team.member.left`, `project.transfer_out.completed`, `audit
actor_vercel_id	User ID of the team member responsible for an event
actor_name	Account responsible for the action. For example, username of the team member
actor_email	Email address of the team member responsible for a specific event
location	IP address from where the action was performed
user_agent	Details about the application, operating system, vendor, and/or browser version used by the team member
previous	Custom metadata (JSON object) showing the object's previous state
next	Custom metadata (JSON object) showing the object's updated state

`actions`

Vercel logs the following list of `actions` performed by team members.

`alias`

Maps a custom domain or subdomain to a specific deployment or URL of a project. To learn more, see the `vercel alias` [docs](/docs/cli/al

Action Name	**Description**
`alias.created`	Indicates that a new alias was created
`alias.deleted`	Indicates that an alias was deleted
`alias.protection-user-access-request-requested`	An external user requested access to a protected deployment alias URL

`auditlog`

Refers to the audit logs of your Vercel team account.

Action Name	**Description**
`auditlog.export.downloaded`	Indicates that an export of the audit logs was downloaded
`auditlog.export.requested`	Indicates that an export of the audit logs was requested

`cert`

A digital certificate to manage SSL/TLS certificates for your custom domains through the [vercel certs](/docs/cli/certs) command. It is u

Action Name	**Description**
`cert.created`	Indicates that a new certificate was created
`cert.deleted`	Indicates that a new certificate was deleted
`cert.renewed`	Indicates that a new certificate was renewed

`deploy_hook`

Create URLs that accept HTTP POST requests to trigger deployments and rerun the build step. To learn more, see the [Deploy Hooks](/docs/d

Action Name	**Description**
`deploy_hook.deduped`	A deploy hook is de-duplicated which means that multiple instances of the same hook have been combined into

`deployment`

Refers to a successful build of your application. To learn more, see the [deployment](/docs/deployments) docs.

Action Name	**Description**
`deployment.deleted`	Indicates that a deployment was deleted
`deployment.job.errored`	Indicates that a job in a deployment has failed with an error

`domain`

A unique name that identifies your website. To learn more, see the [domains](/docs/domains) docs.

Action Name	**Description**
`domain.auto_renew.changed`	Indicates that the auto-renew setting for a domain was changed
`domain.buy`	Indicates that a domain was purchased
`domain.created`	Indicates that a new domain was created
`domain.delegated`	Indicates that a domain was delegated to another account
`domain.deleted`	Indicates that a domain was deleted
`domain.move_out.requested`	Indicates that a request was made to move a domain out of the current account
`domain.moved_in`	Indicates that a domain was moved into the current account
`domain.moved_out`	Indicates that a domain was moved out of the current account
`domain.record.created`	Indicates that a new domain record was created
`domain.record.deleted`	Indicates that a new domain record was deleted
`domain.record.updated`	Indicates that a new domain record was updated
`domain.transfer_in`	Indicates that a request was made to transfer a domain into the current account
`domain.transfer_in.canceled`	Indicates that a request to transfer a domain into the current account was canceled
`domain.transfer_in.completed`	Indicates that a domain was transferred into the current account

`edge_config`

A key-value data store associated with your Vercel account that enables you to read data in the region closest to the user without queryi

Action Name	**Description**
`edge_config.created`	Indicates that a new edge configuration was created
`edge_config.deleted`	Indicates that a new edge configuration was deleted
`edge_config.updated`	Indicates that a new edge configuration was updated

`integration`

Helps you pair Vercel's functionality with a third-party service to streamline installation, reduce configuration, and increase productiv

Action Name	**Description**
`integration.deleted`	Indicates that an integration was deleted
`integration.installed`	Indicates that an integration was installed
`integration.updated`	Indicates that an integration was updated

`password_protection`

[Password Protection](/docs/security/deployment-protection/methods-to-protect-deployments/password-protection) allows visitors to access

Action Name	**Description**
`password_protection.disabled`	Indicates that password protection was disabled
`password_protection.enabled`	Indicates that password protection was enabled

`preview_deployment_suffix`

Customize the appearance of your preview deployment URLs by adding a valid suffix. To learn more, see the [preview deployment suffix](/do

Action Name	**Description**
`preview_deployment_suffix.disabled`	Indicates that the preview deployment suffix was disabled
`preview_deployment_suffix.enabled`	Indicates that the preview deployment suffix was enabled
`preview_deployment_suffix.updated`	Indicates that the preview deployment suffix was updated

`project`

Refers to actions performed on your Vercel [projects](/docs/projects/overview).

Action Name	**Description**
`project.analytics.disabled`	Indicates that analytics were disabled for the project
`project.analytics.enabled`	Indicates that analytics were enabled for the project
`project.deleted`	Indicates that a project was deleted
`project.env_variable`	This field refers to an environment variable within a project
`project.env_variable.created`	Indicates that a new environment variable was created for the project
`project.env_variable.deleted`	Indicates that a new environment variable was deleted for the project
`project.env_variable.updated`	Indicates that a new environment variable was updated for the project

`project.password_protection`

Refers to the password protection settings for a project.

Action Name	**Description**
`project.password_protection.disabled`	Indicates that password protection was disabled for the project
`project.password_protection.enabled`	Indicates that password protection was enabled for the project
`project.password_protection.updated`	Indicates that password protection was updated for the project

`project.sso_protection`

Refers to the [Single Sign-On (SSO)](/docs/saml) protection settings for a project.

Action Name	**Description**
`project.sso_protection.disabled`	Indicates that SSO protection was disabled for the project
`project.sso_protection.enabled`	Indicates that SSO protection was enabled for the project
`project.sso_protection.updated`	Indicates that SSO protection was updated for the project

`project.rolling_release`

Refers to [Rolling Releases](/docs/rolling-releases) for a project, which allow you to gradually roll out deployments to production.

Action Name	**Description**
`project.rolling_release.aborted`	Indicates that a rolling release was aborted
`project.rolling_release.approved`	Indicates that a rolling release was approved to advance to the next stage
`project.rolling_release.completed`	Indicates that a rolling release was completed successfully
`project.rolling_release.configured`	Indicates that the rolling release configuration was updated for the project
`project.rolling_release.deleted`	Indicates that a rolling release was deleted
`project.rolling_release.started`	Indicates that a rolling release was started

`project.transfer`

Refers to the transfer of a project between Vercel accounts.

Action Name	**Description**
`project.transfer_in.completed`	Indicates that a project transfer into the current account was completed successfully
`project.transfer_in.failed`	Indicates that a project transfer into the current account was failed
`project.transfer_out.completed`	Indicates that a project transfer out of the current account was completed successfully
`project.transfer_out.failed`	Indicates that a project transfer out of the current account was
`project.transfer.started`	Indicates that a project transfer was initiated

`project.web-analytics`

Refers to the generation of web [analytics](/docs/analytics) for a Vercel project.

Action Name	**Description**
`project.web-analytics.disabled`	Indicates that web analytics were disabled for the project
`project.web-analytics.enabled`	Indicates that web analytics were enabled for the project

`shared_env_variable`

Refers to environment variables defined at the team level. To learn more, see the [shared environment variables](/docs/environment-variab

| **Action Name** | **Description** |

| ---------------------------------- | -- |
`shared_env_variable.created`	Indicates that a new shared environment variable was created
`shared_env_variable.decrypted`	Indicates that a new shared environment variable was decrypted
`shared_env_variable.deleted`	Indicates that a new shared environment variable was deleted
`shared_env_variable.updated`	Indicates that a new shared environment variable was updated

`team`

Refers to actions performed by members of a Vercel [team](/docs/accounts/create-a-team).

Action Name	**Description**
`team.avatar.updated`	Indicates that the avatar (profile picture) associated with the team was updated
`team.created`	Indicates that a new team was created
`team.deleted`	Indicates that a new team was deleted
`team.name.updated`	Indicates that the name of the team was updated
`team.slug.updated`	Indicates that the team's unique identifier, or "slug," was updated

`team.member`

Refers to actions performed by any [team member](/docs/accounts/team-members-and-roles).

Action Name	**Description**
`team.member.access_request.confirmed`	Indicates that an access request by a team member was confirmed
`team.member.access_request.declined`	Indicates that an access request by a team member was declined
`team.member.access_request.requested`	Indicates that a team member has requested access to the team
`team.member.added`	Indicates that a new member was added to the team
`team.member.deleted`	Indicates that a member was removed from the team
`team.member.joined`	Indicates that a member has joined the team
`team.member.left`	Indicates that a new member has left the team
`team.member.role.updated`	Indicates that the role of a team member was updated

---
title: "Bot Management"
description: "Learn how to manage bot traffic to your site."
last_updated: "2026-01-16T02:19:26.029Z"
source: "https://vercel.com/docs/bot-management"

Bot Management

Bots generate nearly half of all internet traffic. While many bots serve legitimate purposes like search engine crawling and content aggr

How bot management works

Bot management systems analyze incoming traffic to identify and classify requests based on their source and intent. This includes:

- Verifying and allowing legitimate bots that correctly identify themselves
- Monitoring bot traffic patterns and resource consumption
- Detecting and challenging suspicious traffic that behaves abnormally
- Enforcing browser-like behavior by verifying navigation patterns and cache usage

Methods of bot management and protection

To effectively manage bot traffic and protect against harmful bots, various techniques are used, including:

- Signature-based detection: Inspecting HTTP requests for known bot signatures
- Rate limiting: Restricting how often certain actions can be performed to prevent abuse
- Challenges: [Using JavaScript checks to verify human presence](/docs/vercel-firewall/firewall-concepts#challenge)
- Behavioral analysis: Detecting unusual patterns in user activity that suggest automation

With Vercel, you can use:

- [Managed rulesets](/docs/vercel-waf/managed-rulesets#configure-bot-protection-managed-ruleset) to challenge specific bot traffic
- Rate limiting and challenge actions with [WAF custom rules](/docs/vercel-waf/custom-rules) to prevent bot activity from reaching your a
- [DDoS protection](/docs/security/ddos-mitigation) to defend your application against bot driven attacks
- [Observability](/docs/observability) and [Firewall](/docs/vercel-firewall/firewall-observability) to monitor bot patterns, traffic sour

Bot protection managed ruleset

With Vercel, you can use the bot protection managed ruleset to [challenge](/docs/vercel-firewall/firewall-concepts#challenge) non-browser

- It identifies clients that violate browser-like behavior and serves a javascript challenge to them.
- It prevents requests that falsely claim to be from a browser such as a `curl` request identifying as Chrome.
- It automatically excludes [verified bots](#verified-bots), such as Google's crawler, from evaluation.

To learn more about how the ruleset works, review the [Challenge](/docs/vercel-firewall/firewall-concepts#challenge) section of [Firewall

> **💡 Note:** For trusted automated traffic, you can create [custom WAF
> rules](/docs/vercel-waf/custom-rules) with [bypass
> actions](/docs/vercel-firewall/firewall-concepts#bypass) that will allow this
> traffic to skip the bot protection ruleset.

Enable the ruleset

You can apply the ruleset to your project in [log](/docs/vercel-firewall/firewall-concepts#log) or [challenge](/docs/vercel-firewall/firev

Bot protection ruleset with reverse proxies

Bot Protection does not work when a reverse proxy (e.g. Cloudflare, Azure, or other CDNs) is placed in front of your Vercel deployment. T

[Reverse proxies](/docs/security/reverse-proxy) interfere with Vercel's ability to reliably identify bots:

- **Obscured detection signals**: Legitimate users may be incorrectly challenged because the proxy masks signals that Bot Protection reli
- **Frequent re-challenges**: Some proxies rotate their exit node IPs frequently, forcing Vercel to re-initiate the challenge on every IP

AI bots managed ruleset

Vercel's AI bots managed ruleset allows you to control traffic from AI bots that crawl your site for training data, search purposes, or u

- It identifies and filters requests from known AI crawlers and bots.
- It provides options to log or deny these requests based on your preferences.
- The list of known AI bots is automatically maintained and updated by Vercel.

When new AI bots emerge, they are automatically added to Vercel's managed list and will be handled according to your existing configured

Enable the ruleset

You can apply the ruleset to your project in [log](/docs/vercel-firewall/firewall-concepts#log) or [deny](/docs/vercel-firewall/firewall-

Verified bots

Vercel maintains and continuously updates a comprehensive directory of known legitimate bots from across the internet. This directory is

Bot verification methods

To prove that bots are legitimate and verify their claimed identity, several methods are used:

- **IP Address Verification**: Checking if requests originate from known IP ranges owned by legitimate bot operators (e.g., Google's Goog
- **Reverse DNS Lookup**: Performing reverse DNS queries to verify that an IP address resolves back to the expected domain (e.g., an IP c
- **Cryptographic Verification**: Using digital signatures to authenticate bot requests through protocols like [Web Bot Authentication](h

Verified bots directory

[Submit a bot request](https://bots.fyi/new-bot) if you are a SaaS provider and would like to be added to this list.

---
title: "Advanced BotID Configuration"
description: "Fine-grained control over BotID detection levels and backend domain configuration"
last_updated: "2026-01-16T02:19:26.155Z"
source: "https://vercel.com/docs/botid/advanced-configuration"

Advanced BotID Configuration

Route-by-Route configuration

When you need fine-grained control over BotID's detection levels, you can specify `advancedOptions` to choose between basic and deep anal

> **⚠️ Warning:** **Important**: The `checkLevel` in both client and server configurations must
> be identical for each protected route. A mismatch between client and server
> configurations will cause BotID verification to fail, potentially blocking
> legitimate traffic or allowing bots through. This feature is available in
> `botid@1.4.5` and above

Client-side configuration

In your client-side protection setup, you can specify the check level for each protected path:

```ts
initBotId({
  protect: [
    {
      path: '/api/checkout',
      method: 'POST',
      advancedOptions: {
        checkLevel: 'deepAnalysis', // or 'basic'
      },
    },
    {
      path: '/api/contact',
      method: 'POST',
      advancedOptions: {
        checkLevel: 'basic',
      },
    },
  ],
});
```

Server-side configuration

In your server-side endpoint that uses `checkBotId()`, ensure it matches the client-side configuration.

```ts
export async function POST(request: NextRequest) {
  const verification = await checkBotId({
    advancedOptions: {
      checkLevel: 'deepAnalysis', // Must match client-side config
    },
  });

  if (verification.isBot) {
    return NextResponse.json({ error: 'Access denied' }, { status: 403 });
  }

  // Your protected logic here
}
```

Separate backend domains

By default, BotID validates that requests come from the same host that serves the BotID challenge. However, if your application architect

The `extraAllowedHosts` parameter in `checkBotId()` allows you to specify a list of frontend domains that are permitted to send requests

```ts filename="app/api/backend/route.ts"
```

```
export async function POST(request: NextRequest) {
  const verification = await checkBotId({
    advancedOptions: {
      extraAllowedHosts: ['vercel.com', 'app.vercel.com'],
    },
  });

  if (verification.isBot) {
    return NextResponse.json({ error: 'Access denied' }, { status: 403 });
  }

  // Your protected logic here
}
```

> **💡 Note:** Only add trusted domains to `extraAllowedHosts`. Each domain in this list can
> send requests that will be validated by BotID, so ensure these are domains you
> control.

When to use `extraAllowedHosts`

Use this configuration when:

- Your frontend is hosted on a different domain than your API (e.g., `myapp.com` → `api.myapp.com`)
- You have multiple frontend applications that need to access the same protected backend
- Your architecture uses a separate subdomain for API endpoints

Example with advanced options

You can combine `extraAllowedHosts` with other advanced options:

```ts filename="app/api/backend-advanced/route.ts"
const verification = await checkBotId({
  advancedOptions: {
    checkLevel: 'deepAnalysis',
    extraAllowedHosts: ['app.example.com', 'dashboard.example.com'],
  },
});
```

Next.js Pages Router configuration

When using [Pages Router API handlers](https://nextjs.org/docs/pages/building-your-application/routing/api-routes) in development, pass r

```ts filename="pages/api/endpoint.ts"
import type { NextApiRequest, NextApiResponse } from 'next';
import { checkBotId } from 'botid/server';

export default async function handler(
  req: NextApiRequest,
  res: NextApiResponse,
) {
  const result = await checkBotId({
    advancedOptions: {
      headers: req.headers,
    },
  });

  if (result.isBot) {
    return res.status(403).json({ error: 'Access denied' });
  }

  // Your protected logic here
  res.status(200).json({ success: true });
}
```

> **💡 Note:** Pages Router requires explicit headers in development. In production, headers
> are extracted automatically.

---
title: "Form Submissions"
description: "How to properly handle form submissions with BotID protection"
last_updated: "2026-01-16T02:19:26.159Z"
source: "https://vercel.com/docs/botid/form-submissions"

Form Submissions

BotID does **not** support traditional HTML forms that use the `action` and `method` attributes, such as:

```html
<form id="contact-form" method="POST" action="/api/contact">
  <!-- form fields -->
  <button type="submit">Send</button>
</form>
```

Native form submissions don't work with BotID due to how they are handled by the browser.

To ensure the necessary headers are attached, handle the form submission in JavaScript and send the request using `fetch` or `XMLHttpRequ

Enable form submissions to work with BotID

Here's how you can refactor your form to work with BotID:

```tsx
async function handleSubmit(e: React.FormEvent<HTMLFormElement>) {
  e.preventDefault();
```

```
  const formData = new FormData(e.currentTarget);
  const response = await fetch('/api/contact', {
    method: 'POST',
    body: formData,
  });
  const data = await response.json();
  // handle response
}

return (
  <form onSubmit={handleSubmit}>
    {/* form fields */}
    <button type="submit">Send</button>
  </form>
);
```

Form submissions with Next.js

If you're using Next.js, you can [use a server action](https://nextjs.org/docs/app/guides/forms#how-it-works) in your form and use the `c

```ts filename=app/actions/contact.ts
'use server';
import { checkBotId } from 'botid/server';

export async function submitContact(formData: FormData) {
  const verification = await checkBotId();
  if (verification.isBot) {
    throw new Error('Access denied');
  }
  // process formData
  return { success: true };
}
```

And in your form component:

```tsx filename=app/contact/page.tsx
'use client';
import { submitContact } from '../actions/contact';

export default function ContactForm() {
  async function handleAction(formData: FormData) {
    return submitContact(formData);
  }

  return (
    <form action={handleAction}>
      {/* form fields */}
      <button type="submit">Send</button>
    </form>
  );
}
```

---
title: "Get Started with BotID"
description: "Step-by-step guide to setting up BotID protection in your Vercel project"
last_updated: "2026-01-16T02:19:26.181Z"
source: "https://vercel.com/docs/botid/get-started"

Get Started with BotID

This guide shows you how to add BotID protection to your Vercel project. BotID blocks automated bots while allowing real users through, p

The setup involves three main components:

- Client-side component to run challenges.
- Server-side verification to classify sessions.
- Route configuration to ensure requests are routed through BotID.

Step by step guide

Before setting up BotID, ensure you have **a JavaScript [project deployed](/docs/projects/managing-projects#creating-a-project) on Vercel

- ### Install the package
  Add BotID to your project:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i botid
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i botid
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i botid
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i botid
      ```
    </Code>

```
  </CodeBlock>

- ### Configure redirects
  Use the appropriate configuration method for your framework to set up proxy rewrites. This ensures that ad-blockers, third party script
  ```ts filename="next.config.ts" framework=nextjs-app
  import { withBotId } from 'botid/next/config';

  const nextConfig = {
    // Your existing Next.js config
  };

  export default withBotId(nextConfig);
  ```
  ```js filename="next.config.js" framework=nextjs-app
  import { withBotId } from 'botid/next/config';

  const nextConfig = {
    // Your existing Next.js config
  };

  export default withBotId(nextConfig);
  ```
  ```ts filename="nuxt.config.ts" framework=nuxt
  export default defineNuxtConfig({
    modules: ['botid/nuxt'],
  });
  ```
  ```js filename="nuxt.config.js" framework=nuxt
  export default defineNuxtConfig({
    modules: ['botid/nuxt'],
  });
  ```
  > For \['other']:
  For other frameworks, add the following configuration values to your `vercel.json`:
  ```json filename="vercel.json" framework=other
  {
    "rewrites": [
      {
        "source": "/149e9513-01fa-4fb0-aad4-566afd725d1b/2d206a39-8ed7-437e-a3be-862e0f06eea3/a-4-a/c.js",
        "destination": "https://api.vercel.com/bot-protection/v1/challenge"
      },
      {
        "source": "/149e9513-01fa-4fb0-aad4-566afd725d1b/2d206a39-8ed7-437e-a3be-862e0f06eea3/:path*",
        "destination": "https://api.vercel.com/bot-protection/v1/proxy/:path*"
      }
    ],
    "headers": [
      {
        "source": "/149e9513-01fa-4fb0-aad4-566afd725d1b/2d206a39-8ed7-437e-a3be-862e0f06eea3/:path*",
        "headers": [
          {
            "key": "X-Frame-Options",
            "value": "SAMEORIGIN"
          }
        ]
      }
    ]
  }
  ```

- ### Add client-side protection
  Choose the appropriate method for your framework:
  - **Next.js 15.3+**: Use `initBotId()` in `instrumentation-client.ts` for optimal performance
  - **Other Next.js**: Mount the `<BotIdClient/>` component in your layout `head`
  - **Other frameworks**: Call `initBotId()` during application initialization
  > For \['nextjs-app']:
  Next.js 15.3+ (Recommended)
  ```ts filename="instrumentation-client.ts" framework=nextjs-app
  import { initBotId } from 'botid/client/core';

  // Define the paths that need bot protection.
  // These are paths that are routed to by your app.
  // These can be:
  // - API endpoints (e.g., '/api/checkout')
  // - Server actions invoked from a page (e.g., '/dashboard')
  // - Dynamic routes (e.g., '/api/create/*')

  initBotId({
    protect: [
      {
        path: '/api/checkout',
        method: 'POST',
      },
      {
        // Wildcards can be used to expand multiple segments
        // /team/*/activate will match
        // /team/a/activate
        // /team/a/b/activate
        // /team/a/b/c/activate
        // ...
        path: '/team/*/activate',
        method: 'POST',
      },
      {
        // Wildcards can also be used at the end for dynamic routes
        path: '/api/user/*',
        method: 'POST',
      },
    ],
  });
```
```

```
```js filename="instrumentation-client.js" framework=nextjs-app
import { initBotId } from 'botid/client/core';

// Define the paths that need bot protection.
// These are paths that are routed to by your app.
// These can be:
// - API endpoints (e.g., '/api/checkout')
// - Server actions invoked from a page (e.g., '/dashboard')
// - Dynamic routes (e.g., '/api/create/*')

initBotId({
  protect: [
    {
      path: '/api/checkout',
      method: 'POST',
    },
    {
      // Wildcards can be used to expand multiple segments
      // /team/*/activate will match
      // /team/a/activate
      // /team/a/b/activate
      // /team/a/b/c/activate
      // ...
      path: '/team/*/activate',
      method: 'POST',
    },
    {
      // Wildcards can also be used at the end for dynamic routes
      path: '/api/user/*',
      method: 'POST',
    },
  ],
});
```
Next.js < 15.3
```tsx filename="app/layout.tsx" framework=nextjs-app
import { BotIdClient } from 'botid/client';
import { ReactNode } from 'react';

const protectedRoutes = [
  {
    path: '/api/checkout',
    method: 'POST',
  },
];

type RootLayoutProps = {
  children: ReactNode;
};

export default function RootLayout({ children }: RootLayoutProps) {
  return (
    <html lang="en">
      <head>
        <BotIdClient protect={protectedRoutes} />
      </head>
      <body>{children}</body>
    </html>
  );
}
```
```jsx filename="app/layout.js" framework=nextjs-app
import { BotIdClient } from 'botid/client';
import { ReactNode } from 'react';

const protectedRoutes = [
  {
    path: '/api/checkout',
    method: 'POST',
  },
];

type RootLayoutProps = {
  children: ReactNode;
};

export default function RootLayout({ children }: RootLayoutProps) {
  return (
    <html lang="en">
      <head>
        <BotIdClient protect={protectedRoutes} />
      </head>
      <body>{children}</body>
    </html>
  );
}
```
```jsx filename="app/layout.js" framework=nextjs-app
import { BotIdClient } from 'botid/client';
import { ReactNode } from 'react';

const protectedRoutes = [
  {
    path: '/api/checkout',
    method: 'POST',
  },
];

type RootLayoutProps = {
```
```

```
  children: ReactNode;
};

export default function RootLayout({ children }: RootLayoutProps) {
  return (
    <html lang="en">
      <head>
        <BotIdClient protect={protectedRoutes} />
      </head>
      <body>{children}</body>
    </html>
  );
}
```
```ts filename="plugins/botid.client.ts" framework=nuxt
import { initBotId } from 'botid/client/core';

export default defineNuxtPlugin({
  enforce: 'pre',
  setup() {
    initBotId({
      protect: [{ path: '/api/post-data', method: 'POST' }],
    });
  },
});
```
```js filename="plugins/botid.client.js" framework=nuxt
import { initBotId } from 'botid/client/core';

export default defineNuxtPlugin({
  enforce: 'pre',
  setup() {
    initBotId({
      protect: [{ path: '/api/post-data', method: 'POST' }],
    });
  },
});
```
```ts filename="src/hooks.client.ts" framework=sveltekit
import { initBotId } from 'botid/client/core';

export function init() {
  initBotId({
    protect: [
      {
        path: '/api/post-data',
        method: 'POST',
      },
    ],
  });
}
```
```js filename="src/hooks.client.js" framework=sveltekit
import { initBotId } from 'botid/client/core';

export function init() {
  initBotId({
    protect: [
      {
        path: '/api/post-data',
        method: 'POST',
      },
    ],
  });
}
```
```ts filename="client.ts" framework=other
import { initBotId } from 'botid/client/core';

export function init() {
  initBotId({
    protect: [
      {
        path: '/api/post-data',
        method: 'POST',
      },
    ],
  });
}
```
```js filename="client.js" framework=other
import { initBotId } from 'botid/client/core';

export function init() {
  initBotId({
    protect: [
      {
        path: '/api/post-data',
        method: 'POST',
      },
    ],
  });
}
```

- ### Perform BotID checks on the server
  Use `checkBotId()` on the routes configured in the `<BotIdClient/>` component.
  > **💡 Note:** **Important configuration requirements:** - Not adding the protected route to
  > `<BotIdClient />` will result in `checkBotId()` failing. The client side
  > component dictates which requests to attach special headers to for

> classification purposes. - Local development always returns `isBot: false`
> unless you configure the `developmentOptions` option on `checkBotId()`. [Learn
> more about local development
> behavior](/docs/botid/local-development-behavior).
> For \['nextjs-app']:
Using API routes
```ts filename="app/api/sensitive/route.ts" framework=nextjs-app
import { checkBotId } from 'botid/server';
import { NextRequest, NextResponse } from 'next/server';

export async function POST(request: NextRequest) {
  const verification = await checkBotId();

  if (verification.isBot) {
    return NextResponse.json({ error: 'Access denied' }, { status: 403 });
  }

  const data = await processUserRequest(request);

  return NextResponse.json({ data });
}

async function processUserRequest(request: NextRequest) {
  // Your business logic here
  const body = await request.json();
  // Process the request...
  return { success: true };
}
```
```js filename="app/api/sensitive/route.js" framework=nextjs-app
import { checkBotId } from 'botid/server';
import { NextResponse } from 'next/server';

export async function POST(request) {
  const verification = await checkBotId();

  if (verification.isBot) {
    return NextResponse.json({ error: 'Access denied' }, { status: 403 });
  }

  const data = await processUserRequest(request);

  return NextResponse.json({ data });
}

async function processUserRequest(request) {
  // Your business logic here
  const body = await request.json();
  // Process the request...
  return { success: true };
}
```
Using Server Actions
```ts filename="app/actions/create-user.ts" framework=nextjs-app
'use server';

import { checkBotId } from 'botid/server';

export async function createUser(formData: FormData) {
  const verification = await checkBotId();

  if (verification.isBot) {
    throw new Error('Access denied');
  }

  const userData = {
    name: formData.get('name') as string,
    email: formData.get('email') as string,
  };

  const user = await saveUser(userData);
  return { success: true, user };
}

async function saveUser(userData: { name: string; email: string }) {
  // Your database logic here
  console.log('Saving user:', userData);
  return { id: '123', ...userData };
}
```
```js filename="app/actions/create-user.js" framework=nextjs-app
'use server';

import { checkBotId } from 'botid/server';

export async function createUser(formData) {
  const verification = await checkBotId();

  if (verification.isBot) {
    throw new Error('Access denied');
  }

  const userData = {
    name: formData.get('name'),
    email: formData.get('email'),
  };

  const user = await saveUser(userData);
  return { success: true, user };
}
```

```
async function saveUser(userData) {
  // Your database logic here
  console.log('Saving user:', userData);
  return { id: '123', ...userData };
}
```

```ts filename="sensitive.posts.ts" framework=nuxt
import { checkBotId } from 'botid/server';

export default defineEventHandler(async (event) => {
  const verification = await checkBotId();

  if (verification.isBot) {
    throw createError({
      statusCode: 403,
      statusMessage: 'Access denied',
    });
  }

  const data = await processUserRequest(event);

  return { data };
});

async function processUserRequest(event: any) {
  // Your business logic here
  const body = await readBody(event);
  // Process the request...
  return { success: true };
}
```

```js filename="sensitive.posts.js" framework=nuxt
import { checkBotId } from 'botid/server';

export default defineEventHandler(async (event) => {
  const verification = await checkBotId();

  if (verification.isBot) {
    throw createError({
      statusCode: 403,
      statusMessage: 'Access denied',
    });
  }

  const data = await processUserRequest(event);

  return { data };
});

async function processUserRequest(event) {
  // Your business logic here
  const body = await readBody(event);
  // Process the request...
  return { success: true };
}
```

```ts filename="+server.ts" framework=sveltekit
import { checkBotId } from 'botid/server';
import { json, error } from '@sveltejs/kit';
import type { RequestHandler } from './$types';

export const POST: RequestHandler = async ({ request }) => {
  const verification = await checkBotId();

  if (verification.isBot) {
    throw error(403, 'Access denied');
  }

  const data = await processUserRequest(request);

  return json({ data });
};

async function processUserRequest(request: Request) {
  // Your business logic here
  const body = await request.json();
  // Process the request...
  return { success: true };
}
```

```js filename="+server.js" framework=sveltekit
import { checkBotId } from 'botid/server';
import { json, error } from '@sveltejs/kit';
import type { RequestHandler } from './$types';

export const POST: RequestHandler = async ({ request }) => {
  const verification = await checkBotId();

  if (verification.isBot) {
    throw error(403, 'Access denied');
  }

  const data = await processUserRequest(request);

  return json({ data });
};

async function processUserRequest(request) {
  // Your business logic here
```

```
    const body = await request.json();
    // Process the request...
    return { success: true };
  }
  ```
  ```ts filename="api/sensitive.ts" framework=other
  import { checkBotId } from 'botid/server';

  export async function POST(request: Request) {
    const verification = await checkBotId();

    if (verification.isBot) {
      return Response.json({ error: 'Access denied' }, { status: 403 });
    }

    const data = await processUserRequest(request);

    return Response.json({ data });
  }

  async function processUserRequest(request: Request) {
    // Your business logic here
    const body = await request.json();
    // Process the request...
    return { success: true };
  }
  ```
  ```js filename="api/sensitive.js" framework=other
  import { checkBotId } from 'botid/server';

  export async function POST(request) {
    const verification = await checkBotId();

    if (verification.isBot) {
      return Response.json({ error: 'Access denied' }, { status: 403 });
    }

    const data = await processUserRequest(request);

    return Response.json({ data });
  }

  async function processUserRequest(request) {
    // Your business logic here
    const body = await request.json();
    // Process the request...
    return { success: true };
  }
  ```
  > **💡 Note:** BotID actively runs JavaScript on page sessions and sends headers to the
  > server. If you test with `curl` or visit a protected route directly, BotID
  > will block you in production. To effectively test, make a `fetch` request from
  > a page in your application to the protected route.

- ### Enable BotID deep analysis in Vercel (Recommended)
  From the [Vercel dashboard](/dashboard)
  - Select your Project
  - Click the **Firewall** tab
  - Click **Rules**
  - Enable **Vercel BotID Deep Analysis**

Complete examples

Next.js App Router example

Client-side code for the BotID Next.js implementation:

```tsx filename="app/checkout/page.tsx"
'use client';

import { useState } from 'react';

export default function CheckoutPage() {
  const [loading, setLoading] = useState(false);
  const [message, setMessage] = useState('');

  async function handleCheckout(e: React.FormEvent<HTMLFormElement>) {
    e.preventDefault();
    setLoading(true);

    try {
      const formData = new FormData(e.currentTarget);
      const response = await fetch('/api/checkout', {
        method: 'POST',
        body: JSON.stringify({
          product: formData.get('product'),
          quantity: formData.get('quantity'),
        }),
        headers: {
          'Content-Type': 'application/json',
        },
      });

      if (!response.ok) {
        throw new Error('Checkout failed');
      }

      const data = await response.json();
      setMessage('Checkout successful!');
    } catch (error) {
```

```
      setMessage('Checkout failed. Please try again.');
    } finally {
      setLoading(false);
    }
  }

  return (
    <form onSubmit={handleCheckout}>
      <input name="product" placeholder="Product ID" required />
      <input name="quantity" type="number" placeholder="Quantity" required />
      <button type="submit" disabled={loading}>
        {loading ? 'Processing...' : 'Checkout'}
      </button>
      {message && <p>{message}</p>}
    </form>
  );
}
```

Server-side code for the BotID Next.js implementation:

```ts filename="app/api/checkout/route.ts"
import { checkBotId } from 'botid/server';
import { NextRequest, NextResponse } from 'next/server';

export async function POST(request: NextRequest) {
  // Check if the request is from a bot
  const verification = await checkBotId();

  if (verification.isBot) {
    return NextResponse.json(
      { error: 'Bot detected. Access denied.' },
      { status: 403 },
    );
  }

  // Process the legitimate checkout request
  const body = await request.json();

  // Your checkout logic here
  const order = await processCheckout(body);

  return NextResponse.json({
    success: true,
    orderId: order.id,
  });
}

async function processCheckout(data: any) {
  // Implement your checkout logic
  return { id: 'order-123' };
}
```

---
title: "Local Development Behavior"
description: "How BotID behaves in local development environments and testing options"
last_updated: "2026-01-16T02:19:26.185Z"
source: "https://vercel.com/docs/botid/local-development-behavior"

Local Development Behavior

During local development, BotID behaves differently than in production to facilitate testing and development workflows. In development mo

Using developmentOptions

If you need to test BotID's different return values in local development, you can use the `developmentBypass` option:

```ts filename="app/api/sensitive/route.ts"
import { checkBotId } from 'botid/server';
import { NextRequest, NextResponse } from 'next/server';

export async function POST(request: NextRequest) {
  const verification = await checkBotId({
    developmentOptions: {
      bypass: 'BAD-BOT', // default: 'HUMAN'
    },
  });

  if (verification.isBot) {
    return NextResponse.json({ error: 'Access denied' }, { status: 403 });
  }

  // Your protected logic here
}
```

> **💡 Note:** The `developmentOptions` option only works in development mode and is ignored
> in production. In production, BotID always performs real bot detection.

This allows you to:

- Test your bot handling logic without deploying to production
- Verify error messages and fallback behaviors
- Ensure your application correctly handles both human and bot traffic

---

title: "BotID"
description: "Protect your applications from automated attacks with intelligent bot detection and verification, powered by Kasada."
last_updated: "2026-01-16T02:19:26.195Z"
source: "https://vercel.com/docs/botid"

BotID

[Vercel BotID](/botid) is an invisible CAPTCHA that protects against sophisticated bots without showing visible challenges or requiring u

Sophisticated bots are designed to closely mimic real user behavior. They can run JavaScript, solve CAPTCHAs, and navigate interfaces in v

Resources

- [Getting Started](/docs/botid/get-started) - Setup guide with complete code examples
- [Verified Bots](/docs/botid/verified-bots) - Information about verified bots and their handling
- [Bypass BotID](#bypassing-botid) - Configure bypass rules for BotID detection

Validation flow

BotID validates clients with these steps:

1. A **client-side challenge** is sent to the browser.
2. The **browser** solves the challenge and includes the solution in requests to your high-value endpoint.
3. Your **server-side code** calls `checkBotId()`
4. **Vercel** validates the integrity of the challenge response.
5. **Deep Analysis** uses a machine learning model to analyze the client side signals, if configured.
6. The result of the analysis is returned to the **server-side code** where the application can take action.

Check levels

BotID can be configured to run at one of two levels, **Basic** or **Deep Analysis**. Deep Analysis runs only after the Basic validation h

Basic

The **Basic** level validates the integrity and correctness of the challenge response, catching many less sophisticated bots. It is provi

Deep Analysis

BotID includes **Deep Analysis**, powered by [Kasada](https://www.kasada.io/). Kasada is a leading bot protection provider trusted by For

Deep Analysis uses machine learning to analyze thousands of client side signals to further detect bots, in addition to the basic validati

Deep Analysis provides real-time protection against:

- **Automated attacks**: Shield your application from credential stuffing, brute force attacks, and other automated threats
- **Data scraping**: Prevent unauthorized data extraction and content theft
- **API abuse**: Protect your endpoints from excessive automated requests
- **Spam and fraud**: Block malicious bots while allowing legitimate traffic through
- **Expensive resources**: Prevent bots from consuming expensive infrastructure, bandwidth, compute, or inventory

Deep Analysis counters the most advanced bots by:

1. Silently collecting thousands of signals that distinguish human users from bots
2. Changing detection methods on every page load to prevent reverse engineering and sophisticated bypasses
3. Streaming attack data to a global machine learning system that improves protection for all customers

Pricing

Mode	Plans Available	Price
Basic	All Plans	Free
Deep Analysis	Pro and Enterprise	$1/1000 `checkBotId()` Deep Analysis calls

> **💡 Note:** Calling the `checkBotId()` function in your code triggers BotID Deep Analysis
> charges. Passive page views or requests that don't invoke the `checkBotId()`
> function are not charged.

Bypassing BotID

You can add a bypass rule to the [Vercel WAF](https://vercel.com/docs/vercel-firewall/firewall-concepts#bypass) to let through traffic th

BotID observability

You can view BotID checks by selecting BotID on the firewall traffic dropdown filter of the [Firewall tab](/docs/vercel-firewall/firewall

Metrics are also available in [Observability Plus](/docs/observability/observability-plus).

More resources

- [Advanced configuration](/docs/botid/advanced-configuration) - Fine-grained control over detection levels and backend domains
- [Form submissions](/docs/botid/form-submissions) - Handling form submissions with BotID protection
- [Local Development Behavior](/docs/botid/local-development-behavior) - Testing BotID in development environments

---
title: "Handling Verified Bots"
description: "Information about verified bots and their handling in BotID"
last_updated: "2026-01-16T02:19:26.200Z"
source: "https://vercel.com/docs/botid/verified-bots"

Handling Verified Bots

> **💡 Note:** Handling verified bots is available in botid@1.5.0 and above.

BotID allows you to identify and handle [verified bots](/docs/bot-management#verified-bots) differently from regular bots. This feature e

Vercel maintains a directory of known and verified bots across the web at bots.fyi

Checking for Verified Bots

When using `checkBotId()`, the response includes fields that help you identify verified bots:

```javascript
import { checkBotId } from "botid/server";
import { NextResponse } from "next/server";

export async function POST(request: Request) {
  const botResult = await checkBotId();

  const { isBot, verifiedBotName, isVerifiedBot, verifiedBotCategory } = botResult;

  // Check if it's ChatGPT Operator
  const isOperator = isVerifiedBot && verifiedBotName === "chatgpt-operator";

  if (isBot && !isOperator) {
    return Response.json({ error: "Access denied" }, { status: 403 });
  }

  // ... rest of your handler
  return Response.json(botResult);
}
```

Verified Bot response fields

View our directory of verified bot names and categories [here](/docs/bot-management#verified-bots-directory).

The `checkBotId()` function returns the following fields for verified bots:

- **`isVerifiedBot`**: Boolean indicating whether the bot is verified
- **`verifiedBotName`**: String identifying the specific verified bot
- **`verifiedBotCategory`**: String categorizing the type of verified bot

Example use cases

Verified bots are useful when you want to:

- Allow AI assistants to interact with your API while blocking other bots
- Provide different responses or functionality for verified bots
- Track usage by specific verified bot services
- Enable AI-powered features while maintaining security

---
title: "Build Output Configuration"
description: "Learn about the Build Output Configuration file, which is used to configure the behavior of a Deployment."
last_updated: "2026-01-16T02:19:26.353Z"
source: "https://vercel.com/docs/build-output-api/configuration"

Build Output Configuration

Schema (as TypeScript):

```ts
type Config = {
  version: 3;
  routes?: Route[];
  images?: ImagesConfig;
  wildcard?: WildcardConfig;
  overrides?: OverrideConfig;
  cache?: string[];
  crons?: CronsConfig;
};
```

Config Types:

- [Route](#routes)
- [ImagesConfig](#images)
- [WildcardConfig](#wildcard)
- [OverrideConfig](#overrides)
- [CronsConfig](#crons)

The `config.json` file contains configuration information and metadata for a Deployment.
The individual properties are described in greater detail in the sub-sections below.

At a minimum, a `config.json` file with a `"version"` property is *required*.

`config.json` supported properties

version

The `version` property indicates which version of the Build Output API has been implemented.
The version described in this document is version `3`.

`version` example

```json
  "version": 3
```

routes

The `routes` property describes the routing rules that will be applied to the Deployment. It uses the same syntax as the [`routes` proper

Routes may be used to point certain URL paths to others on your Deployment, attach response headers to paths, and various other routing-r

```ts
type Route = Source | Handler;
```

`Source` route

```ts
type Source = {
  src: string;
  dest?: string;
  headers?: Record<string, string>;
  methods?: string[];
  continue?: boolean;
  caseSensitive?: boolean;
  check?: boolean;
  status?: number;
  has?: HasField;
  missing?: HasField;
  locale?: Locale;
  middlewareRawSrc?: string[];
  middlewarePath?: string;
  mitigate?: Mitigate;
  transforms?: Transform[];
};
```

Key		Required	Description
src		Yes	A PCRE-compatible regular expression that matches each incoming pathname (excluding querystr
dest		No	A destination pathname or full URL, including querystring, with the ability to embed capture
headers		No	A set of headers to apply for responses.
methods		No	A set of HTTP method types. If no method is provided, requests with any HTTP method will be a
continue		No	A boolean to change matching behavior. If true, routing will continue even when the src is ma
caseSensitive		No	Specifies whether or not the route `src` should match with case sensitivity.
check		No	If `true`, the route triggers `handle: 'filesystem'` and `handle: 'rewrite'`
status		No	A status code to respond with. Can be used in tandem with Location: header to implement redi
has	HasField		No	Conditions of the HTTP requ
missing	HasField		No	Conditions of the HTTP requ
locale	Locale		No	Conditions of the Locale of
middlewareRawSrc		No	A list containing the original routes used to generate the `middlewarePath`.
middlewarePath		No	Path to an Edge Runtime function that should be invoked as middleware.
mitigate	Mitigate		No	A mitigation action to appl
transforms	Transform\[]		No	A list of transforms to ap

Source route: `MatchableValue`

```ts
type MatchableValue = {
  eq?: string | number;
  neq?: string;
  inc?: string[];
  ninc?: string[];
  pre?: string;
  suf?: string;
  re?: string;
  gt?: number;
  gte?: number;
  lt?: number;
  lte?: number;
};
```

Key		Required	Description
eq		No	Value must equal this exact value.
neq		No	Value must not equal this value.
inc		No	Value must be included in this array.
ninc		No	Value must not be included in this array.
pre		No	Value must start with this prefix.
suf		No	Value must end with this suffix.
re		No	Value must match this regular expression.
gt		No	Value must be greater than this number.
gte		No	Value must be greater than or equal to th
lt		No	Value must be less than this number.
lte		No	Value must be less than or equal to this

Source route: `HasField`

```ts
type HasField = Array<
  | { type: 'host'; value: string | MatchableValue }
  | {
      type: 'header' | 'cookie' | 'query';
      key: string;
      value?: string | MatchableValue;
    }
>;
```

Key		Required	Description
type	"host" | "header" | "cookie" | "query"	Yes	Determines the HasField type.
key		No*	Required for header, cookie, and query types. The key to match against.
value	MatchableValue	No	The value to match against using string or MatchableValue conditions.

Source route: `Locale`

```ts
type Locale = {
  redirect?: Record<string, string>;
```

```
  cookie?: string;
};
```

Key		Required	Description
redirect		Yes	An object of keys that represent locales to check for (`en`, `fr`, etc.) that map to routes to re
cookie		No	Cookie name that can override the Accept-Language header for determining the current locale.

Source route: `Mitigate`

```ts
type Mitigate = {
  action: 'challenge' | 'deny';
};
```

Key		Required	Description
action	"challenge" | "deny"	Yes	The action to take when the route is m

Source route: `Transform`

```ts
type Transform = {
  type: 'request.headers' | 'request.query' | 'response.headers';
  op: 'append' | 'set' | 'delete';
  target: {
    key: string | Omit<MatchableValue, 're'>; // re is not supported for transforms
  };
  args?: string | string[];
};
```

Key		Required	Description
type	"request.headers" | "response.headers" | "request.query"		
op	"append" | "set" | "delete"		
target	`{ key: string | Omit<MatchableValue, 're'> }`		
args		No	The arguments to pass to the transform.

Handler route

The routing system has multiple phases. The `handle` value indicates the start of a phase. All following routes are only checked in that

```ts
type HandleValue =
  | 'rewrite'
  | 'filesystem' // check matches after the filesystem misses
  | 'resource'
  | 'miss' // check matches after every filesystem miss
  | 'hit'
  | 'error'; //  check matches after error (500, 404, etc.)

type Handler = {
  handle: HandleValue;
  src?: string;
  dest?: string;
  status?: number;
};
```

Key		Required	Description
handle	HandleValue	Yes	The phase of routing when all subsequ
src		No	A PCRE-compatible regular expression that matches each incoming pathname (excluding querystring).
dest		No	A destination pathname or full URL, including querystring, with the ability to embed capture groups as
status		No	A status code to respond with. Can be used in tandem with `Location:` header to implement redirects.

Routing rule example

The following example shows a routing rule that will cause the `/redirect` path to perform an HTTP redirect to an external URL:

```json
  "routes": [
    {
      "src": "/redirect",
      "status": 308,
      "headers": { "Location": "https://example.com/" }
    }
  ]
```

images

The `images` property defines the behavior of Vercel's native [Image Optimization API](/docs/image-optimization), which allows on-demand

```ts
type ImageFormat = 'image/avif' | 'image/webp';

type RemotePattern = {
  protocol?: 'http' | 'https';
  hostname: string;
  port?: string;
  pathname?: string;
  search?: string;
};

type LocalPattern = {
  pathname?: string;
```

```
  search?: string;
};

type ImagesConfig = {
  sizes: number[];
  domains: string[];
  remotePatterns?: RemotePattern[];
  localPatterns?: LocalPattern[];
  qualities?: number[];
  minimumCacheTTL?: number; // seconds
  formats?: ImageFormat[];
  dangerouslyAllowSVG?: boolean;
  contentSecurityPolicy?: string;
  contentDispositionType?: string;
};
```

Key		Required	Description
sizes		Yes	Allowed image widths.
domains		Yes	Allowed external domains that can use Image Optimization. Leave empty for only allowing
remotePatterns	RemotePattern\[]	No	Allowed external pat
localPatterns	LocalPattern\[]	No	Allowed local patter
qualities		No	Allowed image qualities. Leave undefined to allow all possibilities, 1 to 100.
minimumCacheTTL		No	Cache duration (in seconds) for the optimized images.
formats	ImageFormat\[]	No	Supported output ima
dangerouslyAllowSVG		No	Allow SVG input image URLs. This is disabled by default for security purposes.
contentSecurityPolicy		No	Change the [Content Security Policy](https://developer.mozilla.org/docs/Web/HTTP/CSP)
contentDispositionType		No	Specifies the value of the `"Content-Disposition"` response header.

`images` example

The following example shows an image optimization configuration that specifies allowed image size dimensions, external domains, caching l

```json
  "images": {
    "sizes": [640, 750, 828, 1080, 1200],
    "domains": [],
    "minimumCacheTTL": 60,
    "formats": ["image/avif", "image/webp"],
    "qualities": [25, 50, 75],
    "localPatterns": [{
      "pathname": "^/assets/.*$",
      "search": ""
    }]
    "remotePatterns": [{
      "protocol": "https",
      "hostname": "^via\\.placeholder\\.com$",
      "port": "",
      "pathname": "^/1280x640/.*$",
      "search": "?v=1"
    }]
  }
```

API

When the `images` property is defined, the Image Optimization API will be available by visiting the `/_vercel/image` path. When the `imag

The API accepts the following query string parameters:

Key		Required	Example	Description
url		Yes	`/assets/me.png`	The URL of the source image that should be optimized. Absolute URLs must match a patte
w		Yes	`200`	The width (in pixels) that the source image should be resized to. Must match a value de
q		Yes	`75`	The quality that the source image should be reduced to. Must be between 1 (lowest quali

wildcard

The `wildcard` property relates to Vercel's Internationalization feature. The way
it works is the domain names listed in this array are mapped to the `$wildcard`
routing variable, which can be referenced by the [`routes` configuration](#routes).

Each of the domain names specified in the `wildcard` configuration will need to
be assigned as [Production Domains in the Project Settings](/docs/domains).

```ts
type WildCard = {
  domain: string;
  value: string;
};

type WildcardConfig = Array<WildCard>;
```

`wildcard` supported properties

Objects contained within the `wildcard` configuration support the following properties:

Key		Required	Description
domain		Yes	The domain name to match for this wildcard configuration.
value		Yes	The value of the `$wildcard` match that will be available for `routes` to utilize.

`wildcard` example

The following example shows a wildcard configuration where the matching
domain name will be served the localized version of the blog post HTML file:

```json
```

```
  "wildcard": [
    {
      "domain": "example.com",
      "value": "en-US"
    },
    {
      "domain": "example.nl",
      "value": "nl-NL"
    },
    {
      "domain": "example.fr",
      "value": "fr"
    }
  ],
  "routes": [
    { "src": "/blog", "dest": "/blog.$wildcard.html" }
  ]
```

overrides

The `overrides` property allows for overriding the output of one or more [static files](/docs/build-output-api/v3/primitives#static-files
within the `.vercel/output/static` directory.

The main use-cases are to override the `Content-Type` header that will be served for a static file,
and/or to serve a static file in the Vercel Deployment from a different URL path than how it is stored on the file system.

```ts
type Override = {
  path?: string;
  contentType?: string;
};

type OverrideConfig = Record<string, Override>;
```

`overrides` supported properties

Objects contained within the `overrides` configuration support the following properties:

Key		Required	Description		
path		No	The URL path where the static file will be accessible from.		
contentType		No	The value of the `Content-Type` HTTP response header that will be served with the static file.		

`overrides` example

The following example shows an override configuration where an HTML file can be accessed
without the `.html` file extension:

```json
  "overrides": {
    "blog.html": {
      "path": "blog"
    }
  }
```

cache

The `cache` property is an array of file paths and/or glob patterns that should be re-populated
within the build sandbox upon subsequent Deployments.

Note that this property is only relevant when Vercel is building a Project from source
code, meaning it is not relevant when building locally or when creating a Deployment
from "prebuilt" build artifacts.

```ts
type Cache = string[];
```

`cache` example

```json
  "cache": [
    ".cache/**",
    "node_modules/**"
  ]
```

framework

The optional `framework` property is an object describing the framework of the built outputs.

This value is used for display purposes only.

```ts
type Framework = {
  version: string;
};
```

`framework` example

```json
  "framework": {
    "version": "1.2.3"
  }
```

crons

The optional `crons` property is an object describing the [cron jobs](/docs/cron-jobs) for the production deployment of a project.

```ts
type Cron = {
  path: string;
  schedule: string;
};

type CronsConfig = Cron[];
```

`crons` example

```json
  "crons": [{
    "path": "/api/cron",
    "schedule": "0 0 * * *"
  }]
```

Full `config.json` example

```json
{
  "version": 3,
  "routes": [
    {
      "src": "/redirect",
      "status": 308,
      "headers": { "Location": "https://example.com/" }
    },
    {
      "src": "/blog",
      "dest": "/blog.$wildcard.html"
    }
  ],
  "images": {
    "sizes": [640, 750, 828, 1080, 1200],
    "domains": [],
    "minimumCacheTTL": 60,
    "formats": ["image/avif", "image/webp"],
    "qualities": [25, 50, 75],
    "localPatterns": [{
      "pathname": "^/assets/.*$",
      "search": ""
    }]
    "remotePatterns": [
      {
        "protocol": "https",
        "hostname": "^via\\.placeholder\\.com$",
        "port": "",
        "pathname": "^/1280x640/.*$",
        "search": "?v=1"
      }
    ]
  },
  "wildcard": [
    {
      "domain": "example.com",
      "value": "en-US"
    },
    {
      "domain": "example.nl",
      "value": "nl-NL"
    },
    {
      "domain": "example.fr",
      "value": "fr"
    }
  ],
  "overrides": {
    "blog.html": {
      "path": "blog"
    }
  },
  "cache": [".cache/**", "node_modules/**"],
  "framework": {
    "version": "1.2.3"
  },
  "crons": [
    {
      "path": "/api/cron",
      "schedule": "* * * * *"
    }
  ]
}
```

---
title: "Features"
description: "Learn how to implement common Vercel platform features through the Build Output API."
last_updated: "2026-01-16T02:19:26.297Z"
source: "https://vercel.com/docs/build-output-api/features"

Features

This section describes how to implement common Vercel platform features through the
Build Output API through a combination of platform primitives, configuration and
helper functions.

High-level routing

The `vercel.json` file supports an [easier-to-use syntax for routing through properties
like `rewrites`, `headers`, etc](/docs/project-configuration). However, the
[`config.json` "routes" property](/docs/build-output-api/v3/configuration#routes) supports a
lower-level syntax.

The `getTransformedRoutes()` function from the [`@vercel/routing-utils` npm package](https://www.npmjs.com/package/@vercel/routing-utils)
can be used to convert this higher-level syntax into the lower-level format that is
supported by the Build Output API. For example:

```typescript
import { writeFileSync } from 'fs';
import { getTransformedRoutes } from '@vercel/routing-utils';

const { routes } = getTransformedRoutes({
  trailingSlash: false,
  redirects: [
    { source: '/me', destination: '/profile.html' },
    { source: '/view-source', destination: 'https://github.com/vercel/vercel' },
  ],
});

const config = {
  version: 3,
  routes,
};
writeFileSync('.vercel/output/config.json', JSON.stringify(config));
```

`cleanUrls`

The [`cleanUrls: true` routing feature](/docs/project-configuration#cleanurls) is a special case because, in addition to the routes
generated with the helper function above, it *also* requires that the static HTML files
have their `.html` suffix removed.

This can be achieved by utilizing the [`"overrides"` property in the `config.json` file](/docs/build-output-api/v3/configuration#override

```typescript
import { writeFileSync } from 'fs';
import { getTransformedRoutes } from '@vercel/routing-utils';

const { routes } = getTransformedRoutes({
  cleanUrls: true,
});

const config = {
  version: 3,
  routes,
  overrides: {
    'blog.html': {
      path: 'blog',
    },
  },
};
writeFileSync('.vercel/output/config.json', JSON.stringify(config));
```

Edge Middleware

An Edge Runtime function can act as a "middleware" in the HTTP request lifecycle for
a Deployment. Middleware is useful for implementing functionality that may be
shared by many URL paths in a Project (e.g. authentication),
before passing the request through to the underlying resource (such as a page or asset)
at that path.

An Edge Middleware is represented on the file system in the same format as an [Edge
Function](/docs/build-output-api/v3/#vercel-primitives/edge-functions). To use the middleware,
add additional rules in the [`routes` configuration](/docs/build-output-api/v3/configuration#routes)
mapping URLs (using the `src` property) to the middleware (using the `middlewarePath` property).

Edge Middleware example

The following example adds a rule that calls the `auth` middleware for any URL that
starts with `/api`, before continuing to the underlying resource:

```json
  "routes": [
    {
      "src": "/api/(.*)",
      "middlewareRawSrc": ["/api"],
      "middlewarePath": "auth",
      "continue": true
    }
  ]
```

Draft Mode

When using [Prerender Functions](/docs/build-output-api/v3/primitives#prerender-functions), you may want to implement "Draft Mode" which

To implement this, the `bypassToken` of the `<name>.prerender-config.json` file should be set to a randomized string that you generate at

To enable "Draft Mode", a cookie with the name `__prerender_bypass` needs to be set (i.e. by a Vercel Function) with the value of the `by

On-Demand Incremental Static Regeneration (ISR)

When using [Prerender Functions](/docs/build-output-api/v3/primitives#prerender-functions), you may want to implement "On-Demand Incremen

To implement this, the `bypassToken` of the `<name>.prerender-config.json` file should be set to a randomized string that you generate at

To trigger "On-Demand Incremental Static Regeneration (ISR)" and revalidate a path to a Prerender Function, make a `GET` or `HEAD` reques

---
title: "Build Output API"
description: "The Build Output API is a file-system-based specification for a directory structure that can produce a Vercel deployment."
last_updated: "2026-01-16T02:19:26.215Z"
source: "https://vercel.com/docs/build-output-api"

Build Output API

The Build Output API is a file-system-based specification for a directory structure that can produce a Vercel deployment.

Framework authors can take advantage of [framework-defined infrastructure](/blog/framework-defined-infrastructure) by implementing this d

Overview

The Build Output API closely maps to the Vercel product features in a logical and understandable format.

It is primarily targeted toward authors of web frameworks who would like to utilize all of the Vercel platform features, such as Vercel F

If you are a framework author looking to integrate with Vercel, you can use
this reference as a way to understand which files the framework should emit to the
`.vercel/output` directory.

If you are not using a framework and would like to still take advantage of any of the features
that those frameworks provide, you can create the `.vercel/output` directory and populate it
according to this specification yourself.

You can find complete examples of Build Output API directories in [vercel/examples](https://github.com/vercel/examples/tree/main/build-ou

Check out our blog post on using the [Build Output API to build your own framework](/blog/build-your-own-web-framework) with Vercel.

Known limitations

Native Dependencies: Please keep in mind that when building locally, your build tools will
compile native dependencies targeting your machine's architecture. This will not necessarily match
what runs in production on Vercel.

For projects that depend
on native binaries, you should build on a host machine running Linux with a `x64` CPU architecture,
ideally the same as the platform [Build Image](/docs/deployments/build-image).

More resources

- [Configuration](/docs/build-output-api/v3/configuration)
- [Vercel Primitives](/docs/build-output-api/v3/primitives)
- [Features](/docs/build-output-api/v3/features)

---
title: "Vercel Primitives"
description: "Learn about the Vercel platform primitives and how they work together to create a Vercel Deployment."
last_updated: "2026-01-16T02:19:26.319Z"
source: "https://vercel.com/docs/build-output-api/primitives"

Vercel Primitives

The following directories, code files, and configuration files represent all Vercel platform primitives.
These primitives are the "building blocks" that make up a Vercel Deployment.

Files outside of these directories are ignored and will not be served to visitors.

Static files

Static files that are *publicly accessible* from the Deployment URL should be placed in the `.vercel/output/static` directory.

These files are served with the [Vercel Edge CDN](/docs/cdn).

Files placed within this directory will be made available at the root (`/`) of the Deployment URL and neither their contents, nor their f

Configuration

There is no standalone configuration file that relates to static files.

However, certain properties of static files (such as the `Content-Type` response header) can be modified by utilizing the [`overrides` pr

Directory structure for static files

The following example shows static files placed into the `.vercel/output/static` directory:

Serverless Functions

A [Vercel Function](/docs/functions) is represented on the file system as
a directory with a `.func` suffix on the name, contained within the `.vercel/output/functions` directory.

Conceptually, you can think of this `.func` directory as a filesystem mount for a Vercel Function:
the files below the `.func` directory are included (recursively) and files above the `.func` directory are not included.
Private files may safely be placed within this directory
because they will not be directly accessible to end-users. However, they can be referenced by code
that will be executed by the Vercel Function.

A `.func` directory may be a symlink to another `.func` directory in cases where you want to have more than one path point to the same un

A configuration file named `.vc-config.json` **must** be included within the `.func` directory,
which contains information about how Vercel should construct the Vercel Function.

The `.func` suffix on the directory name is *not included* as part of the URL path of Vercel Function on the Deployment.
For example, a directory located at `.vercel/output/functions/api/posts.func` will be accessible at the URL path `/api/posts` of the Deplo

Serverless function configuration

The `.vc-config.json` configuration file contains information related to how the Vercel Function will be created by Vercel.

Base config

```ts
type ServerlessFunctionConfig = {
  handler: string;
  runtime: string;
  memory?: number;
  maxDuration?: number;
  environment: Record<string, string>[];
  regions?: string[];
  supportsWrapper?: boolean;
  supportsResponseStreaming?: boolean;
};
```

Key		Required	Description
runtime		Yes	Specifies which "runtime" will be used to execute the Vercel Function. See [Runtime
handler		Yes	Indicates the initial file where code will be executed for the Vercel Function.
memory		No	Amount of memory (RAM in MB) that will be allocated to the Vercel Function. See [siz
architecture		No	Specifies the instruction set "architecture" the Vercel Function supports. Either `
maxDuration		No	Maximum duration (in seconds) that will be allowed for the Vercel Function. See [siz
environment		No	Map of additional environment variables that will be available to the Vercel Fun
regions		No	List of Vercel Regions where the Vercel Function will be deployed to.
supportsWrapper		No	True if a custom runtime has support for Lambda runtime wrappers.
supportsResponseStreaming		No	When true, the Vercel Function will stream the response to the client.

Node.js config

This extends the [Base Config](#base-config) for Node.js Serverless Functions.

```ts
type NodejsServerlessFunctionConfig = ServerlessFunctionConfig & {
  launcherType: 'Nodejs';
  shouldAddHelpers?: boolean; // default: false
  shouldAddSourcemapSupport?: boolean; // default: false
};
```

Key		Required	Description
launcherType	"Nodejs"	Yes	Specifies which la
shouldAddHelpers		No	Enables request and response helpers methods.
shouldAddSourcemapSupport		No	Enables source map support for stack traces at runtime.
awsLambdaHandler		No	[AWS Handler Value](https://docs.aws.amazon.com/lambda/latest/dg/nodejs-handler.htm

Node.js config example

This is what the `.vc-config.json` configuration file could look like in a real scenario:

```json
{
  "runtime": "nodejs22.x",
  "handler": "serve.js",
  "maxDuration": 3,
  "launcherType": "Nodejs",
  "shouldAddHelpers": true,
  "shouldAddSourcemapSupport": true
}
```

Directory structure for Serverless Functions

The following example shows a directory structure where the Vercel Function will be accessible at the `/serverless` URL path of the Deplo

Edge Functions

An [Edge Function](/docs/functions/edge-functions) is represented on the file system as
a directory with a `.func` suffix on the name, contained within the `.vercel/output/functions` directory.

The `.func` directory requires at least one JavaScript or TypeScript source file which will serve as the `entrypoint` of the function. Ad

WebAssembly (Wasm) files may also be placed in this directory for an Edge Function to import.
See [Using a WebAssembly file](/docs/functions/runtimes/wasm) for more information.

A configuration file named `.vc-config.json` **must** be included within the `.func` directory, which contains information about how Verc

The `.func` suffix is *not included* in the URL path. For example, a directory located at `.vercel/output/functions/api/edge.func` will b

Supported content types

Edge Functions will bundle an `entrypoint` and all supported source files that are imported by that `entrypoint`. The following list incl

- `.js`
- `.json`
- `.wasm`

Edge Function configuration

The `.vc-config.json` configuration file contains information related to how the Edge Function will be created by Vercel.

```ts
type EdgeFunctionConfig = {
  runtime: 'edge';
  entrypoint: string;
  envVarsInUse?: string[];
  regions?: 'all' | string | string[];
};
```

Key		Required	Description
runtime		Yes	The `runtime: "edge"` property is required to indicate that this directory represents an Edge Fu
entrypoint		Yes	Indicates the initial file where code will be executed for the Edge Function.
envVarsInUse		No	List of environment variable names that will be available for the Edge Function to utilize.
regions		No	List of regions or a specific region that the edge function will be available in, defaults to `all

Edge Function config example

This is what the `.vc-config.json` configuration file could look like in a real scenario:

```json
{
  "runtime": "edge",
  "entrypoint": "index.js",
  "envVarsInUse": ["DATABASE_API_KEY"]
}
```

Directory structure for Edge Functions

The following example shows a directory structure where the Edge Function will be accessible at the `/edge` URL path of the Deployment:

Prerender Functions

A Prerender asset is a Vercel Function that will be cached by the Vercel CDN
in the same way as a static file. This concept is also known as [Incremental Static Regeneration](/docs/incremental-static-regeneration).

On the file system, a Prerender is represented in the same way as a Vercel Function,
with an additional configuration file that describes the cache invalidation rules for the Prerender asset.

An optional "fallback" static file can also be specified, which will be served when there is no cached version available.

Prerender configuration file

The `<name>.prerender-config.json` configuration file contains information related to how the Prerender Function will be created by Verce

```ts
type PrerenderFunctionConfig = {
  expiration: number | false;
  group?: number;
  bypassToken?: string;
  fallback?: string;
  allowQuery?: string[];
  passQuery?: boolean;
  initialHeaders?: Record<string, string>;
  initialStatus?: number;
};
```

Key		Required	Description
expiration		Yes	Expiration time (in seconds) before the cached asset will be re-generated by invoking the Ve
group		No	Option group number of the asset. Prerender assets with the same group number w
bypassToken		No	Random token assigned to the `__prerender_bypass` cookie when [D
fallback		No	Name of the optional fallback file relative to the configuration file.
allowQuery		No	List of query string parameter names that will be cached independently. If an em
passQuery		No	When true, the query string will be present on the `request` argument passed to
initialHeaders		No	Initial headers to be included with the prerendered response that was generated at build time.
initialStatus		No	Initial HTTP status code to be included with the prerendered response that was

Fallback static file

A Prerender asset may also include a static "fallback" version that is generated at build-time.
The fallback file will be served by Vercel while there is not yet a cached version that was generated during runtime.

When the fallback file is served, the Vercel Function will also be invoked "out-of-band" to
re-generate a new version of the asset that will be cached and served for future HTTP requests.

Prerender config example

This is what an `example.prerender-config.json` file could look like in a real scenario:

```json
{
  "expiration": 60,
  "group": 1,
  "bypassToken": "03326da8bea31b919fa3a31c85747ddc",
  "fallback": "example.prerender-fallback.html",
  "allowQuery": ["id"]
}
```

Directory structure for Prerender Functions

The following example shows a directory structure where the Prerender will be accessible at the `/blog` URL path of the Deployment:

---

title: "Build Features for Customizing Deployments"
description: "Learn how to customize your deployments using Vercel"
last_updated: "2026-01-16T02:19:26.364Z"
source: "https://vercel.com/docs/builds/build-features"

Build Features for Customizing Deployments

Vercel provides the following features to customize your deployments:

- [Private npm packages](#private-npm-packages)
- [Ignored files and folders](#ignored-files-and-folders)
- [Special paths](#special-paths)
- [Git submodules](#git-submodules)

Private npm packages

When your project's code is using private `npm` modules that require authentication, you need to perform an additional step to install pr

To install private `npm` modules, define `NPM_TOKEN` as an [Environment Variable](/docs/environment-variables) in your project. Alternati

To learn more, check out the [guide here](/kb/guide/using-private-dependencies-with-vercel) if you need help configuring private dependen

Ignored files and folders

Vercel ignores certain files and folders by default and prevents them from being uploaded during the deployment process for security and

```bash filename="ignored-files"
.hg
.git
.gitmodules
.svn
.cache
.next
.now
.vercel
.npmignore
.dockerignore
.gitignore
.*.swp
.DS_Store
.wafpicke-*
.lock-wscript
.env.local
.env.*.local
.venv
.yarn/cache
npm-debug.log
config.gypi
node_modules
__pycache__
venv
CVS
```

The `.vercel/output` directory is **not** ignored when [`vercel deploy --prebuilt`](/docs/cli/deploying-from-cli#deploying-from-local-bui

> **💡 Note:** You do not need to add any of the above files and folders to your
> file because it is done automatically
> by Vercel.

Special paths

Vercel allows you to access the source code and build logs for your deployment using special pathnames for **Build Logs and Source Protec

All deployment URLs have two special pathnames to access the source code and the build logs:

- `/_src`
- `/_logs`

By default, these routes are protected so that they can only be accessed by you and the members of your Vercel Team.

Source View

By appending `/_src` to a Deployment URL or [Custom Domain](/docs/domains/add-a-domain) in your web browser, you will be redirected to th

Logs View

By appending `/_logs` to a Deployment URL or [Custom Domain](/docs/domains/add-a-domain) in your web browser, you can see a real-time str

Security considerations

The pathnames `/_src` and `/_logs` redirect to `https://vercel.com` and **require logging into your Vercel account** to access any sensit

You can configure these paths to make them publicly accessible under the Security tab on the Project Settings page. You can learn more ab

Git submodules

On Vercel, you can deploy [Git submodules](https://git-scm.com/book/en/v2/Git-Tools-Submodules) with a [Git provider](/docs/git) as long

---
title: "Build image overview"
description: "Learn about the container image used for Vercel builds."
last_updated: "2026-01-16T02:19:26.377Z"
source: "https://vercel.com/docs/builds/build-image"

Build image overview

When you initiate a deployment, Vercel will [build your project](/docs/builds) within a container using the build image.
Vercel supports [multiple runtimes](/docs/functions/runtimes).

Runtime	[Build image](/docs/builds/build-image)
[Node.js](/docs/functions/runtimes/node-js)	`24.x` `22.x` `20.x`
[Edge](/docs/functions/runtimes/edge-runtime)	
[Python](/docs/functions/runtimes/python)	`3.12`
[Ruby](/docs/functions/runtimes/ruby)	`3.3.x`
[Community Runtimes](/docs/functions/runtimes#community-runtimes)	

The build image uses [Amazon Linux 2023](https://aws.amazon.com/linux/amazon-linux-2023/) as its base image.

Pre-installed packages

The following packages are pre-installed in the build image with `dnf`, the default package manager for Amazon Linux 2023.

Running the build image locally

Vercel does not provide the build image itself, but you can use the Amazon Linux 2023 base image to test things locally:

```bash filename="terminal"
docker run --rm -it amazonlinux:2023.2.20231011.0 sh
```

When you are done, run `exit` to return.

Installing additional packages

You can install additional packages into the build container by configuring the [Install Command](/docs/deployments/configure-a-build#ins

The build image includes access to repositories with stable versions of popular packages. You can list all packages with the following co

```bash filename="terminal"
dnf list
```

You can search for a package by name with the following command:

```bash filename="terminal"
dnf search my-package-here
```

You can install a package by name with the following command:

```bash filename="terminal"
dnf install -y my-package-here
```

---
title: "Build Queues"
description: "Understand how concurrency and same branch build queues manage multiple simultaneous deployments."
last_updated: "2026-01-16T02:19:26.383Z"
source: "https://vercel.com/docs/builds/build-queues"

Build Queues

Build queueing is when a build must wait for resources to become available before starting. This creates more time between when the code

- [With On-Demand Concurrent Builds](#with-on-demand-concurrent-builds), builds will never queue.
- [Without On-Demand Concurrent Builds](#without-on-demand-concurrent-builds), builds can queue under the conditions specified below.

With On-Demand Concurrent Builds

[On-Demand Concurrent Builds](/docs/builds/managing-builds#on-demand-concurrent-builds) prevent build queueing so your team can build fas

You can choose between two modes:

- **Run all builds immediately**: All builds proceed in parallel without waiting. Your builds will never be queued.
- **Run up to one build per branch**: Limit to one active build per branch. New deployments to the same branch won't be processed while t

To configure on-demand concurrent builds, see [Project-level on-demand concurrent builds](/docs/builds/managing-builds#project-level-on-d

**If you're experiencing build queues, we strongly recommend [enabling On-Demand Concurrent Builds](https://vercel.com/d?to=%2F%5Bteam%5D

Without On-Demand Concurrent Builds

When multiple deployments are started concurrently from code changes, Vercel's build system places deployments into one of the following

- [Concurrency queue](#concurrency-queue): The basics of build resource management
- [Git branch queue](#git-branch-queue): How builds to the same branch are managed

Concurrency queue

This queue manages how many builds can run in parallel based on the number of [concurrent build slots](/docs/builds/managing-builds#concu

How concurrent build slots work

Concurrent build slots are the key factor in concurrent build queuing. They control how many builds can run at the same time and ensure e

Each account plan comes with a predefined number of build slots:

- Hobby accounts allow one build at a time.
- Pro accounts support up to 12 simultaneous builds.
- Enterprise accounts can have [custom limits](/docs/deployments/concurrent-builds#usage-and-limits) based on their plan.

Git branch queue

Builds are handled sequentially. If new commits are pushed while a build is in progress:

1. The current build is completed first.
2. Queued builds for earlier commits are skipped.
3. The most recent commit is built and deployed.

This means that commits in between the current build and most recent commit will not produce builds.

> **💡 Note:** Enterprise users can use [Urgent On-Demand
> Concurrency](/docs/deployments/managing-builds#urgent-on-demand-concurrent-builds)
> to skip the Git branch queue for specific builds.

---
title: "Configuring a Build"
description: "Vercel automatically configures the build settings for many front-end frameworks, but you can also customize the build acco
last_updated: "2026-01-16T02:19:26.617Z"
source: "https://vercel.com/docs/builds/configure-a-build"

Configuring a Build

When you make a [deployment](/docs/deployments), Vercel **builds** your project. During this time, Vercel performs a "shallow clone" on y

Vercel automatically configures the build settings for many front-end frameworks, but you can also customize the build according to your

To configure your Vercel build with customized settings, choose a project from the [dashboard](/dashboard) and go to its **Settings** tab

The **Build and Deployment** section of the Settings tab offers the following options to customize your build settings:

- [Framework Settings](#framework-settings)
- [Root Directory](#root-directory)
- [Node.js Version](/docs/functions/runtimes/node-js/node-js-versions#setting-the-node.js-version-in-project-settings)
- [Prioritizing Production Builds](/docs/deployments/concurrent-builds#prioritize-production-builds)
- [On-Demand Concurrent Builds](/docs/deployments/managing-builds#on-demand-concurrent-builds)

Framework Settings

If you'd like to override the settings or specify a different framework, you can do so from
the **Build & Development Settings** section.

Framework Preset

You have a wide range of frameworks to choose from, including Next.js, Svelte, and Nuxt. In several use cases, Vercel automatically detec

Inside the Framework Preset settings, use the drop-down menu to select the framework of your choice. This selection will be used for **al

- **Angular**: Angular is a TypeScript-based cross-platform framework from Google.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/angular) | [View Demo](https
- **Astro**: Astro is a new kind of static site builder for the modern web. Powerful developer experience meets lightweight output.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/astro) | [View Demo](https:/
- **Brunch**: Brunch is a fast and simple webapp build tool with seamless incremental compilation for rapid development.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/brunch) | [View Demo](https:
- **React**: Create React App allows you to get going with React in no time.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/create-react-app) | [View De
- **Docusaurus (v1)**: Docusaurus makes it easy to maintain Open Source documentation websites.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/docusaurus) | [View Demo](ht
- **Docusaurus (v2+)**: Docusaurus makes it easy to maintain Open Source documentation websites.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/docusaurus-2) | [View Demo](
- **Dojo**: Dojo is a modern progressive, TypeScript first framework.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/dojo) | [View Demo](https://
- **Eleventy**: 11ty is a simpler static site generator written in JavaScript, created to be an alternative to Jekyll.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/eleventy) | [View Demo](http
- **Elysia**: Ergonomic framework for humans
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/elysia)
- **Ember.js**: Ember.js helps webapp developers be more productive out of the box.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/ember) | [View Demo](https:/
- **Express**: Fast, unopinionated, minimalist web framework for Node.js
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/express) | [View Demo](https
- **FastAPI**: FastAPI framework, high performance, easy to learn, fast to code, ready for production
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/fastapi) | [View Demo](https
- **FastHTML**: The fastest way to create an HTML app
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/fasthtml) | [View Demo](http
- **Fastify**: Fast and low overhead web framework, for Node.js
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/fastify)
- **Flask**: The Python micro web framework
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/flask)
- **Gatsby.js**: Gatsby helps developers build blazing fast websites and apps with React.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/gatsby) | [View Demo](https:
- **Gridsome**: Gridsome is a Vue.js-powered framework for building websites & apps that are fast by default.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/gridsome) | [View Demo](http
- **H3**: Universal, Tiny, and Fast Servers
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/h3)
- **Hexo**: Hexo is a fast, simple & powerful blog framework powered by Node.js.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/hexo) | [View Demo](https://
- **Hono**: Web framework built on Web Standards
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/hono) | [View Demo](https://
- **Hugo**: Hugo is the world's fastest framework for building websites, written in Go.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/hugo) | [View Demo](https://
- **Hydrogen (v1)**: React framework for headless commerce
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/hydrogen) | [View Demo](http
- **Ionic Angular**: Ionic Angular allows you to build mobile PWAs with Angular and the Ionic Framework.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/ionic-angular) | [View Demo]
- **Ionic React**: Ionic React allows you to build mobile PWAs with React and the Ionic Framework.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/ionic-react) | [View Demo](h

- **Jekyll**: Jekyll makes it super easy to transform your plain text into static websites and blogs.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/jekyll) | [View Demo](https:
- **Middleman**: Middleman is a static site generator that uses all the shortcuts and tools in modern web development.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/middleman) | [View Demo](htt
- **NestJS**: Framework for building efficient, scalable Node.js server-side applications
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/nestjs)
- **Next.js**: Next.js makes you productive with React instantly — whether you want to build static or dynamic sites.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/nextjs) | [View Demo](https:
- **Nitro**: Nitro is a next generation server toolkit.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/nitro) | [View Demo](https:/
- **Nuxt**: Nuxt is the open source framework that makes full-stack development with Vue.js intuitive.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/nuxtjs) | [View Demo](https:
- **Parcel**: Parcel is a zero configuration build tool for the web that scales to projects of any size and complexity.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/parcel) | [View Demo](https:
- **Polymer**: Polymer is an open-source webapps library from Google, for building using Web Components.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/polymer) | [View Demo](https
- **Preact**: Preact is a fast 3kB alternative to React with the same modern API.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/preact) | [View Demo](https:
- **React Router**: Declarative routing for React
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/react-router) | [View Demo](
- **RedwoodJS**: RedwoodJS is a full-stack framework for the Jamstack.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/redwoodjs) | [View Demo](htt
- **Remix**: Build Better Websites
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/remix) | [View Demo](https:/
- **Saber**: Saber is a framework for building static sites in Vue.js that supports data from any source.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/saber)
- **Sanity**: The structured content platform.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/sanity) | [View Demo](https:
- **Sanity (v3)**: The structured content platform.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/sanity-v3) | [View Demo](htt
- **Scully**: Scully is a static site generator for Angular.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/scully) | [View Demo](https:
- **SolidStart (v0)**: Simple and performant reactivity for building user interfaces.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/solidstart) | [View Demo](ht
- **SolidStart (v1)**: Simple and performant reactivity for building user interfaces.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/solidstart-1) | [View Demo](
- **Stencil**: Stencil is a powerful toolchain for building Progressive Web Apps and Design Systems.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/stencil) | [View Demo](https
- **Storybook**: Frontend workshop for UI development
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/storybook)
- **SvelteKit**: SvelteKit is a framework for building web applications of all sizes.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/sveltekit-1) | [View Demo](h
- **TanStack Start**: Full-stack Framework powered by TanStack Router for React and Solid.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/tanstack-start)
- **UmiJS**: UmiJS is an extensible enterprise-level React application framework.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/umijs) | [View Demo](https:/
- **Vite**: Vite is a new breed of frontend build tool that significantly improves the frontend development experience.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/vite) | [View Demo](https://
- **VitePress**: VitePress is VuePress' little brother, built on top of Vite.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/vitepress) | [View Demo](htt
- **Vue.js**: Vue.js is a versatile JavaScript framework that is as approachable as it is performant.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/vue) | [View Demo](https://v
- **VuePress**: Vue-powered Static Site Generator
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/vuepress) | [View Demo](http
- **xmcp**: The MCP framework for building AI-powered tools
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/xmcp) | [View Demo](https://
- **Zola**: Everything you need to make a static site engine in one binary.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/zola) | [View Demo](https://

However, if no framework is detected, "Other" will be selected. In this case, the Override toggle for the Build Command will be enabled b

If you would like to override Framework Preset for a **specific deployment**, add [`framework`](/docs/project-configuration#framework) to

Build Command

Vercel automatically configures the Build Command based on the framework. Depending on the framework, the Build Command can refer to the

For example, if [Next.js](https://nextjs.org) is your framework:

- Vercel checks for the `build` command in `scripts` and uses this to build the project
- If not, the `next build` will be triggered as the default Build Command

If you'd like to override the Build Command for **all deployments** in your Project, you can turn on the Override toggle and specify the

If you would like to override the Build Command for a **specific deployment**, add [`buildCommand`](/docs/project-configuration#buildcomm

> **💡 Note:** If you update the setting, it will be applied on your next
> deployment.

Output Directory

After building a project, most frameworks output the resulting build in a directory. Only the contents of this **Output Directory** will

If Vercel detects a framework, the output directory will automatically be configured.

> **💡 Note:** If you update the setting, it will be applied on your next
> deployment.

For projects that [do not require building](#skip-build-step), you might want to serve the files in the root directory. In this case, do

- Choose "Other" as the Framework Preset. This sets the output directory as `public` if it exists or `.` (root directory of the project)
- If your project doesn't have a `public` directory, it will serve the files from the root directory
- Alternatively, you can turn on the **Override** toggle and leave the field empty (in which case, the build step will be skipped)

If you would like to override the Output Directory for a **specific deployment**, add [`outputDirectory`](/docs/project-configuration#out

Install Command

Vercel auto-detects the install command during the build step. It installs dependencies from `package.json`, including `devDependencies`

The install command can be managed in two ways: through a project override, or per-deployment. See [manually specifying a package manager

To learn what package managers are supported on Vercel, see the [package manager support](/docs/package-managers) documentation.

Corepack

> **⚠️ Warning:** Corepack is considered
> [experimental](https://nodejs.org/docs/latest-v16.x/api/documentation.html#stability-index)
> and therefore, breaking changes or removal may occur in any future release of
> Node.js.

[Corepack](https://nodejs.org/docs/latest-v16.x/api/corepack.html) is an experimental tool that allows a Node.js project to pin a specifi

You can enable Corepack by adding an [environment variable](/docs/environment-variables) with name `ENABLE_EXPERIMENTAL_COREPACK` and val

Then, set the [`packageManager`](https://nodejs.org/docs/latest-v16.x/api/packages.html#packagemanager) property in the `package.json` fi

```json filename="package.json"
{
  "packageManager": "pnpm@7.5.1"
}
```

Custom Install Command for your API

The Install Command defined in the Project Settings will be used for front-end frameworks that support Vercel functions for APIs.

If you're using [Vercel functions](/docs/functions) defined in the natively supported `api` directory, a different Install Command will b

Development Command

This setting is relevant only if you're using `vercel dev` locally to develop your project. Use `vercel dev` only if you need to use Verc

The Development Command settings allow you to customize the behavior of `vercel dev`. If Vercel detects a framework, the development comm

If you'd like to use a custom command for , you can turn on the **Override** toggle. Please note the following:

- If you specify a custom command, your command must pass your framework's `$PORT` variable (which contains the port number). For example
- If the development command is not specified, `vercel dev` will fail. If you've selected "Other" as the framework preset, the default de
- You must create a deployment and have your local project linked to the project on Vercel (using `vercel`). Otherwise, `vercel dev` will

If you would like to override the Development Command, add [`devCommand`](/docs/project-configuration#devcommand) to your `vercel.json` c

Skip Build Step

Some static projects do not require building. For example, a website with only HTML/CSS/JS source files can be served as-is.

In such cases, you should:

- Specify "Other" as the framework preset
- Enable the **Override** option for the Build Command
- Leave the Build Command empty

This prevents running the build, and your content is served directly.

Root Directory

In some projects, the top-level directory of the repository may not be the root directory of the app you'd like to build. For example, yo

For such cases, you can specify the project Root Directory. If you do so, please note the following:

- Your app will not be able to access files outside of that directory. You also cannot use `..` to move up a level
- This setting also applies to [Vercel CLI](/docs/cli). Instead of running `vercel <directory-name>` to deploy, specify `<directory-name>

To configure the Root Directory:

1. Navigate to the **Build and Deployment** page of your **Project Settings**
2. Scroll down to **Root Directory**
3. Enter the path to the root directory of your app
4. Click **Save** to apply the changes

> **💡 Note:** If you update the root directory setting, it will be applied on your next
> deployment.

Skipping unaffected projects

In a monorepo, you can [skip deployments](/docs/monorepos#skipping-unaffected-projects) for projects that were not affected by a commit.

1. Navigate to the **Build and Deployment** page of your **Project Settings**
2. Scroll down to **Root Directory**
3. Enable the **Skip deployment** switch

---
title: "Managing Builds"
description: "Vercel allows you to increase the speed of your builds when needed in specific situations and workflows."
last_updated: "2026-01-16T02:19:26.434Z"
source: "https://vercel.com/docs/builds/managing-builds"

Managing Builds

When you build your application code, Vercel runs compute to install dependencies, run your build script, and upload the build output to

- If you need faster build machines or more memory, you can purchase [Enhanced or Turbo build machines](#larger-build-machines).
- If you are deploying frequently and seeing [build queues](/docs/builds/build-queues), you can enable [On-Demand Concurrent Builds](#on-

[Visit Build Diagnostics in the Observability tab of the Vercel Dashboard](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Fobserv

Your situation	Solution	Best for
Builds are slow or running out of resources	[Enhanced/Turbo build machines](#larger-build-machines)	Large apps, com
Builds are frequently queued	[On-demand Concurrent Builds](#on-demand-concurrent-builds)	Teams with freq
Specific projects are frequently queued	[Project-level on-demand](#project-level-on-demand-concurrent-builds)	Fast-moving pro
Occasional urgent deploy stuck in queue	[Force an on-demand build](#force-an-on-demand-build)	Ad-hoc critical
Production builds stuck behind preview builds	[Prioritize production builds](#prioritize-production-builds)	All production-

Larger build machines

For Pro and Enterprise customers, we offer two higher-tier build machines with more vCPUs, memory and disk space than Standard.

Build machine type	Number of vCPUs	Memory (GB)	Disk size (GB)
Standard	4	8	23
Enhanced	8	16	56
Turbo	30	60	64

You can set the build machine type in [the **Build and Deployment** section of your **Project Settings**](https://vercel.com/d?to=%2F%5Bt

When your team uses Enhanced or Turbo machines, it'll contribute to the "Build Minutes" item of your bill.

Enterprise customers who have Enhanced build machines enabled via contract will always use them by default. You can view if you have this

On-demand concurrent builds

On-demand concurrent builds allow your builds to skip the queue and run immediately. By default, projects have on-demand concurrent build

You are charged for on-demand concurrent builds based on the number of concurrent builds required to allow the builds to proceed as expla

Project-level on-demand concurrent builds

When you enable on-demand build concurrency at the level of a project, any queued builds in that project will automatically be allowed to

You can configure this on the project's [**Build and Deployment Settings**](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Fsetti

\['Dashboard'

1. From your Vercel dashboard, select the project you wish to enable it for.
2. Select the **Settings** tab, and go to the **Build and Deployment** section of your [Project Settings](/docs/projects/overview#project
3. Under **On-Demand Concurrent Builds**, select one of the following:
   - **Run all builds immediately**: Skip the queue for all builds
   - **Run up to one build per branch**: Limit to one active build per branch
4. The standard option is selected by default with 4 vCPUs and 8 GB of memory. You can switch to [Enhanced or Turbo build machines](#larg
5. Click **Save**.

'cURL'

To create an Authorization Bearer token, see the [access token](/docs/rest-api/reference/welcome#creating-an-access-token) section of the

```bash filename="cURL"
curl --request PATCH \
  --url https://api.vercel.com/v9/projects/YOUR_PROJECT_ID?teamId=YOUR_TEAM_ID \
  --header "Authorization: Bearer $VERCEL_TOKEN" \
  --header "Content-Type: application/json" \
  --data '{
    "resourceConfig": {
      "elasticConcurrencyEnabled": true,
      "buildQueue": {
        "configuration": "SKIP_NAMESPACE_QUEUE"
      }
    }
  }'
```

Set `configuration` to one of:

- `SKIP_NAMESPACE_QUEUE`: Run all builds immediately
- `WAIT_FOR_NAMESPACE_QUEUE`: Limit to one active build per branch

'SDK']

To create an Authorization Bearer token, see the [access token](/docs/rest-api/reference/welcome#creating-an-access-token) section of the

```ts filename="updateProject"
import { Vercel } from '@vercel/sdk';

const vercel = new Vercel({
  bearerToken: '<YOUR_BEARER_TOKEN_HERE>',
});

async function run() {
  const result = await vercel.projects.updateProject({
    idOrName: 'YOUR_PROJECT_ID',
    teamId: 'YOUR_TEAM_ID',
    requestBody: {
      resourceConfig: {
        elasticConcurrencyEnabled: true,
        buildQueue: {
          configuration: 'SKIP_NAMESPACE_QUEUE',
        },
      },
    },
  });

  console.log(result);
}

run();
```

Set `configuration` to one of:

- `SKIP_NAMESPACE_QUEUE`: Run all builds immediately
- `WAIT_FOR_NAMESPACE_QUEUE`: Limit to one active build per branch

Force an on-demand build

For individual deployments, you can force build execution using the **Start Building Now** button. Regardless of the reason why this buil

1. Select your project from the [dashboard](/dashboard).

2. From the top navigation, select the **Deployments** tab.

3. Find the queued deployment that you would like to build from the list. You can use the **Status** filter to help find it. You have 2 o
   - Select the three dots to the right of the deployment and select **Start Building Now**.
   - Click on the deployment list item to go to the deployment's detail page and click **Start Building Now**.

4. **Confirm** that you would like to build this deployment in the **Start Building Now** dialog.

Optimizing builds

Some other considerations to take into account when optimizing your builds include:

- [Understand](/docs/deployments/troubleshoot-a-build#understanding-build-cache) and [manage](/docs/deployments/troubleshoot-a-build#mana
- You may choose to [Ignore the Build Step](/docs/project-configuration/git-settings#ignored-build-step) on redeployments if you know tha
- Use the most recent version of your runtime, particularly Node.js, to take advantage of the latest performance improvements. To learn m

Prioritize production builds

If a build has to wait for queued preview deployments to finish, it can delay the production release process. When Vercel queues builds,

> **💡 Note:** For any new projects created after December 12, 2024, Vercel will prioritize
> production builds by default.

To ensure that changes to the [production environment](/docs/deployments/environments#production-environment) are prioritized over [previ

1. From your Vercel dashboard, select the project you wish to enable it for
2. Select the **Settings** tab, and go to the [**Build and Deployment** section](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2F
3. Under **Prioritize Production Builds**, toggle the switch to **Enabled**

Usage and limits

The on-demand build usage is based on the amount of time it took for a deployment to build when using a concurrent build. In Billing, usa

Pro plan

Builds are priced in $ per minute of build time and are based on the type of build machines used. There is no charge for using the Standa

Build machine type	Price per build minute
Standard (billed **only** when On-Demand Concurrent Builds is enabled)	$0.014
Enhanced (always billed)	$0.030
Turbo (always billed)	$0.113

Enterprise plan

On-demand concurrent builds are priced in [MIUs](/docs/pricing/understanding-my-invoice#managed-infrastructure-units-miu) per minute of b

Concurrent builds contracted	Cost ([MIU](/docs/pricing/understanding-my-invoice#managed-infrastructure-units-miu) per minute) for Sta
1-5	0.014 MIUs
6-10	0.012 MIUs
10+	0.010 MIUs

---
title: "Builds"
description: "Understand how the build step works when creating a Vercel Deployment."
last_updated: "2026-01-16T02:19:26.458Z"
source: "https://vercel.com/docs/builds"

Builds

Vercel automatically performs a **build** every time you deploy your code, whether you're pushing to a Git repository, importing a projec

Build infrastructure

When you initiate a build, Vercel creates a secure, isolated virtual environment for your project:

- Your code is built in a clean, consistent environment
- Build processes can't interfere with other users' applications
- Security is maintained through complete isolation
- Resources are efficiently allocated and cleaned up after use

This infrastructure handles millions of builds daily, supporting everything from individual developers to large enterprises, while mainta

Most frontend frameworks—like Next.js, SvelteKit, and Nuxt—are **auto-detected**, with defaults applied for Build Command, Output Directo

How builds are triggered

Builds can be initiated in the following ways:

1. **Push to Git**: When you connect a GitHub, GitLab, or Bitbucket repository, each commit to a tracked branch initiates a new build and

2. **Vercel CLI**: Running `vercel` locally deploys your project. By default, this creates a preview build unless you add the `--prod` fl

3. **Dashboard deploy**: Clicking **Deploy** in the dashboard or creating a new project also triggers a build.

Build customization

Depending on your framework, Vercel automatically sets the **Build Command**, **Install Command**, and **Output Directory**. If needed, y

1. **Build Command**: Override the default (`npm run build`, `next build`, etc.) for custom workflows.

2. **Output Directory**: Specify the folder containing your final build output (e.g., `dist` or `build`).

3. **Install Command**: Control how dependencies are installed (e.g., `pnpm install`, `yarn install`) or skip installing dev dependencies

To learn more, see [Configuring a Build](/docs/deployments/configure-a-build).

Skipping the build step

For static websites—HTML, CSS, and client-side JavaScript only—no build step is required. In those cases:

1. Set **Framework Preset** to **Other**.
2. Leave the build command blank.
3. (Optionally) override the **Output Directory** if you want to serve a folder other than `public` or `.`.

Monorepos

When working in a **monorepo**, you can connect multiple Vercel projects within the same repository. By default, each project will build

1. **Skipping unaffected projects**: Vercel automatically detects whether a project's files (or its dependencies) have changed and skips

2. **Ignored build step**: You can also write a script that cancels the build for a project if no relevant changes are detected. This app

For monorepo-specific build tools, see:

- [Turborepo](/docs/monorepos/turborepo)
- [Nx](/docs/monorepos/nx)

Concurrency and queues

When multiple builds are requested, Vercel manages concurrency and queues for you:

1. **Concurrency Slots**: Each plan has a limit on how many builds can run at once. If all slots are busy, new builds wait until a slot i

2. **Branch-Based Queue**: If new commits land on the same branch, Vercel skips older queued builds and prioritizes only the most recent

3. **On-Demand Concurrency**: If you need more concurrent build slots or want certain production builds to jump the queue, consider enabl

Environment variables

Vercel can automatically inject **environment variables** such as API keys, database connections, or feature flags during the build:

1. **Project-Level Variables**: Define variables under **Settings** for each environment (Preview, Production, or any custom environment)

2. **Pull Locally**: Use `vercel env pull` to download environment variables for local development. This command populates your `.env.loc

3. **Security**: Environment variables remain private within the build environment and are never exposed in logs.

Ignored files and folders

Some files (e.g., large datasets or personal configuration) might not be needed in your deployment:

- Vercel automatically ignores certain files (like `.git`) for performance and security.
- You can read more about how to specify [ignored files and folders](/docs/builds/build-features#ignored-files-and-folders).

Build output and deployment

Once the build completes successfully:

1. Vercel uploads your build artifacts (static files, Vercel Functions, and other assets) to the CDN.
2. A unique deployment URL is generated for **Preview** or updated for **Production** domains.
3. Logs and build details are available in the **Deployments** section of the dashboard.

If the build fails or times out, Vercel provides diagnostic logs in the dashboard to help you troubleshoot. For common solutions, see our

Build infrastructure

Behind the scenes, Vercel manages a sophisticated global infrastructure that:

- Creates isolated build environments on-demand
- Handles automatic regional failover
- Manages hardware resources efficiently
- Pre-warms containers to improve build start times
- Synchronizes OS and runtime environments with your deployment targets

Limits and resources

Vercel enforces certain limits to ensure reliable builds for all users:

- **Build timeout**: The maximum build time is **45 minutes**. If your build exceeds this limit, it will be terminated, and the deploymen
- **Build cache**: Each build cache can be up to **1 GB**. The [cache](/docs/deployments/troubleshoot-a-build#caching-process) is retaine
- **Container resources**: Vercel creates a [build container](/docs/builds/build-image) with different resources depending on your plan:

	Hobby	Pro	Enterprise
Memory	8192 MB	8192 MB	Custom
Disk Space	23 GB	23 GB	Custom
CPUs	2	4	Custom

For more information, visit [Build Container Resources](/docs/deployments/troubleshoot-a-build#build-container-resources) and [Cancelled

Learn more about builds

To explore more features and best practices for building and deploying with Vercel:

- [Configure your build](/docs/builds/configure-a-build): Customize commands, output directories, environment variables, and more.
- [Troubleshoot builds](/docs/deployments/troubleshoot-a-build): Get help with build cache, resource limits, and common errors.
- [Manage builds](/docs/builds/managing-builds): Control how many builds run in parallel and prioritize critical deployments.
- [Working with Monorepos](/docs/monorepos): Set up multiple projects in a single repository and streamline deployments.

Pricing

---
title: "Vercel CDN overview"
description: "Vercel"
last_updated: "2026-01-16T02:19:26.478Z"
source: "https://vercel.com/docs/cdn"

Vercel CDN overview

Vercel's CDN is a globally distributed platform that stores content near your customers and runs compute in [regions](/docs/regions) clos

If you're deploying an app on Vercel, you already use our CDN. These docs will teach you how to optimize your apps and deployment configu

Global network architecture

Vercel's CDN is built on a robust global infrastructure designed for optimal performance and reliability:

- **Points of Presence (PoPs)**: Our network includes 126 PoPs distributed worldwide. These PoPs act as the first point of contact for in
- **Vercel Regions**: Behind these PoPs, we maintain [19 compute-capable regions](/docs/regions) where your code runs close to your data.
- **Private Network**: Traffic flows through private, low-latency connections from PoPs to the nearest region, ensuring fast and efficien

This architecture balances the widespread geographical distribution benefits with the efficiency of concentrated caching and computing re

Features

- [**Redirects**](/docs/redirects): Redirects tell the client to make a new request to a different URL. They are useful for enforcing HTT
- [**Rewrites**](/docs/rewrites): Rewrites change the URL the server uses to fetch the requested resource internally, allowing for dynami
- [**Headers**](/docs/headers): Headers can modify the request and response headers, improving security, performance, and functionality.
- [**Caching**](/docs/cdn-cache): Caching stores responses in the CDN, reducing latency and improving performance
- [**Streaming**](/docs/functions/streaming-functions): Streaming enhances your user's perception of your app's speed and performance.
- [**HTTPS / SSL**](/docs/encryption): Vercel serves every deployment over an HTTPS connection by automatically provisioning SSL certific
- [**Compression**](/docs/compression): Compression reduces data transfer and improves performance, supporting both gzip and brotli compr

Pricing

Vercel's CDN pricing is divided into three resources:

- **Fast Data Transfer**: Data transfer between the Vercel CDN and the user's device.
- **Fast Origin Transfer**: Data transfer between the CDN and Vercel Functions.
- **Edge Requests**: Requests made to the CDN.

All resources are billed based on usage with each plan having an [included allotment](/docs/pricing). Those on the Pro plan are billed ac

The pricing for each resource is based on the region from which requests to your site come. Use the dropdown to select your preferred reg

Usage

The table below shows the metrics for the [**Networking**](/docs/pricing/networking) section of the **Usage** dashboard.

To view information on managing each resource, select the resource link in the **Metric** column. To jump straight to guidance on optimiz

See the [manage and optimize networking usage](/docs/pricing/networking) section for more information on how to optimize your usage.

Supported protocols

The CDN supports the following protocols (negotiated with [ALPN](https://tools.ietf.org/html/rfc7301)):

- [HTTPS](https://en.wikipedia.org/wiki/HTTPS)
- [HTTP/1.1](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)
- [HTTP/2](https://en.wikipedia.org/wiki/HTTP/2)

Using Vercel's CDN locally

Vercel supports 35 [frontend frameworks](/docs/frameworks). These frameworks provide a local development environment used to test your ap

Through [framework-defined infrastructure](https://vercel.com/blog/framework-defined-infrastructure), Vercel then transforms your framewo

If you are using [Vercel Functions](/docs/functions) or other compute on Vercel *without* a framework, you can use the [Vercel CLI](/docs

Using Vercel's CDN with other CDNs

While sometimes necessary, proceed with caution when you place another CDN in front of Vercel:

- Vercel's CDN is designed to deploy new releases of your site without downtime by purging the [CDN Cache](/docs/cdn-cache) globally and
- If you use an additional CDN in front of Vercel, it can cause issues because Vercel has no control over the other provider, leading to
- To avoid these problems while still using another CDN, we recommend you either configure a short cache time or disable the cache entire

---
title: "Vercel CDN Cache"
description: "Vercel"
last_updated: "2026-01-16T02:19:26.590Z"
source: "https://vercel.com/docs/cdn-cache"

Vercel CDN Cache

Vercel's CDN caches your content (including pages, API responses, and static assets) in data centers around the world, closer to your use

CDN caching is available for all deployments and domains on your account, regardless of the [pricing plan](https://vercel.com/pricing).

There are two ways to cache content:

- [Static file caching](#static-files-caching) is automatic for all deployments, requiring no manual configuration
- To cache dynamic content, including SSR content, you can use `Cache-Control` [headers](/docs/headers#cache-control-header). Review [How

To learn about cache keys, manually purging the cache, and the differences between invalidate and delete methods, see [Purging Vercel CDN

How to cache responses

You can cache responses on Vercel with `Cache-Control` headers defined in:

1. Responses from [Vercel Functions](/docs/functions)
2. Route definitions in `vercel.json` or `next.config.js`

You can use any combination of the above options, but if you return `Cache-Control` headers in a Vercel Function, it will override the he

Using Vercel Functions

To cache the response of Functions on Vercel's CDN, you must include [`Cache-Control`](/docs/headers#cache-control-header) headers with *

- `s-maxage=N`
- `s-maxage=N, stale-while-revalidate=Z`

> **💡 Note:** `proxy-revalidate` and `stale-if-error` are not currently supported.

The following example demonstrates a [function](/docs/functions) that caches its response and revalidates it every 1 second:

```ts filename="app/api/cache-control-example/route.ts" framework=nextjs-app
export async function GET() {
  return new Response('Cache Control example', {
    status: 200,
    headers: {
      'Cache-Control': 'public, s-maxage=1',
      'CDN-Cache-Control': 'public, s-maxage=60',
      'Vercel-CDN-Cache-Control': 'public, s-maxage=3600',
    },
  });
}
```

```js filename="app/api/cache-control-example/route.js" framework=nextjs-app
export async function GET() {
  return new Response('Cache Control example', {
    status: 200,
    headers: {
      'Cache-Control': 'public, s-maxage=1',
      'CDN-Cache-Control': 'public, s-maxage=60',
      'Vercel-CDN-Cache-Control': 'public, s-maxage=3600',
    },
  });
}
```

```ts filename="pages/api/cache-control-example.ts" framework=nextjs
import type { NextApiRequest, NextApiResponse } from 'next';

export default function handler(
  request: NextApiRequest,
  response: NextApiResponse,
) {
  response.setHeader('Cache-Control', 'public, s-maxage=1');

  return response.status(200).json({ name: 'John Doe' });
}
```

```js filename="pages/api/cache-control-example.js" framework=nextjs
export default function handler(request, response) {
  response.setHeader('Cache-Control', 'public, s-maxage=1');

  return response.status(200).json({ name: 'John Doe' });
}
```

```ts filename="api/cache-control-example.ts" framework=other
import type { VercelResponse } from '@vercel/node';

export default function handler(response: VercelResponse) {
  response.setHeader('Cache-Control', 'public, s-maxage=1');

  return response.status(200).json({ name: 'John Doe' });
}
```

```js filename="api/cache-control-example.js" framework=other
export default function handler(response) {
  response.setHeader('Cache-Control', 'public, s-maxage=1');

  return response.status(200).json({ name: 'John Doe' });
}
```

For direct control over caching on Vercel and downstream CDNs, you can use [CDN-Cache-Control](#cdn-cache-control) headers.

Using `vercel.json` and `next.config.js`

You can define route headers in `vercel.json` or `next.config.js` files. These headers will be overridden by [headers defined in Function

The following example demonstrates a `vercel.json` file that adds `Cache-Control` headers to a route:

```json filename="vercel.json"
```

```
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "headers": [
    {
      "source": "/about.js",
      "headers": [
        {
          "key": "Cache-Control",
          "value": "s-maxage=1, stale-while-revalidate=59"
        }
      ]
    }
  ]
}
```

If you're building your app with Next.js, you should use `next.config.js` rather than `vercel.json`. The following example demonstrates a

```js filename="next.config.js"
/** @type {import('next').NextConfig} */
const nextConfig = {
  reactStrictMode: true,
  async headers() {
    return [
      {
        source: '/about',
        headers: [
          {
            key: 'Cache-Control',
            value: 's-maxage=1, stale-while-revalidate=59',
          },
        ],
      },
    ];
  },
};

module.exports = nextConfig;
```

See to learn more about `next.config.js`.

Static Files Caching

Static files are **automatically cached on Vercel's global network** for the lifetime of the deployment after the first request.

- If a static file is unchanged, the cached value can persist across deployments due to the hash used in the filename
- Optimized images cached will persist across deployments for both [static images](/docs/image-optimization#local-images-cache-key) and [

Browser

- `max-age=N, public`
- `max-age=N, immutable`

Where `N` is the number of seconds the response should be cached. The response must also meet the [caching criteria](/docs/cdn-cache#how-

Cache control options

You can cache dynamic content through [Vercel Functions](/docs/functions), including SSR, by adding `Cache-Control` [headers](/docs/heade

See [our docs on Cache-Control headers](/docs/headers#cache-control-header) to learn how to best use `Cache-Control` directives on Vercel

CDN-Cache-Control

Vercel supports two [Targeted Cache-Control headers](https://httpwg.org/specs/rfc9213.html "targeted headers for controlling the cache"):

- `CDN-Cache-Control`, which allows you to control the Vercel CDN Cache or other CDN cache *separately* from the browser's cache. The bro
- `Vercel-CDN-Cache-Control`, which allows you to specifically control Vercel's Cache. Neither other CDNs nor the browser will be affecte

By default, the headers returned to the browser are as follows:

- `Cache-Control`
- `CDN-Cache-Control`

`Vercel-CDN-Cache-Control` headers are not returned to the browser or forwarded to other CDNs.

To learn how these headers work in detail, see [our dedicated headers docs](/docs/headers/cache-control-headers#cdn-cache-control-header)

The following example demonstrates `Cache-Control` headers that instruct:

- Vercel's Cache to have a [TTL](https://en.wikipedia.org/wiki/Time_to_live "TTL – Time To Live") of `3600` seconds
- Downstream CDNs to have a TTL of `60` seconds
- Clients to have a TTL of `10` seconds

```js filename="app/api/cache-control-headers/route.js" framework=nextjs
export async function GET() {
  return new Response('Cache Control example', {
    status: 200,
    headers: {
      'Cache-Control': 'max-age=10',
      'CDN-Cache-Control': 'max-age=60',
      'Vercel-CDN-Cache-Control': 'max-age=3600',
    },
  });
}
```

```ts filename="app/api/cache-control-headers/route.ts" framework=nextjs
export async function GET() {
  return new Response('Cache Control example', {
```

```
    status: 200,
    headers: {
      'Cache-Control': 'max-age=10',
      'CDN-Cache-Control': 'max-age=60',
      'Vercel-CDN-Cache-Control': 'max-age=3600',
    },
  });
}
```

```js filename="app/api/cache-control-headers/route.js" framework=nextjs-app
export async function GET() {
  return new Response('Cache Control example', {
    status: 200,
    headers: {
      'Cache-Control': 'max-age=10',
      'CDN-Cache-Control': 'max-age=60',
      'Vercel-CDN-Cache-Control': 'max-age=3600',
    },
  });
}
```

```ts filename="app/api/cache-control-headers/route.ts" framework=nextjs-app
export async function GET() {
  return new Response('Cache Control example', {
    status: 200,
    headers: {
      'Cache-Control': 'max-age=10',
      'CDN-Cache-Control': 'max-age=60',
      'Vercel-CDN-Cache-Control': 'max-age=3600',
    },
  });
}
```

```js filename="api/cache-control-headers.js" framework=other
export default function handler(request, response) {
  response.setHeader('Vercel-CDN-Cache-Control', 'max-age=3600');
  response.setHeader('CDN-Cache-Control', 'max-age=60');
  response.setHeader('Cache-Control', 'max-age=10');

  return response.status(200).json({ name: 'John Doe' });
}
```

```ts filename="api/cache-control-headers.ts" framework=other
import type { VercelResponse } from '@vercel/node';

export default function handler(response: VercelResponse) {
  response.setHeader('Vercel-CDN-Cache-Control', 'max-age=3600');
  response.setHeader('CDN-Cache-Control', 'max-age=60');
  response.setHeader('Cache-Control', 'max-age=10');

  return response.status(200).json({ name: 'John Doe' });
}
```

If you set `Cache-Control` without a `CDN-Cache-Control`, the Vercel CDN strips `s-maxage` and `stale-while-revalidate` from the response

Vary header

The `Vary` response header instructs caches to use specific request headers as part of the cache key. This allows you to serve different

> **💡 Note:** The `Vary` header only has an effect when used in combination with
> `Cache-Control` headers that enable caching (such as `s-maxage`). Without a
> caching directive, the `Vary` header has no behavior.

When Vercel's CDN receives a request, it combines the cache key (described in the [Cache Invalidation](#cache-invalidation) section) with

Use cases

> **💡 Note:** Vercel's CDN already includes the `Accept` and `Accept-Encoding` headers as
> part of the cache key by default. You do not need to explicitly include these
> headers in your `Vary` header.

The most common use case for the `Vary` header is content negotiation, serving different content based on:

- User location (e.g., `X-Vercel-IP-Country`)
- Device type (e.g., `User-Agent`)
- Language preferences (e.g., `Accept-Language`)

Example: Country-specific content

You can use the `Vary` header with Vercel's `X-Vercel-IP-Country` request header to cache different responses for users from different co

```tsx filename="app/api/country-specific/route.ts" framework=nextjs-app
import { type NextRequest } from 'next/server';

export async function GET(request: NextRequest) {
  const country = request.headers.get('x-vercel-ip-country') || 'unknown';

  // Serve different content based on country
  let content;
  if (country === 'US') {
    content = { message: 'Hello from the United States!' };
  } else if (country === 'GB') {
    content = { message: 'Hello from the United Kingdom!' };
  } else {
    content = { message: `Hello from ${country}!` };
```

```
  }

  return Response.json(content, {
    status: 200,
    headers: {
      'Cache-Control': 's-maxage=3600',
      Vary: 'X-Vercel-IP-Country',
    },
  });
}
```

```jsx filename="app/api/country-specific/route.js" framework=nextjs-app
export async function GET(request) {
  const country = request.headers.get('x-vercel-ip-country') || 'unknown';

  // Serve different content based on country
  let content;
  if (country === 'US') {
    content = { message: 'Hello from the United States!' };
  } else if (country === 'GB') {
    content = { message: 'Hello from the United Kingdom!' };
  } else {
    content = { message: `Hello from ${country}!` };
  }

  return Response.json(content, {
    status: 200,
    headers: {
      'Cache-Control': 's-maxage=3600',
      Vary: 'X-Vercel-IP-Country',
    },
  });
}
```

```tsx filename="pages/api/country-specific.ts" framework=nextjs
import type { NextApiRequest, NextApiResponse } from 'next';

export default function handler(req: NextApiRequest, res: NextApiResponse) {
  const country = req.headers['x-vercel-ip-country'] || 'unknown';

  // Serve different content based on country
  let content;
  if (country === 'US') {
    content = { message: 'Hello from the United States!' };
  } else if (country === 'GB') {
    content = { message: 'Hello from the United Kingdom!' };
  } else {
    content = { message: `Hello from ${country}!` };
  }

  // Set caching headers
  res.setHeader('Cache-Control', 's-maxage=3600');
  res.setHeader('Vary', 'X-Vercel-IP-Country');

  res.status(200).json(content);
}
```

```jsx filename="pages/api/country-specific.js" framework=nextjs
export default function handler(req, res) {
  const country = req.headers['x-vercel-ip-country'] || 'unknown';

  // Serve different content based on country
  let content;
  if (country === 'US') {
    content = { message: 'Hello from the United States!' };
  } else if (country === 'GB') {
    content = { message: 'Hello from the United Kingdom!' };
  } else {
    content = { message: `Hello from ${country}!` };
  }

  // Set caching headers
  res.setHeader('Cache-Control', 's-maxage=3600');
  res.setHeader('Vary', 'X-Vercel-IP-Country');

  res.status(200).json(content);
}
```

```tsx filename="api/country-specific.ts" framework=other
export default {
  fetch(request) {
    const country = request.headers.get('x-vercel-ip-country') || 'unknown';

    // Serve different content based on country
    let content;
    if (country === 'US') {
      content = { message: 'Hello from the United States!' };
    } else if (country === 'GB') {
      content = { message: 'Hello from the United Kingdom!' };
    } else {
      content = { message: `Hello from ${country}!` };
    }

    return Response.json(content, {
      status: 200,
      headers: {
```

```
        'Cache-Control': 's-maxage=3600',
        Vary: 'X-Vercel-IP-Country',
      },
    });
  },
};
```

```jsx filename="api/country-specific.js" framework=other
export default {
  fetch(request) {
    const country = request.headers.get('x-vercel-ip-country') || 'unknown';

    // Serve different content based on country
    let content;
    if (country === 'US') {
      content = { message: 'Hello from the United States!' };
    } else if (country === 'GB') {
      content = { message: 'Hello from the United Kingdom!' };
    } else {
      content = { message: `Hello from ${country}!` };
    }

    return Response.json(content, {
      status: 200,
      headers: {
        'Cache-Control': 's-maxage=3600',
        Vary: 'X-Vercel-IP-Country',
      },
    });
  },
};
```

Setting the `Vary` header

You can set the `Vary` header in the same ways you set other response headers:

In Vercel Functions

```tsx filename="app/api/data/route.ts" framework=nextjs-app
import { type NextRequest } from 'next/server';

export async function GET(request: NextRequest) {
  return Response.json(
    { data: 'This response varies by country' },
    {
      status: 200,
      headers: {
        Vary: 'X-Vercel-IP-Country',
        'Cache-Control': 's-maxage=3600',
      },
    },
  );
}
```

```jsx filename="app/api/data/route.js" framework=nextjs-app
export async function GET(request) {
  return Response.json(
    { data: 'This response varies by country' },
    {
      status: 200,
      headers: {
        Vary: 'X-Vercel-IP-Country',
        'Cache-Control': 's-maxage=3600',
      },
    },
  );
}
```

```tsx filename="pages/api/data.ts" framework=nextjs
import type { NextApiRequest, NextApiResponse } from 'next';

export default function handler(req: NextApiRequest, res: NextApiResponse) {
  res.setHeader('Vary', 'X-Vercel-IP-Country');
  res.setHeader('Cache-Control', 's-maxage=3600');
  res.status(200).json({ data: 'This response varies by country' });
}
```

```jsx filename="pages/api/data.js" framework=nextjs
export default function handler(req, res) {
  res.setHeader('Vary', 'X-Vercel-IP-Country');
  res.setHeader('Cache-Control', 's-maxage=3600');
  res.status(200).json({ data: 'This response varies by country' });
}
```

```tsx filename="api/data.ts" framework=other
export default {
  fetch(request) {
    return Response.json(
      { data: 'This response varies by country' },
      {
        status: 200,
        headers: {
          Vary: 'X-Vercel-IP-Country',
          'Cache-Control': 's-maxage=3600',
```

```
        },
      },
    );
  },
};
```

```jsx filename="api/data.js" framework=other
export default {
  fetch(request) {
    return Response.json(
      { data: 'This response varies by country' },
      {
        status: 200,
        headers: {
          Vary: 'X-Vercel-IP-Country',
          'Cache-Control': 's-maxage=3600',
        },
      },
    );
  },
};
```

Using `vercel.json`

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "headers": [
    {
      "source": "/api/data",
      "headers": [
        {
          "key": "Vary",
          "value": "X-Vercel-IP-Country"
        },
        {
          "key": "Cache-Control",
          "value": "s-maxage=3600"
        }
      ]
    }
  ]
}
```

Using `next.config.js`

If you're building your app with Next.js, use `next.config.js`:

```js filename="next.config.js"
/** @type {import('next').NextConfig} */
const nextConfig = {
  async headers() {
    return [
      {
        source: '/api/data',
        headers: [
          {
            key: 'Vary',
            value: 'X-Vercel-IP-Country',
          },
          {
            key: 'Cache-Control',
            value: 's-maxage=3600',
          },
        ],
      },
    ];
  },
};

module.exports = nextConfig;
```

Multiple `Vary` headers

You can specify multiple headers in a single `Vary` value by separating them with commas:

```js
res.setHeader('Vary', 'X-Vercel-IP-Country, Accept-Language');
```

This will create separate cache entries for each unique combination of country and language preference.

Best practices

- Use `Vary` headers selectively, as each additional header exponentially increases the number of cache entries — this doesn't directly i
- Only include headers that meaningfully impact content generation
- Consider combining multiple variations into a single header value when possible

Cacheable response criteria

The `Cache-Control` field is an HTTP header specifying caching rules for client (browser) requests and server responses. A cache must obe

For server responses to be successfully cached with Vercel's CDN, the following criteria must be met:

- Request uses `GET` or `HEAD` method.
- Request does not contain `Range` header.

- Request does not contain `Authorization` header.
- Response uses `200`, `404`, `410`, `301`, `302`, `307` or `308` status code.
- Response does not exceed `10MB` in content length.
- Response does not contain the `set-cookie` header.
- Response does not contain the `private`, `no-cache` or `no-store` directives in the `Cache-Control` header.
- Response does not contain `Vary: *` header, which is treated as equivalent to `Cache-Control: private`.

Vercel **does not allow bypassing the cache for static files** by design.

Cache invalidation

To learn about cache keys, manually purging the cache, and the differences between invalidate and delete methods, see [Purging Vercel CDN

`x-vercel-cache`

The `x-vercel-cache` header is included in HTTP responses to the client, and describes the state of the cache.

See [our headers docs](/docs/headers/response-headers#x-vercel-cache) to learn more.

Limits

Vercel's CDN Cache is segmented [by region](/docs/regions). The following caching limits apply to [Vercel Function](/docs/functions) resp

- Max cacheable response size:
  - Streaming functions: **20MB**
  - Non-streaming functions: **10MB**
- Max cache time: **1 year**
  - `s-maxage`
  - `max-age`
  - `stale-while-revalidate`

While you can put the maximum time for server-side caching, cache times are best-effort and not guaranteed. If an asset is requested ofte

`proxy-revalidate` and `stale-if-error`

Vercel does not currently support using `proxy-revalidate` and `stale-if-error` for server-side caching.

---
title: "Purging Vercel CDN Cache"
description: "Learn how to invalidate and purge cached content on Vercel"
last_updated: "2026-01-16T02:19:26.521Z"
source: "https://vercel.com/docs/cdn-cache/purge"

Purging Vercel CDN Cache

Learn how to [invalidate and purge](#programmatically-purging-vercel-cache) cached content on Vercel's CDN, including cache keys and manu

Cache keys

Each request to Vercel's CDN has a cache key derived from the following:

- The request method (such as `GET`, `POST`, etc)
- The request URL (query strings are ignored for static files)
- The host domain
- The unique [deployment URL](/docs/deployments/generated-urls)
- The scheme (whether it's `https` or `http`)

Since each deployment has a different cache key, you can [promote a new deployment](/docs/deployments/promoting-a-deployment) to productio

> **💡 Note:** The cache key for Image Optimization behaves differently for [static
> images](/docs/image-optimization#local-images-cache-key) and [remote
> images](/docs/image-optimization#remote-images-cache-key).

Understanding cache purging

Invalidating the cache

When you invalidate a cache tag, all cached content associated with that tag is marked as stale. The next request serves the stale conten

Deleting the cache

When you delete a cache tag, the cached entries are marked for deletion. The next request fetches content from your origin before respond

Cache tags

Cache tags are user-defined strings that can be assigned to cached responses. These tags can later be used to purge the the CDN cache.

For example, you may have a product with id `123` that is displayed on multiple pages such as `/products/123/overview`, `/products/123/re

You can add a cache tag by importing [addCacheTag](/docs/functions/functions-api-reference/vercel-functions-package#addcachetag) from the

If you're using Next.js, you can import [cacheTag](https://nextjs.org/docs/app/api-reference/functions/cacheTag) from 'next/cache' instea

Cache tag case sensitivity

Cache tags are case-sensitive, meaning `product` and `Product` are treated as different tags.

Cache tag scope

Cache tags are scoped to your project and environment (production or preview).

When you purge a tag with the REST API, you can optionally provide a target environment such as preview or production (default is all env

When you purge a tag using `@vercel/functions` at runtime, the function's current environment is used which is derived from the deploymen

Programmatically purging CDN Cache

You can purge Vercel CDN cache in any of the following ways:

- [next/cache](https://nextjs.org/docs/app/api-reference/functions/cacheTag): Use helper methods like `revalidatePath()`, `revalidateTag(
- [@vercel/functions](/docs/functions/functions-api-reference/vercel-functions-package): Use helper methods like `invalidateByTag()`, `da
- [Vercel CLI](/docs/cli/cache): Use the `vercel cache invalidate` command or `vercel cache dangerously-delete` command with `--tag` or `
- [REST API](/docs/rest-api/reference/endpoints/edge-cache/invalidate-by-tag): Make direct API calls to the edge cache endpoint like `/in

Manually purging Vercel CDN Cache

In some circumstances, you may need to delete all cached data and force revalidation. For example, you might have set a `Cache-Control` t

1. Under your project, go to the **Settings** tab.
2. In the left sidebar, select **Caches**.
3. In the **CDN Cache** section, click **Purge CDN Cache**.
4. In the dialog, you'll see two options:
   - **Invalidate**: Marks a cache tag as stale, causing cache entries associated with that tag to be revalidated in the background on th
   - **Delete**: Marks a cache tag as deleted, causing cache entries associated with that tag to be revalidated in the foreground on the
5. In the dialog, you'll see a dropdown with two options:
   - **Cache Tag**: Purge cached responses associated with a specific user-defined tag.
   - **Source Image**: Purge [Image Optimization](/docs/image-optimization) transformed images based on the original source image URL.
6. In the dialog, enter a tag or source image in the input. You can use `*` to purge the entire project.
7. Finally, click the **Purge** button in the dialog to confirm.

The purge event itself is not billed but it can temporarily increase Function Duration, Functions Invocations, Edge Function Executions,

> **💡 Note:** Purge is not the same as creating a new deployment because it will also purge
> Image Optimization content, which is usually preserved between deployments, as
> well as ISR content, which is often generated at build time for new
> deployments.

Limits

	Maximum
Characters per tag	256
Tags per cached response	128
Tags per bulk REST API call	16

---
title: "Checks API Reference"
description: "The Vercel Checks API let you create tests and assertions that run after each deployment has been built, and are powered by
last_updated: "2026-01-16T02:19:26.527Z"
source: "https://vercel.com/docs/checks/checks-api"

Checks API Reference

API endpoints allow integrations to interact with the Vercel platform. Integrations can run checks every time you create a deployment.

> **💡 Note:** The and endpoints
> must be called with an OAuth2, or it will produce a
> error.

---
title: "Anatomy of the Checks API"
description: "Learn how to create your own Checks with Vercel Integrations. You can build your own Integration in order to register any a
last_updated: "2026-01-16T02:19:26.655Z"
source: "https://vercel.com/docs/checks/creating-checks"

Anatomy of the Checks API

Checks API extends the build and deploy process once your deployment is ready. Each check behaves like a webhook that triggers specific e

To learn more, see the [Supported Webhooks Events docs](/docs/webhooks/webhooks-api#supported-event-types).

The workflow for registering and running a check is as follows:

1. A check is created after the `deployment.created` event
2. When the `deployment.ready` event triggers, the check updates its `status` to `running`
3. When the check is finished, the `status` updates to `completed`

If a check is "rerequestable", your integration users get an option to [rerequest and rerun the failing checks](#rerunning-checks).

Types of Checks

Depending on the type, checks can block the domain assignment stage of deployments.

- **Blocking Checks**: Prevents a successful deployment and returns a `conclusion` with a `state` value of `canceled` or `failed`. For ex
- **Non-blocking Checks**: Return test results with a successful deployment regardless of the `conclusion`

A blocking check with a `failed` state is configured by the developer (and not the integration).

Associations

Checks are always associated with a specific deployment that is tested and validated.

Body attributes

Attributes	Format	Purpose
`blocking`	Boolean	Tells Vercel if this check needs to block the deployment
`name`	String	Name of the check
`detailsUrl`	String (optional)	URL to display in the Vercel dashboard
`externalID`	String (optional)	ID used for external use
`path`	String (optional)	Path of the page that is being checked
`rerequestable`	Boolean (optional)	Tells Vercel if the check can rerun. Users can trigger a `deployment.check-rerequested` [webhook
`conclusion`	String (optional)	The result of a running check. For [blocking checks](#types-of-checks) the values can be `cancel
`status`	String (optional)	Tells Vercel the status of the check with values: `running` and `completed`

| `output` | Object (optional) | Details about the result of the check. Vercel uses this data to display actionable information f

The check gets a `stale` status if there is no status update for more than one hour (`status = registered`). The same applies if the chec

Response

Response	Format	Purpose
`status`	String	The status of the check. It expects specific values like `running` or `completed`
`state`	String	Tells the current state of the connection
`connectedAt`	Number	Timestamp (in milliseconds) of when the configuration was connected
`type`	String	Name of the integrator performing the check

Response codes

Status	Outcome
`200`	Success
`400`	One of the provided values in the request body is invalid, **OR** one of the provided values in the request query is invalid
`403`	The provided token is not from an OAuth2 client **OR** you do not have permission to access this resource **OR** the API token
`404`	The check was not found **OR** the deployment was not found
`413`	The output provided is too large

Rich results

Output

The `output` property can store any data like [Web Vitals](/docs/speed-insights) and [Virtual Experience Score](/docs/speed-insights/metr

Key		Description
`TBT`		The [Total Blocking Time](/docs/speed-insights/metrics#total-blocking-time-tbt), measured by the
`LCP`		The [Largest Contentful Paint](/docs/speed-insights/metrics#largest-contentful-paint-lcp), measu
`FCP`		The [First Contentful Paint](/docs/speed-insights/metrics#first-contentful-paint-fcp), measured
`CLS`		The [Cumulative Layout Shift](/docs/speed-insights/metrics#cumulative-layout-shift-cls), measure
`virtualExperienceScore`		The overall [Virtual Experience Score](/docs/speed-insights/metrics#predictive-performance-metri

Each of these keys has the following properties:

Key		Description
`value`		The value measured for a particular metric, in milliseconds. For `virtualExperienceScore` this value is the percen
`previousValue`		A previous value for comparison purposes
`source`		`web-vitals`

Metrics

`metrics` makes [Web Vitals](/docs/speed-insights) visible on checks. It is defined inside `output` as follows:

```json filename="checks-metrics.json"
{
  "path": "/",
  "output": {
    "metrics": {
        "FCP": {
          "value": 1200,
          "previousValue": 1400,
          "source": "web-vitals"
        }
        "LCP": {
          "value": 1200,
          "previousValue": 1400,
          "source": "web-vitals"
        },
        "CLS": {
          "value": 1200,
          "previousValue": 1400,
          "source": "web-vitals"
        },
        "TBT": {
          "value": 1200,
          "previousValue": 1400,
          "source": "web-vitals"
        }
      }
    }
  }
}
```

> **💡 Note:** All fields are required except . If
> is present, the delta will be shown.

Rerunning checks

A check can be "rerequested" using the `deployment.check-rerequested` webhook. Add the `rerequestable` attribute, and you can rerequest f

A rerequested check triggers the`deployment.check-rerequested` webhook. It updates the check `status` to `running` and resets the `conclu

Skipping Checks

You can "Skip" to stop and ignore check results without affecting the alias assignment. You cannot skip active checks. They continue runn

Availability of URLs

For "Running Checks", only the [Automatic Deployment URL](/docs/deployments/generated-urls) is available. [Automatic Branch URL](/docs/de

Order of execution

Checks may take different times to run. Each integrator determines the running order of the checks. While [Vercel REST API](/docs/rest-ap

Status and conclusion

When Checks API begins running on your deployment, the `status` is set to `running`. Once it gets a `conclusion`, the `status` updates to

However, your deployment will fail if the `conclusion` updates to one of the following values:

Conclusion	`blocking=true`
`canceled`	Yes
`failed`	Yes
`neutral`	No
`succeeded`	No
`skipped`	No

---
title: "Working with Checks"
description: "Vercel automatically keeps an eye on various aspects of your web application using the Checks API. Learn how to use Checks
last_updated: "2026-01-16T02:19:26.743Z"
source: "https://vercel.com/docs/checks"

Working with Checks

Checks are tests and assertions created and run after every successful deployment. **Checks API** defines your application's quality metr

Most testing and CI/CD flows occur in synthetic environments. This leads to false results, overlooked performance degradation, and missed

Types of flows enabled by Checks API

Flow Type	Description
Core	Checks `200` responses on specific pages or APIs. Determine the deployment's health and identify issues with code, e
Performance	Collects [core web vital](/docs/speed-insights) information for specific pages and compares it with the new deployme
End-to-end	Validates that your deployment has all the required components to build successfully. And identifies any broken page
Optimization	Optimizes information about the bundle size. Ensures that your website manages large assets like package and image s

Checks lifecycle

The diagram shows the complete lifecycle of how a check works:

1. When a [deployment](/docs/deployments) is created, Vercel triggers the `deployment.created` webhook. This tells integrators that check
2. Next, an integrator uses the Checks API to create checks defined in the integration configuration
3. When the deployment is built, Vercel triggers the `deployment.ready` webhook. This notifies integrators to begin checks on the deploym
4. Vercel waits until all the created checks receive an update
5. Once all checks receive a `conclusion`, aliases will apply, and the deployment will go live

Learn more about this process in the [Anatomy of Checks API](/docs/integrations/checks-overview/creating-checks)

Checks integrations

You can create a [native](/docs/integrations#native-integrations) or [connectable account](/docs/integrations#connectable-accounts) integ

Install integrations

Vercel users can find and install your integration from the [Marketplace](/marketplace) under [testing](/marketplace/category/testing), [

Build your Checks integration

Once you have [created your integration](/docs/integrations/create-integration/marketplace-product), [publish](/docs/integrations/create-

- Provide low or no configuration solutions for developers to run checks
- A guided onboarding process for developers from the installation to the end result
- Provide relevant information about the outcome of the test on the Vercel dashboard
- Document how to go beyond the default behavior to build custom tests for advanced users

---
title: "Telemetry"
description: "Vercel CLI collects telemetry data about general usage."
last_updated: "2026-01-16T02:19:26.736Z"
source: "https://vercel.com/docs/cli/about-telemetry"

Telemetry

> **💡 Note:** Participation in this program is optional, and you may
> [opt-out](#how-do-i-opt-out-of-vercel-cli-telemetry) if you would prefer not
> to share any telemetry information.

Why is telemetry collected?

Vercel CLI Telemetry provides an accurate gauge of Vercel CLI feature usage, pain points, and customization across all users. This data e

What is being collected?

Vercel takes privacy and security seriously. Vercel CLI Telemetry tracks general usage information, such as commands and arguments used.
Specifically, the following are tracked:

- Command invoked (`vercel build`, `vercel deploy`, `vercel login`, etc.)
- Version of the Vercel CLI
- General machine information (e.g. number of CPUs, macOS/Windows/Linux, whether or not the command was run within CI)

> **💡 Note:** This list is regularly audited to ensure its accuracy.

You can view exactly what is being collected by setting the following environment variable: `VERCEL_TELEMETRY_DEBUG=1`.

When this environment variable is set, data will **not be sent to Vercel**.
The data will only be printed out to the [*stderr* stream](https://en.wikipedia.org/wiki/Standard_streams), prefixed with `[telemetry]`.

An example telemetry event looks like this:

```json
{
  "id": "cf9022fd-e4b3-4f67-bda2-f02dba5b2e40",
  "eventTime": 1728421688109,
  "key": "subcommand:ls",
  "value": "ls",
  "teamId": "team_9Cdf9AEOj9ef09FaSdEU0f0s",
  "sessionId": "e29b9b32-3edd-4599-92d2-f6886af005f6"
}
```

What about sensitive data?

Vercel CLI Telemetry **does not** collect any metrics which may contain sensitive data, including, but not limited to: environment variab

For more information about Vercel's privacy practices, please see our [Privacy Notice](https://vercel.com/legal/privacy-policy) and if yo

How do I opt-out of Vercel CLI telemetry?

You may use the [vercel telemetry](/docs/cli/telemetry) command to manage the telemetry collection status. This sets a global configurati

You may opt-out of telemetry data collection by running `vercel telemetry disable`:

```bash filename="terminal"
vercel telemetry disable
```

You may check the status of telemetry collection at any time by running `vercel telemetry status`:

```bash filename="terminal"
vercel telemetry status
```

You may re-enable telemetry if you'd like to re-join the program by running the following:

```bash filename="terminal"
vercel telemetry enable
```

Alternatively, you may opt-out by setting an environment variable: `VERCEL_TELEMETRY_DISABLED=1`. This will only apply for runs where the

---
title: "vercel alias"
description: "Learn how to apply custom domain aliases to your Vercel deployments using the vercel alias CLI command."
last_updated: "2026-01-16T02:19:26.758Z"
source: "https://vercel.com/docs/cli/alias"
---

vercel alias

The `vercel alias` command allows you to apply [custom domains](/docs/projects/custom-domains) to your deployments.

When a new deployment is created (with our [Git Integration](/docs/git), Vercel CLI, or the [REST API](/docs/rest-api)), the platform wil

Any custom domain that doesn't have a [custom preview branch](/docs/domains/working-with-domains/assign-domain-to-a-git-branch) configure

Custom domains that do have a custom preview branch configured, however, only get applied when using the [Git Integration](/docs/git).

If you're not using the [Git Integration](/docs/git), `vercel alias` is a great solution if you still need to apply custom domains based

Preferred production commands

The `vercel alias` command is not the recommended way to promote production deployments to specific domains. Instead, you can use the fol

- [`vercel --prod --skip-domain`](/docs/cli/deploy#prod): Use to skip custom domain assignment when deploying to production and creating
- [`vercel promote [deployment-id or url]`](/docs/cli/promote): Use to promote your staged deployment to your custom domains
- [`vercel rollback [deployment-id or url]`](/docs/cli/rollback): Use to alias an earlier production deployment to your custom domains

Usage

In general, the command allows for assigning custom domains to any deployment.

Make sure to **not** include the HTTP protocol (e.g. `https://`) for the `[custom-domain]` parameter.

```bash filename="terminal"
vercel alias set [deployment-url] [custom-domain]
```

```bash filename="terminal"
vercel alias rm [custom-domain]
```

```bash filename="terminal"
vercel alias ls
```

Unique options

These are options that only apply to the `vercel alias` command.

Yes

The `--yes` option can be used to bypass the confirmation prompt when removing an alias.

```bash filename="terminal"
vercel alias rm [custom-domain] --yes
```

```

### Limit

The `--limit` option can be used to specify the maximum number of aliases returned when using `ls`. The default value is `20` and the max

```bash filename="terminal"
vercel alias ls --limit 100
```

Related guides

- [How do I resolve alias related errors on Vercel?](/kb/guide/how-to-resolve-alias-errors-on-vercel)

---
title: "All Vercel CLI Commands"
description: "A complete list of all available Vercel CLI commands."
last_updated: "2026-01-16T02:19:26.680Z"
source: "https://vercel.com/docs/cli/all-cli-commands"

All Vercel CLI Commands

This document provides a complete list of all available Vercel CLI commands.

Installing Vercel CLI

To download and install Vercel CLI, run the following command:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i vercel
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i vercel
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i vercel
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i vercel
    ```
  </Code>
</CodeBlock>

Commands

alias

Apply custom domain aliases to your Vercel deployments.

```bash
vercel alias set [deployment-url] [custom-domain]
vercel alias rm [custom-domain]
vercel alias ls
```

bisect

Perform a binary search on your deployments to help surface issues.

```bash
vercel bisect
vercel bisect --good [deployment-url] --bad [deployment-url]
```

blob

Interact with Vercel Blob storage to upload, list, delete, and copy files.

```bash
vercel blob list
vercel blob put [path-to-file]
vercel blob del [url-or-pathname]
vercel blob copy [from-url] [to-pathname]
```

build

Build a Vercel Project locally or in your own CI environment.

```bash
vercel build
vercel build --prod
```

cache

Manage cache for your project (CDN cache and Data cache).

```bash
vercel cache purge
```
```

```
vercel cache purge --type cdn
vercel cache purge --type data
vercel cache invalidate --tag foo
vercel cache dangerously-delete --tag foo
```

certs

Manage certificates for your domains.

```bash
vercel certs ls
vercel certs issue [domain]
vercel certs rm [certificate-id]
```

curl

Make HTTP requests to your Vercel deployments with automatic deployment protection bypass. This is a beta command.

```bash
vercel curl [path]
vercel curl /api/hello
vercel curl /api/data --deployment [deployment-url]
```

deploy

Deploy your Vercel projects. Default command when no subcommand is specified.

```bash
vercel
vercel deploy
vercel deploy --prod
```

dev

Replicate the Vercel deployment environment locally and test your project.

```bash
vercel dev
vercel dev --port 3000
```

dns

Manage your DNS records for your domains.

```bash
vercel dns ls [domain]
vercel dns add [domain] [name] [type] [value]
vercel dns rm [record-id]
```

domains

Buy, sell, transfer, and manage your domains.

```bash
vercel domains ls
vercel domains add [domain] [project]
vercel domains rm [domain]
vercel domains buy [domain]
```

env

Manage environment variables in your Vercel Projects.

```bash
vercel env ls
vercel env add [name] [environment]
vercel env update [name] [environment]
vercel env rm [name] [environment]
vercel env pull [file]
vercel env run -- <command>
```

git

Manage your Git provider connections.

```bash
vercel git ls
vercel git connect
vercel git disconnect [provider]
```

guidance

Enable or disable guidance messages shown after CLI commands.

```bash
vercel guidance enable
vercel guidance disable
vercel guidance status
```

help

Get information about all available Vercel CLI commands.

```bash
vercel help
vercel help [command]
```

httpstat

Visualize HTTP request timing statistics for your Vercel deployments with automatic deployment protection bypass.

```bash
vercel httpstat [path]
vercel httpstat /api/hello
vercel httpstat /api/data --deployment [deployment-url]
```

init

Initialize example Vercel Projects locally from the examples repository.

```bash
vercel init
vercel init [project-name]
```

inspect

Retrieve information about your Vercel deployments.

```bash
vercel inspect [deployment-id-or-url]
vercel inspect [deployment-id-or-url] --logs
vercel inspect [deployment-id-or-url] --wait
```

install

Install native integrations with the option of adding a product.

```bash
vercel install [integration-name]
```

integration

Perform key integration tasks (add, open, list, remove).

```bash
vercel integration add [integration-name]
vercel integration open [integration-name]
vercel integration list
vercel integration remove [integration-name]
```

integration-resource

Perform native integration product resource tasks (remove, disconnect, create thresholds).

```bash
vercel integration-resource remove [resource-name]
vercel integration-resource disconnect [resource-name]
```

link

Link a local directory to a Vercel Project.

```bash
vercel link
vercel link [path-to-directory]
```

list

List recent deployments for the current Vercel Project.

```bash
vercel list
vercel list [project-name]
```

login

Login to your Vercel account through CLI.

```bash
vercel login
vercel login [email]
vercel login --github
```

logout

Logout from your Vercel account through CLI.

```bash
vercel logout
```

logs

List runtime logs for a specific deployment.

```bash
vercel logs [deployment-url]
vercel logs [deployment-url] --follow
```

mcp

Set up MCP client configuration for your Vercel Project.

```bash
vercel mcp
vercel mcp --project
```

microfrontends

Work with microfrontends configuration.

```bash
vercel microfrontends pull
vercel microfrontends pull --dpl [deployment-id-or-url]
```

open

Open your current project in the Vercel Dashboard.

```bash
vercel open
```

project

List, add, inspect, remove, and manage your Vercel Projects.

```bash
vercel project ls
vercel project add
vercel project inspect [project-name]
vercel project rm
```

promote

Promote an existing deployment to be the current deployment.

```bash
vercel promote [deployment-id-or-url]
vercel promote status [project]
```

pull

Update your local project with remote environment variables and project settings.

```bash
vercel pull
vercel pull --environment=production
```

redeploy

Rebuild and redeploy an existing deployment.

```bash
vercel redeploy [deployment-id-or-url]
```

redirects

Manage project-level redirects.

```bash
vercel redirects list
vercel redirects add /old /new --status 301
vercel redirects upload redirects.csv --overwrite
vercel redirects promote <version-id>
```

remove

Remove deployments either by ID or for a specific Vercel Project.

```bash
vercel remove [deployment-url]
vercel remove [project-name]
```

rollback

Roll back production deployments to previous deployments.

```bash
vercel rollback [deployment-id-or-url]
vercel rollback status [project]
```

```

### rolling-release

Manage your project's rolling releases to gradually roll out new deployments.

```bash
vercel rolling-release configure --cfg='[config]'
vercel rolling-release start --dpl=[deployment-id]
vercel rolling-release approve --dpl=[deployment-id]
vercel rolling-release complete --dpl=[deployment-id]
```

switch

Switch between different team scopes.

```bash
vercel switch
vercel switch [team-name]
```

target

Manage custom environments (targets) and use the `--target` flag on relevant commands.

```bash
vercel target list
vercel target ls
vercel deploy --target=staging
```

teams

List, add, remove, and manage your teams.

```bash
vercel teams list
vercel teams add
vercel teams invite [email]
```

telemetry

Enable or disable telemetry collection.

```bash
vercel telemetry status
vercel telemetry enable
vercel telemetry disable
```

whoami

Display the username of the currently logged in user.

```bash
vercel whoami
```

---
title: "vercel bisect"
description: "Learn how to perform a binary search on your deployments to help surface issues using the vercel bisect CLI command."
last_updated: "2026-01-16T02:19:26.763Z"
source: "https://vercel.com/docs/cli/bisect"
---

vercel bisect

The `vercel bisect` command can be used to perform a [binary search](https://wikipedia.org/wiki/Binary_search_algorithm "What is a binary

This is similar to [git bisect](https://git-scm.com/docs/git-bisect "What is a git bisect?") but faster because you don't need to wait to

Note that if an alias URL is used for either the *good* or *bad* deployment, then the URL will be resolved to the current target of the a

> **💡 Note:** The good and bad deployments provided to `vercel bisect` must be
> **production** deployments.

Usage

```bash filename="terminal"
vercel bisect
```

Unique Options

These are options that only apply to the `vercel bisect` command.

Good

The `--good` option, shorthand `-g`, can be used to specify the initial "good" deployment from the command line. When this option is pres

```bash filename="terminal"
vercel bisect --good https://example.com
```

Bad

The `--bad` option, shorthand `-b`, can be used to specify the "bad" deployment from the command line. When this option is present, the p
```

```bash filename="terminal"
vercel bisect --bad https://example-s93n1nfa.vercel.app
```

Path

The `--path` option, shorthand `-p`, can be used to specify a subpath of the deployment where the issue occurs. The subpath will be appen

```bash filename="terminal"
vercel bisect --path /blog/first-post
```

Open

The `--open` option, shorthand `-o`, will attempt to automatically open each deployment URL in your browser window for convenience.

```bash filename="terminal"
vercel bisect --open
```

Run

The `--run` option, shorthand `-r`, provides the ability for the bisect session to be automated using a shell script or command that will

```bash filename="terminal"
vercel bisect --run ./test.sh
```

Related guides

- [How to determine which Vercel Deployment introduced an issue?](/kb/guide/how-to-determine-which-vercel-deployment-introduced-an-issue)

---
title: "vercel blob"
description: "Learn how to interact with Vercel Blob storage using the vercel blob CLI command."
last_updated: "2026-01-16T02:19:26.778Z"
source: "https://vercel.com/docs/cli/blob"
---

vercel blob

The `vercel blob` command is used to interact with [Vercel Blob](/docs/storage/vercel-blob) storage, providing functionality to upload, l

For more information about Vercel Blob, see the [Vercel Blob documentation](/docs/storage/vercel-blob) and [Vercel Blob SDK reference](/d

Usage

The `vercel blob` command supports the following operations:

- [`list`](#list-ls) - List all files in the Blob store
- [`put`](#put) - Upload a file to the Blob store
- [`del`](#del) - Delete a file from the Blob store
- [`copy`](#copy-cp) - Copy a file in the Blob store
- [`store add`](#store-add) - Add a new Blob store
- [`store remove`](#store-remove-rm) - Remove a Blob store
- [`store get`](#store-get) - Get a Blob store

For authentication, the CLI reads the `BLOB_READ_WRITE_TOKEN` value from your env file or you can use the [`--rw-token` option](#rw-token

list (ls)

```bash filename="terminal"
vercel blob list
```

put

```bash filename="terminal"
vercel blob put [path-to-file]
```

del

```bash filename="terminal"
vercel blob del [url-or-pathname]
```

copy (cp)

```bash filename="terminal"
vercel blob copy [from-url-or-pathname] [to-pathname]
```

store add

```bash filename="terminal"
vercel blob store add [name] [--region <region>]
```

store remove (rm)

```bash filename="terminal"
vercel blob store remove [store-id]
```

store get

```bash filename="terminal"
```

```
vercel blob store get [store-id]
```

Unique Options

These are options that only apply to the `vercel blob` command.

Rw token

You can use the `--rw-token` option to specify your Blob read-write token.

```bash filename="terminal"
vercel blob put image.jpg --rw-token [rw-token]
```

Limit

You can use the `--limit` option to specify the number of results to return per page when using `list`. The default value is `10` and the

```bash filename="terminal"
vercel blob list --limit 100
```

Cursor

You can use the `--cursor` option to specify the cursor from a previous page to start listing from.

```bash filename="terminal"
vercel blob list --cursor [cursor-value]
```

Prefix

You can use the `--prefix` option to filter Blobs by a specific prefix.

```bash filename="terminal"
vercel blob list --prefix images/
```

Mode

You can use the `--mode` option to filter Blobs by either folded or expanded mode. The default is `expanded`.

```bash filename="terminal"
vercel blob list --mode folded
```

Add Random Suffix

You can use the `--add-random-suffix` option to add a random suffix to the file name when using `put` or `copy`.

```bash filename="terminal"
vercel blob put image.jpg --add-random-suffix
```

Pathname

You can use the `--pathname` option to specify the pathname to upload the file to. The default is the filename.

```bash filename="terminal"
vercel blob put image.jpg --pathname assets/images/hero.jpg
```

Content Type

You can use the `--content-type` option to overwrite the content-type when using `put` or `copy`. It will be inferred from the file exten

```bash filename="terminal"
vercel blob put data.txt --content-type application/json
```

Cache Control Max Age

You can use the `--cache-control-max-age` option to set the `max-age` of the cache-control header directive when using `put` or `copy`. T

```bash filename="terminal"
vercel blob put image.jpg --cache-control-max-age 86400
```

Force

You can use the `--force` option to overwrite the file if it already exists when uploading. The default is `false`.

```bash filename="terminal"
vercel blob put image.jpg --force
```

Multipart

You can use the `--multipart` option to upload the file in multiple small chunks for performance and reliability. The default is `true`.

```bash filename="terminal"
vercel blob put large-file.zip --multipart false
```

Region

You can use the `--region` option to specify the region where your Blob store should be created. The default is `iad1`. This option is on

```bash filename="terminal"
```

```
vercel blob store add my-store --region sfo1
```

---
title: "vercel build"
description: "Learn how to build a Vercel Project locally or in your own CI environment using the vercel build CLI command."
last_updated: "2026-01-16T02:19:26.782Z"
source: "https://vercel.com/docs/cli/build"
---

vercel build

The `vercel build` command can be used to build a Vercel Project locally or in your own CI environment.
Build artifacts are placed into the `.vercel/output` directory according to the
[Build Output API](/docs/build-output-api/v3).

When used in conjunction with the `vercel deploy --prebuilt` command, this allows a Vercel Deployment
to be created *without* sharing the Vercel Project's source code with Vercel.

This command can also be helpful in debugging a Vercel Project by receiving error messages for a failed
build locally, or by inspecting the resulting build artifacts to get a better understanding of
how Vercel will create the Deployment.

It is recommended to run the `vercel pull` command before invoking `vercel build` to ensure that
you have the most recent Project Settings and Environment Variables stored locally.

Usage

```bash filename="terminal"
vercel build
```

Unique Options

These are options that only apply to the `vercel build` command.

Production

The `--prod` option can be specified when you want to build the Vercel Project using Production Environment Variables. By default, the Pr

```bash filename="terminal"
vercel build --prod
```

Yes

The `--yes` option can be used to bypass the confirmation prompt and automatically pull environment variables and Project Settings if not

```bash filename="terminal"
vercel build --yes
```

target

Use the `--target` option to define the environment you want to build against. This could be production, preview, or a [custom environmen

```bash filename="terminal"
vercel build --target=staging
```

Output

The `--output` option specifies a custom directory where the build artifacts will be written to, instead of the default `.vercel/output`

```bash filename="terminal"
vercel build --output ./custom-output
```

Related guides

- [How can I use the Vercel CLI for custom workflows?](/kb/guide/using-vercel-cli-for-custom-workflows)

---
title: "vercel cache"
description: "Learn how to manage cache for your project using the vercel cache CLI command."
last_updated: "2026-01-16T02:19:26.795Z"
source: "https://vercel.com/docs/cli/cache"
---

vercel cache

The `vercel cache` command is used to manage the cache for your project, such as [CDN cache](/docs/cdn-cache) and [Data cache](https://ve

Learn more about [purging Vercel cache](/docs/cdn-cache/purge).

Usage

```bash filename="terminal"
vercel cache purge
```

Extended Usage

```bash filename="terminal"
vercel cache purge --type cdn
```

```bash filename="terminal"
```

```
vercel cache purge --type data
```

```bash filename="terminal"
vercel cache invalidate --tag blog-posts
```

```bash filename="terminal"
vercel cache dangerously-delete --tag blog-posts
```

```bash filename="terminal"
vercel cache invalidate --srcimg /api/avatar/1
```

```bash filename="terminal"
vercel cache dangerously-delete --srcimg /api/avatar/1
```

```bash filename="terminal"
vercel cache dangerously-delete --srcimg /api/avatar/1 --revalidation-deadline-seconds 604800
```

Unique Options

These are options that only apply to the `vercel cache` command.

tag

The `--tag` option specifies which tag to invalidate or delete from the cache. You can provide a single tag or multiple comma-separated t

```bash filename="terminal"
vercel cache invalidate --tag blog-posts,user-profiles,homepage
```

srcimg

The `--srcimg` option specifies a source image path to invalidate or delete from the cache. This invalidates or deletes all cached transf

You can't use both `--tag` and `--srcimg` options together. Choose one based on whether you're invalidating cached content by tag or by s

```bash filename="terminal"
vercel cache invalidate --srcimg /api/avatar/1
```

revalidation-deadline-seconds

The `--revalidation-deadline-seconds` option specifies the revalidation deadline in seconds. When used with `dangerously-delete`, cached

```bash filename="terminal"
vercel cache dangerously-delete --tag blog-posts --revalidation-deadline-seconds 3600
```

Yes

The `--yes` option can be used to bypass the confirmation prompt when purging the cache or dangerously deleting cached content.

```bash filename="terminal"
vercel cache purge --yes
```

---
title: "vercel certs"
description: "Learn how to manage certificates for your domains using the vercel certs CLI command."
last_updated: "2026-01-16T02:19:26.785Z"
source: "https://vercel.com/docs/cli/certs"
---

vercel certs

The `vercel certs` command is used to manage certificates for domains, providing functionality to list, issue, and remove them. Vercel ma

Usage

```bash filename="terminal"
vercel certs ls
```

Extended Usage

```bash filename="terminal"
vercel certs issue [domain1, domain2, domain3]
```

```bash filename="terminal"
vercel certs rm [certificate-id]
```

Unique Options

These are options that only apply to the `vercel certs` command.

Challenge Only

The `--challenge-only` option can be used to only show the challenges needed to issue a certificate.

```bash filename="terminal"
vercel certs issue foo.com --challenge-only
```

Limit

The `--limit` option can be used to specify the maximum number of certs returned when using `ls`. The default value is `20` and the maxim

```bash filename="terminal"
vercel certs ls --limit 100
```

---
title: "vercel curl"
description: "Learn how to make HTTP requests to your Vercel deployments with automatic deployment protection bypass using the vercel cur
last_updated: "2026-01-16T02:19:26.926Z"
source: "https://vercel.com/docs/cli/curl"
---

vercel curl

> **⚠️ Warning:** The `vercel curl` command is currently in beta. Features and behavior may change.

The `vercel curl` command works like `curl`, but automatically handles deployment protection bypass tokens for you. When your project has

The command runs the system `curl` command with the same arguments you provide, but adds an [`x-vercel-protection-bypass`](/docs/deployme

> **💡 Note:** This command is available in Vercel CLI v48.8.0 and later. If you're using an older version, see [Updating Vercel CLI](/do

Usage

```bash filename="terminal"
vercel curl [path]
```

Examples

Basic request

Make a GET request to your production deployment:

```bash filename="terminal"
vercel curl /api/hello
```

POST request with data

Send a POST request with JSON data:

```bash filename="terminal"
vercel curl /api/users -X POST -H "Content-Type: application/json" -d '{"name":"John"}'
```

Request specific deployment

Test a specific deployment by its URL:

```bash filename="terminal"
vercel curl /api/status --deployment https://my-app-abc123.vercel.app
```

Verbose output

See detailed request information:

```bash filename="terminal"
vercel curl /api/data -v
```

How it works

When you run `vercel curl`:

1. The CLI finds your linked project (or you can specify one with [`--scope`](/docs/cli/global-options#scope))
2. It gets the latest production deployment URL (or uses the deployment you specified)
3. It retrieves or generates a deployment protection bypass token
4. It runs the system `curl` command with the bypass token in the `x-vercel-protection-bypass` header

The command requires `curl` to be installed on your system.

Unique options

These are options that only apply to the `vercel curl` command.

Deployment

The `--deployment` option, shorthand `-d`, lets you specify a deployment URL to request instead of using the production deployment.

```bash filename="terminal"
vercel curl /api/hello --deployment https://my-app-abc123.vercel.app
```

Protection Bypass

The `--protection-bypass` option, shorthand `-b`, lets you provide your own deployment protection bypass secret instead of automatically

```bash filename="terminal"
vercel curl /api/hello --protection-bypass your-secret-here
```

You can also use the [`VERCEL_AUTOMATION_BYPASS_SECRET`](/docs/deployment-protection/methods-to-bypass-deployment-protection/protection-b

```bash filename="terminal"
export VERCEL_AUTOMATION_BYPASS_SECRET=your-secret-here
vercel curl /api/hello
```

Troubleshooting

curl command not found

Make sure `curl` is installed on your system:

```bash filename="terminal"
# macOS (using Homebrew)
brew install curl

# Ubuntu/Debian
sudo apt-get install curl

# Windows (using Chocolatey)
choco install curl
```

No deployment found for the project

Make sure you're in a directory with a linked Vercel project and that the project has at least one deployment:

```bash filename="terminal"
# Link your project
vercel link

# Deploy your project
vercel deploy
```

Failed to get deployment protection bypass token

If automatic token creation fails, you can create a bypass secret manually in the Vercel Dashboard:

1. Go to your project's **Settings** → **Deployment Protection**
2. Find "Protection Bypass for Automation"
3. Click "Create" or "Generate" to create a new secret
4. Copy the generated secret
5. Use it with the `--protection-bypass` flag or [`VERCEL_AUTOMATION_BYPASS_SECRET`](/docs/deployment-protection/methods-to-bypass-deployı

No deployment found for ID

When using `--deployment`, verify that:

- The deployment ID or URL is correct
- The deployment belongs to your linked project
- The deployment hasn't been deleted

Related

- [Deployment Protection](/docs/security/deployment-protection)
- [vercel deploy](/docs/cli/deploy)
- [vercel inspect](/docs/cli/inspect)

---
title: "vercel deploy"
description: "Learn how to deploy your Vercel projects using the vercel deploy CLI command."
last_updated: "2026-01-16T02:19:26.943Z"
source: "https://vercel.com/docs/cli/deploy"
---

vercel deploy

The `vercel deploy` command deploys Vercel projects, executable from the project's root directory or by specifying a path. You can omit '

Usage

```bash filename="terminal"
vercel
```

Extended usage

```bash filename="terminal"
vercel --cwd [path-to-project]
```

```bash filename="terminal"
vercel deploy --prebuilt
```

Standard output usage

When deploying, `stdout` is always the Deployment URL.

```bash filename="terminal"
vercel > deployment-url.txt
```

Deploying to a custom domain

In the following example, you create a bash script that you include in your CI/CD workflow. The goal is to have all preview deployments b

```bash filename="deployDomain.sh"
# save stdout and stderr to files
```

```
vercel deploy >deployment-url.txt 2>error.txt

# check the exit code
code=$?
if [ $code -eq 0 ]; then
    # Now you can use the deployment url from stdout for the next step of your workflow
    deploymentUrl=`cat deployment-url.txt`
    vercel alias $deploymentUrl my-custom-domain.com
else
    # Handle the error
    errorMessage=`cat error.txt`
    echo "There was an error: $errorMessage"
fi
```

Standard error usage

If you need to check for errors when the command is executed such as in a CI/CD workflow,
use `stderr`. If the exit code is anything other than `0`, an error has occurred. The
following example demonstrates a script that checks if the exit code is not equal to 0:

```bash filename="checkDeploy.sh"
# save stdout and stderr to files
vercel deploy >deployment-url.txt 2>error.txt

# check the exit code
code=$?
if [ $code -eq 0 ]; then
    # Now you can use the deployment url from stdout for the next step of your workflow
    deploymentUrl=`cat deployment-url.txt`
    echo $deploymentUrl
else
    # Handle the error
    errorMessage=`cat error.txt`
    echo "There was an error: $errorMessage"
fi
```

Unique options

These are options that only apply to the `vercel` command.

Prebuilt

The `--prebuilt` option can be used to upload and deploy the results of a previous `vc build` execution located in the .vercel/output dir

When not to use --prebuilt

When using the `--prebuilt` flag, no deployment ID will be made available for supported frameworks (like Next.js) to use, which means [Sk

If you need Skew Protection or System Environment Variables, do not use the `--prebuilt` flag or use Git-based deployments.

```bash filename="terminal"
vercel --prebuilt
```

You should also consider using the [archive](/docs/cli/deploy#archive) option to minimize the number of files uploaded and avoid hitting

```bash filename="terminal"
# Build the project locally
vercel build

# Deploy the pre-built project, archiving it as a .tgz file
vercel deploy --prebuilt --archive=tgz
```

This example uses the `vercel build` command to build your project locally. It then uses the `--prebuilt` and `--archive=tgz` options on

Build env

The `--build-env` option, shorthand `-b`, can be used to provide environment variables to the [build step](/docs/deployments/configure-a-

```bash filename="terminal"
vercel --build-env KEY1=value1 --build-env KEY2=value2
```

Yes

The `--yes` option can be used to skip questions you are asked when setting up a new Vercel project.
The questions will be answered with the provided defaults, inferred from `vercel.json` and the folder name.

```bash filename="terminal"
vercel --yes
```

Env

The `--env` option, shorthand `-e`, can be used to provide [environment variables](/docs/environment-variables) at runtime.

```bash filename="terminal"
vercel --env KEY1=value1 --env KEY2=value2
```

Name

> **💡 Note:** The option has been deprecated in favor of
> [Vercel project linking](/docs/cli/project-linking), which allows you to link
> a Vercel project to your local codebase when you run
> .

The `--name` option, shorthand `-n`, can be used to provide a Vercel project name for a deployment.

```bash filename="terminal"
vercel --name foo
```

Prod

The `--prod` option can be used to create a deployment for a production domain specified in the Vercel project dashboard.

```bash filename="terminal"
vercel --prod
```

Skip Domain

> **⚠️ Warning:** This CLI option will override the [Auto-assign Custom Production
> Domains](/docs/deployments/promoting-a-deployment#staging-and-promoting-a-production-deployment)
> project setting.

Must be used with [`--prod`](#prod). The `--skip-domain` option will disable the automatic promotion (aliasing) of the relevant domains t

```bash filename="terminal"
vercel --prod --skip-domain
```

Public

The `--public` option can be used to ensures the source code is publicly available at the `/_src` path.

```bash filename="terminal"
vercel --public
```

Regions

The `--regions` option can be used to specify which [regions](/docs/regions) the deployments [Vercel functions](/docs/functions) should r

```bash filename="terminal"
vercel --regions sfo1
```

No wait

The `--no-wait` option does not wait for a deployment to finish before exiting from the `deploy` command.

```bash filename="terminal"
vercel --no-wait
```

Force

The `--force` option, shorthand `-f`, is used to force a new deployment without the [build cache](/docs/deployments/troubleshoot-a-build#

```bash filename="terminal"
vercel --force
```

With cache

The `--with-cache` option is used to retain the [build cache](/docs/deployments/troubleshoot-a-build#what-is-cached) when using `--force`

```bash filename="terminal"
vercel --force --with-cache
```

Archive

The `--archive` option compresses the deployment code into one or more files before uploading it. This option should be used when deploym

In some cases, `--archive` makes deployments slower. This happens because the caching of source files to optimize file uploads in future

```bash filename="terminal"
vercel deploy --archive=tgz
```

Logs

The `--logs` option, shorthand `-l`, also prints the build logs.

```bash filename="terminal"
vercel deploy --logs
```

Meta

The `--meta` option, shorthand `-m`, is used to add metadata to the deployment.

```bash filename="terminal"
vercel deploy --meta KEY1=value1
```

> **💡 Note:** Deployments can be filtered using this data with [`vercel list   --meta`](/docs/cli/list#meta).

target

Use the `--target` option to define the environment you want to deploy to. This could be production, preview, or a [custom environment](/

```bash filename="terminal"
vercel deploy --target=staging
```

```

### Guidance

The `--guidance` option displays suggested next steps and commands after deployment completes. This can help you discover relevant CLI co

```bash filename="terminal"
vercel deploy --guidance
```

---
title: "Deploying Projects from Vercel CLI"
description: "Learn how to deploy your Vercel Projects from Vercel CLI using the vercel or vercel deploy commands."
last_updated: "2026-01-16T02:19:26.846Z"
source: "https://vercel.com/docs/cli/deploying-from-cli"

Deploying Projects from Vercel CLI

Deploying from source

The `vercel` command is used to [deploy](/docs/cli/deploy) Vercel Projects and can be used from either the root of the Vercel Project dir

```bash filename="terminal"
vercel
```

You can alternatively use the [`vercel deploy` command](/docs/cli/deploy) for the same effect, if you want to be more explicit.

```bash filename="terminal"
vercel [path-to-project]
```

When deploying, stdout is always the Deployment URL.

```bash filename="terminal"
vercel > deployment-url.txt
```

Relevant commands

- [deploy](/docs/cli/deploy)

Deploying a staged production build

By default, when you promote a deployment to production, your domain will point to that deployment. If you want to create a production de

1. Turn off the auto-assignment of domains for the current production deployment:

```bash filename="terminal"
vercel --prod --skip-domain
```

2. When you are ready, manually promote the staged deployment to production:

```bash filename="terminal"
vercel promote [deployment-id or url]
```

Relevant commands

- [promote](/docs/cli/promote)
- [deploy](/docs/cli/deploy)

Deploying from local build (prebuilt)

You can build Vercel projects locally to inspect the build outputs before they are [deployed](/docs/cli/deploy). This is a great option f

It's also useful for debugging build outputs.

```bash filename="terminal"
vercel build
```

This produces `.vercel/output` in the [Build Output API](/docs/build-output-api/v3) format. You can review the output, then [deploy](/doc

```bash filename="terminal"
vercel deploy --prebuilt
```

> **⚠ Warning:** Review the [When not to use
> \--prebuilt](/docs/cli/deploy#when-not-to-use---prebuilt) section to understand
> when you should not use the `--prebuilt` flag.

See more details at [Build Output API](/docs/build-output-api/v3).

Relevant commands

- [build](/docs/cli/build)
- [deploy](/docs/cli/deploy)

---
title: "vercel dev"
description: "Learn how to replicate the Vercel deployment environment locally and test your Vercel Project before deploying using the ve
last_updated: "2026-01-16T02:19:26.947Z"
source: "https://vercel.com/docs/cli/dev"

```

vercel dev

The `vercel dev` command is used to replicate the Vercel deployment environment locally, allowing you to test your [Vercel Functions](/do

If the [Development Command](/docs/deployments/configure-a-build#development-command) is configured in your Project Settings, it will aff

> **💡 Note:** Before running , make sure to install your
> dependencies by running .

When to Use This Command

If you're using a framework and your framework's [Development Command](/docs/deployments/configure-a-build#development-command) already p

For example, [Next.js](/docs/frameworks/nextjs)'s Development Command (`next dev`) provides native support for Functions, [redirects](/do

Usage

```bash filename="terminal"
vercel dev
```

Unique Options

These are options that only apply to the `vercel dev` command.

Listen

The `--listen` option, shorthand `-l`, can be used to specify which port `vercel dev` runs on.

```bash filename="terminal"
vercel dev --listen 5005
```

Yes

The `--yes` option can be used to skip questions you are asked when setting up a new Vercel Project.
The questions will be answered with the default scope and current directory for the Vercel Project name and location.

```bash filename="terminal"
vercel dev --yes
```

---
title: "vercel dns"
description: "Learn how to manage your DNS records for your domains using the vercel dns CLI command."
last_updated: "2026-01-16T02:19:26.955Z"
source: "https://vercel.com/docs/cli/dns"
---

vercel dns

The `vercel dns` command is used to manage DNS record for domains, providing functionality to list, add, remove, and import records.

> **💡 Note:** When adding DNS records, please wait up to 24 hours for new records to
> propagate.

Usage

```bash filename="terminal"
vercel dns ls
```

Extended Usage

```bash filename="terminal"
vercel dns add [domain] [subdomain] [A || AAAA || ALIAS || CNAME || TXT] [value]
```

```bash filename="terminal"
vercel dns add [domain] '@' MX [record-value] [priority]
```

```bash filename="terminal"
vercel dns add [domain] [name] SRV [priority] [weight] [port] [target]
```

```bash filename="terminal"
vercel dns add [domain] [name] CAA '[flags] [tag] "[value]"'
```

```bash filename="terminal"
vercel dns rm [record-id]
```

```bash filename="terminal"
vercel dns import [domain] [path-to-zonefile]
```

Unique Options

These are options that only apply to the `vercel dns` command.

Limit

The `--limit` option can be used to specify the maximum number of dns records returned when using `ls`. The default value is `20` and the

```bash filename="terminal"
vercel dns ls --limit 100
```

---
title: "vercel domains"
description: "Learn how to buy, sell, transfer, and manage your domains using the vercel domains CLI command."
last_updated: "2026-01-16T02:19:26.959Z"
source: "https://vercel.com/docs/cli/domains"
---

vercel domains

The `vercel domains` command is used to manage domains under the current scope, providing functionality to list, inspect, add, remove, pu

> **💡 Note:** You can manage domains with further options and greater control under a Vercel
> Project's Domains tab from the Vercel Dashboard.

Usage

```bash filename="terminal"
vercel domains ls
```

Extended Usage

```bash filename="terminal"
vercel domains inspect [domain]
```

```bash filename="terminal"
vercel domains add [domain] [project]
```

```bash filename="terminal"
vercel domains rm [domain]
```

```bash filename="terminal"
vercel domains buy [domain]
```

```bash filename="terminal"
vercel domains move [domain] [scope-name]
```

```bash filename="terminal"
vercel domains transfer-in [domain]
```

Unique Options

These are options that only apply to the `vercel domains` command.

Yes

The `--yes` option can be used to bypass the confirmation prompt when removing a domain.

```bash filename="terminal"
vercel domains rm [domain] --yes
```

Limit

The `--limit` option can be used to specify the maximum number of domains returned when using `ls`. The default value is `20` and the max

```bash filename="terminal"
vercel domains ls --limit 100
```

Next

The `--next` option enables pagination when listing domains. Pass the timestamp (in milliseconds since the UNIX epoch) from a previous re

```bash filename="terminal"
vercel domains ls --next 1584722256178
```

Force

The `--force` option forces a domain on a project, removing it from an existing one.

```bash filename="terminal"
vercel domains add my-domain.com my-project --force
```

---
title: "vercel env"
description: "Learn how to manage your environment variables in your Vercel Projects using the vercel env CLI command."
last_updated: "2026-01-16T02:19:26.966Z"
source: "https://vercel.com/docs/cli/env"
---

vercel env

The `vercel env` command is used to manage [Environment Variables](/docs/environment-variables) of a Project, providing functionality to

To leverage environment variables in local tools (like `next dev` or `gatsby dev`) that want them in a file (like `.env`), run `vercel en

To run a command with environment variables without writing them to a file, use `vercel env run -- <command>`. This fetches the environme

Exporting Development Environment Variables

Some frameworks make use of environment variables during local development through CLI commands like `next dev` or `gatsby dev`. The `ver

```bash filename="terminal"
vercel env pull [file]
```

To override environment variable values temporarily, use:

```bash filename="terminal"
MY_ENV_VAR="temporary value" next dev
```

> **💡 Note:** If you are using [](/docs/cli/build) or [
>](/docs/cli/dev), you should use [
>](/docs/cli/pull) instead. Those commands
> operate on a local copy of environment variables and Project settings that are
> saved under , which
> provides.

Usage

```bash filename="terminal"
vercel env ls
```

```bash filename="terminal"
vercel env add
```

```bash filename="terminal"
vercel env rm
```

Extended Usage

```bash filename="terminal"
vercel env ls [environment]
```

```bash filename="terminal"
vercel env ls [environment] [gitbranch]
```

```bash filename="terminal"
vercel env add [name]
```

```bash filename="terminal"
vercel env add [name] [environment]
```

```bash filename="terminal"
vercel env add [name] [environment] [gitbranch]
```

```bash filename="terminal"
vercel env add [name] [environment] < [file]
```

```bash filename="terminal"
echo [value] | vercel env add [name] [environment]
```

```bash filename="terminal"
vercel env add [name] [environment] [gitbranch] < [file]
```

```bash filename="terminal"
vercel env rm [name] [environment]
```

Updating Environment Variables

The `vercel env update` sub-command updates the value of an existing environment variable.

```bash filename="terminal"
vercel env update [name]
```

```bash filename="terminal"
vercel env update [name] [environment]
```

```bash filename="terminal"
vercel env update [name] [environment] [gitbranch]
```

```bash filename="terminal"
cat ~/.npmrc | vercel env update NPM_RC preview
```

```bash filename="terminal"
vercel env pull [file]
```

```bash filename="terminal"
```

vercel env pull --environment=preview
```

```bash filename="terminal"
vercel env pull --environment=preview --git-branch=feature-branch
```

Running Commands with Environment Variables

The `vercel env run` sub-command runs any command with environment variables from your linked Vercel project, without writing them to a f

```bash filename="terminal"
vercel env run -- <command>
```

```bash filename="terminal"
vercel env run -- next dev
```

```bash filename="terminal"
vercel env run -e preview -- npm test
```

```bash filename="terminal"
vercel env run -e production -- next build
```

```bash filename="terminal"
vercel env run -e preview --git-branch feature-x -- next dev
```

> **💡 Note:** The separator is required to distinguish between
> flags for and the command you want to
> run. Flags after are passed to your command.

Options

The following options are available for `vercel env run`:

- `-e, --environment`: Specify the environment to pull variables from. Defaults to `development`. Accepts `development`, `preview`, or `p
- `--git-branch`: Specify a Git branch to pull branch-specific Environment Variables.

Unique Options

These are options that only apply to the `vercel env` command.

Sensitive

The `--sensitive` option marks an environment variable as sensitive. Sensitive variables have additional security measures and their valu

```bash filename="terminal"
vercel env add API_TOKEN --sensitive
```

```bash filename="terminal"
vercel env update API_TOKEN --sensitive
```

Force

The `--force` option overwrites an existing environment variable of the same target without prompting for confirmation.

```bash filename="terminal"
vercel env add API_TOKEN production --force
```

Yes

The `--yes` option can be used to bypass the confirmation prompt when overwriting an environment file, removing an environment variable,

```bash filename="terminal"
vercel env pull --yes
```

```bash filename="terminal"
vercel env rm [name] --yes
```

```bash filename="terminal"
vercel env update API_TOKEN production --yes
```

---
title: "vercel git"
description: "Learn how to manage your Git provider connections using the vercel git CLI command."
last_updated: "2026-01-16T02:19:26.969Z"
source: "https://vercel.com/docs/cli/git"
---

vercel git

The `vercel git` command is used to manage a Git provider repository for a Vercel Project,
enabling deployments to Vercel through Git.

When run, Vercel CLI searches for a local `.git` config file containing at least one remote URL.
```

If found, you can connect it to the Vercel Project linked to your directory.

[Learn more about using Git with Vercel](/docs/git).

Usage

```bash filename="terminal"
vercel git connect
```

```bash filename="terminal"
vercel git disconnect
```

Unique Options

These are options that only apply to the `vercel git` command.

Yes

The `--yes` option can be used to skip connect confirmation.

```bash filename="terminal"
vercel git connect --yes
```

---
title: "Vercel CLI Global Options"
description: "Global options are commonly available to use with multiple Vercel CLI commands. Learn about Vercel CLI"
last_updated: "2026-01-16T02:19:26.900Z"
source: "https://vercel.com/docs/cli/global-options"

Vercel CLI Global Options

Global options are commonly available to use with multiple Vercel CLI commands.

Current Working Directory

The `--cwd` option can be used to provide a working directory (that can be different from the current directory) when running Vercel CLI

This option can be a relative or absolute path.

```bash filename="terminal"
vercel --cwd ~/path-to/project
```

Debug

The `--debug` option, shorthand `-d`, can be used to provide a more verbose output when running Vercel CLI commands.

```bash filename="terminal"
vercel --debug
```

Global config

The `--global-config` option, shorthand `-Q`, can be used set the path to the [global configuration directory](/docs/project-configuratio

```bash filename="terminal"
vercel --global-config /path-to/global-config-directory
```

Help

The `--help` option, shorthand `-h`, can be used to display more information about [Vercel CLI](/cli) commands.

```bash filename="terminal"
vercel --help
```

```bash filename="terminal"
vercel alias --help
```

Local config

The `--local-config` option, shorthand `-A`, can be used to set the path to a local `vercel.json` file.

```bash filename="terminal"
vercel --local-config /path-to/vercel.json
```

Scope

The `--scope` option, shorthand `-S`, can be used to execute Vercel CLI commands from a scope that’s not currently active.

```bash filename="terminal"
vercel --scope my-team-slug
```

Token

The `--token` option, shorthand `-t`, can be used to execute Vercel CLI commands with an [authorization token](/account/tokens).

```bash filename="terminal"
vercel --token iZJb2oftmY4ab12HBzyBXMkp
```

No Color

The `--no-color` option, or `NO_COLOR=1` environment variable, can be used to execute Vercel CLI commands with no color or emoji output.

```bash filename="terminal"
vercel login --no-color
```

Team

The `--team` option, shorthand `-T`, can be used to specify a team slug or ID for the command. This is useful when you need to run a comm

```bash filename="terminal"
vercel list --team my-team-slug
```

```bash filename="terminal"
vercel deploy -T team_abc123def
```

Version

The `--version` option, shorthand `-v`, outputs the current version number of Vercel CLI.

```bash filename="terminal"
vercel --version
```

---
title: "vercel guidance"
description: "Enable or disable guidance messages in the Vercel CLI using the vercel guidance command."
last_updated: "2026-01-16T02:19:26.972Z"
source: "https://vercel.com/docs/cli/guidance"
---

vercel guidance

The `vercel guidance` command allows you to enable or disable guidance messages. Guidance messages are helpful suggestions shown after ce

Usage

```bash filename="terminal"
vercel guidance <subcommand>
```

Subcommands

enable

Enable guidance messages to receive command suggestions after operations complete.

```bash filename="terminal"
vercel guidance enable
```

disable

Disable guidance messages if you prefer a quieter CLI experience.

```bash filename="terminal"
vercel guidance disable
```

status

Check whether guidance messages are currently enabled or disabled.

```bash filename="terminal"
vercel guidance status
```

Examples

Enable guidance after deployment

```bash filename="terminal"
vercel guidance enable
vercel deploy
```

Check current status

```bash filename="terminal"
vercel guidance status
```

---
title: "vercel help"
description: "Learn how to use the vercel help CLI command to get information about all available Vercel CLI commands."
last_updated: "2026-01-16T02:19:26.982Z"
source: "https://vercel.com/docs/cli/help"
---

vercel help

The `vercel help` command generates a list of all available Vercel CLI commands and [options](/docs/cli/global-options) in the terminal.

Alternatively, the [`--help` global option](/docs/cli/global-options#help) can be added to commands to get help information about that co

Usage

```bash filename="terminal"
vercel help
```

Extended Usage

```bash filename="terminal"
vercel help [command]
```

---
title: "vercel httpstat"
description: "Learn how to visualize HTTP request timing statistics for your Vercel deployments using the vercel httpstat CLI command."
last_updated: "2026-01-16T02:19:27.058Z"
source: "https://vercel.com/docs/cli/httpstat"
---

vercel httpstat

> **⚠️ Warning:** The `vercel httpstat` command is currently in beta. Features and behavior may change.

The `vercel httpstat` command works like `httpstat`, but automatically handles deployment protection bypass tokens for you. It provides v

The command runs the `httpstat` tool with the same arguments you provide, but adds an [`x-vercel-protection-bypass`](/docs/deployment-pro

> **💡 Note:** This command is available in Vercel CLI v48.9.0 and later. If you're using an older version, see [Updating Vercel CLI](/do

Usage

```bash filename="terminal"
vercel httpstat [path]
```

Examples

Basic timing analysis

Get timing statistics for your production deployment:

```bash filename="terminal"
vercel httpstat /api/hello
```

POST request timing

Analyze timing for a POST request with JSON data:

```bash filename="terminal"
vercel httpstat /api/users -X POST -H "Content-Type: application/json" -d '{"name":"John"}'
```

Specific deployment timing

Test timing for a specific deployment by its URL:

```bash filename="terminal"
vercel httpstat /api/status --deployment https://my-app-abc123.vercel.app
```

Multiple requests

Run multiple requests to get average timing statistics:

```bash filename="terminal"
vercel httpstat /api/data -n 10
```

How it works

When you run `vercel httpstat`:

1. The CLI finds your linked project (or you can specify one with [`--scope`](/docs/cli/global-options#scope))
2. It gets the latest production deployment URL (or uses the deployment you specified)
3. It retrieves or generates a deployment protection bypass token
4. It runs the `httpstat` tool with the bypass token in the `x-vercel-protection-bypass` header
5. The tool displays a visual breakdown of request timing phases: DNS lookup, TCP connection, TLS handshake, server processing, and conte

The command requires `httpstat` to be installed on your system.

Unique options

These are options that only apply to the `vercel httpstat` command.

Deployment

The `--deployment` option, shorthand `-d`, lets you specify a deployment URL to request instead of using the production deployment.

```bash filename="terminal"
vercel httpstat /api/hello --deployment https://my-app-abc123.vercel.app
```

Protection Bypass

The `--protection-bypass` option, shorthand `-b`, lets you provide your own deployment protection bypass secret instead of automatically

```bash filename="terminal"
vercel httpstat /api/hello --protection-bypass your-secret-here
```

You can also use the [`VERCEL_AUTOMATION_BYPASS_SECRET`](/docs/deployment-protection/methods-to-bypass-deployment-protection/protection-by

```bash filename="terminal"
export VERCEL_AUTOMATION_BYPASS_SECRET=your-secret-here
vercel httpstat /api/hello
```

Understanding the output

The `httpstat` tool displays timing information in a visual format:

- **DNS Lookup**: Time to resolve the domain name
- **TCP Connection**: Time to establish a TCP connection
- **TLS Handshake**: Time to complete the SSL/TLS handshake (for HTTPS)
- **Server Processing**: Time for the server to generate the response
- **Content Transfer**: Time to download the response body

Each phase is color-coded and displayed with its duration in milliseconds, helping you identify which part of the request is taking the m

Troubleshooting

httpstat command not found

Make sure `httpstat` is installed on your system:

```bash filename="terminal"
# Install with pip (Python)
pip install httpstat

# Or install with Homebrew (macOS)
brew install httpstat
```

No deployment found for the project

Make sure you're in a directory with a linked Vercel project and that the project has at least one deployment:

```bash filename="terminal"
# Link your project
vercel link

# Deploy your project
vercel deploy
```

Failed to get deployment protection bypass token

If automatic token creation fails, you can create a bypass secret manually in the Vercel Dashboard:

1. Go to your project's **Settings** → **Deployment Protection**
2. Find "Protection Bypass for Automation"
3. Click "Create" or "Generate" to create a new secret
4. Copy the generated secret
5. Use it with the `--protection-bypass` flag or [`VERCEL_AUTOMATION_BYPASS_SECRET`](/docs/deployment-protection/methods-to-bypass-deployn

No deployment found for ID

When using `--deployment`, verify that:

- The deployment ID or URL is correct
- The deployment belongs to your linked project
- The deployment hasn't been deleted

Related

- [Deployment Protection](/docs/security/deployment-protection)
- [vercel curl](/docs/cli/curl)
- [vercel deploy](/docs/cli/deploy)
- [vercel inspect](/docs/cli/inspect)

---
title: "vercel init"
description: "Learn how to initialize Vercel supported framework examples locally using the vercel init CLI command."
last_updated: "2026-01-16T02:19:27.072Z"
source: "https://vercel.com/docs/cli/init"
---

vercel init

The `vercel init` command is used to initialize [Vercel supported framework](/docs/frameworks) examples locally from the examples found i

Usage

```bash filename="terminal"
vercel init
```

Extended Usage

```bash filename="terminal"
vercel init [framework-name]
```

```bash filename="terminal"
vercel init [framework-name] [new-local-directory-name]
```

```

## Unique Options

These are options that only apply to the `vercel env` command.

### Force

The `--force` option, shorthand `-f`, is used to forcibly replace an existing local directory.

```bash filename="terminal"
vercel init --force
```

```bash filename="terminal"
vercel init gatsby my-project-directory --force
```

---
title: "vercel inspect"
description: "Learn how to retrieve information about your Vercel deployments using the vercel inspect CLI command."
last_updated: "2026-01-16T02:19:27.062Z"
source: "https://vercel.com/docs/cli/inspect"
---

vercel inspect

The `vercel inspect` command is used to retrieve information about a deployment referenced either by its deployment URL or ID.

You can use this command to view either a deployment's information or its [build logs](/docs/cli/inspect#logs).

Usage

```bash filename="terminal"
vercel inspect [deployment-id or url]
```

Unique Options

These are options that only apply to the `vercel inspect` command.

Timeout

The `--timeout` option sets the time to wait for deployment completion. It defaults to 3 minutes.

Any valid time string for the [ms](https://www.npmjs.com/package/ms) package can be used.

```bash filename="terminal"
vercel inspect https://example-app-6vd6bhoqt.vercel.app --timeout=5m
```

Wait

The `--wait` option will block the CLI until the specified deployment has completed.

```bash filename="terminal"
vercel inspect https://example-app-6vd6bhoqt.vercel.app --wait
```

Logs

The `--logs` option, shorthand `-l`, prints the build logs instead of the deployment information.

```bash filename="terminal"
vercel inspect https://example-app-6vd6bhoqt.vercel.app --logs
```

If the deployment is queued or canceled, there will be no logs to display.

If the deployment is building, you may want to specify `--wait` option. The command will wait for build completion, and will display buil

```bash filename="terminal"
vercel inspect https://example-app-6vd6bhoqt.vercel.app --logs --wait
```

---
title: "vercel install"
description: "Learn how to install native integrations with the vercel install CLI command."
last_updated: "2026-01-16T02:19:27.065Z"
source: "https://vercel.com/docs/cli/install"
---

vercel install

The `vercel install` command is used to install a [native integration](/docs/integrations/create-integration#native-integrations) with th

If you have not installed the integration before, you will asked to open the Vercel dashboard and accept the Vercel Marketplace terms. Yo

If you have an existing installation with the provider, you can add a product directly from the CLI by answering a series of questions th

Usage

```bash filename="terminal"
vercel install acme
```

You can get the value of `acme` by looking at the slug of the integration provider from the marketplace URL. For example, for `https://ve
```

---
title: "vercel integration"
description: "Learn how to perform key integration tasks using the vercel integration CLI command."
last_updated: "2026-01-16T02:19:27.068Z"
source: "https://vercel.com/docs/cli/integration"
---

vercel integration

The `vercel integration` command needs to be used with one of the following actions:

- `vercel integration add`
- `vercel integration open`
- `vercel integration list`
- `vercel integration remove`

For the `integration-name` in all the commands below, use the [URL slug](/docs/integrations/create-integration/submit-integration#url-slug

vercel integration add

The `vercel integration add` command initializes the setup wizard for creating an integration resource.
This command is used when you want to add a new resource from one of your installed integrations.
This functionality is the same as `vercel install [integration-name]`.

> **💡 Note:** If you have not installed the integration for the resource or accepted the
> terms & conditions of the integration through the web UI, this command will
> open your browser to the Vercel dashboard and start the installation flow for
> that integration.

```bash filename="terminal"
vercel integration add [integration-name]
```

vercel integration open

The `vercel integration open` command opens a deep link into the provider's dashboard for a specific integration. It's useful when you ne

```bash filename="terminal"
vercel integration open [integration-name]
```

vercel integration list

The `vercel integration list` command displays a list of all installed resources with their associated integrations for the current team

```bash filename="terminal"
vercel integration list
```

The output shows the name, status, product, and integration for each installed resource.

Options:

Option	Shorthand	Description
`--integration`	`-i`	Filter resources to a specific integration
`--all`	`-a`	List all resources regardless of project

Examples:

```bash filename="terminal"
# List all resources for the current project
vercel integration list

# Filter resources to a specific integration
vercel integration list --integration neon
vercel integration list -i upstash

# List all resources across all projects in the team
vercel integration list --all
vercel integration list -a
```

vercel integration remove

The `vercel integration remove` command uninstalls the specified integration from your Vercel account. It's useful in automation workflow

```bash filename="terminal"
vercel integration remove [integration-name]
```

> **💡 Note:** You are required to [remove all installed
> resources](/docs/cli/integration-resource#vercel-integration-resource-remove)
> from this integration before using this command.

---
title: "vercel integration-resource"
description: "Learn how to perform native integration product resources tasks using the vercel integration-resource CLI command."
last_updated: "2026-01-16T02:19:27.075Z"
source: "https://vercel.com/docs/cli/integration-resource"
---

vercel integration-resource

The `vercel integration-resource` command (alias: `vercel ir`) needs to be used with one of the following actions:

- `vercel integration-resource remove`
- `vercel integration-resource disconnect`

For the `resource-name` in all the commands below, use the [URL slug](/docs/integrations/create-integration#create-product-form-details) '

vercel integration-resource remove

The `vercel integration-resource remove` command (alias: `rm`) deletes an integration resource permanently.

```bash filename="terminal"
vercel integration-resource remove [resource-name]
```

Options:

Option	Shorthand	Description
`--disconnect-all`	`-a`	Disconnect all projects before deletion
`--yes`	`-y`	Skip the confirmation prompt

Examples:

```bash filename="terminal"
# Remove a resource
vercel integration-resource remove my-database

# Remove with alias
vercel ir rm my-cache

# Disconnect all projects and remove
vercel ir remove my-database --disconnect-all

# Remove without confirmation
vercel ir rm my-cache -a -y
```

vercel integration-resource disconnect

The `vercel integration-resource disconnect` command disconnects a resource from a project or from all projects.

```bash filename="terminal"
vercel integration-resource disconnect [resource-name] [project-name]
```

Arguments:

Argument	Required	Description
resource-name	Yes	Name or ID of the resource to disconnect
project-name	No	Project to disconnect from (uses linked project if omitted)

Options:

Option	Shorthand	Description
`--all`	`-a`	Disconnect all projects from the resource
`--yes`	`-y`	Skip the confirmation prompt

Examples:

```bash filename="terminal"
# Disconnect from linked project
vercel integration-resource disconnect my-database

# Using alias
vercel ir disconnect my-redis-cache

# Disconnect from a specific project
vercel ir disconnect my-database my-project

# Disconnect all projects from the resource
vercel ir disconnect my-database --all

# Disconnect all without confirmation
vercel ir disconnect my-database -a -y
```

---
title: "vercel link"
description: "Learn how to link a local directory to a Vercel Project using the vercel link CLI command."
last_updated: "2026-01-16T02:19:27.078Z"
source: "https://vercel.com/docs/cli/link"
---

vercel link

The `vercel link` command links your local directory to a [Vercel Project](/docs/projects/overview).

Usage

```bash filename="terminal"
vercel link
```

Extended Usage

```bash filename="terminal"
vercel link [path-to-directory]
```

Unique Options

These are options that only apply to the `vercel link` command.

Repo

The `--repo` option can be used to link all projects in your repository to their respective Vercel projects in one command. This command

```bash filename="terminal"
vercel link --repo
```

Yes

The `--yes` option can be used to skip questions you are asked when setting up a new Vercel Project.
The questions will be answered with the default scope and current directory for the Vercel Project name and location.

```bash filename="terminal"
vercel link --yes
```

Project

The `--project` option can be used to specify a project name. In non-interactive usage, `--project`
allows you to set a project name that does not match the name of the current working directory.

```bash filename="terminal"
vercel link --yes --project foo
```

---
title: "vercel list"
description: "Learn how to list out all recent deployments for the current Vercel Project using the vercel list CLI command."
last_updated: "2026-01-16T02:19:27.087Z"
source: "https://vercel.com/docs/cli/list"
---

vercel list

The `vercel list` command, which can be shortened to `vercel ls`, provides a list of recent deployments for the currently-linked Vercel P

Usage

```bash filename="terminal"
vercel list
```

Extended Usage

```bash filename="terminal"
vercel list [project-name]
```

```bash filename="terminal"
vercel list [project-name] [--status READY,BUILDING]
```

```bash filename="terminal"
vercel list [project-name] [--meta foo=bar]
```

```bash filename="terminal"
vercel list [project-name] [--policy errored=6m]
```

Unique Options

These are options that only apply to the `vercel list` command.

Meta

The `--meta` option, shorthand `-m`, can be used to filter results based on Vercel deployment metadata.

```bash filename="terminal"
vercel list --meta key1=value1 key2=value2
```

A common use case is filtering by the Git commit SHA that created a deployment:

```bash filename="terminal"
vercel ls -m githubCommitSha=de8b89f13b2bc164cf07e735921bf5513e17951d
```

> **💡 Note:** To see the meta values for a deployment, use [GET /deployments/{idOrUrl}
>](https://vercel.com/docs/rest-api/reference/endpoints/deployments/get-a-deployment-by-id-or-url).

Policy

The `--policy` option, shorthand `-p`, can be used to display expiration based on [Vercel project deployment retention policy](/docs/secu

```bash filename="terminal"
vercel list --policy canceled=6m -p errored=6m -p preview=6m -p production=6m
```

Yes

The `--yes` option can be used to skip questions you are asked when setting up a new Vercel Project.
The questions will be answered with the default scope and current directory for the Vercel Project name and location.

```bash filename="terminal"
vercel list --yes
```

Status

The `--status` option, shorthand `-s`, can be used to filter deployments by their status.

```bash filename="terminal"
vercel list --status READY
```

You can filter by multiple status values using comma-separated values:

```bash filename="terminal"
vercel list --status READY,BUILDING
```

The supported status values are:

- `BUILDING` - Deployments currently being built
- `ERROR` - Deployments that failed during build or runtime
- `INITIALIZING` - Deployments in the initialization phase
- `QUEUED` - Deployments waiting to be built
- `READY` - Successfully deployed and available
- `CANCELED` - Deployments that were canceled before completion

environment

Use the `--environment` option to list the deployments for a specific environment. This could be production, preview, or a [custom enviro

```bash filename="terminal"
vercel list my-app --environment=staging
```

Next

The `--next` option enables pagination when listing deployments. Pass the timestamp (in milliseconds since the UNIX epoch) from a previou

```bash filename="terminal"
vercel list --next 1584722256178
```

Prod

The `--prod` option filters the list to show only production deployments.

```bash filename="terminal"
vercel list --prod
```

---
title: "vercel login"
description: "Learn how to login into your Vercel account using the vercel login CLI command."
last_updated: "2026-01-16T02:19:27.096Z"
source: "https://vercel.com/docs/cli/login"
---

vercel login

The `vercel login` command allows you to login to your Vercel account through Vercel CLI.

Usage

```bash filename="terminal"
vercel login
```

Related guides

- [Why is Vercel CLI asking me to log in?](/kb/guide/why-is-vercel-cli-asking-me-to-log-in)

---
title: "vercel logout"
description: "Learn how to logout from your Vercel account using the vercel logout CLI command."
last_updated: "2026-01-16T02:19:27.164Z"
source: "https://vercel.com/docs/cli/logout"
---

vercel logout

The `vercel logout` command allows you to logout of your Vercel account through Vercel CLI.

Usage

```bash filename="terminal"
vercel logout
```

---
title: "vercel logs"
description: "Learn how to list out all runtime logs for a specific deployment using the vercel logs CLI command."
last_updated: "2026-01-16T02:19:27.178Z"
source: "https://vercel.com/docs/cli/logs"
---

vercel logs

The `vercel logs` command displays and follows runtime logs data for a specific deployment.
[Runtime logs](/docs/runtime-logs) are produced by [Middleware](/docs/routing-middleware) and [Vercel Functions](/docs/functions).
You can find more detailed runtime logs on the [Logs](/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Flogs\&title=Open+Logs) page from the Vercel D

From the moment you run this command, all newly emitted logs will display in your terminal, for up to 5 minutes, unless you interrupt it.

Logs are pretty-printed by default, but you can use the `--json` option to display them in JSON format, which makes the output easier to

Usage

```bash filename="terminal"
vercel logs [deployment-url | deployment-id]
```

Unique options

These are options that only apply to the `vercel logs` command.

Json

The `--json` option, shorthand `-j`, changes the format of the logs output from pretty print to JSON objects.
This makes it possible to pipe the output to other command-line tools, such as [jq](https://jqlang.github.io/jq/), to perform your own fi

```bash filename="terminal"
vercel logs [deployment-url | deployment-id] --json | jq 'select(.level == "warning")'
```

Follow

> **💡 Note:** The option has been deprecated since it's
> now the default behavior.

The `--follow` option, shorthand `-f`, can be used to watch for additional logs output.

Limit

> **💡 Note:** The option has been deprecated as the command
> displays all newly emitted logs by default.

The `--limit` option, shorthand `-n`, can be used to specify the number of log lines to output.

Output

> **💡 Note:** The option has been deprecated in favor of
> the option.

The `--output` option, shorthand `-o`, can be used to specify the format of the logs output, this can be either `short` (default) or `raw

Since

> **💡 Note:** The option has been deprecated. Logs are
> displayed from when you started the command.

The `--since` option can be used to return logs only after a specific date, using the ISO 8601 format.

Until

> **💡 Note:** The option has been deprecated. Logs are
> displayed until the command is interrupted, either by you or after 5 minutes.

The `--until` option can be used to return logs only up until a specific date, using the ISO 8601 format.

---
title: "vercel mcp"
description: "Set up Model Context Protocol (MCP) usage with a Vercel project using the vercel mcp CLI command."
last_updated: "2026-01-16T02:19:27.198Z"
source: "https://vercel.com/docs/cli/mcp"
---

vercel mcp

The `vercel mcp` command helps you set up an MCP client to talk to MCP servers you deploy on Vercel. It links your local MCP client confi

Usage

```bash filename="terminal"
vercel mcp [options]
```

Examples

Initialize global MCP configuration

```bash filename="terminal"
vercel mcp
```

Initialize project-specific MCP access

```bash filename="terminal"
vercel mcp --project
```

Unique options

These are options that only apply to the `vercel mcp` command.

Project

The option sets up project-specific MCP access for the currently linked project instead of global configuration.

```bash filename="terminal"
vercel mcp --project
```

---
title: "vercel microfrontends"
description: "Manage microfrontends configuration from the CLI. Learn how to pull configuration for local development."
last_updated: "2026-01-16T02:19:27.202Z"
source: "https://vercel.com/docs/cli/microfrontends"
---

vercel microfrontends

The `vercel microfrontends` command (alias:) provides utilities for working with Vercel Microfrontends from the CLI.

Currently, it supports pulling the remote configuration to your local repository for development.

> **💡 Note:** To learn more about the architecture and config format, see
> .
> For a polyrepo setup walkthrough, see
> .
> This command requires Vercel CLI 44.2.2 or newer.

Usage

```bash filename="terminal"
vercel microfrontends pull [options]
```

Unique options

These are options that only apply to the `vercel microfrontends` command.

Deployment

Use the option to specify a deployment ID or URL
to pull configuration from. If omitted, the CLI uses your project's default
application/deployment.

```bash filename="terminal"
vercel microfrontends pull --dpl https://my-app-abc123.vercel.app
```

Examples

Pull configuration for the linked project

```bash filename="terminal"
vercel microfrontends pull
```

Pull configuration for a specific deployment

```bash filename="terminal"
vercel mf pull --dpl dpl_123xyz
```

---
title: "vercel open"
description: "Learn how to open your current project in the Vercel Dashboard using the vercel open CLI command."
last_updated: "2026-01-16T02:19:27.217Z"
source: "https://vercel.com/docs/cli/open"
---

vercel open

The `vercel open` command opens your current project in the Vercel Dashboard. It automatically opens your default browser to the project'

> **💡 Note:** This command is available in Vercel CLI v48.10.0 and later. If you're using an older version, see [Updating Vercel CLI](/d

This command requires your directory to be [linked to a Vercel project](/docs/cli/project-linking). If you haven't linked your project ye

Usage

```bash filename="terminal"
vercel open
```

How it works

When you run `vercel open`:

1. The CLI checks if your current directory is linked to a Vercel project
2. It retrieves the project information, including the team slug and project name
3. It constructs the dashboard URL for your project
4. It opens the URL in your default browser

The command opens the project's main dashboard page at `https://vercel.com/{team-slug}/{project-name}`, where you can view deployments, c

Examples

Open the current project

From a linked project directory:

```bash filename="terminal"
vercel open
```

This opens your browser to the project's dashboard page.

Troubleshooting

Project not linked

If you see an error that the command requires a linked project:

```bash filename="terminal"
# Link your project first
vercel link

# Then open it
vercel open
```

Make sure you're in the correct directory where your project files are located.

Related

- [vercel link](/docs/cli/link)
- [vercel project](/docs/cli/project)
- [Project Linking](/docs/cli/project-linking)

---
title: "Vercel CLI Overview"
description: "Learn how to use the Vercel command-line interface (CLI) to manage and configure your Vercel Projects from the command line
last_updated: "2026-01-16T02:19:27.135Z"
source: "https://vercel.com/docs/cli"

Vercel CLI Overview

Vercel gives you multiple ways to interact with and configure your Vercel Projects. With the command-line interface (CLI) you can interac

If you'd like to interface with the platform programmatically, check out the [REST API documentation](/docs/rest-api).

Installing Vercel CLI

To download and install Vercel CLI, run the following command:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i vercel
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i vercel
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i vercel
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i vercel
    ```
  </Code>
</CodeBlock>

Updating Vercel CLI

When there is a new release of Vercel CLI, running any command will show you a message letting you know that an update is available.

If you have installed our command-line interface through [npm](http://npmjs.org/) or [Yarn](https://yarnpkg.com), the easiest way to upda

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i vercel
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i vercel
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i vercel
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i vercel
    ```
  </Code>
</CodeBlock>

If you see permission errors, please read npm's [official guide](https://docs.npmjs.com/resolving-eacces-permissions-errors-when-installi

Checking the version

The `--version` option can be used to verify the version of Vercel CLI currently being used.

```bash filename="terminal"
vercel --version
```

Using in a CI/CD environment

Vercel CLI requires you to log in and authenticate before accessing resources or performing administrative tasks. In a terminal environme

Available Commands

---
title: "vercel project"
description: "Learn how to list, add, remove, and manage your Vercel Projects using the vercel project CLI command."
last_updated: "2026-01-16T02:19:27.227Z"
source: "https://vercel.com/docs/cli/project"
---

vercel project

The `vercel project` command is used to manage your Vercel Projects, providing functionality to list, add, inspect, and remove.

Usage

```bash filename="terminal"
vercel project ls

# Output as JSON
vercel project ls --json
```

```bash filename="terminal"
vercel project ls --update-required

# Output as JSON
vercel project ls --update-required --json
```

```bash filename="terminal"
vercel project add
```

```bash filename="terminal"
vercel project inspect
```

```bash filename="terminal"
vercel project inspect my-project
```

```bash filename="terminal"
vercel project rm
```

---
title: "Linking Projects with Vercel CLI"
description: "Learn how to link existing Vercel Projects with Vercel CLI."
last_updated: "2026-01-16T02:19:27.161Z"
source: "https://vercel.com/docs/cli/project-linking"

Linking Projects with Vercel CLI

When running `vercel` in a directory for the first time, Vercel CLI needs to know which [scope](/docs/dashboard-features#scope-selector)
want to [deploy](/docs/cli/deploy) your directory to. You can choose to either [link](/docs/cli/link) an existing Vercel Project or to cr

```bash filename="terminal"
vercel
? Set up and deploy "~/web/my-lovely-project"? [Y/n] y
? Which scope do you want to deploy to? My Awesome Team
? Link to existing project? [y/N] y
? What's the name of your existing project? my-lovely-project
🔗 Linked to awesome-team/my-lovely-project (created .vercel and added it to .gitignore)
```

Once set up, a new `.vercel` directory will be added to your directory. The `.vercel` directory contains
both the organization and `id` of your Vercel Project. If you want [unlink](/docs/cli/link) your directory, you can remove the `.vercel`

You can use the [`--yes` option](/docs/cli/deploy#yes) to skip these questions.

Framework detection

When you create a new Vercel Project, Vercel CLI will [link](/docs/cli/link) the Vercel Project and automatically detect the framework yo
default Project Settings accordingly.

```bash filename="terminal"
vercel
? Set up and deploy "~/web/my-new-project"? [Y/n] y
? Which scope do you want to deploy to? My Awesome Team
? Link to existing project? [y/N] n
? What's your project's name? my-new-project
? In which directory is your code located? my-new-project/
```

Auto-detected project settings (Next.js):
- Build Command: \`next build\` or \`build\` from \`package.json\`
- Output Directory: Next.js default
- Development Command: next dev --port $PORT
? Want to override the settings? [y/N]
```

You will be provided with default **Build Command**, **Output Directory**, and **Development Command** options.

You can continue with the default Project Settings or overwrite them. You can also edit your Project Settings later in your Vercel Projec

## Relevant commands

- [deploy](/docs/cli/deploy)
- [link](/docs/cli/link)

---
title: "vercel promote"
description: "Learn how to promote an existing deployment using the vercel promote CLI command."
last_updated: "2026-01-16T02:19:27.225Z"
source: "https://vercel.com/docs/cli/promote"
---

# vercel promote

The `vercel promote` command is used to promote an existing deployment to be the current deployment.

> **⚠️ Warning:** Deployments built for the Production environment are the typical promote
> target. You can promote Deployments built for the Preview environment, but you
> will be asked to confirm that action and will result in a new production
> deployment. You can bypass this prompt by using the `--yes` option.

## Usage

```bash filename="terminal"
vercel promote [deployment-id or url]
```

Commands

`status`

Show the status of any current pending promotions.

```bash filename="terminal"
vercel promote status [project]
```

Examples:

```bash filename="terminal"
# Check status for the linked project
vercel promote status

# Check status for a specific project
vercel promote status my-project

# Check status with a custom timeout
vercel promote status --timeout 30s
```

Unique Options

These are options that only apply to the `vercel promote` command.

Timeout

The `--timeout` option is the time that the `vercel promote` command will wait for the promotion to complete. When a timeout occurs, it d

When promoting a deployment, a timeout of `0` will immediately exit after requesting the promotion. The default timeout is `3m`.

```bash filename="terminal"
vercel promote https://example-app-6vd6bhoqt.vercel.app --timeout=5m
```

---
title: "vercel pull"
description: "Learn how to update your local project with remote environment variables using the vercel pull CLI command."
last_updated: "2026-01-16T02:19:27.221Z"
source: "https://vercel.com/docs/cli/pull"
---

vercel pull

The `vercel pull` command is used to store [Environment Variables](/docs/environment-variables) and Project Settings in a local cache (un

When environment variables or project settings are updated on Vercel, remember to use `vercel pull` again to update your local environmen

> **💡 Note:** To download [Environment Variables](/docs/environment-variables) to a specific
> file (like `.env`), use [`vercel env
>   pull`](/docs/cli/env#exporting-development-environment-variables)
> instead.

Usage

```bash filename="terminal"
vercel pull
```
```

```bash filename="terminal"
vercel pull --environment=preview
```

```bash filename="terminal"
vercel pull --environment=preview --git-branch=feature-branch
```

```bash filename="terminal"
vercel pull --environment=production
```

Unique Options

These are options that only apply to the `vercel pull` command.

Yes

The `--yes` option can be used to skip questions you are asked when setting up a new Vercel Project.
The questions will be answered with the default scope and current directory for the Vercel Project name and location.

```bash filename="terminal"
vercel pull --yes
```

environment

Use the `--environment` option to define the environment you want to pull environment variables from. This could be production, preview,

```bash filename="terminal"
vercel pull --environment=staging
```

---
title: "vercel redeploy"
description: "Learn how to redeploy your project using the vercel redeploy CLI command."
last_updated: "2026-01-16T02:19:27.282Z"
source: "https://vercel.com/docs/cli/redeploy"
---

vercel redeploy

The `vercel redeploy` command is used to rebuild and [redeploy an existing deployment](/docs/deployments/managing-deployments).

Usage

```bash filename="terminal"
vercel redeploy [deployment-id or url]
```

Standard output usage

When redeploying, `stdout` is always the Deployment URL.

```bash filename="terminal"
vercel redeploy https://example-app-6vd6bhoqt.vercel.app > deployment-url.txt
```

Standard error usage

If you need to check for errors when the command is executed such as in a CI/CD workflow,
use `stderr`. If the exit code is anything other than `0`, an error has occurred. The
following example demonstrates a script that checks if the exit code is not equal to 0:

```bash filename="check-redeploy.sh"
# save stdout and stderr to files
vercel redeploy https://example-app-6vd6bhoqt.vercel.app >deployment-url.txt 2>error.txt

# check the exit code
code=$?
if [ $code -eq 0 ]; then
    # Now you can use the deployment url from stdout for the next step of your workflow
    deploymentUrl=`cat deployment-url.txt`
    echo $deploymentUrl
else
    # Handle the error
    errorMessage=`cat error.txt`
    echo "There was an error: $errorMessage"
fi
```

Unique Options

These are options that only apply to the `vercel redeploy` command.

No Wait

The `--no-wait` option does not wait for a deployment to finish before exiting from the `redeploy` command.

```bash filename="terminal"
vercel redeploy https://example-app-6vd6bhoqt.vercel.app --no-wait
```

target

Use the `--target` option to define the environment you want to redeploy to. This could be production, preview, or a [custom environment]

```bash filename="terminal"
```

```
vercel redeploy https://example-app-6vd6bhoqt.vercel.app --target=staging
```

---
title: "vercel redirects"
description: "Learn how to manage project-level redirects using the vercel redirects CLI command."
last_updated: "2026-01-16T02:19:27.287Z"
source: "https://vercel.com/docs/cli/redirects"
---

vercel redirects

The `vercel redirects` command lets you manage redirects for a project. Redirects managed at the project level apply to all deployments a

> **💡 Note:** Redirects can also be defined and managed in source control using
> `vercel.json`. Project-level redirects are updated without a need for a new
> deployment.

Usage

```bash filename="terminal"
vercel redirects list
```

Commands

The `vercel redirects` command includes several subcommands for managing redirects:

`list`

List all redirects for the current project. These redirects apply to all deployments and environments.

```bash filename="terminal"
vercel redirects list [options]
```

Options:

- `--page <NUMBER>`: Page number to display
- `--per-page <NUMBER>`: Number of redirects per page (default: 50)
- `-s, --search <QUERY>`: Search for redirects by source or destination
- `--staged`: List redirects from the staging version
- `--version <VERSION_ID>`: List redirects from a specific version ID

Examples:

```bash filename="terminal"
# List all redirects
vercel redirects list

# Search for redirects
vercel redirects list --search "/old-path"

# List redirects on page 2
vercel redirects list --page 2

# List redirects with custom page size
vercel redirects list --per-page 25

# List redirects from staging version
vercel redirects list --staged

# List redirects from a specific version
vercel redirects list --version ver_abc123
```

`list-versions`

List all versions of redirects for the current project.

```bash filename="terminal"
vercel redirects list-versions
```

`add`

Add a new redirect to your project.

```bash filename="terminal"
vercel redirects add [source] [destination] [options]
```

Options:

- `--case-sensitive`: Make the redirect case sensitive
- `--name <NAME>`: Version name for this redirect (max 256 characters)
- `--preserve-query-params`: Preserve query parameters when redirecting
- `--status <CODE>`: HTTP status code (301, 302, 307, or 308)
- `-y, --yes`: Skip prompts and use default values

Examples:

```bash filename="terminal"
# Add a new redirect interactively
vercel redirects add

# Add a new redirect with arguments
```

```
vercel redirects add /old-path /new-path

# Add a redirect with all options
vercel redirects add /old-path /new-path --status 301 --case-sensitive --preserve-query-params --name "My redirect"

# Add a redirect non-interactively
vercel redirects add /old-path /new-path --yes
```

`upload`

Upload redirects from a CSV or JSON file.

```bash filename="terminal"
vercel redirects upload file [options]
```

Options:

- `--overwrite`: Replace all existing redirects
- `-y, --yes`: Skip confirmation prompt

Examples:

```bash filename="terminal"
# Upload redirects from CSV file
vercel redirects upload redirects.csv

# Upload redirects from JSON file
vercel redirects upload redirects.json

# Upload and overwrite existing redirects
vercel redirects upload redirects.csv --overwrite

# Upload without confirmation
vercel redirects upload redirects.csv --yes
```

File Formats

CSV Format:

```csv filename="redirects.csv"
source,destination,status,caseSensitive,preserveQueryParams
/old-path,/new-path,301,false,true
/legacy/*,/modern/:splat,308,false,false
/old-blog,/blog,302,false,false
```

JSON Format:

```json filename="redirects.json"
[
  {
    "source": "/old-path",
    "destination": "/new-path",
    "status": 301,
    "caseSensitive": false,
    "preserveQueryParams": true
  },
  {
    "source": "/legacy/*",
    "destination": "/modern/:splat",
    "status": 308,
    "caseSensitive": false,
    "preserveQueryParams": false
  }
]
```

`remove`

Remove a redirect from your project.

```bash filename="terminal"
vercel redirects remove source [options]
```

Options:

- `-y, --yes`: Skip the confirmation prompt when removing a redirect

Example:

```bash filename="terminal"
# Remove a redirect
vercel redirects remove /old-path
```

`promote`

Promote a staged redirects version to production.

```bash filename="terminal"
vercel redirects promote version-id [options]
```

Options:

- `-y, --yes`: Skip the confirmation prompt when promoting

Example:

```bash filename="terminal"
# Promote a redirect version
vercel redirects promote <version-id>
```

`restore`

Restore a previous redirects version.

```bash filename="terminal"
vercel redirects restore version-id [options]
```

Options:

- `-y, --yes`: Skip the confirmation prompt when restoring

Example:

```bash filename="terminal"
# Restore a redirects version
vercel redirects restore <version-id>
```

---
title: "vercel remove"
description: "Learn how to remove a deployment using the vercel remove CLI command."
last_updated: "2026-01-16T02:19:27.295Z"
source: "https://vercel.com/docs/cli/remove"
---

vercel remove

The `vercel remove` command, which can be shortened to `vercel rm`, is used to remove deployments either by ID or for a specific Vercel P

> **💡 Note:** You can also remove deployments from the Project Overview page on the Vercel
> Dashboard.

Usage

```bash filename="terminal"
vercel remove [deployment-url]
```

Extended Usage

```bash filename="terminal"
vercel remove [deployment-url-1 deployment-url-2]
```

```bash filename="terminal"
vercel remove [project-name]
```

> **💡 Note:** By using the [project name](/docs/projects/overview/), the entire Vercel
> Project will be removed from the current scope unless the
> is used.

Unique Options

These are options that only apply to the `vercel remove` command.

Safe

The `--safe` option, shorthand `-s`, can be used to skip the removal of deployments with an active preview URL or production domain when

```bash filename="terminal"
vercel remove my-project --safe
```

Yes

The `--yes` option, shorthand `-y`, can be used to skip the confirmation step for a deployment or Vercel Project removal.

```bash filename="terminal"
vercel remove my-deployment.com --yes
```

---
title: "vercel rollback"
description: "Learn how to roll back your production deployments to previous deployments using the vercel rollback CLI command."
last_updated: "2026-01-16T02:19:27.299Z"
source: "https://vercel.com/docs/cli/rollback"
---

vercel rollback

The `vercel rollback` command is used to [roll back production deployments](/docs/instant-rollback) to previous deployments.

Usage

```bash filename="terminal"
vercel rollback [deployment-id or url]
```

> **💡 Note:** On the hobby plan, you can only [roll
> back](/docs/instant-rollback#who-can-roll-back-deployments) to the previous
> production deployment. If you attempt to pass in a deployment id or url from
> an earlier deployment, you will be given an error:.

Commands

`status`

Show the status of any current pending rollbacks.

```bash filename="terminal"
vercel rollback status [project]
```

Examples:

```bash filename="terminal"
# Check status for the linked project
vercel rollback status

# Check status for a specific project
vercel rollback status my-project

# Check status with a custom timeout
vercel rollback status --timeout 30s
```

Unique Options

These are options that only apply to the `vercel rollback` command.

Timeout

The `--timeout` option is the time that the `vercel rollback` command will wait for the rollback to complete. It does not affect the actu

When rolling back a deployment, a timeout of `0` will immediately exit after requesting the rollback.

```bash filename="terminal"
vercel rollback https://example-app-6vd6bhoqt.vercel.app
```

---
title: "vercel rolling-release"
description: "Learn how to manage your project"
last_updated: "2026-01-16T02:19:27.303Z"
source: "https://vercel.com/docs/cli/rolling-release"
---

vercel rolling-release

The `vercel rolling-release` command (also available as `vercel rr`) is used to manage your project's rolling releases. [Rolling releases

Usage

```bash filename="terminal"
vercel rolling-release [command]
```

Commands

configure

Configure rolling release settings for a project.

```bash filename="terminal"
vercel rolling-release configure --cfg='{"enabled":true, "advancementType":"manual-approval", "stages":[{"targetPercentage":10},{"targetP
```

start

Start a rolling release for a specific deployment.

```bash filename="terminal"
vercel rolling-release start --dpl=dpl_abc
```

Options:

Option	Type	Required	Description
`--dpl`	String	Yes	The deployment ID or URL to target
`--yes`	Boolean	No	Skip confirmation prompt

Examples:

```bash filename="terminal"
vercel rr start --dpl=dpl_123abc456def
vercel rr start --dpl=https://my-project-abc123.vercel.app
vercel rr start --dpl=dpl_123 --yes
```

approve

Approve the current stage of an active rolling release.

```bash filename="terminal"
vercel rolling-release approve --dpl=dpl_abc --currentStageIndex=0
```

abort

Abort an active rolling release.

```bash filename="terminal"
vercel rolling-release abort --dpl=dpl_abc
```

complete

Complete an active rolling release, promoting the deployment to 100% of traffic.

```bash filename="terminal"
vercel rolling-release complete --dpl=dpl_abc
```

fetch

Fetch details about a rolling release.

```bash filename="terminal"
vercel rolling-release fetch
```

Unique Options

These are options that only apply to the `vercel rolling-release` command.

Configuration

The `--cfg` option is used to configure rolling release settings. It accepts a JSON string or the value ``'disable'`` to turn off rolling r

```bash filename="terminal"
vercel rolling-release configure --cfg='{"enabled":true, "advancementType":"automatic", "stages":[{"targetPercentage":10,"duration":5},{"
```

Deployment

The `--dpl` option specifies the deployment ID or URL for rolling release operations.

```bash filename="terminal"
vercel rolling-release start --dpl=https://example.vercel.app
```

Current Stage Index

The `--currentStageIndex` option specifies the current stage index when approving a rolling release stage.

```bash filename="terminal"
vercel rolling-release approve --currentStageIndex=0 --dpl=dpl_123
```

Examples

Configure a rolling release with automatic advancement

```bash filename="terminal"
vercel rolling-release configure --cfg='{"enabled":true, "advancementType":"automatic", "stages":[{"targetPercentage":10,"duration":5},{"
```

This configures a rolling release that starts at 10% traffic, automatically advances after 5 minutes, and then goes to 100%.

Configure a rolling release with manual approval

```bash filename="terminal"
vercel rolling-release configure --cfg='{"enabled":true, "advancementType":"manual-approval","stages":[{"targetPercentage":10},{"targetPe
```

This configures a rolling release that starts at 10% traffic and requires manual approval to advance to 100%.

Configure a multi-stage rolling release

```bash filename="terminal"
vercel rolling-release configure --cfg='{"enabled":true, "advancementType":"manual-approval", "stages":[{"targetPercentage":10},{"targetP
```

This configures a rolling release with three stages: 10%, 50%, and 100% traffic, each requiring manual approval.

Disable rolling releases

```bash filename="terminal"
vercel rolling-release configure --cfg='disable'
```

This disables rolling releases for the project.

---
title: "vercel switch"
description: "Learn how to switch between different team scopes using the vercel switch CLI command."
last_updated: "2026-01-16T02:19:27.305Z"
source: "https://vercel.com/docs/cli/switch"
---

vercel switch

The `vercel switch` command is used to switch to a different team scope when logged in with Vercel CLI. You can choose to select a team f

Usage

```bash filename="terminal"
vercel switch
```

Extended Usage

```bash filename="terminal"
vercel switch [team-name]
```

---
title: "vercel target"
description: "Work with custom environments using the --target flag in Vercel CLI."
last_updated: "2026-01-16T02:19:27.308Z"
source: "https://vercel.com/docs/cli/target"
---

vercel target

The `vercel target` command (alias: `vercel targets`) manages your Vercel project's targets (custom environments). Targets are custom dep

Usage

```bash filename="terminal"
vercel target list
```

Commands

list (ls)

List all targets defined for the current project.

```bash filename="terminal"
vercel target list
vercel target ls
vercel targets ls
```

Using the --target flag

The `--target` flag is available on several commands to specify which environment to target:

```bash filename="terminal"
# Deploy to a custom environment named "staging"
vercel deploy --target=staging
```

Examples

List all targets

```bash filename="terminal"
vercel target list
```

Deploy to a custom environment

```bash filename="terminal"
vercel deploy --target=staging
```

Pull environment variables for a custom environment

```bash filename="terminal"
vercel pull --environment=staging
```

Set and use environment variables for a custom environment

```bash filename="terminal"
vercel env add MY_KEY staging
vercel env ls staging
```

Related

-
-
-

---
title: "vercel teams"
description: "Learn how to list, add, remove, and manage your teams using the vercel teams CLI command."
last_updated: "2026-01-16T02:19:27.311Z"
source: "https://vercel.com/docs/cli/teams"
---

vercel teams

The `vercel teams` command is used to manage [Teams](/docs/accounts/create-a-team), providing functionality to list, add, and invite new

> **💡 Note:** You can manage Teams with further options and greater control from the Vercel
> Dashboard.

Usage

```bash filename="terminal"
vercel teams list
```

Extended Usage

```bash filename="terminal"
vercel teams add
```

```bash filename="terminal"
vercel teams invite [email]
```

---
title: "vercel telemetry"
description: "Learn how to manage telemetry collection."
last_updated: "2026-01-16T02:19:27.313Z"
source: "https://vercel.com/docs/cli/telemetry"
---

vercel telemetry

The `vercel telemetry` command allows you to enable or disable telemetry collection.

Usage

```bash filename="terminal"
vercel telemetry status
```

```bash filename="terminal"
vercel telemetry enable
```

```bash filename="terminal"
vercel telemetry disable
```

---
title: "vercel whoami"
description: "Learn how to display the username of the currently logged in user with the vercel whoami CLI command."
last_updated: "2026-01-16T02:19:27.315Z"
source: "https://vercel.com/docs/cli/whoami"
---

vercel whoami

The `vercel whoami` command is used to show the username of the user currently logged into [Vercel CLI](/cli).

Usage

```bash filename="terminal"
vercel whoami
```

---
title: "Code Owners changelog"
description: "Find out what"
last_updated: "2026-01-16T02:19:27.321Z"
source: "https://vercel.com/docs/code-owners/changelog"

Code Owners changelog

Upgrade instructions

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i @vercel-private/code-owners
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @vercel-private/code-owners
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @vercel-private/code-owners
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i @vercel-private/code-owners
    ```
  </Code>
</CodeBlock>

Releases

`1.0.7`

This patch adds support for underscores in usernames and team slugs to match Github.

`1.0.6`

This patch updates the minimum length of Github username to match Github's validation.

`1.0.5`

This patch updates some dependencies for performance and security.

`1.0.4`

This patch updates some dependencies for performance and security.

`1.0.3`

This patch updates some dependencies for performance and security, and fixes an
issue where CLI output was colorless in GitHub Actions.

`1.0.2`

This patch updates some dependencies for performance and security.

`1.0.1`

This patch delivers improvements to our telemetry. While these improvements
are not directly user-facing, they enhance our ability to monitor and optimize
performance.

`1.0.0`

Initial release of Code Owners.

---
title: "vercel-code-owners"
description: "Learn how to use Code Owners with the CLI."
last_updated: "2026-01-16T02:19:27.326Z"
source: "https://vercel.com/docs/code-owners/cli"
---

vercel-code-owners

The `vercel-code-owners` command provides functionality to initialize and validate
Code Owners in your repository.

Using the CLI

The Code Owners CLI is separate to the [Vercel CLI](/docs/cli). However you
must ensure that the Vercel CLI is
[installed](/docs/cli#installing-vercel-cli) and that you are [logged
in](/docs/cli/login) to use the Code Owners CLI.

Sub-commands

The following sub-commands are available for this CLI.

`init`

The `init` command sets up code owners files in the repository. See
[Getting Started](/docs/code-owners/getting-started#initalizing-code-owners) for more information on
using this command.

`validate`

The `validate` command checks the syntax for all Code Owners files in the
repository for errors.

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i
    ```
  </Code>
</CodeBlock>

---
title: "Code Approvers"
description: "Use Code Owners to define users or teams that are responsible for directories and files in your codebase"
last_updated: "2026-01-16T02:19:27.348Z"
source: "https://vercel.com/docs/code-owners/code-approvers"

Code Approvers

Code Approvers are a list of [GitHub usernames or teams](https://docs.github.com/en/organizations/organizing-members-into-teams/about-tea

You can enable Code Approvers for a directory by adding a `.vercel.approvers` file to that directory in your codebase. For example, this

```sh copy filename="packages/design/.vercel.approvers"
@vercel/ui-team
```

When a team is declared as an approver, all members of that team will be able to approve changes to the directory or file and at least on

Enforcing Code Approvals

Code Approvals by the correct owners are enforced through the `Vercel – Code Owners` GitHub check added by the Vercel GitHub App.

When a pull request is opened, the GitHub App will check if the pull request contains changes to a directory or file that has Code Approv

If no Code Approvers are defined for the changes then the check will pass. Otherwise, the check will fail until the correct Code Approver

To make Code Owners required, follow the [GitHub required status checks](https://docs.github.com/en/pull-requests/collaborating-with-pull

Inheritance

Code Approvers are inherited from parent directories. If a directory does not have a `.vercel.approvers` file, then the approvers from th
Furthermore, even if a directory does have a `.vercel.approvers` file, then the approvers from a parent directory with a `.vercel.approve
This structure allows the most specific approver to review most of the code, but allows other approvers who have broader context and appr

To illustrate the inheritance, the following example has two `.vercel.approvers` files.

The first file defines owners for the `packages/design` directory. The `@vercel/ui-team` can approve any change to a file under `packages

```sh copy filename="packages/design/.vercel.approvers"
@vercel/ui-team
```

A second `.vercel.approvers` file is declared at the root of the codebase and allows users `elmo` and `oscar` to approve changes to any p

```sh copy filename=".vercel.approvers"
@elmo
@oscar
```

The hierarchical nature of Code Owners enables many configurations in larger codebases, such as allowing individuals to approve cross-cut

Reviewer Selection

When a pull request is opened, the Vercel GitHub App will select the approvers for the changed files.
`.vercel.approvers` files allow extensive definitions of file mappings to possible approvers. In many cases, there will be multiple appro
The Vercel GitHub app selects the best reviewers for the pull request based on affinity of `.vercel.approvers` definitions and overall co

Bypassing Reviewer Selection

You can skip automatic assignment of reviewers by adding `vercel:skip:owners` to your pull request description.

To request specific reviewers, you can override the automatic selection by including special text in your pull request description:

```text copy
[vercel:approver:@owner1]
[vercel:approver:@owner2]
```

Code Owners will still ensure that the appropriate code owners have approved the pull request before it can pass. Therefore, make sure to

Modifiers

Modifiers enhance the behavior of Code Owners by giving more control over the behavior of approvals and reviewer selection. The available

- [silent](#silent)
- [notify](#notify)
- [optional](#optional)
- [team](#team)
- [members](#members-default)
  - [not](#excluding-team-members-from-review)
- [required](#required)

Modifiers are appended to the end of a line to modify the behavior of the owner listed for that line:

```sh copy filename=".vercel.approvers"
# Approver with no modifier
@owner1
# Approver with optional modifier
@owner2:optional
```

`silent`

The user or team is an owner for the provided code but is never requested for review. If the user is a non-silent approver in another `.v

```sh copy filename=".vercel.approvers"
# This person will never be requested to review code but can still approve for owners coverage.
@owner:silent
```

`notify`

The user or team is always notified through a comment on the pull request. These owners may still be requested for review as part of [rev

```sh copy filename=".vercel.approvers"
# my-team is always notified even if leerob is selected as the reviewer.
@vercel/my-team:notify
@leerob
```

`optional`

The user or team is never requested for review, and they are ignored as owners when computing review requirements. The owner can still ap

This can be useful while in the process of adding code owners to an existing repository or when you want to designate an owner for a dire

```sh copy filename=".vercel.approvers"
@owner:optional
```

`members` (default)

The `:members` modifier can be used with GitHub teams to select an individual member of the team for reviewer rather than assigning it to

```sh copy filename=".vercel.approvers"
# An individual from the @acme/eng-team will be requested as a reviewer.
@acme/eng-team:members
```

Excluding team members from review

The `:not` modifier can be used with `:members` to exclude certain individuals on the team from review. This can be useful when there is

```sh copy filename=".vercel.approvers"
# An individual from the @acme/eng-team, except for leerob will be requested as a reviewer.
@acme/eng-team:members:not(leerob)
# Both leerob and mknichel will not be requested for review.
@acme/eng-team:members:not(leerob):not(mknichel)
```

`team`

The `:team` modifier can be used with GitHub teams to request the entire team for review instead of individual members from the team. Thi

```sh copy filename=".vercel.approvers"
# The @acme/eng-team will be requested as a reviewer.
@acme/eng-team:team
```

`required`

This user or team is always notified (through a comment) and is a required approver on the pull request regardless of the approvals cover

> **💡 Note:** Most of the time you don't need to specify required approvers. Non-modified
> approvers are usually enough so that correct reviews are enforced.

```sh copy filename=".vercel.approvers"
# Always notifed and are required reviewers.
# The check won't pass until both `owner1` and `owner2` approve.
@owner1:required
@owner2:required
```

When you specify a team as a required reviewer only one member of that team is required to approve.

```sh copy filename=".vercel.approvers"
# The team is notifed and are required reviewers.
# The check won't pass until one member of the team approves.
@vercel/my-team:required
```

Patterns

The `.vercel.approvers` file supports specifying files with a limited set of glob patterns:

- [Directory](#directory-default)
- [Current Directory](#current-directory-pattern)
- [Globstar](#globstar-pattern)
- [Specifying multiple owners](#specifying-multiple-owners-for-the-same-pattern)

The patterns are case-insensitive.

Directory (default)

The default empty pattern represents ownership of the current directory and all subdirectories.

```sh copy filename=".vercel.approvers"
# Matches all files in the current directory and all subdirectories.
@owner
```

Current Directory Pattern

A pattern that matches a file or set of files in the current directory.

```sh copy filename=".vercel.approvers"
# Matches the single `package.json` file in the current directory only.
package.json @package-owner

# Matches all javascript files in the current directory only.
*.js @js-owner
```

Globstar Pattern

The globstar pattern begins with `**/`. And represents ownership of files matching the glob in the current directory and its subdirectori

```sh copy filename=".vercel.approvers"
# Matches all `package.json` files in the current directory and its subdirectories.
**/package.json @package-owner
```

```
# Matches all javascript files in the current directory and its subdirectories.
**/*.js @js-owner
```

Code Owners files are meant to encourage distributed ownership definitions
across a codebase. Thus, the globstar `**/` and `/` can only be used at the
start of a pattern. They cannot be used in the middle of a pattern to enumerate
subdirectories.

For example, the following patterns are not allowed:

```sh copy filename=".vercel.approvers"
# Instead add a `.vercel.approvers` file in the `src` directory.
src/**/*.js @js-owner

# Instead add a `.vercel.approvers` file in the `src/pages` directory.
src/pages/index.js @js-owner
```

Specifying multiple owners for the same pattern

Each owner for the same pattern should be specified on separate lines. All
owners listed will be able to approve for that pattern.

```sh copy filename=".vercel.approvers"
# Both @package-owner and @org/team will be able to approve changes to the
# package.json file.
package.json @package-owner
package.json @org/team
```

Wildcard Approvers

If you would like to allow a certain directory or file to be approved by anyone, you can use the wildcard owner `*`. This is useful for f

```sh copy filename=".vercel.approvers"
# Changes to the `pnpm-lock.yaml` file in the current directory can be approved by anyone.
pnpm-lock.yaml *

# Changes to any README in the current directory or its subdirectories can be approved by anyone.
**/readme.md *

```

---
title: "Getting Started with Code Owners"
description: "Learn how to set up Code Owners for your codebase."
last_updated: "2026-01-16T02:19:27.365Z"
source: "https://vercel.com/docs/code-owners/getting-started"

Getting Started with Code Owners

To [set up Code Owners](#setting-up-code-owners-in-your-repository) in your repository, you'll need to do the following:

- Set up [Vercel's private npm registry](/docs/private-registry) to install the necessary packages
- [Install and initialize](#setting-up-code-owners-in-your-repository) Code Owners in your repository
- [Add your repository](#adding-your-repository-to-the-vercel-dashboard) to your Vercel dashboard

If you've already set up Conformance, you may have already completed some of these steps.

Prerequisites

Get access to Code Owners

To enable Code Owners for your Enterprise team, you'll need to request access through your Vercel account administrator.

Setting up Vercel's private npm registry

Vercel distributes packages with the `@vercel-private` scope through our private npm registry, and requires that each user using the pack

To use the private npm registry, you'll need to follow the documentation to:

- [Set up your local environment](/docs/private-registry#setting-up-your-local-environment) – This should be completed by the team owner,
- [Set up Vercel](/docs/private-registry#setting-up-vercel) – This should be completed by the team owner
- [Set up Code Owners for use with CI](/docs/private-registry#setting-up-your-ci-provider) – This should be completed by the team owner

Setting up Code Owners in your repository

A GitHub App enables Code Owners functionality by adding reviewers and
enforcing review checks for merging PRs.

- ### Set up the Vercel CLI
  The Code Owners CLI is separate to the [Vercel CLI](/docs/cli), however it uses
  the Vercel CLI for authentication.

  Before continuing, please ensure that the Vercel CLI is [installed](/docs/cli#installing-vercel-cli)
  and that you are [logged in](/docs/cli/login).

- ### Initalizing Code Owners
  If you have an existing `CODEOWNERS` file in your repository, you can use the CLI to automatically migrate your repository to use Verce

  Start by running this command in your repository's root:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```

```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>
  > **⚠️ Warning:** `yarn dlx` only works with Yarn version 2 or newer, for Yarn v1 use the npx
  > command.
  After running, check the installation success by executing:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>

- ### Install the GitHub App into a repository
  To install, you must be an organization owner or have the GitHub App Manager permissions.
  1. Go to https://github.com/apps/vercel/installations/new
  2. Choose your organization for the app installation.
  3. Select repositories for the app installation.
  4. Click `Install` to complete the app installation in the chosen repositories.

- ### Define Code Owners files
  After installation, define Code Owners files in your repository. Pull requests
  with changes in specified directories will automatically have reviewers added.

  Start by adding a `.vercel.approvers` file in a directory
  in your repository. List GitHub usernames or team names in the
  file, each on a new line:
  ```text copy filename=".vercel.approvers"
  @username1
  @org/team1
  ```
  Then, run the [`validate`](/docs/code-owners/cli#validate) command to check the syntax and merge your changes into your repository:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>

- ### Test Code Owners on a new pull request
  With the `.vercel.approvers` file merged into the main branch, test the flow by modifying
  any file within the same or child directory. Create a pull request as usual, and the system
  will automatically add one of the listed users as a reviewer.

- ### Add the Code Owners check as required
  This step is optional

  By default, GitHub checks are optional and won't block merging. To make the Code Owners
  check mandatory, go to `Settings > Branches > [Edit] > Require status checks to pass before merging` in your repository settings.

Adding your repository to the Vercel dashboard

Adding your repository to your team's Vercel [dashboard](/dashboard), allows you to access the Conformance dashboard and see an overview
```

- ### Import your repository
  1. Ensure your team is selected in the [scope selector](/docs/dashboard-features#scope-selector).
  2. From your [dashboard](/dashboard), select the **Add New** button and from the dropdown select **Repository**.
  3. Then, from the **Add a new repository** screen, find your Git repository that you wish to import and select **Connect**.

- ### Configure your repository
  Before you can connect a repository, you must ensure that the Vercel GitHub app has been [installed for your team](https://docs.github.

  Once installed, you'll be able to connect your repository.

More resources

- [Code Owners CLI](/docs/code-owners/cli)
- [Conformance](/docs/conformance)

---
title: "Code Owners"
description: "Use Code Owners to define users or teams that are responsible for directories and files in your codebase"
last_updated: "2026-01-16T02:19:27.376Z"
source: "https://vercel.com/docs/code-owners"

Code Owners

As a company grows, it can become difficult for any one person to be familiar with the entire codebase. As growing teams start to special

- **Colocated owners files**: Owners files live right next to the code, making it straightforward to find who owns a piece of code right
- **Mirrored organization dynamics**: **Code Owners** mirrors the structure of your organization. Code owners who are higher up in the di
- **Customizable code review algorithms**: **Modifiers** allow organizations to tailor their code review process to their needs. For exam

Get Started

Code Owners is only available for use with GitHub.

To get started with Code Owners, follow the instructions on the
[Getting Started](/docs/code-owners/getting-started) page.

Code Approvers

Code Approvers are a list of [GitHub usernames or teams](https://docs.github.com/en/organizations/organizing-members-into-teams/about-tea

You can enable Code Approvers by adding a `.vercel.approvers` file to a directory in your codebase. To learn more about how the code appr

---
title: "Enabling and Disabling Comments"
description: "Learn when and where Comments are available, and how to enable and disable Comments at the account, project, and session or
last_updated: "2026-01-16T02:19:27.447Z"
source: "https://vercel.com/docs/comments/how-comments-work"

Enabling and Disabling Comments

Comments are enabled by default for all preview deployments on all new projects. **By default, only members of [your Vercel team](/docs/a

> **💡 Note:** The comments toolbar will only render on sites with **HTML** set as the
> `Content-Type`. Additionally, on Next.js sites, the comments toolbar will only
> render on Next.js pages and **not** on API routes or static files.

At the account level

You can enable or disable comments at the account level with certain permissions:

1. Navigate to [your Vercel dashboard](/dashboard) and make sure that you have selected your team from the [scope selector](/docs/dashboa
2. From your [dashboard](/dashboard), select the **Settings** tab.
3. In the **General** section, find **Vercel Toolbar**.
4. Under each environment (**Preview** and **Production**), select either **On** or **Off** from the dropdown to determine the visibility
5. You can optionally choose to allow the setting to be overridden at the project level.

At the project level

1. From your [dashboard](/dashboard), select the project you want to enable or disable Vercel Toolbar for.
2. Navigate to **Settings** tab.
3. In the **General** section, find **Vercel Toolbar**.
4. Under each environment (**Preview** and **Production**), select either an option from the dropdown to determine the visibility of Verc
   - **Default**: Respect team-level visibility settings.
   - **On**: Enable the toolbar for the environment.
   - **Off**: Disable the toolbar for the environment.

At the session or interface level

To disable comments for the current browser session, you must [disable the toolbar](/docs/vercel-toolbar/managing-toolbar#disable-toolbar

With environment variables

You can enable or disable comments for specific branches or environments with [preview environment variables](/docs/vercel-toolbar/managi

See [Managing the toolbar](/docs/vercel-toolbar/managing-toolbar) for more information.

In production and localhost

To use comments in a production deployment, or link comments in your local development environment to a preview deployment, see [our docs

See [Managing the toolbar](/docs/vercel-toolbar/managing-toolbar) for more information.

Sharing

To learn how to share deployments with comments enabled, see the [Sharing Deployments](/docs/deployments/sharing-deployments) docs.

---
title: "Integrations for Comments"
description: "Learn how Comments integrates with Git providers like GitHub, GitLab, and BitBucket, as well as Vercel"
last_updated: "2026-01-16T02:19:27.398Z"
source: "https://vercel.com/docs/comments/integrations"

Integrations for Comments

Git provider integration

Comments are available for projects using **any** Git provider. Github, BitBucket and GitLab [are supported automatically](/docs/git#supp

Pull requests (PRs) with deployments enabled receive [generated PR messages from Vercel bot](/docs/git/vercel-for-github). These PR messa

The generated PR message will also display an **Add your feedback** URL, which lets people visit the deployment and automatically log in.

Vercel will also add a check to PRs with comments enabled. This check reminds the author of any unresolved comments, and **is not require

To make this check required, check the docs for your favorite Git provider. Docs on required checks for the most popular git providers ar

- [GitHub](https://docs.github.com/en/repositories/configuring-branches-and-merges-in-your-repository/defining-the-mergeability-of-pull-r
- [BitBucket](https://support.atlassian.com/bitbucket-cloud/docs/suggest-or-require-checks-before-a-merge/)
- [GitLab](https://docs.gitlab.com/ee/user/project/merge_requests/status_checks.html#block-merges-of-merge-requests-unless-all-status-che

Vercel CLI deployments

Commenting is available for deployments made with [the Vercel CLI](/docs/cli). The following git providers are supported for comments wit

- GitHub
- GitLab
- BitBucket

See [the section on Git provider integration information](#git-provider-integration) to learn more.

Commenting is available in production and localhost when you use [the Vercel Toolbar package](/docs/vercel-toolbar/in-production-and-loca

Use the Vercel Slack app

The [Vercel Slack app](https://vercel.com/integrations/slack) connects Vercel deployments to Slack channels. Any new activity will create

To get started:

1. Go to [our Vercel Slack app in the Vercel Integrations Marketplace](https://vercel.com/integrations/slack)
2. Select the **Add Integration** button from within the Marketplace, then select which Vercel account and project the integration should
3. Confirm the installation by selecting the **Add Integration** button
4. From the pop-up screen, you'll be prompted to provide permission to access your Slack workspace. Select the **Allow** button
5. In the new pop-up screen, select the **Connect your Vercel account to Slack** button. When successful, the button will change to text

> **💡 Note:** Private Slack channels will not appear in the dropdown list when setting up
> the Slack integration unless you have already invited the Vercel app to the
> channel. Do so by sending `/invite @Vercel` as a message to the channel.

Linking Vercel and Slack users

1. In any channel on your Team's Slack instance enter `/vercel login`
2. Select **Continue with Vercel** to open a new browser window
3. From the new browser window, select **Authorize Vercel to Slack**
4. Once the connection is successful, you'll receive a "Successfully authenticated" message in the Slack channel.
5. You can use `/vercel whoami` at any time to check that you're successfully linked

Linking Slack and Vercel does the following:

- Allows Vercel to translate `@` mentions across messages/platforms
- Allows you to take extra actions
- Allows user replies to be correctly attributed to their Vercel user instead of a `slack-{slackusername}` user when replying in a thread

Updating your Slack integration

If you configured the Slack app before October 4th, 2023, the updated app requires new permissions. You must reconfigure the app to subsc

To do so:

1. Visit your team's dashboard and select the **Integrations** tab
2. Select **Manage** next to Slack in your list of integrations. On the next page, select **Configure**
3. Configure your Slack app and re-authorize it

> **💡 Note:** Your previous linked channels and subscriptions will continue to work even if
> you don't reconfigure the app in Slack.

Connecting a project to a Slack channel

To see a specific project's comments in a Slack channel, send the following command as a message to the channel:

```bash
/vercel subscribe
```

This will open a modal that allows you to configure the subscription, including:

- Subscribing to comments for specific branches
- Subscribing to comments on specific pages

You can specify pages using a [glob pattern](#), and branches with regex, to match multiple options.

You can also configure your subscription with options when using the `/vercel subscribe` command. You can use the `/vercel help` command

Commenting in Slack

When a new comment is created on a PR, the Vercel Slack app will create a matching thread in each of the subscribed Slack channels. The f

- A link to the newly-created comment thread
- A preview of the text of the first comment in the thread
- A ✅ **Resolve** button near the bottom of the Slack post
  - You may resolve comment threads without viewing them
  - You may reopen resolved threads at any time

Replies and edits in either Slack or the original comment thread will be reflected on both platforms.

Your custom Slack emojis will also be available on linked deployments. Search for them by typing `:`, then inputting the name of the emoj

Use the following Slack command to list all available options for your Vercel Slack integration:

```bash
/vercel help
```

Receiving notifications as Slack DMs

To receive comment notifications as DMs from Vercel's Slack app, you must link your Vercel account in Slack by entering the following com

```bash
/vercel login
```

Vercel Slack app command reference

Command	Function
`/vercel help`	List all commands and options
`/vercel subscribe`	Subscribe using the UI interface
`/vercel subscribe team/project`	Subscribe the current Slack channel to a project
`/vercel subscribe list`	List all projects the current Slack channel is subscribed to
`/vercel unsubscribe team/project`	Unsubscribe the current Slack channel from a project
`/vercel whoami`	Check which account you're logged into the Vercel Slack app with
`/vercel logout`	Log out of your Vercel account
`/vercel login` (or `link` or `signin`)	Log into your Vercel account

Adding Comments to your issue tracker

Any member of your team can covert comments to an issue in Linear, Jira, or GitHub. This is useful for tracking bugs, feature requests, a

- ### Install the Vercel integration for your issue tracker
  The following issue trackers are supported:
  - [Linear](/integrations/linear)
  - [Jira Cloud](/integrations/jira)
  - [GitHub](/integrations/github)
  Once you open the integration, select the **Add Integration** button to install it. Select which Vercel team and project(s) the integra
  > **💡 Note:** On Jira, issues will be marked as reported by the user who converted the
  > thread and marked as created by the user who set up the integration. You may
  > want to consider using a dedicated account to connect the integration.

- ### Convert a comment to an issue
  On the top-right hand corner of a comment thread, select the icon for your issue tracker. A **Convert to Issue** dialog will appear.

  If you have more than one issue tracker installed, the most recently used issue tracker will appear on a comment. To select a different

- ### Fill out the issue details
  Fill out the relevant information for the issue. The issue description will be populated with the comment text and any images in the co

  The fields you will see are dependant on the issue tracker you use and the scope it has. When you are done, select **Create Issue**.

  Linear

  Users can set the team, project, and issue title. Only publicly available teams can be selected as Private Linear teams are not support

  Jira

  Users can set the project, issue type, and issue title.

  You can't currently convert a comment into a child issue. After converting a comment into an issue, you may assign it a parent issue in

  GitHub

  Users can set the repository and issue title. If you installed the integration to a Github Organization, there will be an optional fiel

- ### Confirm the issue was created
  Vercel will display a confirmation toast at the bottom-right corner of the page. You can click the toast to open the relevant issue in

  When you create an issue from a comment thread, Vercel will resolve the thread. The thread cannot be unresolved so we recommend only co

  Linear

  If the email on your Linear account matches the Vercel account and you follow a thread converted to an issue, you will be added as a su

  Jira

  On Jira, issues will be marked as *reported* by the user who converted the thread and marked as *created* by the user who set up the in

  GitHub

  The issue will be marked as created by the `vercel-toolbar` bot and will have a label generated based on the Vercel project it was conv

  If selected, the converted issue will be added to the project or board you selected when creating the issue.

---
title: "Managing Comments on Preview Deployments"
description: "Learn how to manage Comments on your Preview Deployments from Team members and invited collaborators."

last_updated: "2026-01-16T02:19:27.408Z"
source: "https://vercel.com/docs/comments/managing-comments"

Managing Comments on Preview Deployments

Resolve comments

You can resolve comments by selecting the **☐ Resolve** checkbox that appears under each thread or comment. You can access this checkbox

Participants in a thread will receive a notification when that thread is resolved.

Notifications

By default, the activity within a comment thread triggers a notification for all participants in the thread. PR owners will also receive

Activities that trigger a notification include:

- Someone creating a comment thread
- Someone replying in a comment thread you have enabled notifications for or participated in
- Someone resolving a comment thread you're receiving notifications for

Whenever there's new activity within a comment thread, you'll receive a new notification. Notifications can be sent to:

- [Your Vercel Dashboard](#dashboard-notifications)
- [Email](#email)
- [Slack](#slack)

Customizing notifications for deployments

To customize notifications for a deployment:

1. Visit the deployment
2. Log into the Vercel toolbar
3. Select the **Menu** button (☰)
4. Select **Preferences** (⚙)
5. In the dropdown beside **Notifications**, select:
   - **Never**: To disable notifications
   - **All**: To enable notifications
   - **Replies and Mentions**: To enable only some notifications

Customizing thread notifications

You can manage notifications for threads in the **Inbox**:

1. Select the three dots (ellipses) near the top of the first comment in a thread
2. Select **Unfollow** to mute the thread, or **Follow** to subscribe to the thread

Dashboard notifications

While logged into Vercel, select the notification bell icon and select the **Comments** tab to see new Comments notifications. To view sp

- **Filter based on**:
  - Author
  - Status
  - Project
  - Page
  - Branch
- **Search**: Search for comments containing specific text

> **💡 Note:** Comments left on pages with query params in the URL may not appear on the page
> when you visit the base URL. Filter by page and search with a `*` wildcard to
> see all pages with similar URLs. For example, you might search for
> `/docs/conformance/rules/req*`.

You can also resolve comments from your notifications.

To reply to a comment, or view the deployment it was made on, select it and select the link to the deployment.

Email

Email notifications will be sent to the email address associated with your Vercel account. Multiple notifications within a short period w

Slack

When you configure Vercel's Slack integration, comment threads on linked branches will create Slack threads. New activity on Slack or in

Troubleshooting comments

Sometimes, issues appear on a webpage for certain browsers and devices, but not for others. It's also possible for users to leave comment

To get around this issue, you can select the screen icon beside a commenter's name to copy their session info to your clipboard. Doing so

```json filename="session-data"
{
  "browserInfo": {
    "ua": "Mozilla/5.0 (Macintosh; Intel Mac OS X 9_10_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36",
    "browser": {
      "name": "Chrome",
      "version": "106.0.0.0",
      "major": "106"
    },
    "engine": {
      "name": "Blink",
      "version": "106.0.0.0"
    },
    "os": {
      "name": "Mac OS",
      "version": "10.15.7"
    },
```

```
    "device": {},
    "cpu": {}
  },
  "screenWidth": 1619,
  "screenHeight": 1284,
  "devicePixelRatio": 1.7999999523162842,
  "deploymentUrl": "vercel-site-7p6d5t8vq.vercel.sh"
}
```


On desktop, you can hover your cursor over a comment's timestamp to view less detailed session information at a glance, including:

- Browser name and version
- Window dimensions in pixels
- Device pixel ratio
- Which deployment they were viewing

---
title: "Comments Overview"
description: "Comments allow teams and invited participants to give direct feedback on preview deployments. Learn more about Comments in
last_updated: "2026-01-16T02:19:27.421Z"
source: "https://vercel.com/docs/comments"

Comments Overview

Comments allow teams [and invited participants](/docs/comments/how-comments-work#sharing) to give direct feedback on [preview deployments

Pull request owners receive emails when a new comment is created. Comment creators and participants in comment threads will receive email

When changes are pushed to a PR, and a new preview deployment has been generated, a popup modal in the bottom-right corner of the deploym

Comments are a feature of the [Vercel Toolbar](/docs/vercel-toolbar) and the toolbar must be active to see comments left on a page. You c

To leave a comment:

1. Open the toolbar menu and select **Comment** or the comment bubble icon in shortcuts.
2. Then, click on the page or highlight text to place your comment.

More resources

- [Enabling or Disabling Comments](/docs/comments/how-comments-work)
- [Using Comments](/docs/comments/using-comments)
- [Managing Comments](/docs/comments/managing-comments)
- [Comments Integrations](/docs/comments/integrations)
- [Using Comments in production and localhost](/docs/vercel-toolbar/in-production-and-localhost)

---
title: "Using Comments with Preview Deployments"
description: "This guide will help you get started with using Comments with your Vercel Preview Deployments."
last_updated: "2026-01-16T02:19:27.437Z"
source: "https://vercel.com/docs/comments/using-comments"

Using Comments with Preview Deployments

Add comments

You must be logged in to create a comment. You can press `c` to enable the comment placement cursor.

Alternatively, select the **Comment** option in the toolbar menu. You can then select a location to place your comment with your cursor.

Mention users

You can use `@` to mention team members and alert them to your comment. For example, you might want to request Jennifer's input by writing

Add emojis to a comment

You can add emojis by entering `:` (the colon symbol) into your comment input box, then entering the name of the emoji. For example, add

To add a reaction, select the emoji icon to the right of the name of the commenter whose comment you want to react to. You can then searc

> **💡 Note:** Custom emoji from your Slack organization are supported when you integrate the
> [Vercel Slack app](/docs/comments/integrations#use-the-vercel-slack-app).

Add screenshots to a comment

You can add screenshots to a comment in any of the following ways:

- Click the plus icon that shows when drafting a comment to upload a file.
- Click the camera icon to take a screenshot of the page you are on.
- Click and drag while in commenting mode to automatically screenshot a portion of the page and start a comment with it attached.

The latter two options are only available to users with the [browser extension](/docs/vercel-toolbar/in-production-and-localhost/add-to-p

Use Markdown in a comment

Markdown is a markup language that allows you to format text, and you can use it to make your comments more readable and visually pleasing

Supported formatting includes:

Supported markdown formatting options

Command	Keyboard Shortcut (Windows)	Keyboard Shortcut (Mac)	Example Input	Example Output
Bold	`Ctrl+B`	`⌘+B`	`*Bold text*`	**Bold text**
Italic	`Ctrl+I`	`⌘+I`	`_Italic text_`	*Italic text*
Strikethrough	`Ctrl+Shift+X`	`⌘+⇧+X`	`~Strikethrough text~`	~~Strikethrough text~~

| Code-formatted text | `Ctrl+E` | `⌘+E` | `` `Code-formatted text` `` | `Code-formatted text`
| Bulleted list | `-` or `*` | `-` or `*` | `- Item 1 - Item 2` | • Item 1 • Item 2
| Numbered list | `1.` | `1.` | `1. Item 1 2. Item 2` | 1. Item 1 2. Item 2
| Embedded links | N/A | N/A | `[A link](https://example.com)` | [A link](#supported-mar
| Quotes | `>` | `>` | `> Quote` | | Quote

Comment threads

Every new comment placed on a page begins a thread. The comment author, PR owner, and anyone participating in the conversation will see t

The Inbox can be opened by selecting the **Inbox** option in the toolbar menu. A small badge will indicate if any comments have been adde

You can move the **Inbox** to the left or right side of the screen by selecting the top of the Inbox modal and dragging it.

Thread filtering

You can filter threads by selecting the branch name at the top of the **Inbox**. A modal will appear, with the following filter options:

- **Filter by page**: Show comments across all pages in the inbox, or only those that appear on the page you're currently viewing
- **Filter by status**: Show comments in the inbox regardless of status, or either show resolved or unresolved

Copy comment links

You can copy a link to a comment in two ways:

- Select a comment in the **Inbox**. When you do, the URL will update with an anchor to the selected comment
- Select the ellipses (three dots) icon to the right of the commenter's name, then select the **Copy Link** option in the menu that pops

---
title: "Vercel CDN Compression"
description: "Vercel helps reduce data transfer and improve performance by supporting both Gzip and Brotli compression"
last_updated: "2026-01-16T02:19:27.523Z"
source: "https://vercel.com/docs/compression"

Vercel CDN Compression

Vercel helps reduce data transfer and improve performance by supporting both Gzip and Brotli compression. These algorithms are widely use

Compression algorithms

While `gzip` has been around for quite some time, `brotli` is a newer compression algorithm built by Google that best serves text compres

- `brotli` compressed JavaScript files are 14% smaller than `gzip`
- HTML files are 21% smaller than `gzip`
- CSS files are 17% smaller than `gzip`

`brotli` has an advantage over `gzip` since it uses a dictionary of common keywords on both the client and server-side, which gives a bet

Compression negotiation

Many clients (e.g., browsers like Chrome, Firefox, and Safari) include the `Accept-Encoding` [request header](https://developer.mozilla.o

You can verify if a response was compressed by checking the `Content-Encoding` [response header](https://developer.mozilla.org/en-US/docs

Clients that don't use `Accept-Encoding`

The following clients may not include the `Accept-Encoding` header by default:

- Custom applications, such as Python scripts, Node.js servers, or other software that can send HTTP requests to your deployment
- HTTP libraries, such as [`http`](https://nodejs.org/api/http.html) in Node.js, and networking tools, like `curl` or `wget`
- Older browsers. Check [MDN's browser compatibility list](https://developer.mozilla.org/docs/Web/HTTP/Headers/Accept-Encoding#browser_co
- Bots and crawlers sometimes do not specify `Accept-Encoding` in their headers by default when visiting your deployment

When writing a client that doesn't run in a browser, for example a CLI, you will need to set the `Accept-Encoding` request header in your

Automatically compressed MIME types

When the `Accept-Encoding` request header is present, only the following list of MIME types will be automatically compressed.

Application types

- `json`
- `x-web-app-manifest+json`
- `geo+json`
- `manifest+json`
- `ld+json`
- `atom+xml`
- `rss+xml`
- `xhtml+xml`
- `xml`
- `rdf+xml`
- `javascript`
- `tar`
- `vnd.ms-fontobject`
- `wasm`

Font types

- `otf`
- `ttf`

Image types

- `svg+xml`
- `bmp`
- `x-icon`

Text types

- `cache-manifest`
- `css`
- `csv`
- `dns`
- `javascript`
- `plain`
- `markdown`
- `vcard`
- `calendar`
- `vnd.rim.location.xloc`
- `vtt`
- `x-component`
- `x-cross-domain-policy`

Why doesn't Vercel compress all MIME types?

The compression allowlist above is necessary to avoid accidentally increasing the size of non-compressible files, which can negatively im

For example, most image formats are already compressed such as JPEG, PNG, WebP, etc. If you want to compress an image even further, consi

---
title: "Conformance Allowlists"
description: "Learn how to use allowlists to bypass your Conformance rules to merge changes into your codebase."
last_updated: "2026-01-16T02:19:27.585Z"
source: "https://vercel.com/docs/conformance/allowlist"

Conformance Allowlists

Conformance allowlists enable developers to integrate code into the codebase, bypassing specific Conformance rules when necessary. This h

Anatomy of an allowlist entry

An allowlist entry looks like the following:

```json filename="my-site/.allowlists"
{
  "testName": "NEXTJS_MISSING_SECURITY_HEADERS",
  "entries": [
    {
      "testName": "NEXTJS_MISSING_SECURITY_HEADERS",
      "reason": "TODO: This existed before the Conformance test was added but should be fixed.",
      "location": {
        "workspace": "dashboard",
        "filePath": "next.config.js"
      },
      "details": {
        "missingField": "headers"
      }
    }
  ]
}
```

The allowlist entry contains the following fields:

- `testName`: The name of the triggered test
- `needsResolution`: Whether the allowlist entry needs to be resolved
- `reason`: Why this code instance is allowed despite Conformance catching it
- `location`: The file path containing the error
- `details` (optionally): Details about the Conformance error

An allowlist entry will match an existing one when the `testName`, `location`,
and `details` fields all match. The `reason` is only used for documentation
purposes.

The `needsResolution` field

This field is used by the CLI and our metrics to assess if an allowlisted issue
is something that needs to be resolved. The default value is `true`. When set
to `false`, this issue is considered to be "accepted" by the team and will not
show up in future metrics.

As this field was added after the release of Conformance, the value of this
field is considered `true` when the field is missing from an allowlist entry.

Allowlists location

In a monorepo, Conformance allowlists are located in an `.allowlists/` directory
in the root directory of each workspace. For repository-wide rules, place allowlist entries in the top-level `.allowlists/` directory.

Allowlisting all errors

The Conformance CLI can add an allowlist entry for all the active errors. This
can be useful when adding a new entry to the allowlist for review, or when a
new check is being added to the codebase. To add an allowlist entry for all
active errors in a package:

From the package directory:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i
    ```

    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i
    ```
  </Code>
</CodeBlock>

From the root of a monorepo:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i
    ```
  </Code>
</CodeBlock>

Configuring Code Owners for Allowlists

You can use [Code Owners](/docs/code-owners) with allowlists for specific team reviews on updates. For instance, have the security team r

To configure Code Owners for all tests at the top level for the entire repository:

```text copy filename=".vercel.approvers"
**/*.allowlist.json @org/team:required
**/NO_CORS_HEADERS.* @org/security-team:required
```

For a specific workspace, add a `.vercel.approvers` file in the `.allowlists` sub-directory:

```text copy filename="apps/docs/.allowlists/.vercel.approvers"
NO_EXTERNAL_CSS_AT_IMPORTS.* @org/performance-team:required
```

The `:required` check ensures any modifications need the specified owners' review.

---
title: "Conformance changelog"
description: "Find out what"
last_updated: "2026-01-16T02:19:27.610Z"
source: "https://vercel.com/docs/conformance/changelog"

Conformance changelog

Upgrade instructions

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i @vercel-private/conformance
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @vercel-private/conformance
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @vercel-private/conformance
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i @vercel-private/conformance
    ```
  </Code>
</CodeBlock>

Releases

`1.12.3`

- Support for Turborepo v2 configuration

`1.12.2`
    ```

- Update dependencies listed in `THIRD_PARTY_LICENSES.md` file
- Update `NEXTJS_NO_CLIENT_DEPS_IN_MIDDLEWARE` rule to not treat `react` as just a client dependency

`1.12.1`

- Adds a `THIRD_PARTY_LICENSES.md` file listing third party licenses

`1.12.0`

- Update `NO_SERIAL_ASYNC_CALLS` rule to highlight the awaited call expression instead of the entire function

`1.11.0`

- Update rule logic for detecting duplicate allowlist entries based on the details field

`1.10.3`

This patch update has the following changes:

- Optimize checking allowlists for existing Conformance issues
- Isolate some work by moving it to a worker thread
- Fix error when trying to parse empty JavaScript/TypeScript files

`1.10.2`

This patch update has the following changes:

- Parse ESLint JSON config with a JSONC parser
- Fix retrieving latest version of CLI during `init`

`1.10.1`

This patch update has the following changes:

- Fix updating allowlist files when entries conflict or already exist

`1.10.0`

This minor update has the following changes:

- Replace [`NEXTJS_MISSING_MODULARIZE_IMPORTS`](/docs/conformance/rules/NEXTJS_MISSING_MODULARIZE_IMPORTS) Next.js rule with [`NEXTJS_MIS
- Fix showing error messages for rules
- Update allowlist entry details for [`REQUIRE_CARET_DEPENDENCIES`](/docs/conformance/rules/REQUIRE_CARET_DEPENDENCIES)

`1.9.0`

This minor update has the following changes:

- Ensure in-memory objects are cleaned up after each run
- Fix detection of Next.js apps in certain edge cases
- Bump dependencies for performance and security

`1.8.1`

This patch update has the following changes:

- Fix the init command for Yarn classic (v1)
- Update AST caching to prevent potential out of memory issues
- Fix requesting git authentication when sending Conformance metrics

`1.8.0`

This minor update has the following changes:

- Support non-numeric Node version numbers like `lts` in [`REQUIRE_NODE_VERSION_FILE`](/docs/conformance/rules/REQUIRE_NODE_VERSION_FILE)
- Add version range support for [`forbidden-packages`](/docs/conformance/custom-rules/forbidden-packages) custom rules.
- Updates dependencies for performance and security.

New rules:

- [`REQUIRE_DOCS_ON_EXPORTED_FUNCTIONS`](/docs/conformance/rules/REQUIRE_DOCS_ON_EXPORTED_FUNCTIONS).
  Requires that all exported functions have JSDoc comments.

`1.7.0`

This minor update captures and sends Conformance runs metrics to Vercel.
Your team will be able to view those metrics in the Vercel dashboard.

The following rules also include these fixes:

- [`NEXTJS_REQUIRE_EXPLICIT_DYNAMIC`](/docs/conformance/rules/NEXTJS_REQUIRE_EXPLICIT_DYNAMIC):
  Improved error messaging.
- [`NEXTJS_SAFE_NEXT_PUBLIC_ENV_USAGE`](/docs/conformance/rules/NEXTJS_SAFE_NEXT_PUBLIC_ENV_USAGE):
  Improved error messaging.

`1.6.0`

This minor update introduces multiple new rules, fixes and improvements for
existing rules and the CLI, and updates to some dependencies for performance
and security.

Notably, this release introduces a new `needsResolution` flag. This is used
by the CLI and will be used in future metrics as a mechanism to opt-out of
further tracking of this issue.

The following new rules have been added:

- [`NO_UNNECESSARY_PROP_SPREADING`](/docs/conformance/rules/NO_UNNECESSARY_PROP_SPREADING):
  Disallows the usage of object spreading in JSX components.

The following rules had fixes and improvements:

- [`REQUIRE_CARET_DEPENDENCIES`](/docs/conformance/rules/REQUIRE_CARET_DEPENDENCIES):
  Additional cases are now covered by this rule.
- [`NO_INSTANCEOF_ERROR`](/docs/conformance/rules/NO_INSTANCEOF_ERROR):
  Multiple issues in the same file are no longer reported as a single issue.
- [`NO_INLINE_SVG`](/docs/conformance/rules/NO_INLINE_SVG):
  Multiple issues in the same file are no longer reported as a single issue.
- [`REQUIRE_ONE_VERSION_POLICY`](/docs/conformance/rules/REQUIRE_ONE_VERSION_POLICY):
  Multiple issues in the same file are now differentiated by the package name
  and the location of the entry in `package.json`.

`1.5.0`

This minor update introduces a new rule and improvements to our telemetry.

The following new rules have been added:

- [`NO_INSTANCEOF_ERROR`](/docs/conformance/rules/NO_INSTANCEOF_ERROR):
  Disallows using `error instanceof Error` comparisons due to risk of false negatives.

`1.4.0`

This minor update introduces multiple new rules, fixes and improvements for
existing rules and the CLI, and updates to some dependencies for performance
and security.

The following new rules have been added:

- [`NEXTJS_SAFE_NEXT_PUBLIC_ENV_USAGE`](/docs/conformance/rules/NEXTJS_SAFE_NEXT_PUBLIC_ENV_USAGE):
  Requires allowlist entries for any usage of `NEXT_PUBLIC_*` environment variables.
- [`NO_POSTINSTALL_SCRIPT`](/docs/conformance/rules/NO_POSTINSTALL_SCRIPT):
  Prevents the use of `"postinstall"` script in package for performance reasons.
- [`REQUIRE_CARET_DEPENDENCIES`](/docs/conformance/rules/REQUIRE_CARET_DEPENDENCIES):
  Requires that all `dependencies` and `devDependencies` have a `^` prefix.

The following rules had fixes and improvements:

- [`PACKAGE_MANAGEMENT_REQUIRED_README`](/docs/conformance/rules/PACKAGE_MANAGEMENT_REQUIRED_README):
  Lowercase `readme.md` files are now considered valid.
- [`REQUIRE_NODE_VERSION_FILE`](/docs/conformance/rules/REQUIRE_NODE_VERSION_FILE):
  Resolved an issue preventing this rule from correctly reporting issues.
- [`NO_INLINE_SVG`](/docs/conformance/rules/NO_INLINE_SVG):
  Detection logic now handles template strings alongside string literals.
- The [`forbidden-imports`](/docs/conformance/custom-rules/forbidden-imports)
  custom rule type now supports `paths` being defined in [rule configuration](/docs/conformance/custom-rules/forbidden-imports#configurin

`1.3.0`

This minor update introduces new rules to improve Next.js app performance,
resolves an issue where TypeScript's `baseUrl` wasn't respected when traversing
files, and fixes an issue with dependency traversal which caused some rules to
return false positives in specific cases.

The following new rules have been added:

- [`NEXTJS_REQUIRE_EXPLICIT_DYNAMIC`](/docs/conformance/rules/NEXTJS_REQUIRE_EXPLICIT_DYNAMIC):
  Requires explicitly setting the `dynamic` route segment option for Next.js pages and routes.
- [`NO_INLINE_SVG`](/docs/conformance/rules/NO_INLINE_SVG):
  Prevents the use of `svg` tags inline, which can negatively impact the
  performance of both browser and server rendering.

`1.2.1`

This patch updates some Conformance dependencies for performance and security,
and improves handling of edge case for both [`NEXTJS_NO_ASYNC_LAYOUT`](/docs/conformance/rules/NEXTJS_NO_ASYNC_LAYOUT)
and [`NEXTJS_NO_ASYNC_PAGE`](/docs/conformance/rules/NEXTJS_NO_ASYNC_PAGE).

`1.2.0`

This minor update introduces a new rule, and improvements to both
`NEXTJS_NO_ASYNC_LAYOUT` and `NEXTJS_NO_ASYNC_PAGE`.

The following new rules have been added:

- [`REQUIRE_NODE_VERSION_FILE`](/docs/conformance/rules/REQUIRE_NODE_VERSION_FILE):
  Requires that workspaces have a valid Node.js version file (`.node-version` or `.nvmrc`) file defined.

`1.1.0`

This minor update introduces new rules to improve Next.js app performance,
enhancements to the CLI output, and improvements to our telemetry. While
telemetry improvements are not directly user-facing, they enhance our ability
to monitor and optimize performance.

The following new rules have been added:

- [`NEXTJS_NO_ASYNC_PAGE`](/docs/conformance/rules/NEXTJS_NO_ASYNC_PAGE):
  Ensures that the exported Next.js page component and its transitive dependencies are not asynchronous,
  as that blocks the rendering of the page.
- [`NEXTJS_NO_ASYNC_LAYOUT`](/docs/conformance/rules/NEXTJS_NO_ASYNC_LAYOUT):
  Ensures that the exported Next.js layout component and its transitive dependencies are not asynchronous,
  as that can block the rendering of the layout and the rest of the page.
- [`NEXTJS_USE_NATIVE_FETCH`](/docs/conformance/rules/NEXTJS_USE_NATIVE_FETCH):
  Requires using native `fetch` which Next.js polyfills, removing the need for
  third-party fetch libraries.
- [`NEXTJS_USE_NEXT_FONT`](/docs/conformance/rules/NEXTJS_USE_NEXT_FONT):
  Requires using `next/font` (when possible), which optimizes fonts for
  improved privacy and performance.

- [`NEXTJS_USE_NEXT_IMAGE`](/docs/conformance/rules/NEXTJS_USE_NEXT_IMAGE):
  Requires that `next/image` is used for all images for improved performance.
- [`NEXTJS_USE_NEXT_SCRIPT`](/docs/conformance/rules/NEXTJS_USE_NEXT_SCRIPT):
  Requires that `next/script` is used for all scripts for improved performance.

`1.0.0`

Initial release of Conformance.

---
title: "vercel-conformance"
description: "Learn how Conformance improves collaboration, productivity, and software quality at scale."
last_updated: "2026-01-16T02:19:27.615Z"
source: "https://vercel.com/docs/conformance/cli"
---

vercel-conformance

The `vercel-conformance` command is used to run
[Conformance](/docs/conformance) on your code.

Using the CLI

The Conformance CLI is separate to the [Vercel CLI](/docs/cli). However you
must ensure that the Vercel CLI is
[installed](/docs/cli#installing-vercel-cli) and that you are [logged
in](/docs/cli/login) to use the Conformance CLI.

Sub-commands

The following sub-commands are available for this CLI.

`audit`

The `audit` command runs Conformance on code without needing to install any NPM
dependencies or build any of the code. This is useful for viewing Conformance
results on a repository that you don't own and may not have permissions to
modify or build.

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i
    ```
  </Code>
</CodeBlock>

> **⚠️ Warning:** `yarn dlx` only works with Yarn version 2 or newer, for Yarn v1 use the npx
> command.

If you would like to store the results of the conformance audit in a file, you
can redirect `stderr` to a file:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i
    ```
  </Code>
</CodeBlock>

`init`

The `init` command installs Conformance in the repository. See
[Getting Started](/docs/conformance/getting-started#initialize-conformance) for more information on
using this command.

---

title: "forbidden-code"
description: "Learn how to set custom rules to disallow code and code patterns through string and regular expression matches."
last_updated: "2026-01-16T02:19:27.554Z"
source: "https://vercel.com/docs/conformance/custom-rules/forbidden-code"
---

forbidden-code

The `forbidden-code` rule type enables you to disallow code and code patterns through string and regular expression matches.

When to use this rule type

- **Disallowing comments**
  - You want to disallow `// TODO` comments
  - You want to disallow usage of `@ts-ignore`
- **Disallowing specific strings**
  - You want to enforce a certain casing for one or more strings
  - You want to disallow specific strings from being used within code

If you want to disallow specific operations on a property, you should instead
use the [`forbidden-properties`](/docs/conformance/custom-rules/forbidden-properties) rule type.

Configuring this rule type

To create a custom `forbidden-code` rule, you'll need to configure the below
required properties:

Property	Type	Description
`ruleType`	`"forbidden-code"`	The custom rule's type.
`ruleName`	`string`	The custom rule's name.
`categories`	`("nextjs" | "performance" | "security" | "code-health")[]` (optional)	The custom rule's categories. Default is `
`errorMessage`	`string`	The error message, which is shown to users
`errorLink`	`string` (optional)	An optional link to show alongside the err
`description`	`string` (optional)	The rule description, which is shown in th
`severity`	`"major" | "minor"` (optional)	The rule severity added to the allowlists
`patterns`	`(string | { pattern: string, flags: string })[]`	An array of regular expression patterns to
`strings`	`string[]`	An array of exact string to match against

> **⚠️ Warning:** Multi-line strings and patterns are currently unsupported by this custom rule
> type.

Example configuration

The example below configures a rule named `NO_DISALLOWED_USAGE` that disallows:

- Any usage of `"and"` at the start of a line (case-sensitive).
- Any usage of `"but"` in any case.
- Any usage of `"TODO"` (case-sensitive).

```jsonc copy filename="conformance.config.jsonc" {4-11}
{
  "customRules": [
    {
      "ruleType": "forbidden-imports",
      "ruleName": "NO_DISALLOWED_USAGE",
      "categories": ["code-health"],
      "errorMessage": "References to \"and\" at the start of a line are not allowed.",
      "description": "Disallows using \"and\" at the start of a line.",
      "severity": "major",
      "patterns": ["^and", { "pattern": "but", "flags": "i" }],
      "strings": ["TODO"],
    },
  ],
}
```

Using flags with patterns

This custom rule type always sets the `"g"` (or global) flag for regular
expressions. This ensures that all regular expression matches are reported,
opposed to only reporting on the first match.

When providing flags through an object in `patterns`, you can omit the `"g"` as
this will automatically be set.

To learn more about regular expression flags, see [the MDN guide](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_e

Writing patterns

If you're not familiar with regular expressions, you can use tools like
[regex101](https://regex101.com/) and/or [RegExr](https://regexr.com/) to help
you understand and write regular expressions.

Regular expressions can vary in complexity, depending on what you're trying to
achieve. We've added some examples below to help you get started.

Pattern	Description
`^and`	Matches `"and"`, but only if it occurs at the start of a line (`^`).
`(B|a)ar$`	Matches `"But"` and `"but"`, but only if it occurs at the end of a line (`$`).
`regexp?`	Matches `"regexp"` and `"regex"`, with or without the `"p"` (`?`).
`(?<!-)so`	Matches `"so"`, but only when not following a hyphen (`(?<!-)`).

Enabling this rule type

To enable this rule type, you can set the rule to `true`, or provide the
following configuration.

| Property | Type | Description

| -------- | ---------------------- | ---
| `paths` | `string[]` (optional) | An optional array of exact paths or glob expressions*. |

The example below enables the `NO_DISALLOWED_USAGE` custom rule for all files in the
`src/` directory, excluding files in `src/legacy/`. In this example, the custom
rule is also restricted to the `dashboard` and `marketing-site` workspaces,
which is optional.

```jsonc copy filename="conformance.config.jsonc" {4-9}
{
  "overrides": [
    {
      "restrictTo": {
        "workspaces": ["dashboard", "marketing-site"],
      },
      "rules": {
        "CUSTOM.NO_DISALLOWED_USAGE": {
          "paths": ["src", "!src/legacy"],
        },
      },
    },
  ],
  "customRules": [
    // ...
  ],
}
```

This next example enables the `NO_DISALLOWED_USAGE` custom rule for all files,
and without workspace restrictions.

```jsonc copy filename="conformance.config.jsonc" {4-6}
{
  "overrides": [
    {
      "rules": {
        "CUSTOM.NO_DISALLOWED_USAGE": true,
      },
    },
  ],
  "customRules": [
    // ...
  ],
}
```

;

---
title: "forbidden-dependencies"
description: "Learn how to set custom rules to disallow one or more files from depending on one or more predefined module"
last_updated: "2026-01-16T02:19:27.636Z"
source: "https://vercel.com/docs/conformance/custom-rules/forbidden-dependencies"
---

forbidden-dependencies

The `forbidden-dependencies` rule type enables you to disallow one or more files from depending on one or more predefined modules.

Unlike [`forbidden-imports`](/docs/conformance/custom-rules/forbidden-imports), this rule type will check for
indirect (or transitive) dependencies, where a module may not directly import
the disallowed dependency, but the disallowed dependency is present in the
dependency chain. This makes it slower, but more powerful than the
`forbidden-imports` rule type.

For example, below we have a `logger` utility that imports a package that may
cause security keys to be exposed.

```ts filename="src/utils/logger.ts"
import { SECURITY_KEY } from 'secret-package';
```

We can use this rule type to create a custom rule that prevents any module in
`src/app` from importing any file that depends on our potentially dangerous
`secret-package`.

```ts filename="src/app/page.ts"
import { log } from '../utils/logger';
// Would result in an error
```

When to use this rule type

- **Performance**
  - You want to prevent importing packages that are known to increase the size
    of your client side code
  - You want to prevent using a package that is known to perform poorly in
    specific environments
- **Security**
  - You want to disallow client-side code from depending on a file that
    exposes secrets
- **Error prevention**
  - You want to prevent errors by disallowing server-side code from importing
    a module where some methods require browser APIs

Configuring this rule type

To create a custom `forbidden-dependencies` rule, you'll need to configure
the required properties below:

Property	Type	Description
`ruleType`	`"forbidden-dependencies"`	The custom rule's type.
`ruleName`	`string`	The custom rule's name.
`categories`	`("nextjs" | "performance" | "security" | "code-health")[]` (optional)	The custom rule's categories. Defau
`errorMessage`	`string`	The error message, which is shown t
`errorLink`	`string` (optional)	An optional link to show alongside
`description`	`string` (optional)	The rule description, which is show
`severity`	`"major" | "minor"` (optional)	The rule severity added to the allo
`moduleNames`	`string[]`	An array of exact module names or g
`paths`	`string[]` (optional)	An optional array of exact paths or
`traverseNodeModules`	`boolean` (optional)	When `true`, this rule will also tr

> **⚠️ Warning:** When using `traverseNodeModules`, module names currently need to be prefixed
> with `node_modules` (i.e., `["disallowed", "node_modules/disallowed"]`). We're
> working to improve this.

Example configuration

The example below configures a rule named `NO_SUPER_SECRET_IN_CLIENT` that
disallows depending on any package from the `super-secret` workspace except for
`@super-secret/safe-exports`.

```jsonc copy filename="conformance.config.jsonc" {4-10}
{
  "customRules": [
    {
      "ruleType": "forbidden-dependencies",
      "ruleName": "NO_SUPER_SECRET_IN_CLIENT",
      "categories": ["code-health"],
      "errorMessage": "Depending on packages from the 'super-secret' workspace may result in secrets being exposed in client-side code. P
      "description": "Prevents depending on packages from the 'super-secret' workspace.",
      "severity": "major",
      "moduleNames": ["@super-secret/*", "!@super-secret/safe-exports"],
    },
  ],
}
```

Enabling this rule type

To enable this rule type, you can set the rule to `true`, or provide the
following configuration.

Property	Type	Description
`paths`	`string[]` (optional)	An optional array of exact paths or glob expressions, which restricts the paths that this custom rul

The example below enables the `NO_SUPER_SECRET_IN_CLIENT` custom rule for all
files in the `src/` directory, excluding test files. In this example, the
custom rule is also restricted to the `dashboard` and `marketing-site`
workspaces, which is optional.

```jsonc copy filename="conformance.config.jsonc" {4-10}
{
  "overrides": [
    {
      "restrictTo": {
        "workspaces": ["dashboard", "marketing-site"],
      },
      "rules": {
        "CUSTOM.NO_SUPER_SECRET_IN_CLIENT": {
          "paths": ["src", "!src/**/*.test.ts"],
        },
      },
    },
  ],
  "customRules": [
    // ...
  ],
}
```

This next example enables the `NO_SUPER_SECRET_IN_CLIENT` custom rule for all
files, and without workspace restrictions.

```jsonc copy filename="conformance.config.jsonc" {4-6}
{
  "overrides": [
    {
      "rules": {
        "CUSTOM.NO_SUPER_SECRET_IN_CLIENT": true,
      },
    },
  ],
  "customRules": [
    // ...
  ],
}
```

---
title: "forbidden-imports"
description: "Learn how to set custom rules to disallow one or more files from importing one or more predefined modules"
last_updated: "2026-01-16T02:19:27.655Z"
source: "https://vercel.com/docs/conformance/custom-rules/forbidden-imports"
---

forbidden-imports

The `forbidden-imports` rule type enables you to disallow one or more files from importing one or more predefined modules.

Unlike [`forbidden-dependencies`](/docs/conformance/custom-rules/forbidden-dependencies), this rule type won't
check for indirect (transitive) dependencies. This makes this rule faster, but
limits its effectiveness.

When to use this rule type

- **Deprecating packages or versions**
  - You want to disallow importing a deprecated package, and to recommend a
    different approach
- **Recommending an alternative package**
  - You want to require that users import custom/wrapped methods from
    `test-utils` instead of directly from a testing library

If you want to prevent depending on a module for performance or security
reasons, you should instead use the
[`forbidden-dependencies`](/docs/conformance/custom-rules/forbidden-dependencies) rule type.

Configuring this rule type

To create a custom `forbidden-imports` rule, you'll need to configure the below
required properties:

Property	Type	Description
`ruleType`	`"forbidden-imports"`	The custom rule's type.
`ruleName`	`string`	The custom rule's name.
`categories`	`("nextjs" | "performance" | "security" | "code-health")[]` (optional)	The custom rule's categories.
`errorMessage`	`string`	The error message, which is sh
`errorLink`	`string` (optional)	An optional link to show along
`description`	`string` (optional)	The rule description, which is
`severity`	`"major" | "minor"` (optional)	The rule severity added to the
`moduleNames`	`string[]`	An array of exact module names
`importNames`	`string[]` (optional)	An array of exact module names
`paths`	`string[]` (optional)	**Added in Conformance `1.4.0`
`disallowDefaultImports`	`boolean` (optional)	Flags default imports (i.e. `i
`disallowNamespaceImports`	`boolean` (optional)	Flags namespace imports (i.e.

Note that when using `moduleNames` alone, imports are not allowed at all from
that module. When used with conditions like `importNames`, the custom rule will
only report an error when those conditions are also met.

Example configuration

The example below configures a rule named `NO_TEAM_IMPORTS` that disallows
importing any package from the `team` workspace except for `@team/utils`. It also
configures a rule that disallows importing `oldMethod` from `@team/utils`, but
restricts that rule to the `src/new/` directory.

```jsonc copy filename="conformance.config.jsonc" {4-20}
{
  "customRules": [
    {
      "ruleType": "forbidden-imports",
      "ruleName": "NO_TEAM_IMPORTS",
      "categories": ["security"],
      "errorMessage": "Packages from the team workspace have been deprecated in favour of '@team/utils'.",
      "description": "Disallows importing packages from the team workspace.",
      "severity": "major",
      "moduleNames": ["@team/*", "!@team/utils"],
    },
    {
      "ruleType": "forbidden-imports",
      "ruleName": "NO_TEAM_OLD_METHOD_IMPORTS",
      "categories": ["performance"],
      "errorMessage": "'oldMethod' has been deprecated in favour of 'newMethod'.",
      "description": "Disallows using the deprecated method 'oldMethod' from '@team/utils'.",
      "severity": "minor",
      "moduleNames": ["@team/utils"],
      "importNames": ["oldMethod"],
      "paths": ["src/new/**"],
    },
  ],
}
```

Enabling this rule type

To enable this rule type, you can set the rule to `true`, or provide the
following configuration.

Property	Type	Description
`paths`	`string[]` (optional)	An optional array of exact paths or glob expressions, which restricts the paths that this custom rul

The example below enables the `NO_TEAM_IMPORTS` custom rule for all files in the
`src/` directory, excluding files in `src/legacy/`. In this example, the custom
rule is also restricted to the `dashboard` and `marketing-site` workspaces,
which is optional.

```jsonc copy filename="conformance.config.jsonc" {4-10}
{
  "overrides": [
    {
      "restrictTo": {
        "workspaces": ["dashboard", "marketing-site"],
      },
```

```
      "rules": {
        "CUSTOM.NO_TEAM_IMPORTS": {
          "paths": ["src", "!src/legacy"],
        },
      },
    },
  ],
  "customRules": [
    // ...
  ],
}
```

;

---
title: "forbidden-packages"
description: "Learn how to set custom rules to disallow packages from being listed as dependencies."
last_updated: "2026-01-16T02:19:27.664Z"
source: "https://vercel.com/docs/conformance/custom-rules/forbidden-packages"
---

forbidden-packages

The `forbidden-packages` rule type enables you to disallow packages from being listed as dependencies in `package.json`.

When to use this rule type

- **Deprecating packages**
  - You want to disallow importing a deprecated package, and to recommend a different approach
- **Standardization**
  - You want to ensure that projects depend on the same set of packages when performing similar tasks (i.e. using `jest` or `vitest` consistently across a monorepo)
- **Visibility and approval**
  - You want to enable a workflow where team-owned packages can't be depended upon without acknowledgement or approval from that team. This helps owning teams to better plan and understand the impacts of their work

Configuring this rule type

To create a custom `forbidden-packages` rule, you'll need to configure the below required properties:

Property	Type	Description
`ruleType`	`"forbidden-packages"`	The custom rule's type.
`ruleName`	`string`	The custom rule's name.
`categories`	`("nextjs" | "performance" | "security" | "code-health")[]` (optional)	The custom rule's categories. Default i
`errorMessage`	`string`	The error message, which is shown to us
`errorLink`	`string` (optional)	An optional link to show alongside the
`description`	`string` (optional)	The rule description, which is shown in
`severity`	`"major" | "minor"` (optional)	The rule severity added to the allowlis
`packageNames`	`string[]`	An array of exact package names or glob
`packageVersions`	`string[]` (optional)	**Added in Conformance `1.8.0`.** An op

Example configuration

The example below configures a rule named `NO_TEAM_PACKAGES` that disallows importing any package from the `team` workspace except for `@team/utils`.

```jsonc copy filename="conformance.config.jsonc" {4-9}
{
  "customRules": [
    {
      "ruleType": "forbidden-packages",
      "ruleName": "NO_TEAM_PACKAGES",
      "errorMessage": "Packages from the team workspace have been deprecated in favour of '@team/utils'.",
      "description": "Disallow importing packages from the team workspace.",
      "severity": "major",
      "packageNames": ["@team/*", "!@team/utils"],
    },
  ],
}
```

The next example restricts the `utils` package, only allowing versions equal to or above `2.0.0`. This option requires Conformance `1.8.0` or later.

```jsonc copy filename="conformance.config.jsonc" {4-10}
{
  "customRules": [
    {
      "ruleType": "forbidden-packages",
      "ruleName": "NO_OLD_UTIL_PACKAGES",
      "errorMessage": "Versions of `utils` below `2.0.0` are not allowed for security reasons.",
      "description": "Disallow importing `utils` versions below version `2.0.0`.",
      "severity": "major",
      "packageNames": ["utils"],
      "packageVersions: ["<=2.0.0"]
    },
  ],
}
```

Enabling this rule type

The example below enables the `NO_TEAM_PACKAGES` custom rule. In this example,
the custom rule is also restricted to the `dashboard` and `marketing-site`
workspaces, which is optional.

```jsonc copy filename="conformance.config.jsonc" {4-9}
{
  "overrides": [
    {
      "restrictTo": {
        "workspaces": ["dashboard", "marketing-site"],
      },
      "rules": {
        "CUSTOM.NO_TEAM_PACKAGES": true,
      },
    },
  ],
  "customRules": [
    // ...
  ],
}
```

---
title: "forbidden-properties"
description: "Learn how to set custom rules to disallow reading from,
writing to, and/or calling one or more properties"
last_updated: "2026-01-16T02:19:27.561Z"
source: "https://vercel.com/docs/conformance/custom-rules/forbidden-properties"
---

forbidden-properties

The `forbidden-properties` rule type enables you to disallow reading from,
writing to, and/or calling one or more properties.

When to use this rule type

- **Disallowing use of global properties**
  - You want to disallow calling `document.write`
  - You want to disallow using browser-only APIs in a component library that
    may be server-rendered
  - You want to disallow calls to usage of `window.location` in favor of another solution.
- **Disallowing use of deprecated features**
  - You want to disallow using `event.keyCode`
  - You want to disallow specific strings from being used within code

Configuring this rule type

To create a custom `forbidden-properties` rule, you'll need to configure the below
required properties:

Property	Type	Description
`ruleType`	`"forbidden-properties"`	The custom rule's type.
`ruleName`	`string`	The custom rule's name.
`errorMessage`	`string`	The error message, which is shown to users when they encounter th
`errorLink`	`string` (optional)	An optional link to show alongside the error message.
`description`	`string` (optional)	The rule description, which is shown in the Vercel Compass dashbo
`severity`	`"major" | "minor"` (optional)	The rule severity added to the allowlists and used to calculate a
`forbiddenProperties`	[`ForbiddenProperty[]`](#forbiddenproperty)	One or more properties and their forbidden operations.

`ForbiddenProperty`

Property	Type	Description
`property`	`string`	The property to target.
`operations`	`{ call?: boolean, read?: boolean, write?: boolean }`	The operation(s) to target. At least one operation is required.

Example configuration

The example below configures a rule named `NO_DOCUMENT_WRITE_CALLS` that
disallows calling `document.write`.

```jsonc copy filename="conformance.config.jsonc" {4-14}
{
  "customRules": [
    {
      "ruleType": "forbidden-properties",
      "ruleName": "NO_DOCUMENT_WRITE_CALLS",
      "errorMessage": "Calling 'document.write' is not allowed.",
      "description": "Disallows calls to `document.write`.",
      "severity": "major",
      "forbiddenProperties": [
        {
          "property": "document.write",
          "operations": {
            "call": true,
          },
        },
      ],
    },
  ],
}
```

Property assignments

Note that a property's assignments are tracked by this custom rule type.

Using our example `NO_DOCUMENT_WRITE_CALLS` rule (above), the following calls
will both result in errors.

```ts {1,4}
document.write();

const writer = document.write;
writer();
```

Enabling this rule type

The example below enables the `NO_DOCUMENT_WRITE_CALLS` custom rule. In this
example, the custom rule is also restricted to the `dashboard` and
`marketing-site` workspaces, which is optional.

```jsonc copy filename="conformance.config.jsonc" {4-9}
{
  "overrides": [
    {
      "restrictTo": {
        "workspaces": ["dashboard", "marketing-site"],
      },
      "rules": {
        "CUSTOM.NO_DOCUMENT_WRITE_CALLS": true,
      },
    },
  ],
  "customRules": [
    // ...
  ],
}
```

;

---
title: "Conformance Custom Rules"
description: "Learn how Conformance improves collaboration, productivity, and software quality at scale."
last_updated: "2026-01-16T02:19:27.578Z"
source: "https://vercel.com/docs/conformance/custom-rules"

Conformance Custom Rules

Vercel's built-in Conformance rules are crafted from extensive experience in developing large-scale codebases and high-quality web applic

Custom rules in Vercel feature unique error names and messages, providing deeper context and actionable resolution guidance. For example,

- Links to internal documentation
- Alternative methods for logging issues
- Information on who to contact for help

You can use custom rules to proactively prevent future issues, to reactively
prevent issues from reoccuring, and/or as a mitigation tool.

Available custom rule types

We support the following custom rules types:

Type	Description
[`forbidden-code`](/docs/conformance/custom-rules/forbidden-code)	Disallows code and code patterns through string and
[`forbidden-properties`](/docs/conformance/custom-rules/forbidden-properties)	Disallows properties from being read, written, and/
[`forbidden-dependencies`](/docs/conformance/custom-rules/forbidden-dependencies)	Disallows one or more files from depending on one o
[`forbidden-imports`](/docs/conformance/custom-rules/forbidden-imports)	Disallows one or more files from importing one or m
[`forbidden-packages`](/docs/conformance/custom-rules/forbidden-packages)	Disallows packages from being listed as dependencie

Getting started

The no-code custom rules are defined and [configured](/docs/conformance/customize) in `conformance.config.jsonc`.

In this example, you will set up a custom rule with the [`forbidden-imports`](/docs/conformance/custom-rules/forbidden-imports) type. Thi
called `api-utils`, and suggests to users that they should instead use a newer
version of that package.

- ### Create your config file
  At the root of your directory, create a file named `conformance.config.jsonc`. If one already exists, skip to the next step.

- ### Define a custom rule
  First, define a new custom rule in `conformance.customRules`.

  All custom rules require the properties:
  - `ruleType`
  - `ruleName`
  - `errorMessage`
  Other required and optional configuration depends on the custom
  rule type. In this example, we're using the `forbidden-imports`
  type, which requires an `moduleNames` property.
  ```jsonc copy filename="conformance.config.jsonc" {4-11}
  {
    "customRules": [
      {
        "ruleType": "forbidden-imports",
        "ruleName": "NO_API_UTILS",
        "categories": ["code-health"],
        "errorMessage": "The `api-utils` package has been deprecated. Please use 'api-utils-v2' instead, which includes more features.",
        "errorLink": "https://vercel.com/docs",
        "description": "Don't allow importing the deprecated `api-utils` package.",
  ```

````
        "severity": "major",
        "moduleNames": ["my-utils"],
      },
    ],
  }
  ```

- ### Enable the custom rule
  As all custom rules are disabled by default, you'll need to [enable rules](/docs/conformance/customize#managing-a-conformance-rule)
  in `conformance.overrides`. Refer to the documentation for each custom rule
  type for more information.

  Rule names must be prefixed with `"CUSTOM"` when enabled, and any allowlist
  files and entries will also be prefixed with `"CUSTOM"`. This prefix is added
  to ensure that the names of custom rules don't conflict with built-in rules.

  In the example below, we're enabling the rule for the entire project by
  providing it with the required configuration (targeting all files in `src`).
  ```jsonc copy filename="conformance.config.jsonc" {4-6}
  {
    "overrides": [
      {
        "rules": {
          "CUSTOM.NO_API_UTILS": {
            "paths": ["src"],
          },
        },
      },
    ],
    "customRules": [
      // ...
    ],
  }
  ```
  In this example, we've used the same configuration as above, but have also
  restricted the rule and configuration to the `api-teams` workspace.
  ```jsonc copy filename="conformance.config.jsonc" {4-9}
  {
    "overrides": [
      {
        "restrictTo": {
          "workspaces": ["api-teams"],
        },
        "rules": {
          "CUSTOM.NO_API_UTILS": {
            "paths": ["src", "!src/**/*.test.ts"],
          },
        },
      },
    ],
    "customRules": [
      // ...
    ],
  }
  ```

- ### Restrict the rule to a workspace
  In this example used the same configuration as above, but have also
  restricted the rule and configuration to the `api-teams` workspace:
  ```jsonc copy filename="conformance.config.jsonc" {4-9}
  {
    "overrides": [
      {
        "restrictTo": {
          "workspaces": ["api-teams"],
        },
        "rules": {
          "CUSTOM.NO_API_UTILS": {
            "paths": ["src", "!src/**/*.test.ts"],
          },
        },
      },
    ],
    "customRules": [
      // ...
    ],
  }
  ```


---
title: "Customizing Conformance"
description: "Learn how to manage and configure your Conformance rules."
last_updated: "2026-01-16T02:19:27.725Z"
source: "https://vercel.com/docs/conformance/customize"
---

# Customizing Conformance

The Conformance framework may be customized so that you can manage
rules for different workspaces in your repository or to pass configuration to
the rules.

To customize Conformance, first define a `conformance.config.jsonc` file in the root of your directory.

> **💡 Note:** Both `conformance.config.jsonc` and `conformance.config.json` are supported,
> and both support JSONC (JSON with JavaScript-style comments). We recommend
> using the `.jsonc` extension as it helps other tools (for example, VS Code) to
> provide syntax highlighting and validation.
````
````

Enabling all rules By default

To enable all Conformance rules by default, add the `defaultRules` field to the
top level `configuration` section of the config file:

```jsonc copy filename="conformance.config.jsonc" {3}
{
  "configuration": {
    "defaultRules": "all",
  },
}
```

Ignoring files

To exclude one or more files from Conformance, use the `ignorePatterns` field in the top level of the config file:

```jsonc copy filename="conformance.config.jsonc"
{
  "ignorePatterns": ["generated/**/*.js"],
}
```

This field accepts an array of glob patterns as strings.

Configuring specific workspaces

Each Conformance override accepts a `restrictTo` parameter which controls what
workspaces the configuration will apply to. If no `restrictTo` is specified,
then the configuration will apply globally to every workspace.

```jsonc copy filename="conformance.config.jsonc" {5}
{
  "overrides": [
    {
      // NOTE: No `restrictTo` is specified here so this applies globally.
      "rules": {},
    },
  ],
}
```

Conformance configuration can be applied to specific workspaces using either
the name of the workspace or the directory of the workspace on the `restrictTo` field:

- Use the `workspaces` field, which accepts a list of workspace names:
  ```jsonc copy filename="conformance.config.jsonc" {4-7}
  {
    "overrides": [
      {
        "restrictTo": {
          "workspaces": ["eslint-config-custom"],
        },
        "rules": {},
      },
    ],
  }
  ```
- Use the `directories` field to specify a directory. All workspaces that live under that directory will be matched:
  ```jsonc copy filename="conformance.config.json" {4-7}
  {
    "overrides": [
      {
        "restrictTo": {
          "directories": ["configs/"],
        },
        "rules": {},
      },
    ],
  }
  ```
  This will match `configs/tsconfig` and `configs/eslint-config-custom`.
- Set the `root` field to true to match the root of the repository:
  ```jsonc copy filename="conformance.config.jsonc" {4-7}
  {
    "overrides": [
      {
        "restrictTo": {
          "root": true,
        },
        "rules": {},
      },
    ],
  }
  ```

Configuration cascade

If multiple overrides are specified that affect the same workspace, the
configurations will be unioned together. If there are conflicts between the
overrides, the last specified value will be used.

Managing a Conformance rule

To enable or disable a Conformance rule, use the `rules` field. This
field is an object literal where the keys are the name of the [rule](/docs/conformance/rules) and the
values are booleans or another object literal containing a [rule-specific
configuration](#configuring-a-conformance-rule).

For example, this configuration will disable the `TYPESCRIPT_CONFIGURATION` rule:

```jsonc copy filename="conformance.config.jsonc" {5}
{
  "overrides": [
    {
      "rules": {
        "TYPESCRIPT_CONFIGURATION": false,
      },
    },
  ],
}
```

All rules are enabled by default unless explicitly disabled in the config.

Configuring a Conformance rule

Some Conformance rules can be configured to alter behavior based on the project
settings. Instead of a `boolean` being provided in the `rules` configuration,
an object literal could be passed with the configuration for that rule.

For example, this configuration will require a specific list of
ESLint plugins in every workspace:

```jsonc copy filename="conformance.config.jsonc" {6}
{
  "overrides": [
    {
      "rules": {
        "ESLINT_CONFIGURATION": {
          "requiredPlugins": ["@typescript-eslint"],
        },
      },
    },
  ],
}
```

Adding custom error messages to Conformance rules

If you want to specify additional information or link to project-specific
documentation, you can add custom error messages to the output of any
conformance rule. These messages can be added globally to all rules or on a
per-rule basis.

To add an error message to the output of **all rules**, add `globalErrorMessage` to
the `configuration` section of the override:

```jsonc copy filename="conformance.config.jsonc" {5}
{
  "overrides": [
    {
      "configuration": {
        "globalErrorMessage": "See link_to_docs for more information.",
      },
    },
  ],
}
```

To add an error message to the output of **one
specific rule**, add an entry for that test to the `additionalErrorMessages`
field:

```jsonc copy filename="conformance.config.jsonc" {5-7}
{
  "overrides": [
    {
      "configuration": {
        "additionalErrorMessages": {
          "TYPESCRIPT_CONFIGURATION": "Please see project_link_to_typescript_docs for more information.",
        },
      },
    },
  ],
}
```

---
title: "Getting Started with Conformance"
description: "Learn how to set up Conformance for your codebase."
last_updated: "2026-01-16T02:19:27.767Z"
source: "https://vercel.com/docs/conformance/getting-started"

Getting Started with Conformance

To [set up Conformance](#setting-up-conformance-in-your-repository) in your repository, you must:

- Set up [Vercel's private npm registry](/docs/private-registry) to install the necessary packages
- [Install and initialize](/docs/conformance/getting-started#setting-up-conformance-in-your-repository) Conformance in your repository

If you've already set up Code Owners, you may have already completed some of these steps.

Prerequisites

Get access to Conformance

To enable Conformance for your Enterprise team, you'll need to request access through your Vercel account administrator.

Setting up Vercel's private npm registry

Vercel distributes packages with the `@vercel-private` scope through our private npm registry, and requires that each user using the pack

To use the private npm registry, you'll need to follow the documentation to:

- [Set up your local environment](/docs/private-registry#setting-up-your-local-environment) – This should be completed by the team owner,
- [Set up Vercel](/docs/private-registry#setting-up-vercel) – This should be completed by the team owner
- [Optionally, set up Conformance for use with CI](/docs/private-registry#setting-up-your-ci-provider) – This should be completed by the

Setting up Conformance in your repository

This section guides you through setting up Conformance for your repository.

- ### Set up the Vercel CLI
  The Conformance CLI is separate to the [Vercel CLI](/docs/cli), however it
  uses the Vercel CLI for authentication.

  Before continuing, please ensure that the Vercel CLI is [installed](/docs/cli#installing-vercel-cli)
  and that you are [logged in](/docs/cli/login).

- ### Initialize Conformance
  Use the CLI to automatically initialize Conformance in your project. Start by running this command in your repository's root:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>
  > **⚠️ Warning:** `yarn dlx` only works with Yarn version 2 or newer, for Yarn v1 use
  > `yarn -DW add @vercel-private/conformance && yarn vercel-conformance init`
  After running, check the installation success by executing:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>

- ### Review the generated changes
  The Conformance `init` command creates the following changes:
  - First, it installs the CLI package in your root `package.json` and every workspace `package.json`, if your monorepo uses workspaces.
  - It also adds a `conformance` script to the `scripts` field of every
    `package.json`. This script runs Conformance.
  - It adds any existing Conformance errors to allowlists, letting you start using Conformance without immediate fixes and allowing you t
  Once you've reviewed these, open a pull request with the changes and merge it.

- ### Add owners for allowlist files
  ** This step assumes you have [set up Code Owners](/docs/code-owners/getting-started).**

  Conformance allows specific individuals to review modifications to allowlist files.
  Add a `.vercel.approvers` file at your repository's root:
  ```text copy filename=".vercel.approvers"
  **/*.allowlist.json @org/team:required
  ```
  Now, changes to allowlist files need a review from someone on
  `@org/team` before merging.

  Learn more about [wildcard syntax](/docs/code-owners/code-approvers#globstar-pattern)
  and [`:required` syntax](/docs/code-owners/code-approvers#required) from Code Owners.

- ### Add Conformance to your CI system
  You can integrate Conformance in your CI to avoid merging errors into your code. To learn more, see [Setting up your CI provider](/docs

More resources

- [Code Owners](/docs/code-owners)

- [Conformance](/docs/conformance)

---
title: "Introduction to Conformance"
description: "Learn how Conformance improves collaboration, productivity, and software quality at scale."
last_updated: "2026-01-16T02:19:27.888Z"
source: "https://vercel.com/docs/conformance"

Introduction to Conformance

Conformance provides tools that run automated checks on your code for product critical issues, such as performance, security, and code he

- **Prevent issues from being merged into your codebase**: Conformance runs locally and on Continuous Integration (CI) to notify develope
- **Learn from expert guidance directly in your development workflow**: Conformance rules were created based on years of experience in la
- **Burn down existing issues over time**: Conformance allowlists enable you to identify and allowlist all existing errors, unblocking de

Getting Started

To get started with Conformance, follow the instructions on the
[Getting Started](/docs/conformance/getting-started) page.

Conformance Rules

Conformance comes with a curated suite of rules that look
for common issues. These rules were created based on the decades of combined
experience that we have building high quality web applications.

You can lean more about the built-in Conformance rules on the
[Conformance Rules](/docs/conformance/rules) page.

Conformance Allowlists

A core feature in Conformance is the ability to provide allowlists. This mechanism allows organizations to have developers review their c

Learn more about how this mechanism works on the
[Allowlists](/docs/conformance/allowlist) page.

Customizing Conformance

Conformance can be customized to meet your repository's
needs. See [Customizing Conformance](/docs/conformance/customize) for more
information.

More resources

- [Learn how Vercel helps organizations grow with Conformance and Code owners](https://www.youtube.com/watch?v=IFkZz3_7Poo)

---
title: "BFCACHE_INTEGRITY_NO_UNLOAD_LISTENERS"
description: "Disallows the use of the unload and beforeunload events to eliminate a source of eviction from the browser"
last_updated: "2026-01-16T02:19:27.779Z"
source: "https://vercel.com/docs/conformance/rules/BFCACHE_INTEGRITY_NO_UNLOAD_LISTENERS"

BFCACHE_INTEGRITY_NO_UNLOAD_LISTENERS

This rule disallows the use of the `unload` and `beforeunload` events to improve the integrity of the Back-Forward Cache in browsers.

The Back-Forward Cache (bfcache) is a browser feature that allows pages to be cached in memory when the user navigates
away from them. When the user navigates back to the page, it can be loaded almost instantly from the cache instead of
having to be reloaded from the network. Breaking the bfcache's integrity can cause a page to be reloaded from the network
when the user navigates back to it, which can be slow and jarring.

The most important rule for maintaining the integrity of the bfcache is to not use the `unload` event. This event is fired
when the user navigates away from the page, but it is unreliable and disables the cache on most browsers.

The `beforeunload` event can also make your page ineligible for the cache in browsers so it is better to avoid using.
However there are some legitimate use cases for this event, such as checking if the user has unsaved work before they exit
the page. In this case it is recommended to add the listener conditionally and remove it as soon as the work as been saved.

Alternative events that can be considered are `pagehide` or `visibilitychange`, which are more reliable
events that do not break the bfcache and will fire when the user navigates away from or unfocuses the page.

To learn more about the bfcache, see the [web.dev docs](https://web.dev/bfcache).

Related Rules

- [BFCACHE_INTEGRITY_REQUIRE_NOOPENER_ATTRIBUTE](/docs/conformance/rules/BFCACHE_INTEGRITY_REQUIRE_NOOPENER_ATTRIBUTE)

Example

Two examples of when this check would fail:

```ts filename="src/utils/handle-user-navigation.ts"
export function handleUserNavigatingAway() {
  window.onunload = (event) => {
    console.log('Page has unloaded.');
  };
}

export function handleUserAboutToNavigateAway() {
  window.onbeforeunload = (event) => {
    console.log('Page is about to be unloaded.');
  };
}
```

```ts filename="src/utils/handle-user-navigation.ts"
export function handleUserNavigatingAway() {
  window.addEventListener('unload', (event) => {
    console.log('Page has unloaded.');
  });
}

export function handleUserAboutToNavigateAway() {
  window.addEventListener('beforeunload', (event) => {
    console.log('Page is about to be unloaded.');
  });
}
```

How to fix

Instead, we can use the `pagehide` event to detect when the user navigates away from the page.

```ts filename="src/utils/handle-user-navigation.ts"
export function handleUserNavigatingAway() {
  window.onpagehide = (event) => {
    console.log('Page is about to be hidden.');
  };
}
```

```ts filename="src/utils/handle-user-navigation.ts"
export function handleUserNavigatingAway() {
  window.addEventListener('pagehide', (event) => {
    console.log('Page is about to be hidden.');
  });
}
```

---
title: "BFCACHE_INTEGRITY_REQUIRE_NOOPENER_ATTRIBUTE"
description: "Requires that links opened with window.open use the noopener attribute to eliminate a source of eviction from the browser"
last_updated: "2026-01-16T02:19:27.801Z"
source: "https://vercel.com/docs/conformance/rules/BFCACHE_INTEGRITY_REQUIRE_NOOPENER_ATTRIBUTE"

BFCACHE_INTEGRITY_REQUIRE_NOOPENER_ATTRIBUTE

The Back-Forward Cache (bfcache) is a browser feature that allows pages to be cached in memory when the user navigates
away from them. When the user navigates back to the page, it can be loaded almost instantly from the cache instead of
having to be reloaded from the network. Breaking the bfcache's integrity can cause a page to be reloaded from the network
when the user navigates back to it, which can be slow and jarring.

Pages opened with `window.open` that do not use the `noopener` attribute can both be a security risk and also will
prevent browsers from caching the page in the bfcache. This is because the new window can access the `window.opener` property
of the original window, so putting the original page into the bfcache could break the new window when attempting to access it.

Using the `noreferrer` attribute will also set the `noopener` attribute to true, so it can also be used to ensure
the page is placed into the bfcache.

To learn more about the bfcache, see the [web.dev docs](https://web.dev/bfcache).

Related Rules

- [BFCACHE_INTEGRITY_NO_UNLOAD_LISTENERS](/docs/conformance/rules/BFCACHE_INTEGRITY_NO_UNLOAD_LISTENERS)

Example

Examples of when this check would fail:

```ts
window.open('https://example.com', '_blank');
window.open('https://example.com');
```

How to fix

Instead, use the `noopener` or `noreferrer` attributes:

```ts
window.open('https://example.com', '_blank', 'noopener');
window.open('https://example.com', '_top', 'noreferrer');
```

---
title: "ESLINT_CONFIGURATION"
description: "Requires that a workspace package has ESLint installed and configured correctly"
last_updated: "2026-01-16T02:19:27.817Z"
source: "https://vercel.com/docs/conformance/rules/ESLINT_CONFIGURATION"

ESLINT_CONFIGURATION

[ESLint](https://eslint.org/) is a tool to statically analyze code to find and
report problems. ESLint is required to be enabled for every workspace package
in a monorepo so that all code in the monorepo is checked for these problems.
Additionally, repositories can enforce that particular ESLint plugins are
installed and that specific rules are treated as errors.

This rule requires that:

- An ESLint config exists in the current workspace.
- A script to run ESLint exists in `package.json` in the current workspace.

- `reportUnusedDisableDirectives` is set to `true`, which detects and can
  autofix unused ESLint disable comments.
- `root` is set to `true`, which ensures that workspaces don't inherit
  unintended rules and configuration from ESLint configuration files in parent
  directories.

Example

```sh
A Conformance error occurred in test "ESLINT_CONFIGURATION".

ESLint configuration must specify `reportUnusedDisableDirectives` to be `true`

To find out more information and how to fix this error, visit
/docs/conformance/rules/ESLINT_CONFIGURATION.

If this violation should be ignored, add the following entry to
/apps/dashboard/.allowlists/ESLINT_CONFIGURATION.allowlist.json and get approval from the appropriate person.

{
  "testName": "ESLINT_CONFIGURATION",
  "reason": "TODO: Add reason why this violation is allowed to be ignored.",
  "location": {
    "workspace": "dashboard"
  }
}
```

See the [ESLint docs](https://eslint.org/docs/latest/use/configure/) for more information on how to configure ESLint, including plugins a

How To Fix

The recommended approach for configuring ESLint in a monorepo is to have a
shared ESLint config in an internal package. See the [Turbo docs on ESLint](https://turborepo.com/docs/handbook/linting/eslint) to get st

Once your monorepo has a shared ESLint config, you can add a `.eslintrc.cjs`
file to the root folder of your workspace with the contents:

```js copy filename=".eslintrc.cjs"
module.exports = {
  root: true,
  extends: ['eslint-config-custom/base'],
};
```

You should also add `"eslint-config-custom": "workspace:*"` to your
`devDependencies`.

---
title: "ESLINT_NEXT_RULES_REQUIRED"
description: "Requires that a workspace package is configured with required Next.js plugins and rules"
last_updated: "2026-01-16T02:19:27.824Z"
source: "https://vercel.com/docs/conformance/rules/ESLINT_NEXT_RULES_REQUIRED"

ESLINT_NEXT_RULES_REQUIRED

This Conformance check requires that ESLint plugins for Next.js are configured
correctly in your application, including:

- [@next/next](https://nextjs.org/docs/basic-features/eslint#eslint-plugin)

These plugins help to catch common Next.js issues, including performance.

Example

```sh
A Conformance error occurred in test "ESLINT_NEXT_RULES_REQUIRED".

These ESLint plugins must have rules configured to run: @next/next

To find out more information and how to fix this error, visit
https://vercel.com/docs/conformance/rules/ESLINT_NEXT_RULES_REQUIRED.

If this violation should be ignored, add the following entry to
/apps/dashboard/.allowlists/ESLINT_NEXT_RULES_REQUIRED.allowlist.json and
get approval from the appropriate person.

{
  "testName": "ESLINT_NEXT_RULES_REQUIRED",
  "reason": "TODO: Add reason why this violation is allowed to be ignored.",
  "location": {
    "workspace": "dashboard"
  },
}
```

This check requires that certain ESLint plugins are installed and rules within
those plugins are configured to be errors. If you are missing required plugins,
you will receive an error such as:

```sh
ESLint configuration is missing required security plugins:
  Missing plugins: @next/next
  Registered plugins: import and @typescript-eslint
```

For more information on ESLint plugins and rules, see [plugins](https://eslint.org/docs/latest/user-guide/configuring/plugins) and [rules

How To Fix

The recommended approach for configuring ESLint in a monorepo is to have a
shared ESLint config in an internal package. See the [Turbo docs on ESLint](https://turborepo.com/docs/handbook/linting/eslint) to get st

Once your monorepo has a shared ESLint config, you can add a `.eslintrc.cjs`
file to the root folder of your workspace with the contents:

```js copy filename=".eslintrc.cjs"
module.exports = {
  root: true,
  extends: ['eslint-config-custom/base'],
};
```

You should also add `"eslint-config-custom": "workspace:*"` to your
`devDependencies`.

---
title: "ESLINT_REACT_RULES_REQUIRED"
description: "Requires that a workspace package is configured with required React plugins and rules"
last_updated: "2026-01-16T02:19:27.841Z"
source: "https://vercel.com/docs/conformance/rules/ESLINT_REACT_RULES_REQUIRED"

ESLINT_REACT_RULES_REQUIRED

This Conformance check requires that ESLint plugins for React are configured
correctly in your application, including:

- [react](https://github.com/jsx-eslint/eslint-plugin-react)
- [react-hooks](https://github.com/facebook/react/tree/main/packages/eslint-plugin-react-hooks)
- [jsx-a11y](https://github.com/jsx-eslint/eslint-plugin-jsx-a11y)

These plugins help to catch common React issues, such as incorrect React hooks
usage, helping to reduce bugs and to improve application accessibility.

Example

```sh
A Conformance error occurred in test "ESLINT_REACT_RULES_REQUIRED".

These ESLint plugins must have rules configured to run: @next/next

To find out more information and how to fix this error, visit
https://vercel.com/docs/conformance/rules/ESLINT_REACT_RULES_REQUIRED.

If this violation should be ignored, add the following entry to
/apps/dashboard/.allowlists/ESLINT_REACT_RULES_REQUIRED.allowlist.json and
get approval from the appropriate person.

{
  "testName": "ESLINT_REACT_RULES_REQUIRED",
  "reason": "TODO: Add reason why this violation is allowed to be ignored.",
  "location": {
    "workspace": "dashboard"
  },
}
```

This check requires that certain ESLint plugins are installed and rules within
those plugins are configured to be errors. If you are missing required plugins,
you will receive an error such as:

```sh
ESLint configuration is missing required security plugins:
  Missing plugins: react, react-hooks, and jsx-a11y
  Registered plugins: import and @typescript-eslint
```

For more information on ESLint plugins and rules, see [plugins](https://eslint.org/docs/latest/user-guide/configuring/plugins) and [rules

How To Fix

The recommended approach for configuring ESLint in a monorepo is to have a
shared ESLint config in an internal package. See the [Turbo docs on ESLint](https://turborepo.com/docs/handbook/linting/eslint) to get st

Once your monorepo has a shared ESLint config, you can add a `.eslintrc.cjs`
file to the root folder of your workspace with the contents:

```js copy filename=".eslintrc.cjs"
module.exports = {
  root: true,
  extends: ['eslint-config-custom/base'],
};
```

You should also add `"eslint-config-custom": "workspace:*"` to your
`devDependencies`.

---
title: "ESLINT_RULES_REQUIRED"
description: "Requires that a workspace package is configured with required ESLint plugins and rules"
last_updated: "2026-01-16T02:19:27.865Z"
source: "https://vercel.com/docs/conformance/rules/ESLINT_RULES_REQUIRED"

ESLINT_RULES_REQUIRED

This Conformance check requires that ESLint plugins are configured correctly
in your application, including:

- [@typescript-eslint](https://typescript-eslint.io/)
- [eslint-comments](https://mysticatea.github.io/eslint-plugin-eslint-comments/)
- [import](https://github.com/import-js/eslint-plugin-import)

These plugins help to catch common issues, and ensure that ESLint is set
up to work with TypeScript where applicable.

Example

```sh
A Conformance error occurred in test "ESLINT_RULES_REQUIRED".

These ESLint plugins must have rules configured to run: @typescript-eslint and import

To find out more information and how to fix this error, visit
https://vercel.com/docs/conformance/rules/ESLINT_RULES_REQUIRED.

If this violation should be ignored, add the following entry to
/apps/dashboard/.allowlists/ESLINT_RULES_REQUIRED.allowlist.json and
get approval from the appropriate person.

{
  "testName": "ESLINT_RULES_REQUIRED",
  "reason": "TODO: Add reason why this violation is allowed to be ignored.",
  "location": {
    "workspace": "dashboard"
  },
}
```

This check requires that certain ESLint plugins are installed and rules within
those plugins are configured to be errors. If you are missing required plugins,
you will receive an error such as:

```sh
ESLint configuration is missing required security plugins:
  Missing plugins: eslint-comments
  Registered plugins: import and @typescript-eslint
```

If all the required plugins are installed but some rules are not configured to
run or configured to be errors, you will receive an error such as:

```sh
`eslint-comments/no-unlimited-disable` must be specified as an error in the ESLint configuration, but is specified as off.
```

As a part of this test, some rules are forbidden from being disabled. If you
disable those rules, you will receive an error such as:

```sh
Disabling these ESLint rules is not allowed.
Please see the ESLint documentation for each rule for how to fix.
eslint-comments/disable-enable-pair
eslint-comments/no-restricted-disable
```

For more information on ESLint plugins and rules, see [plugins](https://eslint.org/docs/latest/user-guide/configuring/plugins) and [rules

How To Fix

The recommended approach for configuring ESLint in a monorepo is to have a
shared ESLint config in an internal package. See the [Turbo docs on ESLint](https://turborepo.com/docs/handbook/linting/eslint) to get st

Once your monorepo has a shared ESLint config, you can add a `.eslintrc.cjs`
file to the root folder of your workspace with the contents:

```js copy filename=".eslintrc.cjs"
module.exports = {
  root: true,
  extends: ['eslint-config-custom/base'],
};
```

You should also add `"eslint-config-custom": "workspace:*"` to your
`devDependencies`.

---
title: "NEXTJS_MISSING_MODULARIZE_IMPORTS"
description: "modularizeImports can improve dev compilation speed for packages that use barrel files."
last_updated: "2026-01-16T02:19:27.880Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_MISSING_MODULARIZE_IMPORTS"

NEXTJS_MISSING_MODULARIZE_IMPORTS

`modularizeImports` is a feature of Next 13 that can reduce dev compilation times
when importing packages that are exported as barrel files. Barrel files are
convenient ways to export code from a package from a single file to make it
straightforward to import any of the code from the package. However, since they export a
lot of code from the same file, importing these packages can cause tools to do
a lot of additional work analyzing files that are unused in the application.

How to fix

To fix this, you can add a `modularizeImports` config to `next.config.js` for

the package that uses barrel files. For example:

```js filename="next.config.js"
modularizeImports: {
  lodash: {
    transform: 'lodash/{{member}}';
  }
}
```

The exact format of the transform may differ by package, so double check how
the package uses barrel files first.

See the [Next.js docs](https://nextjs.org/docs/architecture/nextjs-compiler#modularize-imports) for
more information.

Custom configuration

You can also specify required `modularizeImports` config for your own packages.

In your `conformance.config.jsonc` file, add:

```js filename="conformance.config.jsonc"
NEXTJS_MISSING_MODULARIZE_IMPORTS: {
  requiredModularizeImports: [
    {
      moduleDependency: 'your-package-name',
      requiredConfig: {
        transform: 'your-package-name/{{member}}',
      },
    },
  ];
}
```

This will require that any workspace in your monorepo that uses the
`your-package-name` package must use the provided `modularizeImports` config
in their `next.config.js` file.

See [Customizing Conformance](/docs/conformance/customize) for more information.

---
title: "NEXTJS_MISSING_NEXT13_TYPESCRIPT_PLUGIN"
description: "Applications using Next 13 should use the "
last_updated: "2026-01-16T02:19:27.899Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_MISSING_NEXT13_TYPESCRIPT_PLUGIN"

NEXTJS_MISSING_NEXT13_TYPESCRIPT_PLUGIN

Next 13 introduced a TypeScript plugin to provide richer information for
Next.js applications using TypeScript. See the [Next.js docs](https://nextjs.org/docs/app/building-your-application/configuring/typescrip

How to fix

Add the following to `plugins` in the `compilerOptions` of your `tsconfig.json`
file.

```json filename="tsconfig.json"
  "compilerOptions": {
    "plugins": [{ "name": "next" }]
  }
```

---
title: "NEXTJS_MISSING_OPTIMIZE_PACKAGE_IMPORTS"
description: "optimizePackageImports improves compilation speed for packages that use barrel files or export many modules."
last_updated: "2026-01-16T02:19:27.904Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_MISSING_OPTIMIZE_PACKAGE_IMPORTS"

NEXTJS_MISSING_OPTIMIZE_PACKAGE_IMPORTS

[`optimizePackageImports`](https://nextjs.org/docs/pages/api-reference/next-config-js/optimizePackageImports)
is a feature added in Next 13.5 that improves compilation speed when importing packages that use barrel
exports and export many named exports. This replaces the [`modularizeImports`](https://nextjs.org/docs/architecture/nextjs-compiler#modul
configuration option as it optimizes many of the most popular open source libraries automatically.

Barrel files make the process of exporting code from a package convenient by allowing all the code to be exported from a single file. Thi

For further reading, see:

- [How we optimized package imports in Next.js](https://vercel.com/blog/how-we-optimized-package-imports-in-next-js)
- [`optimizePackageImports`](https://nextjs.org/docs/pages/api-reference/next-config-js/optimizePackageImports)

> **⚠️ Warning:** As of Next.js 14.2.3, this configuration option is still experimental. Check
> the Next.js documentation for the latest information here:
> [`optimizePackageImports`](https://nextjs.org/docs/pages/api-reference/next-config-js/optimizePackageImports).

How to fix

To fix this, you can add a `modularizeImports` config to `next.config.js` for
the package that uses barrel files. For example:

```js filename="next.config.js"
experimental: {
  optimizePackageImports: ['@vercel/geistcn/components'];
}
```

```


---
title: "NEXTJS_MISSING_REACT_STRICT_MODE"
description: "Applications using Next.js should enable React Strict Mode"
last_updated: "2026-01-16T02:19:27.906Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_MISSING_REACT_STRICT_MODE"
---

# NEXTJS_MISSING_REACT_STRICT_MODE

We strongly suggest you enable Strict Mode in your Next.js application
to better prepare your application for the future of React. See the [Next.js doc on React Strict Mode](https://nextjs.org/docs/api-refere
for more information.

## How to fix

Add the following to your `next.config.js` file.

```json filename="next.config.js"
module.exports = {
  reactStrictMode: true,
}
```


---
title: "NEXTJS_MISSING_SECURITY_HEADERS"
description: "Requires that security headers are set correctly for Next.js apps and contain valid directives."
last_updated: "2026-01-16T02:19:27.920Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_MISSING_SECURITY_HEADERS"

NEXTJS_MISSING_SECURITY_HEADERS

Security headers are important to set to improve the security of your application.
Security headers can be set for all routes in \[`next.config.js` files]
(https://nextjs.org/docs/advanced-features/security-headers). This
conformance check requires that the security headers are set and use a valid
value.

Required headers:

- Content-Security-Policy
- Strict-Transport-Security
- X-Frame-Options
- X-Content-Type-Options
- Referrer-Policy

Example

```sh
Conformance errors found!

A Conformance error occurred in test "NEXTJS_MISSING_SECURITY_HEADERS".

The security header "Strict-Transport-Security" is not set correctly. The "includeSubDomains" directive should be used in conjunction wit

To find out more information and how to fix this error, visit
/docs/conformance/rules/NEXTJS_MISSING_SECURITY_HEADERS.

If this violation should be ignored, add the following entry to
/apps/docs/.allowlists/NEXTJS_MISSING_SECURITY_HEADERS.allowlist.json
and get approval from the appropriate person.

{
  "testName": "NEXTJS_MISSING_SECURITY_HEADERS",
  "reason": "TODO: Add reason why this violation is allowed to be ignored.",
  "location": {
    "workspace": "docs"
  },
  "details": {
    "header": "Strict-Transport-Security"
  }
}
```

How to fix

Follow the [Next.js security headers documentation](https://nextjs.org/docs/advanced-features/security-headers)
to fix this Conformance test. That document will walk through each of the
headers and also links to further documentation to understand what the headers
do and how to set the best values for your application.

---
title: "NEXTJS_NO_ASYNC_LAYOUT"
description: "Ensures that the exported Next.js "
last_updated: "2026-01-16T02:19:27.926Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_NO_ASYNC_LAYOUT"

NEXTJS_NO_ASYNC_LAYOUT

This rule examines all Next.js app router layout files and their transitive dependencies to ensure
none are asynchronous or return new Promise instances. Even if the layout component itself is not
asynchronous, importing an asynchronous component somewhere in the layout's dependency tree can
silently cause the layout to render dynamically. This can cause a blank layout to be displayed to
the user while Next.js waits for long promises to resolve.
```

By default, this rule is disabled. To enable it, refer to
[customizing Conformance](/docs/conformance/customize).

For further reading, these resources may be helpful:

- [Loading UI and Streaming in Next.js](https://nextjs.org/docs/app/building-your-application/routing/loading-ui-and-streaming):
  This guide discusses strategies for loading UI components and streaming content in Next.js applications.
- [Next.js Layout File Conventions](https://nextjs.org/docs/app/api-reference/file-conventions/layout):
  This document provides an overview of file conventions related to layout in Next.js.
- [Next.js Parallel Routes](https://nextjs.org/docs/app/building-your-application/routing/parallel-routes):
  This guide discusses how to use parallel routes to improve performance in Next.js applications.
- [Next.js Route Segment Config](https://nextjs.org/docs/app/api-reference/file-conventions/route-segment-config#dynamic):
  This document provides an overview of the `dynamic` export and how it can be used to force the dynamic behavior of a layout.

Examples

This rule will catch the following code.

```tsx filename="app/layout.tsx"
export default async function RootLayout() {
  const data = await fetch();
  return <div>{data}</div>;
}
```

```jsx filename="app/layout.jsx"
async function AuthButton() {
  const isAuthorized = await auth();
  return <div>{isAuthorized ? 'Authorized' : 'Unauthorized'}</div>;
}

export default function Layout() {
  return <AuthButton />;
}
```

How to fix

You can fix this error by wrapping your async component with a `<Suspense/>` boundary that has
a fallback UI to indicate to Next.js that it should use the fallback until the promise resolves.

You can also move the asynchronous component to a [parallel route](https://nextjs.org/docs/app/building-your-application/routing/parallel
which allows Next.js to render one or more pages within the same layout.

Alternatively, you can manually force the dynamic behavior of the layout by exporting a `dynamic` value.
This rule will only error if `dynamic` is not specified or is set to `auto`.
Read more [here](https://nextjs.org/docs/app/api-reference/file-conventions/route-segment-config#dynamic).

```tsx filename="app/layout.tsx"
export const dynamic = 'force-static';

export default async function RootLayout() {
  const data = await fetch();
  return <div>{data}</div>;
}
```

---
title: "NEXTJS_NO_ASYNC_PAGE"
description: "Ensures that the exported Next.js page component and its transitive dependencies are not asynchronous, as that blocks the r
last_updated: "2026-01-16T02:19:27.940Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_NO_ASYNC_PAGE"

NEXTJS_NO_ASYNC_PAGE

This rule examines all Next.js app router page files and their transitive dependencies to ensure
none are asynchronous or return new Promise instances. Even if the page component itself is not
asynchronous, importing an asynchronous component somewhere in the page's dependency tree can
silently cause the page to render dynamically. This can cause a blank page to be displayed to
the user while Next.js waits for long promises to resolve.

This rule will not error if it detects a sibling [loading.js](https://nextjs.org/docs/app/api-reference/file-conventions/loading)
file beside the page.

By default, this rule is disabled. To enable it, refer to
[customizing Conformance](/docs/conformance/customize).

For further reading, you may find these resources helpful:

- [Loading UI and Streaming in Next.js](https://nextjs.org/docs/app/building-your-application/routing/loading-ui-and-streaming):
  This guide discusses strategies for loading UI components and streaming content in Next.js applications.
- [Next.js Loading File Conventions](https://nextjs.org/docs/app/api-reference/file-conventions/loading):
  This document provides an overview of file conventions related to loading in Next.js.
- [Next.js Route Segment Config](https://nextjs.org/docs/app/api-reference/file-conventions/route-segment-config#dynamic):
  This document provides an overview of the `dynamic` export and how it can be used to force the dynamic behavior of a layout.

Examples

This rule will catch the following code.

```tsx filename="app/page.tsx"
export default async function Page() {
  const data = await fetch();
  return <div>{data}</div>;
}
```

```jsx filename="app/page.jsx"
async function AuthButton() {
  const isAuthorized = await auth();
  return <div>{isAuthorized ? 'Authorized' : 'Unauthorized'}</div>;
}

export default function Page() {
  return <AuthButton />;
}
```

How to fix

You can fix this error by wrapping your async component with a `<Suspense/>` boundary that has
a fallback UI to indicate to Next.js that it should use the fallback until the promise resolves.

Alternatively, you can manually force the dynamic behavior of the page by exporting a `dynamic` value.
This rule will only error if `dynamic` is not specified or is set to `auto`.
Read more [here](https://nextjs.org/docs/app/api-reference/file-conventions/route-segment-config#dynamic).

```tsx filename="app/page.tsx"
export const dynamic = 'force-static';

export default async function Page() {
  const data = await fetch();
  return <div>{data}</div>;
}
```

---
title: "NEXTJS_NO_BEFORE_INTERACTIVE"
description: "Requires review of usage of the beforeInteractive strategy in Script (next/script) elements."
last_updated: "2026-01-16T02:19:27.944Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_NO_BEFORE_INTERACTIVE"

NEXTJS_NO_BEFORE_INTERACTIVE

The default [loading strategy](https://nextjs.org/docs/basic-features/script#strategy)
for [`next/script`](https://nextjs.org/docs/basic-features/script) is optimised
for fast page loads.

Setting the strategy to [`beforeInteractive`](https://nextjs.org/docs/api-reference/next/script#beforeinteractive)
forces the script to load before any Next.js code and before hydration occurs,
which delays the page from becoming interactive.

For further reading, see:

- [Loading strategy in Next.js](https://nextjs.org/docs/basic-features/script#strategy)
- [`next/script` docs](https://nextjs.org/docs/api-reference/next/script#beforeinteractive)
- [Chrome blog on the Next.js Script component](https://developer.chrome.com/blog/script-component/#the-nextjs-script-component)

Examples

This rule will catch the following code.

```ts {5}
import Script from 'next/script';

export default function MyPage() {
  return (
    <Script src="https://example.com/script.js" strategy="beforeInteractive" />
  );
}
```

How to fix

This rule flags any usage of `beforeInteractive` for review. If approved, the
exception should be added to the allowlist.

---
title: "NEXTJS_NO_CLIENT_DEPS_IN_MIDDLEWARE"
description: "Disallows dependency on client libraries inside of middleware to improve performance of middleware."
last_updated: "2026-01-16T02:19:27.947Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_NO_CLIENT_DEPS_IN_MIDDLEWARE"

NEXTJS_NO_CLIENT_DEPS_IN_MIDDLEWARE

This check disallows dependencies on client libraries, such as `react` and
`next/router` in Next.js middleware. Since middleware runs on the server and
runs on every request, this code is not able to run any client side code and it
should have a small bundle size to improve loading and execution times.

Example

An example of when this check could manifest is when middleware transitively
depends on a file that also uses `react` within the same file.

For example:

```ts filename="experiments.ts"
import { createContext, type Context } from 'react';

export function createExperimentContext(): Context<ExperimentContext> {
  return createContext<ExperimentContext>({
    experiments: () => {
```

````
      return EXPERIMENT_DEFAULTS;
    },
  });
}

export async function getExperiments() {
  return activeExperiments;
}
```

```ts filename="middleware.ts"
export async function middleware(
  request: NextRequest,
  event: NextFetchEvent,
): Promise<Response> {
  const experiments = await getExperiments();

  if (experiments.includes('new-marketing-page)) {
    return NextResponse.rewrite(MARKETING_PAGE_URL);
  }
  return NextResponse.next();
}
```

In this example, the `experiments.ts` file both fetches the active experiments
as well as provides helper functions to use experiments on the client in React.

## How to fix

Client dependencies used or transitively depended on by middleware files should
be refactored to avoid depending on the client libraries. In the example above,
the code that is used by middleware to fetch experiments should be moved to a
separate file from the code that provides the React functionality.


---
title: "NEXTJS_NO_DYNAMIC_AUTO"
description: "Prevent usage of force-dynamic as a dynamic page rendering strategy."
last_updated: "2026-01-16T02:19:27.955Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_NO_DYNAMIC_AUTO"
---

# NEXTJS_NO_DYNAMIC_AUTO

Changing the dynamic behavior of a layout or page using "force-dynamic" is
not recommended in App Router. This is because this will force only dynamic rendering
of those pages and opt-out "fetch" request from the fetch cache. Furthermore, opting
out will also prevent future optimizations such as partially static subtrees and
hybrid server-side rendering, which can significantly improve performance.

See [Next.js Segment Config docs](https://nextjs.org/docs/app/api-reference/file-conventions/route-segment-config)
for more information on the different migration strategies that can be used and how
they work.

## How to fix

Usage of `force-dynamic` can be avoided and instead `no-store` or `fetch` calls
can be used instead. Alternatively, usage of `cookies()` can also avoid the need
to use `force-dynamic`.

```js
// Example of how to use `no-store` on `fetch` calls.
const data = fetch(someURL, { cache: 'no-store' });
```


---
title: "NEXTJS_NO_FETCH_IN_SERVER_PROPS"
description: "Prevent relative fetch calls in getServerSideProps from being added to Next.js applications."
last_updated: "2026-01-16T02:19:27.957Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_NO_FETCH_IN_SERVER_PROPS"
---

# NEXTJS_NO_FETCH_IN_SERVER_PROPS

Since both `getServerSideProps` and API routes run on the server, calling `fetch` on a non-relative
URL will trigger an additional network request.

## How to fix

Instead of using `fetch` to make a call to the API route, you can instead share the code in a shared
library or module to avoid another network request. You can then import this hared logic and call directly
within your `getServerSideProps` function, avoiding additional network requests entirely.


---
title: "NEXTJS_NO_GET_INITIAL_PROPS"
description: "Requires any use of getInitialProps in Next.js pages be reviewed and approved, and encourages using getServerSideProps or g
last_updated: "2026-01-16T02:19:27.962Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_NO_GET_INITIAL_PROPS"
---

# NEXTJS_NO_GET_INITIAL_PROPS

`getInitialProps` is an older Next.js API for server-side rendering that can usually be replaced with
`getServerSideProps` or `getStaticProps` for more performant and secure code.

`getInitialProps` runs on both the server and the client after page load, so the JavaScript bundle will
contain any dependencies used by `getInitialProps`. This means that it is possible for unintended code to
be included in the client side bundle, for example, code that should only be used on the server such as
````
````

database connections.

If you need to avoid a server-round trip when performing a client side transition, `getInitialProps` could be used.
However, if you do not, `getServerSideProps` is a good API to use instead so that the code remains on the server and
does not bloat the JavaScript bundle, or `getStaticProps` can be used if the page can be statically generated at build time.

This rule is for highlighting these concerns and while there are still valid use cases for using `getInitialProps` if you do need to
do data fetching on both the client and the server, they should be reviewed and approved.

Example

An example of when this check would fail:

```ts filename="src/pages/index.tsx"
import { type NextPage } from 'next';

const Home: NextPage = ({ users }) => {
  return (
    <ul>
      {users.map((user) => (
        <li key={user.id}>{user.name}</li>
      ))}
    </ul>
  );
};

Home.getInitialProps = async () => {
  const res = await fetch('https://api.github.com/repos/vercel/next.js');
  const json = await res.json();
  return { stars: json.stargazers_count };
};

export default Home;
```

In this example, the `getInitialProps` function is used to fetch data from an API,
but it isn't necessary that we fetch the data on both the client and the server so we can fix it below.

How to fix

Instead, we should use `getServerSideProps` instead of `getInitialProps`:

```ts filename="src/pages/index.tsx"
import { type GetServerSideProps } from 'next';

const Home = ({ users }) => {
  return (
    <ul>
      {users.map((user) => (
        <li key={user.id}>{user.name}</li>
      ))}
    </ul>
  );
};

export getServerSideProps: GetServerSideProps = async () => {
  const res = await fetch('https://api.github.com/repos/vercel/next.js');
  const json = await res.json();
  return {
    props: {
      stars: json.stargazers_count
    },
  };
};

export default Home;
```

---
title: "NEXTJS_NO_PRODUCTION_SOURCE_MAPS"
description: "Applications using Next.js should not enable production source maps so that they don"
last_updated: "2026-01-16T02:19:27.975Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_NO_PRODUCTION_SOURCE_MAPS"

NEXTJS_NO_PRODUCTION_SOURCE_MAPS

Enabling production source maps in your Next.js application will publicly share your
application's source code and should be done with caution. This rule flags any
usage of `productionBrowserSourceMaps` for review. If intentional, the exception
should be added to an allowlist.

For further reading, see:

- [`productionBrowserSourceMaps` documentation](https://nextjs.org/docs/app/api-reference/next-config-js/productionBrowserSourceMaps)

Examples

This rule will catch the following code.

```next.config.js {2}
module.exports = {
  productionBrowserSourceMaps: true,
};
```

How to fix

To fix this issue, either set the value of `productionBrowserSourceMaps` configuration to false,

or if intentional add an exception to an allowlist.

Considerations

Tradeoffs of Disabling Source Maps

Disabling source maps in production has the benefit of not exposing your source code publicly, but it also means that errors in productio

Protected Deployments

For [protected deployments](/docs/security/deployment-protection/methods-to-protect-deployments), it is generally safe to enable source m

Third-Party Error Tracking Services

If you use a third-party error tracking service like [Sentry](https://sentry.io/), you can safely enable source maps by:

1. Uploading the source maps to your error tracking service
2. Emptying or deleting the source maps before deploying to production

Many third-party providers like Sentry offer built-in configuration options to automatically delete sourcemaps after uploading them. Chec

If you need to implement this manually, you can use an approach like this:

```ts
// Empty the source maps after uploading them to your error tracking service
const sourcemapFiles = await findFiles('.next', /\.js\.map$/);
await Promise.all(
  sourcemapFiles.map(async (file) => {
    await writeFile(file, '', 'utf8');
  }),
);
```

---
title: "NEXTJS_NO_SELF_HOSTED_VIDEOS"
description: "Prevent video files from being added to Next.js applications."
last_updated: "2026-01-16T02:19:27.979Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_NO_SELF_HOSTED_VIDEOS"

NEXTJS_NO_SELF_HOSTED_VIDEOS

Video files, which are typically large, can consume a lot of bandwidth for
your Next.js application. Video files are better served from a dedicated video
CDN that is optimized for serving videos.

How to fix

Vercel Blob can be used for storing and serving large files such as videos.

You can use either [server uploads or client uploads](/docs/storage/vercel-blob#server-and-client-uploads) depending on the file size:

- [Server uploads](/docs/storage/vercel-blob/server-upload) are suitable for files up to **4.5 MB**
- [Client uploads](/docs/storage/vercel-blob/client-upload) allow for uploading larger files directly from the browser to Vercel Blob, su

See the [best practices for hosting videos on Vercel](/kb/guide/best-practices-for-hosting-videos-on-vercel-nextjs-mp4-gif) guide to lear

---
title: "NEXTJS_NO_TURBO_CACHE"
description: "Prevent Turborepo from caching the Next.js .next/cache folder to prevent an oversized cache."
last_updated: "2026-01-16T02:19:27.982Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_NO_TURBO_CACHE"

NEXTJS_NO_TURBO_CACHE

This rule prevents the `.next/cache` folder from being added to the Turborepo cache.
This is important because including the `.next/cache` folder in the Turborepo cache can cause
the cache to grow to an excessive size. Vercel also already includes this cache in the build
container cache.

Examples

The following `turbo.json` config will be caught by this rule for Next.js apps:

```json filename="turbo.json" {5}
{
  "extends": ["//"],
  "pipeline": {
    "build": {
      "outputs": [".next/**"]
    }
  }
}
```

How to fix

To fix, add `"!.next/cache/**"` to the list of outputs for the task.

```json filename="turbo.json" {5}
{
  "extends": ["//"],
  "pipeline": {
    "build": {
      "outputs": [".next/**", "!.next/cache/**"]
    }
  }
```

}
```

---
title: "NEXTJS_REQUIRE_EXPLICIT_DYNAMIC"
description: "Requires explicitly setting the "
last_updated: "2026-01-16T02:19:27.996Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_REQUIRE_EXPLICIT_DYNAMIC"
---

# NEXTJS_REQUIRE_EXPLICIT_DYNAMIC

> **⚠️ Warning:** This rule conflicts with the experimental Next.js feature [Partial
> Prerendering
> (PPR)](https://vercel.com/blog/partial-prerendering-with-next-js-creating-a-new-default-rendering-model).
> If you enable PPR in your Next.js app, you should not enable this rule.

For convenience, Next.js defaults to automatically selecting the rendering mode
for pages and routes.

Whilst this works well, it also means that rendering modes can be changed
unintentionally (i.e. through an update to a component that a page depends on).
These changes can lead to unexpected behaviors, including performance issues.

To mitigate the chance that rendering modes change unexpectedly, you should
explicitly set the `dynamic` route segment option to the desired mode. Note
that the default value is `auto`, which will not satisfy this rule.

By default, this rule is disabled. To enable it, refer to
[customizing Conformance](/docs/conformance/customize).

For further reading, see:

- [Next.js File Conventions: Route Segment Config](https://nextjs.org/docs/app/api-reference/file-conventions/route-segment-config#dynami

## Examples

This rule will catch any pages or routes that:

- Do not have the `dynamic` option set to a valid value.
- Have the `dynamic` option set to `'auto'` (which is the default value).

In the following example, the page component does not have the `dynamic` route
segment option set.

```tsx filename="app/page.tsx"
export default function Page() {
  // ...
}
```

The next example sets the `dynamic` route segment option, however it sets it to
`'auto'`, which is already the default behavior and will not satisfy this rule.

```tsx filename="app/dashboard/page.tsx" {1}
export const dynamic = 'auto';

export default function Page() {
  // ...
}
```

How to fix

If you see this issue in your codebase, you can resolve it by explicitly
setting the `dynamic` route segment option for the page or route.

In this example, the `dynamic` route segment option is set to `error`, which
forces the page to static, and will throw an error if any components use
[dynamic functions](https://nextjs.org/docs/app/building-your-application/rendering/server-components#server-rendering-strategies#dynamic
or uncached data.

```tsx filename="app/page.tsx" {1}
export const dynamic = 'error';

export default function Page() {
  const text = 'Hello world';
  return <div>{text}</div>;
}
```

---
title: "NEXTJS_SAFE_NEXT_PUBLIC_ENV_USAGE"
description: "Usage process.env.NEXT_PUBLIC_* environment variables must be allowlisted."
last_updated: "2026-01-16T02:19:28.001Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_SAFE_NEXT_PUBLIC_ENV_USAGE"

NEXTJS_SAFE_NEXT_PUBLIC_ENV_USAGE

The use of `process.env.NEXT_PUBLIC_*` environment variables may warrant a review from other developers to ensure there are no unintended

When enabled, this rule requires that all usage of `NEXT_PUBLIC_*` must be included in the [allowlist](https://vercel.com/docs/conformanc

Examples

This rule will catch any pages or routes that are using `process.env.NEXT_PUBLIC_*` environment variables.
```

In the following example, we are using a local variable to initialize our analytics service. As the variable will be visible in the clien

```tsx filename="app/dashboard/page.tsx" {1}
setupAnalyticsService(process.env.NEXT_PUBLIC_ANALYTICS_ID);

function HomePage() {
  return <h1>Hello World</h1>;
}

export default HomePage;
```

How to fix

If you hit this issue, include the entry in the [Conformance allowlist file](https://vercel.com/docs/conformance/allowlist).

---
title: "NEXTJS_SAFE_SVG_IMAGES"
description: "Prevent dangerouslyAllowSVG without Content Security Policy in Next.js applications."
last_updated: "2026-01-16T02:19:28.005Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_SAFE_SVG_IMAGES"

NEXTJS_SAFE_SVG_IMAGES

SVG can do many of the same things that HTML/JS/CSS can, meaning that it can be dangerous to execute SVG
as this can lead to vulnerabilities without proper [Content Security Policy](https://nextjs.org/docs/advanced-features/security-headers)

How to fix

If you need to serve SVG images with the default Image Optimization API, you
can set `dangerouslyAllowSVG` inside your `next.config.js`:

```js filename="next.config.js"
module.exports = {
  images: {
    dangerouslyAllowSVG: true,
    contentDispositionType: 'attachment',
    contentSecurityPolicy: "default-src 'self'; script-src 'none'; sandbox;",
  },
};
```

In addition, it is strongly recommended to also set `contentDispositionType` to
force the browser to download the image, as well as `contentSecurityPolicy` to
prevent scripts embedded in the image from executing.

---
title: "NEXTJS_SAFE_URL_IMPORTS"
description: "Prevent unsafe URL Imports from being added to Next.js applications."
last_updated: "2026-01-16T02:19:28.018Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_SAFE_URL_IMPORTS"

NEXTJS_SAFE_URL_IMPORTS

URL imports are an experimental feature that allows you to import modules directly
from external servers (instead of from the local disk). When you opt-in, and
supply URL prefixes inside `next.config.js`, like so:

```js filename="next.config.js"
module.exports = {
  experimental: {
    urlImports: ['https://example.com/assets/', 'https://cdn.skypack.dev'],
  },
};
```

If any of the URLs have not been added to the safe import comformance configuration,
then this will cause this rule to fail.

How to fix

Engineers should reach out to the appropriate engineer(s) or team(s) for a
security review of the URL import configuration.

When requesting a review, please provide as much information as possible around
the proposed URL being added, and if there any security implications for using
the URL.

If this URL is deemed safe for general use, it can be added to the list of approved URL imports. This can be done by following the [Custo

```json filename="conformance.config.jsonc"
"NEXTJS_SAFE_URL_IMPORTS": {
  urlImports: [theUrlToAdd],
}
```

---
title: "NEXTJS_UNNEEDED_GET_SERVER_SIDE_PROPS"
description: "Catches usages of getServerSideProps that could use static rendering instead, improving the performance of those pages."
last_updated: "2026-01-16T02:19:28.035Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_UNNEEDED_GET_SERVER_SIDE_PROPS"

NEXTJS_UNNEEDED_GET_SERVER_SIDE_PROPS

This rule will analyze each Next.js page's `getServerSideProps` to see if the context parameter is being used and if not then it will fail.

When using `getServerSideProps` to render a Next.js page on the server, if the page doesn't require any information from the request, consider using [SSG](https://nextjs.org/docs/basic-features/data-fetching/get-static-props) with `getStaticProps`. If you are using `getServerSideProps` to refresh the data on each page load, consider using [ISR](https://nextjs.org/docs/basic-features/data-fetching/incremental-static-regeneration) instead with a `revalidate` property to control how often the page is regenerated. If you are using `getServerSideProps` to randomize the data on each page load, consider moving that logic to the client instead and use `getStaticProps` to reuse the statically generated page.

Example

An example of when this check would fail:

```tsx filename="src/pages/index.tsx"
import { type GetServerSideProps } from 'next';

export const getServerSideProps: GetServerSideProps = async () => {
  const res = await fetch('https://api.github.com/repos/vercel/next.js');
  const json = await res.json();
  return {
    props: { stargazersCount: json.stargazers_count },
  };
};

function Home({ stargazersCount }) {
  return <h1>The Next.js repo has {stargazersCount} stars.</h1>;
}

export default Home;
```

In this example, the `getServerSideProps` function is used to pass data from an API to the page, but it isn't using any information from the context argument so `getServerSideProps` is unnecessary.

How to fix

Instead, we can convert the page to use [SSG](https://nextjs.org/docs/basic-features/data-fetching/get-static-props) with `getStaticProps`. This will generate the page at build time and serve it statically. If you need the page to be updated more frequently, then you can also use [ISR](https://nextjs.org/docs/basic-features/data-fetching/incremental-static-regenerat with the revalidate option:

```tsx filename="src/pages/index.tsx"
import { type GetStaticProps } from 'next';

export const getStaticProps: GetStaticProps = async () => {
  const res = await fetch('https://api.github.com/repos/vercel/next.js');
  const json = await res.json();
  return {
    props: { stargazersCount: json.stargazers_count },
    revalidate: 60, // Using ISR, regenerate the page every 60 seconds
  };
};

function Home({ stargazersCount }) {
  return <h1>The Next.js repo has {stargazersCount} stars.</h1>;
}

export default Home;
```

Or, you can use information from the context argument to customize the page:

```tsx filename="src/pages/index.tsx"
import { type GetServerSideProps } from 'next';

export const getServerSideProps: GetServerSideProps = async (context) => {
  const res = await fetch(
    `https://api.github.com/repos/vercel/${context.query.repoName}`,
  );
  const json = await res.json();
  return {
    props: {
      repoName: context.query.repoName,
      stargazersCount: json.stargazers_count,
    },
  };
};

function Home({ repoName, stargazersCount }) {
  return (
    <h1>
      The {repoName} repo has {stargazersCount} stars.
    </h1>
  );
}

export default Home;
```

---
title: "NEXTJS_USE_NATIVE_FETCH"
description: "Requires using native "
last_updated: "2026-01-16T02:19:28.043Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_USE_NATIVE_FETCH"

NEXTJS_USE_NATIVE_FETCH

Next.js extends the native [Web `fetch` API](https://nextjs.org/docs/app/api-reference/functions/fetch)
with additional caching capabilities which means third-party fetch libraries are not needed.
Including these libraries in your app can increase bundle size and negatively impact performance.

This rule will detect any usage of the following third-party fetch libraries:

- `isomorphic-fetch`
- `whatwg-fetch`
- `node-fetch`
- `cross-fetch`
- `axios`

If there are more libraries you would like to restrict,
consider using a [custom rule](https://vercel.com/docs/conformance/custom-rules).

By default, this rule is disabled. You can enable it by
[customizing Conformance](/docs/conformance/customize).

For further reading, see:

- https://nextjs.org/docs/app/api-reference/functions/fetch
- https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API

Examples

This rule will catch the following code.

```tsx {1}
import fetch from 'isomorphic-fetch';

export async function getAuth() {
  const auth = await fetch('/api/auth');
  return auth.json();
}
```

How to fix

Replace the third-party fetch library with the native `fetch` API Next.js provides.

---
title: "NEXTJS_USE_NEXT_FONT"
description: "Requires using next/font to load local fonts and fonts from supported CDNs."
last_updated: "2026-01-16T02:19:28.058Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_USE_NEXT_FONT"

NEXTJS_USE_NEXT_FONT

[`next/font`](https://nextjs.org/docs/pages/api-reference/components/font)
automatically optimizes fonts and removes external network requests for
improved privacy and performance.

By default, this rule is disabled. Enable it by
[customizing Conformance](/docs/conformance/customize).

This means you can optimally load web fonts with zero layout shift, thanks to
the underlying CSS size-adjust property used.

For further reading, see:

- https://nextjs.org/docs/basic-features/font-optimization
- https://nextjs.org/docs/pages/api-reference/components/font
- https://www.lydiahallie.io/blog/optimizing-webfonts-in-nextjs-13

Examples

This rule will catch the following code.

```css {3-4}
@font-face {
  font-family: Foo;
  src:
    url(https://fonts.gstatic.com/s/roboto/v30/KFOiCnqEu92Fr1Mu51QrEzOdL-vwnYh2eg.woff2)
      format('woff2'),
    url(/custom-font.ttf) format('truetype');
  font-display: block;
  font-style: normal;
  font-weight: 400;
}
```

```ts {3-6}
function App() {
  return (
    <link
      href="https://fonts.googleapis.com/css2?family=Krona+One&display=optional"
      rel="stylesheet"
    />
  );
}
```

How to fix

Replace any `@font-face` at-rules and `link` elements that are caught by this
rule with [`next/font`](https://nextjs.org/docs/api-reference/next/font).

---
title: "NEXTJS_USE_NEXT_IMAGE"
description: "Requires that next/image is used for all images."
last_updated: "2026-01-16T02:19:28.068Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_USE_NEXT_IMAGE"

NEXTJS_USE_NEXT_IMAGE

The Next.js Image component ([`next/image`](https://nextjs.org/docs/pages/api-reference/components/image))
extends the HTML `<img>` element with features for automatic image optimization.

It optimizes image sizes for different devices using modern image formats,
improves visual stability by preventing layout shifts during image loading,
and speeds up page loads with lazy loading and optional blur-up placeholders.

Additionally, it provides the flexibility of on-demand image resizing, even for
images hosted on remote servers. This may incur costs from your managed hosting
provider (see [below](#important-note-on-costs) for more information)

By default, this rule is disabled. Enable it by
[customizing Conformance](/docs/conformance/customize).

For further reading, see:

- https://nextjs.org/docs/app/building-your-application/optimizing/images
- https://nextjs.org/docs/pages/api-reference/components/image

Important note on costs

Using image optimization may incur costs from your managed hosting
provider. You can opt out of image optimization by setting the optional
[`unoptimized` prop](https://nextjs.org/docs/pages/api-reference/components/image#unoptimized).

Please check with your hosting provider for details.

- [Vercel pricing](https://vercel.com/pricing)
- [Cloudinary pricing](https://cloudinary.com/pricing)
- [imgix pricing](https://imgix.com/pricing)

Important note on self-hosting

If self-hosting, you'll need to install the optional package
[`sharp`](https://www.npmjs.com/package/sharp), which Next.js will use to
optimize images. Optimized images will require more available storage on your
server.

Examples

This rule will catch the following code.

```tsx {2}
function App() {
  return <img src="/media/image.png" alt="Example" />;
}
```

The following code will not be caught by this rule.

```tsx
function App() {
  return (
    <picture>
      <source srcSet="/hero.avif" type="image/avif" />
      <source srcSet="/hero.webp" type="image/webp" />
      <img src="/hero.jpg" alt="Landscape picture" width={800} height={500} />
    </picture>
  );
}
```

How to fix

Replace any `<img>` elements that are caught by this rule with
[`next/image`](https://nextjs.org/docs/pages/api-reference/components/image).

Again, please check with your managed hosting provider for image optimization
costs.

---
title: "NEXTJS_USE_NEXT_SCRIPT"
description: "Requires that next/script is used for all scripts."
last_updated: "2026-01-16T02:19:28.074Z"
source: "https://vercel.com/docs/conformance/rules/NEXTJS_USE_NEXT_SCRIPT"

NEXTJS_USE_NEXT_SCRIPT

[`next/script`](https://nextjs.org/docs/pages/api-reference/components/script)
automatically optimizes scripts for improved performance through customizable
loading strategies. By default, `next/script` loads scripts so that they're
non-blocking, meaning that they load after the page has loaded.

Additionally, `next/script` has built in event handlers for common events such
as `onLoad` and `onError`.

By default, this rule is disabled. Enable it by
[customizing Conformance](/docs/conformance/customize).

For further reading, see:

- https://nextjs.org/docs/pages/building-your-application/optimizing/scripts
- https://nextjs.org/docs/pages/api-reference/components/script

Examples

This rule will catch the following code.

```tsx {2}
function insertScript() {
  const script = document.createElement('script');
  script.src = process.env.SCRIPT_PATH;
  document.body.appendChild(script);
}
```

```tsx {3-5}
function App() {
  return (
    <script
      dangerouslySetInnerHTML={{ __html: "console.log('Hello world');" }}
    />
  );
}
```

How to fix

Replace any `document.createElement('script')` calls and `<script>`
elements that are caught by this rule with [`next/script`](https://nextjs.org/docs/pages/api-reference/components/script).

---
title: "NO_ASSIGN_WINDOW_LOCATION"
description: "Prevent unsafe assignment to window.location.href in your application."
last_updated: "2026-01-16T02:19:28.079Z"
source: "https://vercel.com/docs/conformance/rules/NO_ASSIGN_WINDOW_LOCATION"

NO_ASSIGN_WINDOW_LOCATION

Direct assignments to "window.location.href" or "window.location" should be avoided due to possible XSS attacks that can occur from lack of sanitization of input to the "href".

How to fix

The recommended approach for Next.js applications is to use a custom `redirectTo` function. This provides a clear way to use `router.push` or `window.location.href` to provide an experience that is best for the user (client-side navigation only, or a full page refresh). Here's an example of how you might do this using Next.js:

Before:

```js filename="my-site.js"
windows.location.href = '/login';
```

After:

```js filename="my-site.js"
router.push('/login');
```

---
title: "NO_CORS_HEADERS"
description: "Warns when CORS header (or header-like) configuration is detected, requiring that configuration to be allowlisted."
last_updated: "2026-01-16T02:19:28.086Z"
source: "https://vercel.com/docs/conformance/rules/NO_CORS_HEADERS"

NO_CORS_HEADERS

Misconfiguring CORS ([Cross Origin Resource Sharing](https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS)) headers can introduce security risks, potentially exposing private and/or secure information such as API keys and user data.

> **💡 Note:** This rule is not meant to block usage of CORS. Instead, it is designed to flag
> potentially risky configuration for review by the appropriate engineer(s) or
> team(s).

For more information around the risks associated with CORS, including testing for potential vulnerabilities, see:

- [OWASP: HTML5 Security Cheat Sheet - Cross Origin Resource Sharing](https://cheatsheetseries.owasp.org/cheatsheets/HTML5_Security_Cheat
- [OWASP: Web Security Testing Guide - Testing Cross Origin Resource Sharing](https://owasp.org/www-project-web-security-testing-guide/la
- [OWASP: CORS OriginHeaderScrutiny](https://owasp.org/www-community/attacks/CORS_OriginHeaderScrutiny)

Examples

The examples below are common approaches to settings CORS headers in JavaScript applications. All of these examples will be caught by this rule.

```ts
response.headers.set('Access-Control-Allow-Origin', '*');

const headers = {
  'Access-Control-Allow-Credentials': true,
};
```

```
const options = {
  headers: [
    {
      key: 'Access-Control-Max-Age',
      value: 600,
    },
  ],
};

const headers = new Headers();
headers.append('Access-Control-Allow-Methods', '*');
```

Additionally, this rule will catch partial matches, such as a template literal.
In this example, the rule will match the `"Access-Control-"` part of the
template literal.

```ts
const headers = new Headers();
headers.append(`Access-Control-${HEADER_TYPE}`, '*');
```

How to fix

Engineers should reach out to the appropriate engineer(s) or team(s) for a
security review of the configuration.

When requesting a review, please provide as much information as possible around
the proposed CORS configuration. Where applicable, include information around
alternative approaches, and why this approach is preferable.

As there are many ways to configure CORS headers in applications, this rule
will match any string that looks like a possible CORS header. We've tried to
mitigate the risk of false-positives, but if they occur they will need to be
added to the allowlists.

---
title: "NO_DANGEROUS_HTML"
description: "Prevent the unsafe creation of DOM via HTML methods in your application."
last_updated: "2026-01-16T02:19:28.095Z"
source: "https://vercel.com/docs/conformance/rules/NO_DANGEROUS_HTML"

NO_DANGEROUS_HTML

Unsafe creation of DOM can be done a variety of ways:

- `element.innerHTML`
- `element.outerHTML`
- `DOMParser.parseFromString()`
- `element.insertAdjacentHTML()`
- `srcdoc` on iframe elements
- `dangerouslySetInnerHTML` prop in React apps

Usage of these methods is deemed an unsafe coding practice as the HTML might result in security vulnerabilities.

How to fix

It is recommended to instead use alternative approaches for HTML construction - such as `document.createElement()` or a HTML sanitizer.

---
title: "NO_DOCUMENT_WRITE"
description: "Prevent unsafe usage of document.write() in your application."
last_updated: "2026-01-16T02:19:28.099Z"
source: "https://vercel.com/docs/conformance/rules/NO_DOCUMENT_WRITE"

NO_DOCUMENT_WRITE

Calls to `document.write()` or `document.writeln()` manipulate DOM directly without any sanitization and should be avoided.

Furthermore, these APIs can also cause performance issues and trigger will clear the page contents if used after page load.

How to fix

Avoid usage of `document.write()` entirely in your application, and instead either use UI framework like React to handle writing to the d
or use safer DOM APIs, such as `document.createElement()` instead.

---
title: "NO_EVAL"
description: "Prevent unsafe usage of eval() in your application."
last_updated: "2026-01-16T02:19:28.105Z"
source: "https://vercel.com/docs/conformance/rules/NO_EVAL"

NO_EVAL

JavaScript's `eval()` function is potentially dangerous, is often misused, and
might cause security issues. Using `eval()` on untrusted code can open an
application up to several different injection attacks.

This rule will also catch eval-like function usage (or *implied eval*), such as
passing a string as the first argument to `setTimeout`.

This is especially dangerous when working with data from external sources.

```ts
const dontDoThis = req.body;
setTimeout(dontDoThis, 1000);
```

For more information on why you should never use evaluation, see the [MDN docs](https://developer.mozilla.org/en-US/docs/Web/JavaScript/R

Example

The lines below (and variations of those) will all be caught by this rule.

```ts
eval('() => console.log("DROP TABLE")');

setTimeout('() => console.log("DROP TABLE")', 1000);

window.setInterval('() => console.log("DROP TABLE")', 1000);

new Function('() => console.log("DROP TABLE")');
```

References

Conformance rules are not type-aware, but will follow variable references
within the current module (or file).

```ts
import { importedVar } from 'foo';

// No error reported, as this rule doesn't have access to the value.
setTimeout(importedVar, 100);

const localVar = 'bar';

// An error will be reported, as the variable was declared in this file.
setTimeout(localVar, 100);
```

How to fix

Avoid usage of this type of evaluation entirely in your application. Instead,
you should write the same functionality as raw code (not within a string).

```ts
setTimeout(() => {
  console.log('Safe usage');
});
```

---
title: "NO_EXTERNAL_CSS_AT_IMPORTS"
description: "Disallows @import at-rules that import from URLs."
last_updated: "2026-01-16T02:19:28.120Z"
source: "https://vercel.com/docs/conformance/rules/NO_EXTERNAL_CSS_AT_IMPORTS"

NO_EXTERNAL_CSS_AT_IMPORTS

Importing CSS through ([`@import`](https://developer.mozilla.org/en-US/docs/Web/CSS/@import))
is render blocking, causing browsers to sequentially download and parse the
imported CSS (a [critical request chain](https://developer.chrome.com/en/docs/lighthouse/performance/critical-request-chains/)).

```css filename="app.module.css"
@import url('https://fonts.googleapis.com/css2?family=Inter');
```

This can result in a [flash of unstyled content (FOUC)](https://en.wikipedia.org/wiki/Flash_of_unstyled_content),
where page content is briefly shown without complete styles until all required
CSS has been downloaded and parsed, along with slower page load times.

Imports to relative paths are processed by frameworks like Next.js, and will
not be affected by this issue.

```css filename="app.module.css"
/* This import is safe. */
@import './globals.css';
```

> **💡 Note:** Note that this rule currently only parses CSS and not CSS written in Less,
> Sass, or other CSS preprocessor syntaxes.

How to fix

If you're importing a font, you can use [`next/font`](https://nextjs.org/docs/basic-features/font-optimization)
which will automatically optimize your fonts (including custom fonts) and
remove external network requests.

If you're importing CSS, such as Bootstrap, avoid loading it from external
sources, such as a CDN or the [Next.js public folder](https://nextjs.org/docs/basic-features/static-file-serving).
Instead, you can import that CSS relatively, or from a package.

```ts filename="layout.tsx"
// CSS imported relatively from a local file.
import './globals.css';
// CSS from a package in `node_modules`.
import 'bootstrap/dist/css/bootstrap.css';

interface RootLayoutProps {
  children: React.ReactNode;
```

```
}

export default function RootLayout({ children }: RootLayoutProps) {
  return (
    <html lang="en">
      <head />
      <body>{children}</body>
    </html>
  );
}
```

---
title: "NO_FETCH_FROM_MIDDLEWARE"
description: "Requires that any fetch call that is depended on transitively by Next.js middleware be reviewed and approved before use."
last_updated: "2026-01-16T02:19:28.125Z"
source: "https://vercel.com/docs/conformance/rules/NO_FETCH_FROM_MIDDLEWARE"

NO_FETCH_FROM_MIDDLEWARE

[Next.js middleware](https://nextjs.org/docs/advanced-features/middleware) runs
code at the Edge. This means that the code is globally distributed. When
middleware makes a `fetch` call, it may be to a backend that is not globally
distributed, in which case the latency of the middleware function will be
really slow. To prevent this, `fetch` calls that can be made from middleware are
flagged and reviewed to make sure that it looks like an appropriate use.

Example

This check will fail when a `fetch` call is detected from Next.js middleware or
transitive dependencies used by the middleware file.

In this example, there are two files. An experiments file asynchronously
fetches experiments using `fetch`. The middleware file uses the experiments
library to fetch the experiments and then decide to rewrite the URL.

```ts filename="experiments.ts"
export async function getExperiments() {
  const res = await fetch('/experiments');
  return res.json();
}
```

```ts filename="middleware.ts"
export async function middleware(
  request: NextRequest,
  event: NextFetchEvent,
): Promise<Response> {
  const experiments = await getExperiments();

  if (experiments.includes('new-marketing-page')) {
    return NextResponse.rewrite(MARKETING_PAGE_URL);
  }
  return NextResponse.next();
}
```

How to fix

The correct fix will depend on the specific situation. If the server that is
being called is globally distributed, then this asynchronous call may be okay.
If not, then the code should try to remove the `fetch` statement to avoid
making a request that would add latency to middleware.

---
title: "NO_INLINE_SVG"
description: "Prevent the use of "
last_updated: "2026-01-16T02:19:28.141Z"
source: "https://vercel.com/docs/conformance/rules/NO_INLINE_SVG"

NO_INLINE_SVG

Preventing the use of `<svg></svg>` inline improves the health of your codebase at the page level.
Using inlined `svg` tags in excess can cause hydration issues, negatively impact the performance of both
the browser and the server rendering.

By default, this rule is disabled. To enable it, refer to
[customizing Conformance](/docs/conformance/customize).

How to fix

If you hit this issue, you can resolve it by using SVGs as an [`<Image>`](https://nextjs.org/docs/pages/api-reference/components/image)
component. Don't forget to enable [`dangerouslyAllowSVG`](https://nextjs.org/docs/pages/api-reference/components/image#dangerouslyallowsv
in your application's `next.config.js` file, and use the `unoptimized` component prop.

```JSX filename=".app/page.js"
import Image from 'next/image'

export default function Page() {
  return (
    <Image
      src="/logo.svg"
      width={100}
      height={100}
      alt="Logo of ACME"
      unoptimized
```

```
    />
  )
}
```

---
title: "NO_INSTANCEOF_ERROR"
description: "Disallows using "
last_updated: "2026-01-16T02:19:28.159Z"
source: "https://vercel.com/docs/conformance/rules/NO_INSTANCEOF_ERROR"

NO_INSTANCEOF_ERROR

A common pattern for checking if an object is an error is to use
`error instanceof Error`.

This pattern is problematic because errors can come from other [realms](https://tc39.es/ecma262/#realm).
Errors from other realms are instantiated from the realm's global `Error`
constructor, and are therefore not instances of the current realm's global
`Error` constructor and will not pass the `instanceof` check.

Some examples of where you might hit this include:

- In Node.js, errors from a workers are instances of `Error` from the worker's
  global environment.
- In browser environments, errors from `iframe` are instances of `Error` from
  the `iframe`'s global environment (i.e. `iframe.contentWindow.Error`).

By default, this rule is disabled. To enable it, refer to
[customizing Conformance](/docs/conformance/customize).

Examples

In this example, an error is returned from a [`vm`](https://nodejs.org/api/vm.html) context. As this error was created in a different rea

```tsx {6}
const vm = require('node:vm');

const context = vm.createContext({});
const error = vm.runInContext('new Error()', context);

if (error instanceof Error) {
  // Returns `false` because `error` is from a different realm.
}
```

How to fix

Node.js

You can use [`isNativeError`](https://nodejs.org/api/util.html#utiltypesisnativeerrorvalue)
in Node.js environments, which will return `true` for errors from other realms.

```tsx {7}
import { isNativeError } from 'node:util/types';
const vm = require('node:vm');

const context = vm.createContext({});
const error = vm.runInContext('new Error()', context);

if (isNativeError(error)) {
  // ...
}
```

Browsers

Use a library like [`is-error`](https://www.npmjs.com/package/is-error) to
ensure you cover errors from other realms.

You can also use `Object.prototype.toString.call(error) === '[object Error]'`
in some cases. This method will not work for custom errors, and you'll need to
traverse the prototype chain (i.e. `Object.getPrototypeOf(error)`)to handle
those cases.

The following code is a simplified version of the code used in the `is-error`
library:

```ts
function isError(error) {
  if (typeof error !== 'object') {
    return false;
  }

  if (error instanceof Error) {
    return true;
  }

  let currentError = error;
  while (currentError) {
    if (Object.prototype.toString.call(currentError) === '[object Error]') {
      return true;
    }
    currentError = Object.getPrototypeOf(currentError);
  }

  return false;
}
```

```

---
title: "NO_MIXED_ASYNC_MODULES"
description: "Prevent imports to modules that contain top-level awaits in your applications."
last_updated: "2026-01-16T02:19:28.164Z"
source: "https://vercel.com/docs/conformance/rules/NO_MIXED_ASYNC_MODULES"
---

# NO_MIXED_ASYNC_MODULES

Top-level await expressions in modules that are imported by other modules in sync
prevent possible lazy module optimizations from being deployed on the module containing
the top-level await.

One such optimization this prevents is inline lazy imports. Inline lazy imports allow
for modules to be lazily evaluated and executed when they're used, rather than at
initialization time of the module that uses them, improving initialization performance.

This is particularly impactful for modules that might only be used conditionally or
given a user's interaction which might happen much latter in an application. Without this optimization, the module initialization times,

## How to fix

Consider refactoring the import to a dynamic import instead, or removing the top-level await
in favor of standard import.

If a top-level await is important, then it's important that any other modules importing the
module with the top-level await do so dynamically, as to avoid affecting initialization performance.

For example, this can be refactored:

```js
// Contains a top-level await
import { asyncConfig } from 'someModule';

function doSomething(data) {
  processData(data, asyncConfig);
}
```

To this:

```js
function doSomething(data) {
  import('someModule').then(({ asyncConfig }) => {
    processData(data, asyncConfig);
  });
}
```

Or this:

```js
import { asyncConfig } from 'someModule';

// Note the async keyword on the function
async function doSomething(data) {
  processData(data, asyncConfig);
}
```

---
title: "NO_POSTINSTALL_SCRIPT"
description: "Prevent the use of "
last_updated: "2026-01-16T02:19:28.176Z"
source: "https://vercel.com/docs/conformance/rules/NO_POSTINSTALL_SCRIPT"

NO_POSTINSTALL_SCRIPT

Modifying, adding, or updating any dependencies in your application triggers the execution of the `"postinstall"` script. Consequently, i

How to fix

If you hit this issue, you can resolve it by removing the `"postinstall"` script in the `package.json` file.

```JSX filename="package.json" {3}
{
  "scripts": {
    "postinstall": "sleep 360"
  },
}
```

---
title: "NO_SERIAL_ASYNC_CALLS"
description: "Prevent blocking serial async await calls in your applications."
last_updated: "2026-01-16T02:19:28.183Z"
source: "https://vercel.com/docs/conformance/rules/NO_SERIAL_ASYNC_CALLS"

NO_SERIAL_ASYNC_CALLS

Sequential execution of async/await calls can significantly impact performance because
each await call prevents further execution until resolving its Promise. This rule aims to
refactor sequential async/await calls into parallel executions to enhance performance.
```

You should note that this rule might not flag some async/await usage patterns. For example:

- Patterns involving conditional statements
- Call expressions
- Patterns that await in a manner that suggests non-serial dependencies between calls

For instance, scenarios where async calls depend conditionally on each other or are part of
complex expressions are not flagged. This includes cases where one async call's outcome is
necessary for subsequent calls, requiring serial execution due to logical or dependency reasons.

The following example **will not** be flagged by this rule:

```ts
async function updateDatabase() {
  const result1 = await async1();
  const result2 = await async2();
  doSomething(result1, result2);
}
```

These patterns fall outside the scope of this rule because safely suggesting parallelization requires more context,
and the rule uses conservative heuristics to avoid false positives.

How to fix

Instead, of executing async logic sequentially, opt to refactor the logic so it can be run parallel.

This can be fixed using `Promise.all`:

```js
export async function getStaticProps() {
  const firstThing = await getFirstThing();
  const secondThing = await getSecondThing();

  return {
    props: {
      firstThing,
      secondThing,
    },
  };
}
```

We can extract both `await` expressions into a single `Promise.all`, as follows:

```js
export async function getStaticProps() {
  const [firstThing, secondThing] = await Promise.all([
    getFirstThing(),
    getSecondThing(),
  ]);

  return {
    props: {
      firstThing,
      secondThing,
    },
  };
}
```

---
title: "NO_UNNECESSARY_PROP_SPREADING"
description: "Disallows the usage of object spreading in a JSX component."
last_updated: "2026-01-16T02:19:28.196Z"
source: "https://vercel.com/docs/conformance/rules/NO_UNNECESSARY_PROP_SPREADING"

NO_UNNECESSARY_PROP_SPREADING

This rule detects the usage of the spread operator when spreading an object as a prop within a JSX component.

When spreading an object in the component, the data types of the object's properties are not validated, potentially causing unexpected ru

Examples

In the following example, the `Header` component contains an object with the spread operator being applied to it.

We don't know if the props that the `Header` component reads will accept all the values contained in the `foo` object.

```tsx filename="app/dashboard/page.tsx" {2}
function HomePage() {
  return <Header {...{ foo }}>Hello World</Header>;
}

export default HomePage;
```

How to fix

You can remove the spread operator from the JSX component, and list all props explicitly.

```tsx filename="app/dashboard/page.tsx" {2}
function HomePage() {
  return (
    <Header bar={foo.bar} baz={foo.baz}>
      Hello World
```

```
    </Header>
  );
}

export default HomePage;
```

In the example above, [TypeScript](https://www.typescriptlang.org/) will be able to type-check all props.

---
title: "NO_VARIABLE_IMPORT_REFERENCES"
description: "import and require statements must be passed string literals to avoid arbitrary user access to code."
last_updated: "2026-01-16T02:19:28.201Z"
source: "https://vercel.com/docs/conformance/rules/NO_VARIABLE_IMPORT_REFERENCES"

NO_VARIABLE_IMPORT_REFERENCES

`import` and `require` statements load code from another file. When the
location of the import is influenced by user input, the user may be able to
load code that would otherwise be inaccessible to them. Such imports should
protect against this by adding guards to make sure that arbitrary code can not
be loaded from the import statement.

Example

The following code would be flagged by this rule:

```typescript
function loadDynamicCode(moduleName: string) {
  return import(moduleName);
}
```

In this example, it can not be guaranteed that the `moduleName` that is
provided would not be arbitrary input that could load unintended code.

How to fix

Instances of this rule should be reviewed by a knowledgeable security person.
If user input is used to select which module is loaded, guards against arbitrary
strings should be added, such as only allowing access to a list of valid options.
If no user input is involved in the import, then this code could be allowlisted
after being reviewed by a security team member, but developers should be careful
to ensure that only the desired code can be loaded.

---
title: "PACKAGE_JSON_DESCRIPTION_REQUIRED"
description: "Requires that every package.json file has the description field set."
last_updated: "2026-01-16T02:19:28.223Z"
source: "https://vercel.com/docs/conformance/rules/PACKAGE_JSON_DESCRIPTION_REQUIRED"

PACKAGE_JSON_DESCRIPTION_REQUIRED

This check ensures that every `package.json` has a `description` field.
This field is used to describe the workspace's purpose within the monorepo.

See the [Node.js docs](https://nodejs.org/api/packages.html#description) for more information.

Related Rules

- [PACKAGE_JSON_NAME_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_NAME_REQUIRED)
- [PACKAGE_JSON_PRIVATE_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_PRIVATE_REQUIRED)
- [PACKAGE_JSON_TYPE_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_TYPE_REQUIRED)
- [PACKAGE_JSON_SIDE_EFFECTS_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_SIDE_EFFECTS_REQUIRED)

How to fix

Add the `description` field to the `package.json` file that explains
what the package does and when it should be used.

---
title: "PACKAGE_JSON_DUPLICATE_DEPENDENCIES"
description: "Found duplicate dependencies between the list of dependencies and devDependencies or peerDependencies in a package.json fil
last_updated: "2026-01-16T02:19:28.237Z"
source: "https://vercel.com/docs/conformance/rules/PACKAGE_JSON_DUPLICATE_DEPENDENCIES"

PACKAGE_JSON_DUPLICATE_DEPENDENCIES

Packages that are listed in the `dependencies` section of `package.json` should
not be listed in `devDependencies` or `peerDependencies`. A package in the
`dependencies` section says that the package are required for the functionality
of your workspace, in which case there is no reason to also list it in
`devDependencies` or `peerDependencies`.

Example

This `package.json` file would cause the check to fail:

```json filename="package.json"
{
  "name": "workspace",
  "dependencies": {
    "@next/mdx": "13.1.5"
  },
```

```
  "devDependencies": {
    "@next/mdx": "13.1.5"
  }
}
```

How to fix

If the package is needed to use the package from your workspace, you can remove
the package from the `devDependencies` and `peerDependencies` sections. If the
package is only needed for development of your workspace or if the package is
only needed to express version compatibility requirements and it is not needed
for the functionality of your workspace when people use your package, then it
can be left in `devDependencies` or `peerDependencies` and be removed from
`dependencies`.

---
title: "PACKAGE_JSON_NAME_REQUIRED"
description: "Requires that every package.json file has the name field set to ensure each workspace has a unique identifier."
last_updated: "2026-01-16T02:19:28.259Z"
source: "https://vercel.com/docs/conformance/rules/PACKAGE_JSON_NAME_REQUIRED"

PACKAGE_JSON_NAME_REQUIRED

This check ensures that every `package.json` has a `name` field. This field is important because
it used to identify the workspace in the monorepo.

See the [Node.js docs](https://nodejs.org/api/packages.html#name) for more information.

Related Rules

- [PACKAGE_JSON_DESCRIPTION_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_DESCRIPTION_REQUIRED)
- [PACKAGE_JSON_PRIVATE_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_PRIVATE_REQUIRED)
- [PACKAGE_JSON_TYPE_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_TYPE_REQUIRED)
- [PACKAGE_JSON_SIDE_EFFECTS_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_SIDE_EFFECTS_REQUIRED)

How to fix

Add the `name` field to the `package.json` file that contains a unique name for
this package. The name should be understandable by someone viewing or using the
package as to what it does.

---
title: "PACKAGE_JSON_PRIVATE_REQUIRED"
description: "Requires that every package.json file has the private field set to prevent accidental publishing to npm."
last_updated: "2026-01-16T02:19:28.294Z"
source: "https://vercel.com/docs/conformance/rules/PACKAGE_JSON_PRIVATE_REQUIRED"

PACKAGE_JSON_PRIVATE_REQUIRED

This check ensures that every `package.json` has the `private` field set to true or false.
This field ensures that the workspace is not accidentally published to npm. In a monorepo,
this should be the default to prevent packages from being accidentally published and can be explicitly set to
`false` to indicate that the package can be published.

Related Rules

- [PACKAGE_JSON_NAME_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_NAME_REQUIRED)
- [PACKAGE_JSON_DESCRIPTION_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_DESCRIPTION_REQUIRED)
- [PACKAGE_JSON_TYPE_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_TYPE_REQUIRED)
- [PACKAGE_JSON_SIDE_EFFECTS_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_SIDE_EFFECTS_REQUIRED)

How to fix

Packages should set `private` to `true` unless the package is
intended to be published in which case it can be explicitly set to `false`.

---
title: "PACKAGE_JSON_PRIVATE_REQUIREDPACKAGE_JSON_SIDE_EFFECTS_REQUIRED"
description: "Requires that every package.json file has the sideEffects field set to ensure tree-shaking works optimally."
last_updated: "2026-01-16T02:19:28.298Z"
source: "https://vercel.com/docs/conformance/rules/PACKAGE_JSON_SIDE_EFFECTS_REQUIRED"

PACKAGE_JSON_PRIVATE_REQUIREDPACKAGE_JSON_SIDE_EFFECTS_REQUIRED

This check ensures that every `package.json` has a `sideEffects` field.
The `sideEffects` field is required for shared packages. This field helps bundlers
make assumptions about packages that improve tree shaking, or pruning
files that aren't used and don't have any global side effects.

See https://webpack.js.org/guides/tree-shaking/ for more information.

Related Rules

- [PACKAGE_JSON_NAME_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_NAME_REQUIRED)
- [PACKAGE_JSON_DESCRIPTION_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_DESCRIPTION_REQUIRED)
- [PACKAGE_JSON_PRIVATE_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_PRIVATE_REQUIRED)
- [PACKAGE_JSON_TYPE_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_TYPE_REQUIRED)

How to fix

The `sideEffects` field should be set to `false` unless the code in that workspace has
global side effects, in which case it should be set to `true` or an array of glob
patterns for files that do have side effects.

---
title: "PACKAGE_JSON_TYPE_REQUIRED"
description: "Requires that every package.json file has the type field set to encourage using ES Modules since commonjs is the default."
last_updated: "2026-01-16T02:19:28.302Z"
source: "https://vercel.com/docs/conformance/rules/PACKAGE_JSON_TYPE_REQUIRED"

PACKAGE_JSON_TYPE_REQUIRED

This check ensures that every `package.json` has a `type` field. This field determines
how files within the workspace are treated by default. Files are treated as
[CommonJS](https://nodejs.org/api/modules.html) by default. However, the new recommendation
is to use [ES Modules](https://nodejs.org/api/esm.html).

This field is required so that packages explicitly choose which module format to use,
preferring ES Modules when possible.

See the [Node.js docs](https://nodejs.org/api/packages.html#type) for more information.

Related Rules

- [PACKAGE_JSON_NAME_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_NAME_REQUIRED)
- [PACKAGE_JSON_DESCRIPTION_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_DESCRIPTION_REQUIRED)
- [PACKAGE_JSON_PRIVATE_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_PRIVATE_REQUIRED)
- [PACKAGE_JSON_SIDE_EFFECTS_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_SIDE_EFFECTS_REQUIRED)

How to fix

The `type` field should be set to `module` when possible, although there are still situations
where `commonjs` has to be used.

---
title: "PACKAGE_MANAGEMENT_NO_CIRCULAR_IMPORTS"
description: "Circular imports between two files are not allowed."
last_updated: "2026-01-16T02:19:28.305Z"
source: "https://vercel.com/docs/conformance/rules/PACKAGE_MANAGEMENT_NO_CIRCULAR_IMPORTS"

PACKAGE_MANAGEMENT_NO_CIRCULAR_IMPORTS

This check ensures that there is no path through import statements back to the
original file. This helps to keep dependencies between files clean, which aids
in dependency analysis and refactoring.

Example

```ts filename="component-a.ts"
import Badge from './component-b';

export function withHigherOrderComponent({ children }) {
  return <div>{children}</div>;
}

export function Page() {
  return (
    <div>
      <Badge />
    </div>
  );
}
```

```ts filename="component-b.ts"
import { withHigherOrderComponent } from './component-a';

function Badge() {
  return <div>Badge</div>;
}

export default withHigherOrderComponent(Badge);
```

How to fix

The exports in the file that has a circular import should be refactored so that
the circular import doesn't exist anymore. This might be fixed by moving some
of the exports in a file to a separate file so that the imports don't cause a
circular import. In some cases, it may be necessary to refactor the code to
avoid the circular import.

---
title: "PACKAGE_MANAGEMENT_NO_UNRESOLVED_IMPORTS"
description: "Import statements that can not be resolved to a local file or a package from package.json dependencies are not allowed."
last_updated: "2026-01-16T02:19:28.308Z"
source: "https://vercel.com/docs/conformance/rules/PACKAGE_MANAGEMENT_NO_UNRESOLVED_IMPORTS"

PACKAGE_MANAGEMENT_NO_UNRESOLVED_IMPORTS

All imports must be able to be resolved to a file local to the workspace or a
package declared as a dependency within the `package.json` file. This ensures
that the workspace has not missed any dependencies and is not relying on
global dependencies.

Example

```ts filename="component.ts"
import { useState } from 'react';
import { useRouter } from 'next/router';
```

The `package.json` is missing a dependency on the `next` package.

```json filename="package.json"
{
  "name": "shared-component-pkg",
  "dependencies": {
    "react": "19.0.0-beta-4508873393-20240430"
  }
}
```

How to fix

If the workspace is missing a package dependency, add the appropriate one to
the `package.json` file of the workspace. In the example above, a dependency
on the `next` package should be added.

---
title: "PACKAGE_MANAGEMENT_REQUIRED_README"
description: "Every workspace is required to have a README.md file in the root of the workspace."
last_updated: "2026-01-16T02:19:28.311Z"
source: "https://vercel.com/docs/conformance/rules/PACKAGE_MANAGEMENT_REQUIRED_README"

PACKAGE_MANAGEMENT_REQUIRED_README

A `README.md` file helps orient readers to the purpose of a workspace and
instructions how to use it, which makes it straightforward for people browsing the code
to understand its purpose, whether they should use it, and how to make changes
to the code.

How to fix

Add a `README.md` file in the workspace directory. This file can contain a
description of the package, and any instructions for developers or users to
build or use the package.

---
title: "REACT_NO_STATIC_IMPORTS_IN_EVENT_HANDLERS"
description: "Prevent static imports that are referenced only in React event handlers from being eagerly loaded in React components."
last_updated: "2026-01-16T02:19:28.314Z"
source: "https://vercel.com/docs/conformance/rules/REACT_NO_STATIC_IMPORTS_IN_EVENT_HANDLERS"

REACT_NO_STATIC_IMPORTS_IN_EVENT_HANDLERS

React event handlers are async, and as such, this means we can defer loading the
associated code until we interact with the UI, triggering that event handler. Specifically, this
means we can improve initial code size and the overhead of loading the code until it is actually needed.

How to fix

Instead of using static imports at the top of your module, you can use dynamic imports as needed in your React event handlers.

Before:

```js
import foo from 'foo';

const onClick = () => {
  foo.doSomething();
};
```

After:

```js
const onClick = () => {
  import('foo').then((foo) => {
    foo.doSomething();
  });
};
```

Additionally, you can [configure](/docs/conformance/customize) the rule for only specific React event handlers:

```json
"REACT_NO_STATIC_IMPORTS_IN_EVENT_HANDLERS": {
  eventAllowList: ['onClick'],
}
```

---
title: "REACT_STABLE_CONTEXT_PROVIDER_VALUE"
description: "Prevent non-stable values from being used in React Context providers that could cause unnecessary re-renders."
last_updated: "2026-01-16T02:19:28.317Z"
source: "https://vercel.com/docs/conformance/rules/REACT_STABLE_CONTEXT_PROVIDER_VALUE"

REACT_STABLE_CONTEXT_PROVIDER_VALUE

When non-stable values (i.e. object identities) are used as the `value` prop for `Context.Provider`,

React will trigger cascading updates to all components that use this context value on each
render, causing needless re-renders (affecting application performance) or causing
unintended consequences that may negatively affect the user-experience.

Examples

Examples of incorrect code for this rule:

```jsx
return <SomeContext.Provider value={{ foo: 'bar' }}>...</SomeContext.Provider>;
```

Examples of correct code for this rule:

```jsx
const foo = useMemo(() => ({ foo: 'bar' }), []);

return <SomeContext.Provider value={foo}>...</SomeContext.Provider>;
```

How to fix

One way to fix this issue may be to wrap the value in a `useMemo()`. If the value is a function
then `useCallback()` can be used as well. See the above examples for a reference on how you might
fix this by wrapping with `useMemo`.

---
title: "REQUIRE_CARET_DEPENDENCIES"
description: "Prevent the use of dependencies without a caret ("
last_updated: "2026-01-16T02:19:28.320Z"
source: "https://vercel.com/docs/conformance/rules/REQUIRE_CARET_DEPENDENCIES"

REQUIRE_CARET_DEPENDENCIES

Using a caret ("^") as a prefix in the version of your dependencies is recommended. [Caret Ranges](https://github.com/npm/node-semver?tab

By default, this rule is disabled. To enable it, refer to
[customizing Conformance](/docs/conformance/customize).

Examples

This rule will catch any `package.json` files:

- Using `~` or `*` as a prefix of the version, like `~1.0.0`.
- Version without a prefix, such as `1.0.0`.

```JSX filename="package.json" {3-4} {7}
{
  "dependencies": {
    "chalk": "~5.3.0",
    "ms": "*2.1.3",
  },
  "devDependencies": {
    "semver": "7.6.0"
  },
}
```

How to fix

If you hit this issue, you can resolve it by adding a `"^"` to the version of your dependency. If you want to keep using a pinned version

```JSX filename="package.json" {3}
{
  "dependencies": {
    "semver": "^7.6.0"
  },
}
```

---
title: "REQUIRE_DOCS_ON_EXPORTED_FUNCTIONS"
description: "Requires that all exported functions have JSDoc comments."
last_updated: "2026-01-16T02:19:28.324Z"
source: "https://vercel.com/docs/conformance/rules/REQUIRE_DOCS_ON_EXPORTED_FUNCTIONS"

REQUIRE_DOCS_ON_EXPORTED_FUNCTIONS

Adding JSDoc to exported functions helps engineers to quickly understand the
purpose and application of those functions when reviewing or using them.

This is particularly important in packages where the source code may be
minified and/or obfuscated, and can save users time by avoiding the need to
find usage information in external documentation.

For more information on JSDoc, see [Getting started with JSDoc](https://jsdoc.app/about-getting-started).

Additionally, for non-TypeScript projects, JSDoc can be used to declare type
information for function parameters and return values. For packages, these
declarations can provide type information for both JavaScript and TypeScript
consumers.

Examples

The below function is a minimal example of a function that would be caught by
this rule.

```ts
export function appendWorld(str: string): string {
  return str + ' world';
}
```

This rule will also catch references within the same file, and different ways
of declaring functions. For example:

```ts
const appendWorld = function (str: string): string {
  return str + ' world';
};

export default appendWorld;
```

This rule non-function exports and re-exports of functions.

How to fix

To resolve this issue, add a JSDoc comment to the exported function.

```ts
/**
 * Modifies a string by appending `' world'` to it.
 */
export function appendWorld(str: string): string {
  return str + ' world';
}
```

You can add additional information to the JSDoc comment, such as descriptions
of the function's parameters and return value.

```ts
/**
 * Modifies a string by appending `' world'` to it.
 *
 * @param str - The string to modify.
 * @returns The modified string.
 */
export function appendWorld(str: string): string {
  return str + ' world';
}
```

The example above doesn't provide type information in the JSDoc comment, as
this information is already provided by the function signature. When working
without TypeScript, you can also provide this information using JSDoc.

```js
/**
 * Modifies a string by appending `' world'` to it.
 *
 * @param {string} str - The string to modify.
 * @returns {string} The modified string.
 */
export function appendWorld(str) {
  return str + ' world';
}
```

---
title: "REQUIRE_NODE_VERSION_FILE"
description: "Requires that workspaces have a valid Node.js version file ("
last_updated: "2026-01-16T02:19:28.329Z"
source: "https://vercel.com/docs/conformance/rules/REQUIRE_NODE_VERSION_FILE"

REQUIRE_NODE_VERSION_FILE

Using a Node.js version file (`.node-version` or `.nvmrc`) ensures that all
developers and tooling (e.g., CI systems) use the same version of Node.js. This
practice helps to avoid inconsistencies between environments and can even
prevent bugs from being shipped to production.

As another benefit, committing a Node.js version file improves developer
experience, as many Node.js version management tools can automatically detect
and use the version defined in the file. This includes [GitHub Actions](https://docs.github.com/en/actions),
and popular Node.js version managers such as [`fnm`](https://github.com/Schniz/fnm)
and [`nvm`](https://github.com/nvm-sh/nvm).

This rule also validates to ensure that the version in the file is defined
in a way that is compatible with common tooling.

By default, this rule is disabled. To enable it, refer to
[customizing Conformance](/docs/conformance/customize).

How to fix

If you hit this issue, you can resolve it by adding a Node.js version file at
the root of your workspace.

The example `.node-version` file below requires that Node.js `20.9` is used in
the workspace, allowing for any patch version (i.e. `20.9.1`). The level of
strictness can be adjusted based on your teams needs.

```text filename=".node-version"
v20.9
```

---
title: "REQUIRE_ONE_VERSION_POLICY"
description: "Requires all dependencies in a monorepo to have the same version policy."
last_updated: "2026-01-16T02:19:28.332Z"
source: "https://vercel.com/docs/conformance/rules/REQUIRE_ONE_VERSION_POLICY"

REQUIRE_ONE_VERSION_POLICY

Dependency mismatch is a common and easily preventable problem. When there
are multiple versions of a single dependency, not only is there additional complexity
in maintaining different versions of that dependency, there are also code size complications
with bundling. Having multiple versions of a given dependency will likely result in bloated
code size as each dependency version will need to be bundled independently. Having multiple
versions might also leave developers confused and lead to possible security implications.

Additionally – keeping versions consistent reduces the possibility of type mismatches across
the monorepo.

By default, this rule is disabled. Enable it by [customizing Conformance](/docs/conformance/customize).

How to fix

Ensure all `package.json` files in your monorepo that have a `dependency` are
aligned to all have the same version. Version ranges are not always reliable, so it's recommended that
you pin all versions to the same given version to ensure consistency.

Exceptions

Sometimes it is useful to temporarily have two or more versions of a dependency whilst incrementally
migrating a monorepo to having the same version policy. In which case, it's acceptable to allowlist
this rule on specific parts of the codebase using by [customizing Conformance](/docs/conformance/customize)
until all packages have been successfully migrated.

---
title: "SET_COOKIE_VALIDATION"
description: "Prevents usage of cookies that do not conform to the allowed cookie policy."
last_updated: "2026-01-16T02:19:28.335Z"
source: "https://vercel.com/docs/conformance/rules/SET_COOKIE_VALIDATION"

SET_COOKIE_VALIDATION

It's a good practice to enforce a cookie policy across a workspace to ensure only
certain cookies are allowed to be set.

How to fix

Engineers should reach out to the appropriate engineer(s) or team(s) for a
review of the defined cookie and request the cookie name be added to the
allowed cookie policy list. This can be set in the `conformance.config.jsonc` configuration
file as follows:

```json filename="conformance.config.jsonc"
"SET_COOKIE_VALIDATION": {
  "cookieAllowList": ["some-cookie-name"]
}
```

---
title: "TESTS_NO_CONDITIONAL_ASSERTIONS"
description: "Requires that assertions are not conditional, or that expect.assertions is used."
last_updated: "2026-01-16T02:19:28.340Z"
source: "https://vercel.com/docs/conformance/rules/TESTS_NO_CONDITIONAL_ASSERTIONS"

TESTS_NO_CONDITIONAL_ASSERTIONS

When possible, conditional test assertions should be avoided as they can lead
to false test passes if and when conditions are not evaluated as expected.

If you can't avoid using a condition in your test, you can satisfy this rule by
using an `expect.assertions` statement.

Example

In this abstract example, there are two potential points of failure:

1. The button could throw a ButtonError during `render(Button)`, causing the
   first (`try`) assertion to be skipped.
2. The `throwError()` function could fail to throw, causing the second
   (`catch`) assertion to be skipped.

```ts filename="src/button/button.test.ts" {5,8}
describe('button', () => {
  it('should render', () => {
    try {
      const button = render(Button);
      expect(button).not.toBe(null);
      button.throwAnError();
    } catch (error) {
      expect(error).toBeInstanceOf(ButtonError);
    }
```

  });
});
```

## How to fix

There are two ways to resolve this error:

1. Refactor the test code to ensure that assertions are no longer conditional.
2. Use `expect.assertions` to inform the test runner that it should fail if the
   required number of assertions were not called during the test.

Taking our previous example, we can apply the second fix:

```ts filename="src/button/button.test.ts" {10}
describe('button', () => {
  it('should render', () => {
    try {
      const button = render(Button);
      expect(button).not.toBe(null);
      button.throwAnError();
    } catch (error) {
      expect(error).toBeInstanceOf(ButtonError);
    }
    expect.assertions(2);
  });
});
```

Using `expect.assertions`

Most test frameworks and runners support `expect.assertions`, and this is the
preferred approach to resolving this error if you can't refactor your test
code.

To satisfy this rule, the test must not conditionally call `expect.assertions`.
This rule doesn't count or report on the number of assertions.

What to do when you can't use `expect.assertions`

There may be cases where you can't use `expect.assertions` (i.e. your test
framework or runner doesn't support it), and refactoring the test code is not
a viable solution. In those cases, you have the following options:

1. You can use allowlists to allow individual violations (see: [Conformance
   Allowlists](/docs/conformance/allowlist)).
2. You can disable this test (see: [Customizing Conformance](/docs/conformance/customize)).

Customization

The default pattern matches the default patterns for Jest and Vitest, however
you can provide your own patterns through the `paths` property.

The default configuration is:

```jsonc filename="conformance.config.jsonc" {2-4}
{
  "configuration": [
    "testPatterns": ["**/unit-tests/**/*.{js,jsx}"]
  ]
}
```

---
title: "TESTS_NO_ONLY"
description: "Requires that focused tests (i.e. it.only()) are unfocused."
last_updated: "2026-01-16T02:19:28.344Z"
source: "https://vercel.com/docs/conformance/rules/TESTS_NO_ONLY"

TESTS_NO_ONLY

Focusing tests can help to write and debug test suites, but focused tests
should be unfocused before committing changes.

This rule disallows focused tests so that they can't be committed without an
allowlist entry.

Example

```ts filename="src/button/button.test.ts" {2}
describe('button', () => {
  it.only('should render', () => {
    // ...
  });
});
```

Note that the following patterns (and variants of these patterns) will be
reported as errors by this test. These should cover popular test frameworks and
runners, including:

- [`jest`](https://jestjs.io/)
- [`node:test`](https://nodejs.org/api/test.html#test-runner)
- [`vitest`](https://vitest.dev/)
- [`cypress`](https://www.cypress.io/)
- [`@playwright/test`](https://playwright.dev/docs/api/class-test)

```ts

```
// Most test frameworks and runners
describe.only(/* ... */);
it.concurrent.only(/* ... */);
test.only.each([])(/* ... */);
// Jest - supported in addition to the above
fdescribe(/* ... */);
fit.each([])(/* ... */);
ftest(/* ... */);
```

How to fix

This error will be resolved when debugging is complete and the test has been
unfocused.

Customization

The default pattern matches the default patterns for Jest and Vitest, however
you can provide your own patterns through the `paths` property.

The default configuration is:

```jsonc filename="conformance.config.jsonc" {3-5}
{
  "configuration": [
    "testPatterns": ["**/unit-tests/**/*.{js,jsx}"]
  ]
}
```

---
title: "TYPESCRIPT_CONFIGURATION"
description: "Requires that a workspace package that uses TypeScript files has configured TypeScript correctly for that workspace."
last_updated: "2026-01-16T02:19:28.348Z"
source: "https://vercel.com/docs/conformance/rules/TYPESCRIPT_CONFIGURATION"

TYPESCRIPT_CONFIGURATION

Using TypeScript in a workspace requires a few items to be set up correctly:

- There should be a `tsconfig.json` file at the root of the workspace.
- The `tsconfig.json` should extend from the repo's shared `tsconfig.json`
  file.
- The `tsconfig.json` file should specify a `tsBuildInfoFile` to speed up
  incremental compilation.
- The `tsconfig.json` file should have certain compiler options set for
  improved type safety.
- The workspace should have a `type-check` command that runs the TypeScript
  compiler to check for type issues.

These changes will ensure that the TypeScript compiler picks up the right
compiler settings for the project and that the TypeScript type checking
will run when the `type-check` command is run for the entire repository.

Example

```sh
Conformance errors found!

A Conformance error occurred in test "TYPESCRIPT_CONFIGURATION".

package.json in "docs" should have a "type-check" script that runs TypeScript type checking.

To find out more information and how to fix this error, visit
/docs/conformance/rules/TYPESCRIPT_CONFIGURATION.

If this violation should be ignored, add the following entry to
/apps/docs/.allowlists/TYPESCRIPT_CONFIGURATION.allowlist.json
and get approval from the appropriate person.

{
  "testName": "TYPESCRIPT_CONFIGURATION",
  "reason": "TODO: Add reason why this violation is allowed to be ignored.",
  "location": {
    "workspace": "docs"
  }
}
```

How To Fix

The shared `tsconfig.json` should have at least the following defined:

```json filename="tsconfig.json"
{
  "compilerOptions": {
    "incremental": true,
    "noUncheckedIndexedAccess": true,
    "strict": true
  }
}
```

For other configuration issues, the project's `tsconfig.json` may need to be
updated. Most files that don't require customization should look like:

```json filename="tsconfig.json"
{
```

```
  "extends": "your_shared_tsconfig/base.json",
  "exclude": ["dist", "node_modules"],
  "compilerOptions": {
    "tsBuildInfoFile": "node_modules/.cache/tsbuildinfo.json"
  }
}
```

Additionally, the project's `package.json` file may need to be updated. A
`type-check` command needs to be added to the `scripts` section:

```json filename="package.json"
{
  "scripts": {
    ...,
    "type-check": "tsc -p tsconfig.json --noEmit"
  }
}
```

The dependency on the repository's shared TypeScript must also exist:

```json
{
  "devDependencies": {
    "your_shared_tsconfig": "workspace:*"
  }
}
```

---
title: "TYPESCRIPT_ONLY"
description: "Requires that a workspace package may only contain TypeScript files and no JavaScript or JSX files."
last_updated: "2026-01-16T02:19:28.352Z"
source: "https://vercel.com/docs/conformance/rules/TYPESCRIPT_ONLY"

TYPESCRIPT_ONLY

[TypeScript](https://typescriptlang.org) is a superset of JavaScript that adds optional static typing. Using TypeScript in your codebase

- Type Safety: TypeScript is a strongly-typed language, which means that it
  allows you to catch errors at compile-time rather than at runtime. This can
  help you catch bugs earlier in the development process, making your code more
  reliable and easier to maintain over time.
- Tooling: TypeScript has excellent tooling support, including autocompletion,
  type checking, and refactoring tools. This can help you write code faster
  and with fewer errors.
- JavaScript Compatibility: TypeScript is a superset of JavaScript, which
  means that any valid JavaScript code is also valid TypeScript code. This
  means that you can gradually introduce TypeScript into your project without
  having to rewrite your entire codebase.
- Scalability: TypeScript is designed to work well with large-scale
  applications. With features like interfaces and classes, it allows you to
  write code that is easier to read and maintain, even as your project grows
  in complexity.

Example

```sh
Conformance errors found!

A Conformance error occurred in test "TYPESCRIPT_ONLY".

JavaScript files are not allowed. Please convert the file to TypeScript.

To find out more information and how to fix this error, visit
/docs/conformance/rules/TYPESCRIPT_ONLY.

If this violation should be ignored, add the following entry to
/apps/docs/.allowlists/TYPESCRIPT_ONLY.allowlist.json
and get approval from the appropriate person.

{
  "testName": "TYPESCRIPT_ONLY",
  "reason": "TODO: Add reason why this violation is allowed to be ignored.",
  "location": {
    "filePath": "apps/docs/src/add-numbers.js"
  }
}
```

How To Fix

To fix this error, you must convert the JavaScript file to TypeScript.
You can do this by changing the file extension from `.js` to `.ts` or `.jsx` to `.tsx` and
adding the appropriate type annotations.

```sh filename="diff"
--- a/apps/docs/src/add-numbers.js
+++ b/apps/docs/src/add-numbers.ts
-export function addNumbers(a, b) {
+export function addNumbers(a: number, b: number): number {
  return a + b;
}
```

Customization

The check supports custom file globs and ignore file globs that can be specified on `conformance.config.jsonc`.
The globs take effect from the root of the workspace package.

```json filename="conformance.config.jsonc"
{
  "rules": {
    "TYPESCRIPT_ONLY": {
      "files": ["**/*.js", "**/*.jsx"],
      "ignoreFiles": ["**/*.custom-config.js"]
    }
  }
}
```

The default configuration is:

```jsonc filename="conformance.config.jsonc"
{
  "rules": {
    "TYPESCRIPT_ONLY": {
      "files": ["**/*.{cjs,mjs,js,jsx}"],
      "ignoreFiles": [
        "dist/**",
        "node_modules/**",
        ".next/**", // Next.js output
        ".eslintrc.{cjs,js}", // Common ESLint config file name
        "*.config.{cjs,mjs,js}", // Common config file name
        "*.setup.{cjs,mjs,js}", // Common setup file name
      ],
    },
  },
}
```

---

title: "WORKSPACE_MISSING_CONFORMANCE_SCRIPT"
description: "All packages must define a conformance script that invokes the Conformance package."
last_updated: "2026-01-16T02:19:28.355Z"
source: "https://vercel.com/docs/conformance/rules/WORKSPACE_MISSING_CONFORMANCE_SCRIPT"

WORKSPACE_MISSING_CONFORMANCE_SCRIPT

Conformance requires a script to exist in every workspace in the
repository. This makes sure that Conformance rules are running on all code.
This test throws an error if a workspace does not define a `conformance` script
in the `package.json` file.

Example

A workspace contains a `package.json` file that looks like:

```json filename="package.json"
{
  "name": "test-workspace",
  "scripts": {
    "build": "tsc -b"
  }
}
```

It does not contain a `conformance` script, so this check will fail.

How to fix

Install the `@vercel-private/conformance` package in this workspace and define
a `conformance` script in the `package.json` file.

```json filename="package.json"
{
  "name": "test-workspace",
  "scripts": {
    "build": "tsc -b",
    "conformance": "vercel conformance"
  },
  "devDependencies": {
    "@vercel-private/conformance": "^1.0.0"
  }
}
```

---

title: "WORKSPACE_MISSING_PACKAGE_JSON"
description: "All directories that match a workspace glob must include a package.json file."
last_updated: "2026-01-16T02:19:28.358Z"
source: "https://vercel.com/docs/conformance/rules/WORKSPACE_MISSING_PACKAGE_JSON"

WORKSPACE_MISSING_PACKAGE_JSON

All directories that match a glob used to configure package manager workspaces
must be defined as a package and contain a `package.json` file. This check
prevents confusion where a new directory may be placed within a directory that
is configured to be a workspace but the new directory is not actually a
workspace.

Example

The repository configures pnpm workspaces in this file:

```yaml filename="pnpm-workspace.yaml"
packages:
  - 'apps/*'
  - 'packages/*'
```

If a directory is defined in `packages/not-a-package`, then this test will fail
saying that the `not-a-package` directory must contain a `package.json` file.

How to fix

Directories that match a workspace glob but do not have a `package.json` file
should either be converted to a package, be moved to a different directory, or
be excluded in the workspaces configuration.

---
title: "Conformance Rules"
description: "Learn how Conformance improves collaboration, productivity, and software quality at scale."
last_updated: "2026-01-16T02:19:28.383Z"
source: "https://vercel.com/docs/conformance/rules"

Conformance Rules

This page lists all the built-in rules that Conformance will check for by
default in your application.

\['Next.js'

These Conformance rules catch common issues that can happen in Next.js
applications.

Test Name	Description
[ESLINT_NEXT_RULES_REQUIRED](/docs/conformance/rules/ESLINT_NEXT_RULES_REQUIRED)	Requires that ESLin
[NEXTJS_MISSING_MODULARIZE_IMPORTS](/docs/conformance/rules/NEXTJS_MISSING_MODULARIZE_IMPORTS)	Requires that Next.
[NEXTJS_MISSING_OPTIMIZE_PACKAGE_IMPORTS](/docs/conformance/rules/NEXTJS_MISSING_OPTIMIZE_PACKAGE_IMPORTS)	Requires that Next
[NEXTJS_MISSING_NEXT13_TYPESCRIPT_PLUGIN](/docs/conformance/rules/NEXTJS_MISSING_NEXT13_TYPESCRIPT_PLUGIN)	Applications using
[NEXTJS_MISSING_REACT_STRICT_MODE](/docs/conformance/rules/NEXTJS_MISSING_REACT_STRICT_MODE)	Applications using
[NEXTJS_MISSING_SECURITY_HEADERS](/docs/conformance/rules/NEXTJS_MISSING_SECURITY_HEADERS)	Requires that impor
[NEXTJS_NO_ASYNC_LAYOUT](/docs/conformance/rules/NEXTJS_NO_ASYNC_LAYOUT)	Ensures that the ex
[NEXTJS_NO_ASYNC_PAGE](/docs/conformance/rules/NEXTJS_NO_ASYNC_PAGE)	Ensures that the ex
[NEXTJS_NO_BEFORE_INTERACTIVE](/docs/conformance/rules/NEXTJS_NO_BEFORE_INTERACTIVE)	Requires review of
[NEXTJS_NO_CLIENT_DEPS_IN_MIDDLEWARE](/docs/conformance/rules/NEXTJS_NO_CLIENT_DEPS_IN_MIDDLEWARE)	Disallows depende
[NEXTJS_NO_DYNAMIC_AUTO](/docs/conformance/rules/NEXTJS_NO_DYNAMIC_AUTO)	Prevent usage of `f
[NEXTJS_NO_FETCH_IN_SERVER_PROPS](/docs/conformance/rules/NEXTJS_NO_FETCH_IN_SERVER_PROPS)	Prevent relative
[NEXTJS_NO_GET_INITIAL_PROPS](/docs/conformance/rules/NEXTJS_NO_GET_INITIAL_PROPS)	Requires any use o
[NEXTJS_NO_PRODUCTION_SOURCE_MAPS](/docs/conformance/rules/NEXTJS_NO_PRODUCTION_SOURCE_MAPS)	Applications using
[NEXTJS_NO_SELF_HOSTED_VIDEOS](/docs/conformance/rules/NEXTJS_NO_SELF_HOSTED_VIDEOS)	Prevent video file
[NEXTJS_NO_TURBO_CACHE](/docs/conformance/rules/NEXTJS_NO_TURBO_CACHE)	Prevent Turborepo f
[NEXTJS_REQUIRE_EXPLICIT_DYNAMIC](/docs/conformance/rules/NEXTJS_REQUIRE_EXPLICIT_DYNAMIC)	Requires explicitly
[NEXTJS_SAFE_NEXT_PUBLIC_ENV_USAGE](/docs/conformance/rules/NEXTJS_SAFE_NEXT_PUBLIC_ENV_USAGE)	Usage process.env
[NEXTJS_SAFE_SVG_IMAGES](/docs/conformance/rules/NEXTJS_SAFE_SVG_IMAGES)	Prevent `dangerously
[NEXTJS_SAFE_URL_IMPORTS](/docs/conformance/rules/NEXTJS_SAFE_URL_IMPORTS)	Prevent unsafe URL
[NEXTJS_UNNEEDED_GET_SERVER_SIDE_PROPS](/docs/conformance/rules/NEXTJS_UNNEEDED_GET_SERVER_SIDE_PROPS)	Catches usages of
[NEXTJS_USE_NATIVE_FETCH](/docs/conformance/rules/NEXTJS_USE_NATIVE_FETCH)	Requires using nati
[NEXTJS_USE_NEXT_FONT](/docs/conformance/rules/NEXTJS_USE_NEXT_FONT)	Requires using `nex
[NEXTJS_USE_NEXT_IMAGE](/docs/conformance/rules/NEXTJS_USE_NEXT_IMAGE)	Requires that `next
[NEXTJS_USE_NEXT_SCRIPT](/docs/conformance/rules/NEXTJS_USE_NEXT_SCRIPT)	Requires that `next
[NO_FETCH_FROM_MIDDLEWARE](/docs/conformance/rules/NO_FETCH_FROM_MIDDLEWARE)	Requires that any `
[REACT_NO_STATIC_IMPORTS_IN_EVENT_HANDLERS](/docs/conformance/rules/REACT_NO_STATIC_IMPORTS_IN_EVENT_HANDLERS)	Prevent static i

'Performance'

These Conformance rules catch issues that negatively affect the performance
of your website.

Test Name	Description
[BFCACHE_INTEGRITY_NO_UNLOAD_LISTENERS](/docs/conformance/rules/BFCACHE_INTEGRITY_NO_UNLOAD_LISTENERS)	Disallows th
[BFCACHE_INTEGRITY_REQUIRE_NOOPENER_ATTRIBUTE](/docs/conformance/rules/BFCACHE_INTEGRITY_REQUIRE_NOOPENER_ATTRIBUTE)	Requires tha
[NEXTJS_NO_ASYNC_LAYOUT](/docs/conformance/rules/NEXTJS_NO_ASYNC_LAYOUT)	Ensures that
[NEXTJS_NO_ASYNC_PAGE](/docs/conformance/rules/NEXTJS_NO_ASYNC_PAGE)	Ensures that
[NEXTJS_NO_BEFORE_INTERACTIVE](/docs/conformance/rules/NEXTJS_NO_BEFORE_INTERACTIVE)	Requires revi
[NEXTJS_NO_CLIENT_DEPS_IN_MIDDLEWARE](/docs/conformance/rules/NEXTJS_NO_CLIENT_DEPS_IN_MIDDLEWARE)	Disallows d
[NEXTJS_NO_DYNAMIC_AUTO](/docs/conformance/rules/NEXTJS_NO_DYNAMIC_AUTO)	Prevent usage
[NEXTJS_NO_FETCH_IN_SERVER_PROPS](/docs/conformance/rules/NEXTJS_NO_FETCH_IN_SERVER_PROPS)	Prevent rel
[NEXTJS_NO_GET_INITIAL_PROPS](/docs/conformance/rules/NEXTJS_NO_GET_INITIAL_PROPS)	Requires any
[NEXTJS_REQUIRE_EXPLICIT_DYNAMIC](/docs/conformance/rules/NEXTJS_REQUIRE_EXPLICIT_DYNAMIC)	Requires expl
[NEXTJS_UNNEEDED_GET_SERVER_SIDE_PROPS](/docs/conformance/rules/NEXTJS_UNNEEDED_GET_SERVER_SIDE_PROPS)	Catches usag
[NEXTJS_USE_NATIVE_FETCH](/docs/conformance/rules/NEXTJS_USE_NATIVE_FETCH)	Requires usin
[NEXTJS_USE_NEXT_IMAGE](/docs/conformance/rules/NEXTJS_USE_NEXT_IMAGE)	Requires that
[NEXTJS_USE_NEXT_SCRIPT](/docs/conformance/rules/NEXTJS_USE_NEXT_SCRIPT)	Requires that
[NO_EXTERNAL_CSS_AT_IMPORTS](/docs/conformance/rules/NO_EXTERNAL_CSS_AT_IMPORTS)	Disallows `@
[NO_FETCH_FROM_MIDDLEWARE](/docs/conformance/rules/NO_FETCH_FROM_MIDDLEWARE)	Requires that
[NO_INLINE_SVG](/docs/conformance/rules/NO_INLINE_SVG)	Prevent the us
[NO_MIXED_ASYNC_MODULES](/docs/conformance/rules/NO_MIXED_ASYNC_MODULES)	Prevent impor
[NO_POSTINSTALL_SCRIPT](/docs/conformance/rules/NO_POSTINSTALL_SCRIPT)	Prevent the us
[NO_SERIAL_ASYNC_CALLS](/docs/conformance/rules/NO_SERIAL_ASYNC_CALLS)	Prevent block
[REACT_NO_STATIC_IMPORTS_IN_EVENT_HANDLERS](/docs/conformance/rules/REACT_NO_STATIC_IMPORTS_IN_EVENT_HANDLERS)	Prevent st
[REACT_STABLE_CONTEXT_PROVIDER_VALUE](/docs/conformance/rules/REACT_STABLE_CONTEXT_PROVIDER_VALUE)	Prevent non-

'Security'

These Conformance rules catch issues that could become security vulnerabilities
in your application.

Test Name	Description
[NEXTJS_MISSING_SECURITY_HEADERS](/docs/conformance/rules/NEXTJS_MISSING_SECURITY_HEADERS)	Requires that important security he
[NEXTJS_NO_PRODUCTION_SOURCE_MAPS](/docs/conformance/rules/NEXTJS_NO_PRODUCTION_SOURCE_MAPS)	Applications using Next.js should
[NEXTJS_SAFE_NEXT_PUBLIC_ENV_USAGE](/docs/conformance/rules/NEXTJS_SAFE_NEXT_PUBLIC_ENV_USAGE)	Usage process.env.NEXT*PUBLIC**
[NEXTJS_SAFE_SVG_IMAGES](/docs/conformance/rules/NEXTJS_SAFE_SVG_IMAGES)	Prevent `dangerouslyAllowSVG` witho
[NEXTJS_SAFE_URL_IMPORTS](/docs/conformance/rules/NEXTJS_SAFE_URL_IMPORTS)	Prevent unsafe URL Imports from bei
[NO_ASSIGN_WINDOW_LOCATION](/docs/conformance/rules/NO_ASSIGN_WINDOW_LOCATION)	Prevent unsafe assignment to `windo
[NO_CORS_HEADERS](/docs/conformance/rules/NO_CORS_HEADERS)	Requires that CORS header configurat
[NO_DANGEROUS_HTML](/docs/conformance/rules/NO_DANGEROUS_HTML)	Prevent the unsafe creation of DOM t
[NO_DOCUMENT_WRITE](/docs/conformance/rules/NO_DOCUMENT_WRITE)	Prevent unsafe usage of `document.wr
[NO_EVAL](/docs/conformance/rules/NO_EVAL)	Prevent unsafe usage of `eval()` in y
[NO_VARIABLE_IMPORT_REFERENCES](/docs/conformance/rules/NO_VARIABLE_IMPORT_REFERENCES)	Prevents loading of arbitrary modul
[REQUIRE_CARET_DEPENDENCIES](/docs/conformance/rules/REQUIRE_CARET_DEPENDENCIES)	Prevent the use of dependencies with
[SET_COOKIE_VALIDATION](/docs/conformance/rules/SET_COOKIE_VALIDATION)	Prevents usage of cookies that do no

'Code Health']

These Conformance rules catch issues that can negatively affect your codebase
or code health.

Test Name	Description
[ESLINT_CONFIGURATION](/docs/conformance/rules/ESLINT_CONFIGURATION)	Requires that a workspa
[ESLINT_REACT_RULES_REQUIRED](/docs/conformance/rules/ESLINT_REACT_RULES_REQUIRED)	Requires that ESLint
[ESLINT_RULES_REQUIRED](/docs/conformance/rules/ESLINT_RULES_REQUIRED)	Requires that ESLint h
[NEXTJS_MISSING_MODULARIZE_IMPORTS](/docs/conformance/rules/NEXTJS_MISSING_MODULARIZE_IMPORTS)	Requires that Next.js
[NO_ASSIGN_WINDOW_LOCATION](/docs/conformance/rules/NO_ASSIGN_WINDOW_LOCATION)	Prevent unsafe assign
[NO_INSTANCEOF_ERROR](/docs/conformance/rules/NO_INSTANCEOF_ERROR)	Disallows using `error
[NO_UNNECESSARY_PROP_SPREADING](/docs/conformance/rules/NO_UNNECESSARY_PROP_SPREADING)	Prevent the use of ob
[PACKAGE_JSON_DESCRIPTION_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_DESCRIPTION_REQUIRED)	Requires that every `
[PACKAGE_JSON_DUPLICATE_DEPENDENCIES](/docs/conformance/rules/PACKAGE_JSON_DUPLICATE_DEPENDENCIES)	Found duplicate depen
[PACKAGE_JSON_NAME_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_NAME_REQUIRED)	Requires that every `
[PACKAGE_JSON_PRIVATE_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_PRIVATE_REQUIRED)	Requires that every `
[PACKAGE_JSON_SIDE_EFFECTS_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_SIDE_EFFECTS_REQUIRED)	Requires that every
[PACKAGE_JSON_TYPE_REQUIRED](/docs/conformance/rules/PACKAGE_JSON_TYPE_REQUIRED)	Requires that every `
[PACKAGE_MANAGEMENT_NO_CIRCULAR_IMPORTS](/docs/conformance/rules/PACKAGE_MANAGEMENT_NO_CIRCULAR_IMPORTS)	Circular imports bet
[PACKAGE_MANAGEMENT_NO_UNRESOLVED_IMPORTS](/docs/conformance/rules/PACKAGE_MANAGEMENT_NO_UNRESOLVED_IMPORTS)	Import statements th
[PACKAGE_MANAGEMENT_REQUIRED_README](/docs/conformance/rules/PACKAGE_MANAGEMENT_REQUIRED_README)	Requires that every w
[REQUIRE_DOCS_ON_EXPORTED_FUNCTIONS](/docs/conformance/rules/REQUIRE_DOCS_ON_EXPORTED_FUNCTIONS)	Requires that all ex
[REQUIRE_NODE_VERSION_FILE](/docs/conformance/rules/REQUIRE_NODE_VERSION_FILE)	Requires that workspa
[REQUIRE_ONE_VERSION_POLICY](/docs/conformance/rules/REQUIRE_ONE_VERSION_POLICY)	Requires all dependen
[TESTS_NO_CONDITIONAL_ASSERTIONS](/docs/conformance/rules/TESTS_NO_CONDITIONAL_ASSERTIONS)	Requires that asserti
[TESTS_NO_ONLY](/docs/conformance/rules/TESTS_NO_ONLY)	Requires that focused
[TYPESCRIPT_CONFIGURATION](/docs/conformance/rules/TYPESCRIPT_CONFIGURATION)	Requires that a workspa
[TYPESCRIPT_ONLY](/docs/conformance/rules/TYPESCRIPT_ONLY)	Requires that a workspa
[WORKSPACE_MISSING_CONFORMANCE_SCRIPT](/docs/conformance/rules/WORKSPACE_MISSING_CONFORMANCE_SCRIPT)	All packages must def
[WORKSPACE_MISSING_PACKAGE_JSON](/docs/conformance/rules/WORKSPACE_MISSING_PACKAGE_JSON)	All directories that

---
title: "Connectivity"
description: "Connect your Vercel projects to backend services with static IPs and secure networking options."
last_updated: "2026-01-16T02:19:28.390Z"
source: "https://vercel.com/docs/connectivity"

Connectivity

Connect your projects to backend services that require IP allowlisting or private network access.

Static IPs (shared pool)

When your database or API needs to see traffic from known IP addresses, Static IPs give you shared static egress IPs that won't change. P

- **Use case**: IP allowlisting for databases, APIs, and legacy systems
- **Network**: Shared VPC with subnet-level isolation
- [**Pricing**](/docs/connectivity/static-ips#pricing): $100/month per project + $0.15/GB Private Data Transfer

[Learn more about Static IPs](/docs/connectivity/static-ips)

Secure Compute

For when you need your own private Virtual Private Cloud (VPC). Secure Compute gives you dedicated networks with VPC peering — your infra

- **Use case**: Full network isolation and VPC peering
- **Network**: Dedicated VPC per customer

[Learn more about Secure Compute](/docs/connectivity/secure-compute)

Pricing

Both connectivity options are billed on **Private Data Transfer** priced regionally based on the [regional pricing documentation](/docs/p

Feature	Static IPs (Pro)	Secure Compute (Enterprise)
Monthly cost	$100/month per project	Custom
Private Data Transfer	[Regional pricing](/docs/pricing/regional-pricing)	Custom
Network isolation	Shared VPC with subnet-level isolation	Dedicated VPC and subnet per customer

Understanding data transfer costs

Data transfer costs kick in for all outbound traffic from your Vercel Functions to external services:

- Database queries and responses
- API calls to third-party services
- File uploads and downloads
- Any other outbound network traffic

Keep tabs on your usage in the **Team Settings** **Usage** tab under the **Private Data Transfer** section.

---
title: "Secure Compute"
description: "Secure Compute provides dedicated private networks with VPC peering for Enterprise teams."
last_updated: "2026-01-16T02:19:28.417Z"
source: "https://vercel.com/docs/connectivity/secure-compute"

Secure Compute

Secure Compute creates private connections between your [Vercel Functions](/docs/functions) and your backend infrastructure like database

By default, Vercel deployments can come from [any IP address](/kb/guide/how-to-allowlist-deployment-ip-address). Secure Compute gives you

When you enable Secure Compute on your [project](/docs/projects), your deployments and build container get their own [dedicated network w

You can provision and manage your own Secure Compute networks directly from the Vercel dashboard. Create networks for different teams, pr

> **💡 Note:** If you only need static IP addresses for IP allowlisting, without features like dedicated infrastructure, VPC peering, or

How Secure Compute works

Here's what you get with Secure Compute:

- Your own dedicated private network inside a VPC
- Static IPs that won't change, plus a NAT Gateway
- Complete isolation — only your specified resources can reach your Vercel Functions

Enabling Secure Compute

You can create Secure Compute networks directly from the Vercel dashboard:

1. Navigate to your team's **Settings** → [**Connectivity**](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fsettings%2Fconnectivity\&title
2. Click **Create Network** to start the setup process.
3. Select your desired **Region**: choose the region closest to your backend infrastructure for best performance.
4. Optionally expand **Advanced options** to configure:
   - **CIDR Address Block**: Specify a custom private IPv4 address range.
   - **Availability Zones**: Select specific AWS Availability Zones within your chosen region.
5. Click **Next** to review your settings.
6. Click **Create Network** to provision your network.

Once created, your network includes:

- A pair of dedicated IP addresses
- AWS account ID
- AWS region based on your selection
- AWS VPC ID
- CIDR block based on your selection

When you enable Secure Compute on a project, Vercel attaches your project's [build container](/docs/builds) and subsequent deployment ins

Secure Compute networks and dedicated IP addresses

Each private network has its own dedicated IP pair and is isolated from others, ensuring no sharing across teams. You can assign multiple

You can create multiple Secure Compute networks for your team directly from the dashboard. For example, separate networks for different p

Once your IP pair is ready, add it to your backend's access control list. You'll still need to use a username/password or authentication

Specific region

When you create a Secure Compute network, you select the [Vercel Function region](/docs/functions/configuring-functions/region) where it

Vercel applies Secure Compute to [Vercel Functions](/docs/functions) using the following runtimes:

- [Node.js](/docs/functions/runtimes/node-js)
- [Ruby](/docs/functions/runtimes/ruby)
-
- [Python](/docs/functions/runtimes/python)

The [Edge Runtime](/docs/functions/runtimes/edge) **is not supported** meaning features like [Routing Middleware](/docs/routing-middlewar

Region failover

Secure Compute supports automatic region failover using the active and passive network concept. Each project environment can have:

- **Active Network**: The primary Secure Compute network where your functions run
- **Passive Network**: A secondary network in a different region for automatic failover

To set up region failover:

1. **Create networks in different regions**: Navigate to your team's **Settings** → [**Connectivity**](https://vercel.com/d?to=%2F%5Bteam
2. **Connect project environments**: In your project's **Settings** → [**Connectivity**](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproje
3. **Automatic failover**: When enabled, Vercel automatically switches to the passive network if the primary region becomes unavailable,

Add a project to your Secure Compute network

To add a project to your Secure Compute network:

1. Navigate to your project's **Settings** → [**Connectivity**](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Fsettings%2Fconnec
2. For every environment you want to connect to Secure Compute:
   - Select an **Active Network**.
   - Optionally select a **Passive Network** to enable passive failover.
   - Optionally enable **Builds** to include the project's build container in the network.
3. Click **Save** to persist your changes.

To change multiple environments at once:

1. Select the environments using checkboxes or use the checkbox in the table header to select all environments.

2. Click **Edit Selected**.
3. In bulk edit modal:
   - Select an **Active Network**.
   - Optionally select a **Passive Network** to enable passive failover.
   - Optionally check **Include Builds** to include the project's build container in the network.
   - Click **Apply** to modify the selected environments.
4. Click **Save** to persist your changes.

Managing the build container

When you add a project to a Secure Compute network, you can choose to include the project's build container in the network. This is usefu

You can opt the [build container](/docs/builds) out of using the dedicated IP addresses. This is useful if your application **only** call

By opting out of including the build container, you will not incur the 5s delay when provisioning a secure build container.

To manage the build container during the [project connection](#add-a-project-to-your-secure-compute-network) process select **Include Bui

To manage the build container *after* the project is connected to the Secure Compute network:

1. Navigate to your team's **Settings** → [**Connectivity**](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fsettings%2Fconnectivity\&title
2. Select a private network from the list.
3. Select the **Projects** tab.
4. Click the icon to the right of your connected project and click **Edit**.
5. Check/uncheck **Include Builds** to include/exclude the project's build container in the network.
6. Click **Save**.

Multiple Secure Compute networks

You can use one network with multiple projects in the same team. In this case, the same IP pair is shared across multiple projects.

If you require additional security or have a large team, you can have one network for each project so that each project will have its own

Connecting a project to multiple networks across different regions is currently not supported. Each project environment can only be linke

VPC peering

Virtual private cloud (VPC) peering is a method of connecting two VPCs in the same or different region. When you use Secure Compute, Verc

To set up VPC peering:

1. **Create a Secure Compute network**: Navigate to your team's **Settings** → [**Connectivity**](https://vercel.com/d?to=%2F%5Bteam%5D%2
2. **Set up peering in AWS**: In your AWS VPC dashboard, configure the peering connection by copying the values from your Secure Compute
   - **Requester VPC ID**: Your VPC ID
   - **Account ID**: The AWS account ID
   - **Accepter VPC ID**: Your Vercel Secure Compute network's VPC Peering ID
   - **Region**: Your Vercel Secure Compute network's region
3. **Create peering connection**: In the AWS VPC peering connection settings, click **Create Peering Connection** to establish the connec
4. **Accept peering connection**: Go back to your Vercel dashboard and click **Accept** to accept the connection.
5. **Update route tables**: Go to AWS's VPC dashboard, select **Route Tables**, and configure routing to allow traffic from Vercel's CIDR

The connection can be deleted from either the Vercel dashboard, or the AWS VPC dashboard.

VPN Support

If your current security and compliance obligations require more than dedicated IP addresses, contact us for guidance related to your spe

Pricing

Secure Compute starts at **$6.5K/year** for Enterprise teams, plus **Secure Connect Data Transfer** at **$0.15/GB**.

Understanding data transfer costs

Data transfer costs apply to all outbound traffic from your Vercel Functions to external services:

- Database queries and responses
- API calls to third-party services
- File uploads and downloads
- Any other outbound network traffic

Monitor your usage in the **Team Settings** **Usage** tab under the **Secure Connect Data Transfer** section.

Limits

Build delay

When connected to a Secure Compute network, builds experience up to a 5s delay as they provision a secure build container. When this happ

Max number of VPC peering connections

The maximum number of VPC peering connections that can be established per network is **50**.

---
title: "Getting Started with Static IPs"
description: "Learn how to set up Static IPs for your Vercel projects to connect to IP-restricted backend services."
last_updated: "2026-01-16T02:19:28.423Z"
source: "https://vercel.com/docs/connectivity/static-ips/getting-started"

Getting Started with Static IPs

This guide walks you through setting up Static IPs so you can access backend services that require IP allowlisting.

Prerequisites

Before you dive in, make sure you have:

- A project deployed on Vercel
- A backend service that supports IP allowlisting

- [Pro](/docs/plans/pro-plan) or [Enterprise](/docs/plans/enterprise) plan

- ### Access the Connectivity settings
  1. Go to your **Project Dashboard**
  2. Navigate to **Project Settings**
  3. Click the **Connectivity** section

- ### Configure your region
  1. Click **Manage Active Regions**
  2. Pick a **region** close to your backend services to keep latency down. You can pick up to 3 regions
  3. Your project gets assigned static IPs within a shared VPC for each configured region

- ### Get your static IP addresses and configure your backend service
  1. Copy the static IP addresses from the dashboard
  2. Add the static IPs to your backend service's allowlist so it knows which IP addresses are allowed to connect

- ### Verify your connection
  To test your connection, redeploy your project that connects to your backend service. All your outbound traffic will now go through tho

Next steps

- Learn how to [monitor usage and billing](/docs/connectivity/static-ips#managing-your-static-ips) for your Static IPs
- Understand [how Static IPs work](/docs/connectivity/static-ips#how-it-works)
- Review [limits and pricing](/docs/connectivity/static-ips#limits-and-pricing)

---
title: "Static IPs"
description: "Access IP-restricted backend services through shared static egress IPs for Pro and Enterprise teams."
last_updated: "2026-01-16T02:19:28.434Z"
source: "https://vercel.com/docs/connectivity/static-ips"

Static IPs

With Static IPs (shared pool), you can access backend services that require IP allowlisting through static egress IPs. It's designed for

> **💡 Note:** If you need dedicated infrastructure, VPC peering, or complete network isolation, consider [Secure Compute](/docs/connecti

When to use Static IPs

- Connect to databases such as Amazon RDS, Google Cloud SQL, Azure SQL, and MongoDB Atlas
- Connect to APIs such as Auth0, PayPal, Stripe, internal corporate APIs
- Connect to systems such as on-premises databases and services behind firewalls
- Support compliance and business requirements

When not to use Static IPs

Static IP is a service provided by Vercel that assigns a set of fixed outbound IP addresses used for egress traffic from your deployments
Therefore, Static IPs should not be used if you need your app to be reachable through a fixed inbound IP or require ingress traffic suppo

Static IPs or Secure Compute

Feature	Static IPs (Pro & Enterprise)	Secure Compute (Enterprise only)
IP type	Static in shared Virtual Private Cloud (VPC)	Static in dedicated VPC
Network isolation	Shared VPC for a small group of customers with subnet-level isolation	Dedicated VPC and subnet per customer
Use cases	IP allowlisting, database access	IP allowlisting, VPC Peering, full isol
Pricing	$100/month per project, plus Private Data Transfer at $0.15/GB	Custom pricing

Static IPs with Secure Compute

If your project uses [Secure Compute](/docs/connectivity/secure-compute) and you have enabled Static IPs, Static IPs will be ignored.

Getting started

Read our [getting started guide](/docs/connectivity/static-ips/getting-started) to learn how to set up Static IPs.

How it works

When you enable Static IPs, you get:

- **Shared infrastructure**: Each VPC serves a small group of customers
- **Static egress**: All outbound traffic routes through shared static IP pairs
- **Logical isolation**: Subnet-level isolation maintains security between customers on the same VPC
- **NAT gateway**: Traffic exits through a managed NAT gateway for consistent IPs
- **Build traffic**: Traffic from both deployed functions and builds will route through the static IPs

Managing your static IPs

Routing build traffic

If your application calls data sources at build time, you can route its build traffic through your static IPs to keep your data sources s

To enable this, go to your [project's connectivity settings](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Fsettings%2Fconnectiv

1. Go to your project's **Settings**
2. Navigate to **Connectivity**
3. Toggle **Use Static IPs for builds** under **Static IPs**

This setting is disabled by default. When enabled, both your project's build and deployed function traffic will route through static IPs

Routing Middleware support

Static IPs (region-specific) don't apply to [middleware](/docs/routing-middleware) (which are deployed at the [edge](/docs/glossary#edge)

Checking usage

1. Go to your **Team** and click the **Usage** tab
2. Scroll down to the **Content, Caching & Optimization** section. Static IPs data transfer is metered by **Private Data Transfer**

3. Click **Private Data Transfer** for more detail about direction, regions, and projects

Static IPs with deployment environments

When you configure static IPs in a project, they apply to all the [environments](/docs/deployments/environments) set up in this project.

Regional considerations

- Choose regions close to your backend services to reduce latency
- Each configured region has its own static IP pair

Limits and pricing

Limits

- Static IP addresses are shared across a small group of customers in the same region
- Project-level configuration: You cannot isolate static IPs to specific deployment environments

Pricing

Static IPs are priced at **$100/month per project** for Pro plus **Private Data Transfer** priced regionally based on [regional pricing d

---
title: "Vercel Enterprise Managed Infrastructure"

last_updated: "2026-01-16T02:19:28.440Z"
source: "https://vercel.com/docs/contentful/managed-infrastructure"

Vercel Enterprise Managed Infrastructure

Vercel prices its [CDN](/docs/cdn) resources by region to help optimize costs and performance for your projects. This is to ensure you ar

Managed Infrastructure Units

Managed Infrastructure Units (MIUs) serve as both a financial commitment and a measurement of the infrastructure consumption of an Enterp

MIUs are billed monthly and do not roll over from month to month.

Regional pricing

The following table lists the usage amounts for each resource in Managed Infrastructure Units. Resources that depend on the region of you

Use the dropdown to select the region you are interested in.

Fluid compute regional pricing

The following table shows the regional pricing for fluid compute resources on Vercel. The prices are per hour for CPU and per GB-hr for m

Region	Active CPU time (per hour)	Provisioned Memory (GB-hr)
Washington, D.C., USA (iad1)	0.128 MIUs	0.0106 MIUs
Cleveland, USA (cle1)	0.128 MIUs	0.0106 MIUs
San Francisco, USA (sfo1)	0.177 MIUs	0.0147 MIUs
Portland, USA (pdx1)	0.128 MIUs	0.0106 MIUs
Cape Town, South Africa (cpt1)	0.200 MIUs	0.0166 MIUs
Hong Kong (hkg1)	0.176 MIUs	0.0146 MIUs
Mumbai, India (bom1)	0.140 MIUs	0.0116 MIUs
Osaka, Japan (kix1)	0.202 MIUs	0.0167 MIUs
Seoul, South Korea (icn1)	0.169 MIUs	0.0140 MIUs
Singapore (sin1)	0.160 MIUs	0.0133 MIUs
Sydney, Australia (syd1)	0.180 MIUs	0.0149 MIUs
Tokyo, Japan (hnd1)	0.202 MIUs	0.0167 MIUs
Frankfurt, Germany (fra1)	0.184 MIUs	0.0152 MIUs
Dublin, Ireland (dub1)	0.168 MIUs	0.0139 MIUs
London, UK (lhr1)	0.177 MIUs	0.0146 MIUs
Paris, France (cdg1)	0.177 MIUs	0.0146 MIUs
Stockholm, Sweden (arn1)	0.160 MIUs	0.0133 MIUs
Dubai, UAE (dxb1)	0.185 MIUs	0.0153 MIUs
São Paulo, Brazil (gru1)	0.221 MIUs	0.0183 MIUs
Montréal, Canada (yul1)	0.147 MIUs	0.0121 MIUs

Additional usage based products

The following table lists the MIUs for additional usage based products in Managed Infrastructure.

---
title: "Managing Cron Jobs"
description: "Learn how to manage Cron Jobs effectively in Vercel. Explore cron job duration, error handling, deployments, concurrency co
last_updated: "2026-01-16T02:19:28.489Z"
source: "https://vercel.com/docs/cron-jobs/manage-cron-jobs"

Managing Cron Jobs

Viewing cron jobs

To view your active cron jobs:

1. Select your project from the Vercel dashboard
2. Select the **Settings** tab
3. Select the **Cron Jobs** tab from the left sidebar

Cron jobs maintenance

- **Updating Cron Jobs**: Change the [expression](/docs/cron-jobs#cron-expressions) in `vercel.json` file or the function's configuration
- **Deleting Cron Jobs**: Remove the configuration from the `vercel.json` file or the function's configuration, and then redeploy
- **Disabling Cron Jobs**: Navigate to the **Cron Jobs** tab and then click the **Disable Cron Jobs** button

> **⚠️ Warning:** Disabled cron jobs will still be listed and will count towards your [cron jobs
> limits](/docs/cron-jobs/usage-and-pricing)

Securing cron jobs

It is possible to secure your cron job invocations by adding an environment variable called `CRON_SECRET` to your Vercel project. We reco

The value of the variable will be automatically sent as an `Authorization` header when Vercel invokes your cron job. Your endpoint can th

> **💡 Note:** You can use App Router [Route
> Handlers](https://nextjs.org/docs/app/building-your-application/routing/route-handlers)
> to secure your cron jobs, even when using the Pages Router.

```ts filename="app/api/cron/route.ts" framework=nextjs
import type { NextRequest } from 'next/server';

export function GET(request: NextRequest) {
  const authHeader = request.headers.get('authorization');
  if (authHeader !== `Bearer ${process.env.CRON_SECRET}`) {
    return new Response('Unauthorized', {
      status: 401,
    });
  }

  return Response.json({ success: true });
}
```

```js filename="app/api/cron/route.js" framework=nextjs
export function GET(request) {
  const authHeader = request.headers.get('authorization');
  if (authHeader !== `Bearer ${process.env.CRON_SECRET}`) {
    return new Response('Unauthorized', {
      status: 401,
    });
  }

  return Response.json({ success: true });
}
```

```ts filename="app/api/cron/route.ts" framework=nextjs-app
import type { NextRequest } from 'next/server';

export function GET(request: NextRequest) {
  const authHeader = request.headers.get('authorization');
  if (authHeader !== `Bearer ${process.env.CRON_SECRET}`) {
    return new Response('Unauthorized', {
      status: 401,
    });
  }

  return Response.json({ success: true });
}
```

```js filename="app/api/cron/route.js" framework=nextjs-app
export function GET(request) {
  const authHeader = request.headers.get('authorization');
  if (authHeader !== `Bearer ${process.env.CRON_SECRET}`) {
    return new Response('Unauthorized', {
      status: 401,
    });
  }

  return Response.json({ success: true });
}
```

```ts filename="api/cron/route.ts" framework=other
import type { VercelRequest, VercelResponse } from '@vercel/node';

export default function handler(
  request: VercelRequest,
  response: VercelResponse,
) {
  const authHeader = request.headers.get('authorization');
  if (
    !process.env.CRON_SECRET ||
    authHeader !== `Bearer ${process.env.CRON_SECRET}`
  ) {
    return response.status(401).json({ success: false });
  }

  response.status(200).json({ success: true });
}
```

```js filename="api/cron/route.js" framework=other
export default function handler(request) {
  const authHeader = request.headers.get('authorization');
  if (
    !process.env.CRON_SECRET ||
    authHeader !== `Bearer ${process.env.CRON_SECRET}`
  ) {
    return response.status(401).json({ success: false });
  }

  response.status(200).json({ success: true });
```

}
```

The `authorization` header will have the `Bearer` prefix for the value.

> For \['nextjs-app', 'nextjs']:

For those using TypeScript versions below 5.2, it's important to adapt the code to `import NextResponse from 'next/server'` and use `Next

## Cron job duration

The duration limits for Cron jobs are identical to those of [Vercel Functions](/docs/functions#limits). See the [`maxDuration`](/docs/fun

In most cases, these limits are sufficient. However, if you need more processing time, it's recommended to split your cron jobs into diff

## Cron job error handling

Vercel will not retry an invocation if a cron job fails. You can check for error [logs](/docs/runtime-logs) through the **View Log** butt

## Cron jobs with dynamic routes

Cron jobs can be created for [dynamic routes](https://nextjs.org/docs/routing/dynamic-routes):

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "crons": [
    {
      "path": "/api/sync-slack-team/T0CAQ10TZ",
      "schedule": "0 5 * * *"
    },
    {
      "path": "/api/sync-slack-team/T4BOE34OP",
      "schedule": "0 5 * * *"
    }
  ]
}
```

Handling nonexistent paths

If you create a cron job for a path that doesn't exist, it generates a [404 error](/docs/errors/platform-error-codes#404:-not_found). How

Cron jobs and deployments

Creating a new deployment will not interrupt your running cron jobs; they will continue until they finish.

Controlling cron job concurrency

If your cron job runs longer than the interval between invocations, Vercel can trigger a second instance while the first is still running

To prevent concurrent runs, use a lock mechanism like [Redis distributed locks](https://redis.io/docs/latest/develop/clients/patterns/dis

You can also prevent overlapping runs by:

- **Reducing execution time**: Optimize your job to finish before the next invocation
- **Setting timeouts**: Use [`maxDuration`](/docs/functions/runtimes#max-duration) to force long-running jobs to stop
- **Increasing the interval**: Run your cron job less frequently

Cron jobs and idempotency

Vercel's event-driven system can occasionally deliver the same cron event more than once. This means your job might run twice for a singl

Design your operations to be **idempotent** so they produce the same result even when executed multiple times. For example:

- **Good**: "Set user status to active" (running twice has the same effect)
- **Bad**: "Increment user credit by 10" (running twice doubles the credit)

To make operations idempotent:

- Use unique IDs to track which events you've already processed
- Check state before making changes (e.g., "if not already active, then activate")
- Store results with timestamps or version numbers

Use both locks (to prevent concurrent runs) and idempotency (to handle duplicate events safely) together for the most reliable cron jobs

Running cron jobs locally

Cron jobs are API routes. You can run them locally by making a request to their endpoint. For example, if your cron job is in `/api/cron`

There is currently no support for `vercel dev`, `next dev`, or other framework-native local development servers.

Cron jobs and redirects

Cron jobs do not follow redirects. When a cron-triggered endpoint returns a 3xx redirect status code, the job completes without further r

The view logs button on the cron job overview can be used to verify the response of the invocations and gain further insights.

Cron jobs logs

Cron jobs are logged as function invocations from the **Logs** tab of your projects [dashboard](/dashboard). You can view the logs for a

1. From the list of cron jobs, select **View Logs**.
2. This will take you to the [runtime logs](/docs/runtime-logs#request-path) view with a `requestPath` filter to your cron job such as `r

See [how to view runtime logs](/docs/runtime-logs#view-runtime-logs) for more information.

Note that when cron jobs respond with a redirect or a cached response, they will not be shown in the logs.

Cron jobs accuracy
```

Hobby users can only create cron jobs with [hourly accuracy](/docs/cron-jobs/usage-and-pricing#hobby-scheduling-limits). Vercel may invok

For all other teams, cron jobs will be invoked within the minute specified. For instance, the expression `5 8 * * *` would trigger an inv

Rollbacks with cron jobs

If you [Instant Rollback](/docs/instant-rollback) to a previous deployment, active cron jobs **will not** be updated. They will continue

---
title: "Cron Jobs"
description: "Learn about cron jobs, how they work, and how to use them on Vercel."
last_updated: "2026-01-16T02:19:28.450Z"
source: "https://vercel.com/docs/cron-jobs"

Cron Jobs

Cron jobs are time-based scheduling tools used to automate repetitive tasks. By using a specific syntax called a [cron expression](#cron-

Some common use cases of cron jobs are:

- Automating backups and archiving them
- Sending email and Slack notifications
- Updating Stripe subscription quantities

Vercel supports cron jobs for [Vercel Functions](/docs/functions). Cron jobs can be added through [`vercel.json`](/docs/project-configura

See [Managing Cron Jobs](/docs/cron-jobs/manage-cron-jobs) for information on duration, error handling, deployments, concurrency control,

Getting started with cron jobs

Learn how to set up and configure cron jobs for your project using our [Quickstart](/docs/cron-jobs/quickstart) guide.

How cron jobs work

To trigger a cron job, Vercel makes an HTTP GET request to your project's production deployment URL, using the `path` provided in your pr

Vercel Functions triggered by a cron job on Vercel will always contain `vercel-cron/1.0` as the user agent.

Cron expressions

Vercel supports the following cron expressions format:

Field	Value Range	Example Expression	Description
Minute	0 - 59	`5 * * * *`	Triggers at 5 minutes past the hour
Hour	0 - 23	`* 5 * * *`	Triggers every minute, between 05:00 AM and 05:59 AM
Day of Month	1 - 31	`* * 5 * *`	Triggers every minute, on day 5 of the month
Month	1 - 12	`* * * 5 *`	Triggers every minute, only in May
Day of Week	0 - 6 (Sun-Sat)	`* * * * 5`	Triggers every minute, only on Friday

Validate cron expressions

To validate your cron expressions, you can use the following tool to quickly verify the syntax and timing of your scheduled tasks to ensu

You can also use [crontab guru](https://crontab.guru/) to validate your cron expressions.

Cron expression limitations

- Cron jobs on Vercel do not support alternative expressions like `MON`, `SUN`, `JAN`, or `DEC`
- You cannot configure both day of the month and day of the week at the same time. When one has a value, the other must be `*`
- The timezone is always UTC

More resources

- [Managing Cron Jobs](/docs/cron-jobs/manage-cron-jobs)
- [Usage and Pricing](/docs/cron-jobs/usage-and-pricing)

---
title: "Getting started with cron jobs"
description: "Learn how to schedule cron jobs to run at specific times or intervals."
last_updated: "2026-01-16T02:19:28.536Z"
source: "https://vercel.com/docs/cron-jobs/quickstart"

Getting started with cron jobs

This guide will help you get started with using cron jobs on Vercel. Cron jobs are scheduled tasks that run at specific times or interval

Prerequisites

- [A Vercel account](/signup)
- [A project](/docs/projects/overview#creating-a-project) with a [Vercel Function](/docs/functions)

- ### Create a function
  This function contains the code that will be executed by the cron job. This example uses a simple function that returns the user's regi
  > For \['nextjs']:
  ```ts v0="build" filename="app/api/hello/route.ts" framework=nextjs
  export function GET(request: Request) {
    return new Response('Hello from Vercel!');
  }
  ```
  ```js v0="build" filename="app/api/hello/route.js" framework=nextjs
  export function GET(request) {
    return new Response('Hello from Vercel!');
  }
  ```

  ```ts filename="api/hello.ts" framework=other
  export function GET(request: Request) {
    return new Response('Hello from Vercel!');
  }
  ```
  ```js filename="api/hello.js" framework=other
  export function GET(request) {
    return new Response('Hello from Vercel!');
  }
  ```
  ```ts v0="build" filename="app/api/hello/route.ts" framework=nextjs-app
  export function GET(request: Request) {
    return new Response('Hello from Vercel!');
  }
  ```
  ```js v0="build" filename="app/api/hello/route.js" framework=nextjs-app
  export function GET(request) {
    return new Response('Hello from Vercel!');
  }
  ```

- ### Create or update your file
  Create or go to your [`vercel.json`](/docs/project-configuration#functions) file and add the following code:
  ```json filename="vercel.json"
  {
    "$schema": "https://openapi.vercel.sh/vercel.json",
    "crons": [
      {
        "path": "/api/hello",
        "schedule": "0 5 * * *"
      }
    ]
  }
  ```
  The `crons` property is an array of cron jobs. Each cron job has two properties:
  - The `path`, which must start with `/`
  - The `schedule` property, which must be a string that represents a [cron expression](/docs/cron-jobs#cron-expressions). In this exampl

- ### Deploy your project.
  When you deploy your project, Vercel's build process creates the cron job. Vercel invokes cron jobs only for [production](/docs/deploym

  You can also deploy to your production domain using the CLI:
  ```bash filename="terminal"
  vercel deploy --prod
  ```

Your cron job is now active and will call the `/api/hello` path every day at 5:00 am UTC.

Next steps

Now that you have created a cron job, you can learn more about how to manage and configure them:

- [Learn about managing cron jobs](/docs/cron-jobs/manage-cron-jobs)
- [Explore usage and pricing](/docs/cron-jobs/usage-and-pricing)

---
title: "Usage & Pricing for Cron Jobs"
description: "Learn about cron jobs usage and pricing details."
last_updated: "2026-01-16T02:19:28.502Z"
source: "https://vercel.com/docs/cron-jobs/usage-and-pricing"

Usage & Pricing for Cron Jobs

Cron jobs invoke [Vercel Functions](/docs/functions). This means the same [usage](/docs/limits) and [pricing](/pricing) limits will apply

	Number of cron jobs per account	**Schedule**
Hobby	2 cron jobs	Triggered once a day
Pro	40 cron jobs	Unlimited cron invocations
Enterprise	100 cron jobs	Unlimited cron invocations

Each project has a hard limit of **20 cron jobs per project**.

Hobby scheduling limits

On the Hobby plan, Vercel cannot assure a timely cron job invocation. For example, a cron job configured as `0 1 * * *` (every day at 1 a

For more specific cron job executions, **upgrade to our [Pro](/docs/plans/pro-plan) plan**.

Pricing

Cron jobs are included in **all plans**.

You use a function to invoke a cron job, and therefore [usage](/docs/limits) and [pricing](/pricing) limits for these functions apply to

- [Functions limits and pricing](/docs/functions/usage-and-pricing)

---
title: "Using the Command Menu "
description: "Learn how to quickly navigate through the Vercel dashboard with your keyboard using the Command Menu."
last_updated: "2026-01-16T02:19:28.522Z"
source: "https://vercel.com/docs/dashboard-features/command-menu"

Using the Command Menu

Vercel provides a menu with shortcuts, called the **Command Menu**, to navigate through the dashboard and perform common actions using on

You can access the menu by pressing + on macOS or + on Windows and Linux. Alternatively, you can access it by clicking on **Command M

Once opened, the Command Menu will offer you a list of commonly used shortcuts. For example, you can quickly navigate to a specific [Proj

The Command Menu is only available on desktop and tablet devices, but not on smartphones, as it provides the biggest efficiency advantage

Recently Used Items

By default, the list is comprised of shortcuts that the Vercel Team has found to be most useful for you. However, over time, the list wil

Up to 3 suggestions for recently used shortcuts will appear, and be ordered by the latest time you used them, with the most recently used

When the dashboard is closed, the suggestions will reset.

Context-Based Items

Because the purpose of the Command Menu is to get you to your desired goal in the quickest way possible, it also changes its behavior bas

If you're currently looking at the dashboard for a [Pro](/docs/plans/pro-plan) or [Enterprise](/docs/plans/enterprise) Team, for example,

Whereas, if you're on a [Hobby plan](/docs/plans/hobby) instead, you will not be offered that option because Hobby plans don't support co

Additional Keyboard Shortcuts

In addition to + (instead of , use on Windows or Linux) for opening the overview of the Command Menu, Vercel also offers direct keyboa

- Search [Projects](/docs/projects/overview) with +
- Search [Teams](/docs/accounts/create-a-team) with +
- Switch between light and dark mode with

They are also shown next to each of the supported items in the list.

Thanks to the shortcuts mentioned above, you often won't even have to navigate through the items offered by the Command Menu to get to yo

Instead, you can use these shortcuts to skip the overview of items and perform the action directly. Therefore, it is recommended to embed

Searching documentation

The Command Menu allows you to search through the documentation on the Vercel, [Next.js](https://nextjs.org/docs) and [Turborepo](https:/

Searching from the Vercel documentation

When on the Vercel documentation site:

1. Open the command menu using +
2. Begin typing a search query
3. The results will then be shown from across all three websites where there is a query match
   - Initially you will see the top three most relevant results
   - Followed by the rest of the results split by website
4. You can filter results by website using the following badges:
   - **All**: Results from all three websites (default)
   - **Vercel**: Results from the Vercel documentation site
   - **Next.js**: Results from the Next.js documentation site
   - **Turborepo**: Results from the Turborepo documentation site

Searching from the Vercel dashboard

When on the Vercel dashboard:

1. Open the command menu using +
2. Scroll down to **Search Docs...** and select, or use the shortcut +
3. Begin typing a search query
4. The results will then be shown from across all three websites where there is a query match
   - Initially you will see the top three most relevant results
   - Followed by the rest of the results split by website
5. You can filter results by website using the following badges:
   - **All**: Results from all three websites (default)
   - **Vercel**: Results from the Vercel documentation site
   - **Next.js**: Results from the Next.js documentation site
   - **Turborepo**: Results from the Turborepo documentation site

What about regular menus?

If you want, the Command Menu can be a complete replacement for the traditional dashboard navigation.

Regular menus will continue to exist for the purpose of navigating the dashboard with your mouse or fingers (on touch-based devices), but

Over time, the Command Menu will offer increasingly intelligent suggestions and allow for performing more actions inline to increase your

---
title: "Navigating the Dashboard"
description: "Learn how to select a scope, change the Project view, use search, or create a new project, all within the Vercel dashboard.
last_updated: "2026-01-16T02:19:28.544Z"
source: "https://vercel.com/docs/dashboard-features/overview"

Navigating the Dashboard

When you sign in to Vercel through your browser, you'll be presented with the dashboard. Any subsequent visits to [vercel.com](https://ve

Projects and repositories

Your dashboard view shows a list of all projects and repositories that belong to the [selected team](/docs/dashboard-features#scope-selec

You can click on the button to filter by a specific Repository and to choose whether to sort by Activity (which projects you have most r

You can use the toggle to change the view between a grid view and list view. Your viewing preference is saved to your account, so if you

Each project in the view shows:

- The deployed URL
- Information about the last commit
- The Real Experience Score for any deployments using Analytics

You can click on the button to:

- Add the project to your Favorites
- Visit the production deployment for the project with the [toolbar](/docs/vercel-toolbar)
- View Logs
- Manage Domains
- Transfer the project
- Go to Project Settings
- Access pages for the repository or [Conformance](/docs/conformance)

Project Dashboard

You can select any project to bring up its **Project Dashboard**, which allows you to view information about its deployments and configur

Learn more in [our project dashboard docs](/docs/projects/project-dashboard).

Search

Use the searchbar to search for the name of any deployed project.

Create

For accounts on a Hobby plan, you can either create a new team or a new project.

For members of a team, depending on your permissions, you can use the **Add New…** button to add a new project, domain, or team Member.

---
title: "Dashboard Overview"
description: "Learn how to use the Vercel dashboard to view and manage all aspects of the Vercel platform, including your Projects and De
last_updated: "2026-01-16T02:19:28.608Z"
source: "https://vercel.com/docs/dashboard-features"

Dashboard Overview

You can use the [Vercel dashboard](/dashboard) to view and manage all aspects of the Vercel platform, including your [Projects](/docs/pro

Scope selector

What you see in each tab is dependant on the *scope* that is selected.

The scope selector allows you to switch between your Hobby team and any teams that you may be part of. To switch between accounts and tea

To go back to your Team dashboard at any time, click the Vercel logo or the scope selector.

Find

The Find bar allows you to search for:

- Teams
- Projects
- Deployments (by branch)
- Pages
- Settings

Access this feature by clicking on the **Find** search input on the top right of the Vercel dashboard or pressing on your keyboard.

Overview

When you first create an account and log on to Vercel, you'll be greeted by your team overview. This shows information on all projects th

You can click on the button to filter by a specific Repository and to choose whether to sort by Activity (which projects you have most r

- [My dashboard](/dashboard)

Integrations

Integrations allow you to extend the capabilities of Vercel and connect with third-party platforms or services. Users and Teams on all pl

Through the Integrations section on the dashboard, you can [view and manage a list of all integrations](/docs/integrations/install-an-int

- [Integrations overview](/docs/integrations)
- [My integrations](/dashboard/integrations)

Activity

The Activity Log provides a list of all events on a Hobby team, chronologically organized since its creation. The [events](/docs/observab

- [Activity log overview](/docs/dashboard-features/activity-log)
- [My activity log](/dashboard/activity)

Recent Previews

The recent previews panel gives you a quick way to access recently deployed and viewed previews within your teams. It's scoped to the tea

Each listed preview shows the latest deployment ID and status. Any associated pull request to your git provider is also shown or the rele

Selecting a preview from the list will navigate to the live preview.

You can also navigate to related items for a preview deployment:

- The associated pull request or code repository page by clicking the label that will have the word "Code" or the pull request ID
- The deployment details by clicking the label with the deployment ID and status icon

Each preview deployment item also has a context menu where you can see further details and also remove the listing.

Domains

By default, all deployments are assigned a domain with the suffix: `.vercel.app`. This domain can be replaced with a Custom Domain of you

The Domains section of the dashboard lets you view all domains related to your account or Team, and allows you to **Buy**, **Add**, or **

- [Add a domain](/docs/domains/add-a-domain)
- [My domains](/dashboard/domains)

Usage

The Usage tab on the Dashboard provides detailed insight into the actual resource usage of all projects relating to your account or Team.

From the dashboard, you can filter the usage by billing cycle, date, project, or function.

- [Usage overview](/docs/limits/usage)
- [My usage](/dashboard/usage)

Settings

There are two different types of settings pages:

- **Personal Account / Team Settings** - These settings allow you to manage account details, billing, invoicing, membership, security, an

- **Project Settings** - You can view this by selecting a project in the dashboard and then selecting its settings. From there you can ma

- [My settings dashboard](/account)

Command Menu

Vercel provides a **Command Menu** that enables you to navigate through the dashboard and perform common actions using only the keyboard.

You can access the menu using by pressing + on macOS or + on Windows and Linux. Note that you must be logged in to access the Command

- [Command Menu overview](/docs/dashboard-features/command-menu)

---
title: "Support Center"
description: "Learn how to communicate securely with the Vercel support team"
last_updated: "2026-01-16T02:19:28.619Z"
source: "https://vercel.com/docs/dashboard-features/support-center"

Support Center

The Vercel Support Center provides a secure and streamlined way for you to submit support cases. The Support Center allows you to create

Submit a ticket

To submit a ticket to Vercel Support, do the following:

1. From your team's dashboard, select the **Support** tab and then click the **Contact Support** button
2. Select **Start Chat**, then select your team the issue is related to in the dropdown.
3. Start a conversation with the AI support agent.
4. If the AI is unable to resolve your issue, you can submit a ticket by clicking **Create Case** under the chat.
5. The AI agent will create a case for you and fill out the initial details based on your conversation with it. You can view the proposed
6. Once you are happy with the case, click **Submit Case**.

The team aims to respond to tickets as described in the [Support Terms](/legal/support-terms#when).

Manage tickets

You can see a list of all support cases, regardless of status, in the **Support** tab. This list shows the ticket name, number, and the s

Case correspondence

Each ticket displays all correspondences with the support team. Correspondence on your case is handled both over email and within the cas

Manage a ticket status

To manage the status of any ticket, do the following:

1. From your team's dashboard, select the **Support** tab
2. Find the ticket from the list and click the ticket name to open it
3. If the ticket was closed, click **Reopen case**. If it was open, click **Close case**. In either scenario, you may want to provide add

Send attachments to the support team

The support team may request additional logs or other information from you that you'll need to attach to your support ticket. To upload a

1. From your team's dashboard, select the **Support** tab
2. Find the ticket from the list and click the ticket name to open it
3. Click **Attach a File** to bring up your system file dialog. Select the relevant file and click **Upload**

---
title: "Data Cache for Next.js"
description: "Vercel Data Cache is a specialized cache that stores responses from data fetches in Next.js App Router"
last_updated: "2026-01-16T02:19:28.630Z"
source: "https://vercel.com/docs/data-cache"

Data Cache for Next.js

The Vercel Data Cache is a specialized, granular cache that was introduced with Next.js 13 for storing [segment-level data](https://nextj

When using [Next.js caching APIs](https://nextjs.org/docs/app/getting-started/caching-and-revalidating) such as `fetch` or `unstable_cach

Features

- **Globally available, regional cache**: Every region in which your function runs has an independent cache, so any data used in server s
- **Time-based revalidation**: All cached data can define a revalidation interval, after which the data will be marked as stale, triggeri
- **On-demand revalidation**: Any data can be triggered for revalidation on-demand, regardless of the revalidation interval. The revalida
- **Tag based revalidation**: Next.js allows associating tags with data, which can be used to revalidate all data with the same tag at on

Comparing with ISR and Vercel CDN Cache

Next.js combines Vercel Data Cache with [Incremental Static Regeneration](/docs/incremental-static-regeneration) (ISR) to provide an optii

When a page contains *entirely static data*, Vercel uses ISR to generate the whole page. However, when a page contains a *mix of static a

Both ISR and Vercel Data Cache support time-based revalidation, on-demand revalidation, and tag based revalidation.

[Vercel's Cache](/docs/cdn-cache) is used for caching entire static assets on the CDN, such as images, fonts, and JavaScript bundles. The

Managing Data Cache

When you deploy a Next.js project that uses [App Router](https://nextjs.org/docs/app) to Vercel, the Vercel Data Cache is automatically e

Observing your Data Cache usage

You can observe your project's Data Cache usage using the [**Observability**](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fobservability
To view your usage for Data Cache:

You can also track Data Cache usage per request in the [**Logs**](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Flogs\&title=Log

Manually purging Data Cache

In some circumstances, you may need to delete all cached data and force revalidation. You can do this by purging the Data Cache:

1. Under your project, go to the
   Settings tab.
2. In the left sidebar, select **Caches**.
3. In the **Data Cache** section, click **Purge Data Cache**.
4. In the dialog, confirm that you wish to delete and click the **Continue & Purge Data Cache** button.

Purging your Data Cache will create a temporary increase in request times for users as new data needs to be refetched.

Using Data Cache examples

These examples use the [Next.js App Router](https://nextjs.org/docs/app/building-your-application/data-fetching/fetching-caching-and-reva

Time-based revalidation

```ts v0="build" filename="app/page.tsx" framework=nextjs
type BlogPosts = Awaited<ReturnType<typeof getStaticProps>>['props']['blog'];

export default function Page({ blog }: { blog: BlogPosts }) {
  return (
    <main>
      <pre>{JSON.stringify(blog, null, 2)}</pre>
    </main>
  );
}

export async function getStaticProps() {
  const res = await fetch('https://api.vercel.app/blog');
  const blog = await res.json();

  return {
    props: {
      blog,
    },
    revalidate: 3600, // 1 hour
  };
}
```

```js v0="build" filename="app/page.jsx" framework=nextjs
export default function Page({ blog }) {
  return (
    <main>
      <pre>{JSON.stringify(blog, null, 2)}</pre>
    </main>
  );
}

export async function getStaticProps() {
  const res = await fetch('https://api.vercel.app/blog');
  const blog = await res.json();

  return {
    props: {
      blog,
    },
    revalidate: 3600, // 1 hour
  };
}
```

```ts v0="build" filename="app/page.tsx" framework=nextjs-app
export default async function Page() {
  const res = await fetch('https://api.vercel.app/blog', {
    next: {
```

```
      revalidate: 3600, // 1 hour
    },
  });
  const data = await res.json();

  return (
    <main>
      <pre>{JSON.stringify(data, null, 2)}</pre>
    </main>
  );
}
```

```js v0="build" filename="app/page.jsx" framework=nextjs-app
export default async function Page() {
  const res = await fetch('https://api.vercel.app/blog', {
    next: {
      revalidate: 3600, // 1 hour
    },
  });
  const data = await res.json();

  return (
    <main>
      <pre>{JSON.stringify(data, null, 2)}</pre>
    </main>
  );
}
```

Tag-based revalidation

```ts v0="build" filename="app/page.tsx" framework=all
export default async function Page() {
  const res = await fetch('https://api.vercel.app/blog', {
    next: {
      tags: ['blog'], // Invalidate with revalidateTag('blog') on-demand
    },
  });
  const data = await res.json();

  return '...';
}
```

```js v0="build" filename="app/page.jsx" framework=all
export default async function Page() {
  const res = await fetch('https://api.vercel.app/blog', {
    next: {
      tags: ['blog'], // Invalidate with revalidateTag('blog') on-demand
    },
  });
  const data = await res.json();

  return '...';
}
```

```ts v0="build" filename="app/actions.ts" framework=all
'use server';

import { revalidateTag } from 'next/cache';

export default async function action() {
  revalidateTag('blog');
}
```

```js v0="build" filename="app/actions.js" framework=all
'use server';

import { revalidateTag } from 'next/cache';

export default async function action() {
  revalidateTag('blog');
}
```

Revalidation behavior

The Vercel Data Cache infrastructure isolates cached data per Vercel project and [deployment environment](/docs/deployments/environments)

Vercel persists cached data across deployments, unless you explicitly invalidate it using framework api's like `res.revalidate`, `revalid

When the system triggers a revalidation, Vercel marks the corresponding path or cache tag as stale in every Vercel CDN region. The next r

Limits

Data Cache property	Limit
Item size	2 MB (items larger won't be cached)
Tags per item	128 tags
Maximum tag length	256 bytes

More resources

- [Explore Vercel regions](/docs/regions)
- [Next.js App Router template](/templates/next.js/app-directory)
- [Usage of Data Cache with ISR](/docs/data-cache#comparing-with-isr-and-vercel-cache)
- [Learn how Data Cache works in Next.js](https://nextjs.org/docs/app/deep-dive/caching#data-cache)

---
title: "Build Settings"
description: "Learn how to configure the Build & Development settings for your Vercel Deploy Button."
last_updated: "2026-01-16T02:19:28.556Z"
source: "https://vercel.com/docs/deploy-button/build-settings"

Build Settings

Build Command

Parameter	Type	Description
`build-command`	`string`	Setting this value is equivalent to enabling the **Override** toggle for that field in the dashboard.

This allows you to define a custom Build command that is normally automatically configured based on your Project's framework.

The example below shows a source URL using the `build-command` parameter to set the Build command to `npm run build`:

```bash filename="source url"
https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&build-co
```

Install Command

Parameter	Type	Description
`install-command`	`string`	Setting this value is equivalent to enabling the **Override** toggle for that field in the dashboard.

This allows you to define a custom Install command that is normally automatically configured based on the following:

Lock File	Install Command	Package Manager Version
`pnpm-lock.yaml`	`pnpm install`	`pnpm 6/7/8/9/10` See [supported package managers](/docs/package-managers#supported-package-man
`package-lock.json`	`npm install`	`npm`
`bun.lockb`	`bun install`	`bun 1`
`bun.lock`	`bun install`	`bun 1`
None	`npm install`	N/A

The example below shows a source URL using the `install-command` parameter to set the Install command to `npm install`:

```bash filename="source url"
https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&install-
```

Development Command

Parameter	Type	Description
`dev-command`	`string`	Setting this value is equivalent to enabling the **Override** toggle for that field in the dashboard.

This allows you to define a custom development command if you are using `vercel dev` to test your project locally. Each framework has its

The example below shows a source URL using the `dev-command` parameter to set the Development command to `next dev --port $PORT`:

```bash filename="source url"
https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&dev-comm
```

Ignored Build Command

Parameter	Type	Description
`ignore-command`	`string`	Setting this value is equivalent to enabling the **Override** toggle for that field in the dashboard.

This allows you to define an Ignored Build Step to determine when your project should build and deploy.

The example below shows a source URL using the `ignore-command` parameter to set the Ignored Build Step command to `npx turbo-ignore`:

```bash filename="source url"
https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&ignore-c
```

Root Directory

Parameter	Type	Description
`root-directory`	`string`	Setting this value is equivalent to enabling the **Override** toggle for that field in the dashboard.

This allows you to define the path of the directory relative to the root of the Project folder where your source code is located. By defa

The example below shows a source URL using the `root-directory` parameter to set the Root Directory to `apps/frontend`:

```bash filename="source url"
https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel-support%2Fyarn-ws-monorepo&root-directory=apps%2Ffrontend
```

Output Directory

Parameter	Type	Description
`output-directory`	`string`	Setting this value is equivalent to enabling the **Override** toggle for that field in the dashboard.

This allows you to define the output directory's path relative to the Project folder's root. Usually, this is automatically configured ba

The example below shows a source URL using the `output-directory` parameter for a monorepo where the application output is generated to `

```bash filename="source url"
```

https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&output-d
```

---
title: "Using Callbacks with the Deploy Button"
description: "Learn how to use the Deploy Button"
last_updated: "2026-01-16T02:19:28.565Z"
source: "https://vercel.com/docs/deploy-button/callback"
---

# Using Callbacks with the Deploy Button

## Redirect URL

| Parameter | Type | Value |
| --- | --- | --- |
| `redirect-url` | `string` | The URL to redirect the user to in the event of a successful deployment. |

The Redirect URL parameter allows you to define a URL, other than the newly created Vercel project, to send the user to after a successfu

This parameter is helpful if you are sending a user from an application, to deploy a project with Vercel, but want the user to continue w

The example below shows a Deploy Button source URL using the Redirect URL parameter:

```bash filename="redirect url"
https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&redirect
```

Provide a custom name and logo for the redirect UI by using the [Developer ID](#developer-id) parameter.

Callback Parameters

Vercel additionally attaches some "Callback Parameters" to the defined Redirect URL when the user is redirected. The following parameters

Parameter	Description
	The URL to view the Project that was created through the Project creation flow on the Vercel Dash
	The Name of the Project that was created through the Project creation flow.
	The URL to view the Deployment that was created through the Project creation flow on the Vercel Dash
	The URL of the deployment that was created through the Project creation flow. This contain
	The URL of the Git repository that was created through the Project creation flow, within t
([conditional](#deploy-hook))	The URL of a Deploy Hook. Requires [the `production-deploy-hook` parameter](#deploy-hook).

Developer ID

Parameter	Type	Value
`developer-id`	`string`	The Client ID of an Integration.

The Developer ID parameter allows you to define a [Vercel Integration](/docs/integrations) Client ID which will then attach your logo and

You can find the Developer ID listed as "Client ID" in your [Integrations Developer Console](/dashboard/integrations/console).

This parameter requires the [Redirect URL](#redirect-url) parameter to be set and also that the Integration website field matches the Red

The example below shows a Deploy Button source URL using the Redirect URL and Developer ID parameters:

```bash filename="redirect url"
https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&redirect
```

External ID

Parameter	Type	Value
`external-id`	`string`	An external ID or reference of your choice.

This parameter allows you to pass the ID or reference of your choice to the Project creation flow.

The query parameter will be relayed to the [Redirect URL](/docs/integrations/create-integration) of
each required [Integration](/docs/integrations/deploy-button/integrations) when the user adds them in the Project creation flow.

To use this parameter, you also need to specify at least one [Integration](/docs/integrations/deploy-button/integrations).

The example below shows a Deploy Button source URL using the Integration ID and External ID parameters:

```bash filename="external id"
https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&integrat
```

Deploy Hook

Parameter	Type	Value
`production-deploy-hook`	`string`	The name of the Deploy Hook to set up.

The Deploy Hook parameter allows you to receive [a URL](/docs/deploy-hooks) when also using the Redirect URL parameter, which you can use

This is useful if you are directing a user to deploy a project that works with your application, for example a headless CMS, and you need

The value of this parameter should be the name of the [Deploy Hook](/docs/deploy-hooks) you want to create for the user.

When redirected back to your application upon a successful deployment for the user, you will get the `production-deploy-hook-url` callbac

This parameter requires the [Redirect URL](#redirect-url) parameter to also be set.

The example below shows a Deploy Button source URL using the Redirect URL and production Deploy Hook URL parameters:

```bash filename="deploy hook"

https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&redirect
```

---
title: "Deploy Button Demo"
description: "Learn how to use the Deploy Button Demo parameters to showcase an example of a successful deployment to the user when click
last_updated: "2026-01-16T02:19:28.582Z"
source: "https://vercel.com/docs/deploy-button/demo"

Deploy Button Demo

Demo Title

Parameter	Type	Value	Required
`demo-title`	`string`	The title of an example deployment.	Yes

This parameter allows you to specify the title of an
example of a successful deployment.

The parameter is part of the Demo Card parameters. The Demo Card should
showcase an example of a successful deployment to the user clicking
the Deploy Button and entering the Project creation flow.

> **💡 Note:** The Demo card is displayed only when all
> parameters are provided.

The example below shows how to use the `demo-title` parameter in the Deploy Button source URL:

```bash filename="demo title"
https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&demo-tit
```

Demo Description

Parameter	Type	Value	Required
`demo-description`	`string`	The description of an example deployment.	Yes

This parameter allows you to specify the description of an
example of a successful deployment.

The parameter is part of the Demo Card parameters. The Demo Card should
showcase an example of a successful deployment to the user clicking
the Deploy Button and entering the Project creation flow.

> **💡 Note:** The Demo card is displayed only when all
> parameters are provided.

The example below shows how to use the `demo-description` parameter in the Deploy Button source URL:

```bash filename="demo description"
https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&demo-des
```

Demo URL

Parameter	Type	Value	Required
`demo-url`	`string`	The URL of an example deployment.	Yes

This parameter allows you to specify the URL of an example of a successful deployment.

The parameter is part of the Demo Card parameters. The Demo Card should showcase an example of a successful deployment to the user clicki

> **💡 Note:** The Demo card is displayed only when all
> parameters are provided.

The example below shows how to use the `demo-url` parameter in the Deploy Button source URL:

```bash filename="demo url"
https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&demo-url
```

Demo Image

Parameter	Type	Value	Required
`demo-image`	`string`	The URL of the screenshot of an example deployment.	Yes

This parameter allows you to specify the URL of the screenshot of an example of a successful deployment.

The parameter is part of the Demo Card parameters. The Demo Card should showcase an example of a successful deployment to the user clicki

> **💡 Note:** The Demo card is displayed only when all
> parameters are provided.

The example below shows how to use the `demo-image` parameter in the Deploy Button source URL:

```bash filename="demo image"
https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&demo-ima
```
```

---
title: "Using Environment Variables with the Deploy Button"
description: "Learn how to use Environment Variables with the Vercel Deploy Button."
last_updated: "2026-01-16T02:19:28.589Z"
source: "https://vercel.com/docs/deploy-button/environment-variables"

Using Environment Variables with the Deploy Button

Required environment variables

Parameter	Type	Value
`env`	`string[]`	A comma-separated list of required environment variable keys.

Use the `env` parameter to require users to fill in values for environment variables that your project needs to run.

The example below shows how to use the `env` parameter in a Deploy Button source URL:

```bash filename="env"
https://vercel.com/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&env=PUBLIC_A
```

> **⚠️ Warning:** You cannot pass environment variable values using this parameter because the
> URL is saved in the browser history, making it insecure.

Environment variables default values

Parameter	Type	Value
`envDefaults`	`string`	A JSON-encoded object mapping environment variable keys to default values.

Set non-sensitive default values for required environment variables with the `envDefaults` parameter. When users click the Deploy Button,

Default values should only be used for non-sensitive configuration. Examples of appropriate use cases:

- Feature flags (e.g., `ENABLE_ANALYTICS=true`)
- Public API endpoints (e.g., `API_BASE_URL=https://api.example.com`)
- Default configuration values (e.g., `MAX_ITEMS_PER_PAGE=10`)
- Non-sensitive application settings

> **⚠️ Warning:** Never use default values for sensitive data like passwords, API keys, tokens,
> database credentials, or any secret values. Users should always enter these
> manually.

The parameter expects a JSON object where keys are the environment variable names (which must also be listed in the `env` parameter), and

The example below shows how to use the `envDefaults` parameter:

```bash filename="envDefaults"
https://vercel.com/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&env=NEXT_PUB
```

The decoded JSON in this example is:

```json
{
  "NEXT_PUBLIC_API_URL": "https://api.example.com",
  "ENABLE_FEATURE_X": "true"
}
```

> **💡 Note:** Default values only apply if the environment variable is listed in the `env`
> parameter. Users can still modify or clear these values before deploying.

Environment variables description

Parameter	Type	Value
`envDescription`	`string`	A short description of the required environment variables

Add a description that explains what the required environment variables are used for with the `envDescription` parameter. The description

> **💡 Note:** The description provided through this parameter only shows if required
> environment variables are set.

The example below shows how to use the `envDescription` parameter in a Deploy Button source URL:

```bash filename="envDescription"
https://vercel.com/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&env=PUBLIC_A
```

Environment variables link

Parameter	Type	Value
`envLink`	`string`	A link to an explanation of the required environment variables

Attach a link to external documentation that helps users find the values they need with the `envLink` parameter. This link should point t

> **💡 Note:** The link provided through this parameter only shows if required environment
> variables are set.

The example below shows how to use the `envLink` parameter in a Deploy Button source URL. Make sure you provide users with a specific lin

```bash filename="envLink"
https://vercel.com/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&env=PUBLIC_A
```

---
title: "Using Integrations with the Deploy Button"
description: "Learn how to use Integrations with the Vercel Deploy Button."
last_updated: "2026-01-16T02:19:28.597Z"
source: "https://vercel.com/docs/deploy-button/integrations"

Using Integrations with the Deploy Button

Required Integrations

Parameter	Type	Value
`integration-ids`	`string[]`	A comma-separated list of required Integrations IDs: `oac_4mkAfc68cuDV4suZRlgkn3R9, oac_JI9dt8xHo7UXmV`

This parameter allows you to specify a list of Integration IDs. When specified, the corresponding Integrations will be required to be add

You can find the IDs of your Integrations in the [Integrations Console](/dashboard/integrations/console).

The example below shows how to use the `integration-ids` parameter in a Deploy Button source URL:

```bash filename="integration id"
https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&integrat
```

Skippable Integrations

Parameter	Type	Value
`skippable-integrations`	`number`	Mark the list of provided Integrations as optional

If this parameter is present, the user will be able to add one of the provided Integrations or skip them entirely, instead of being force

Because the user will only be able to select one (not multiple) of the optional Integrations, they should all serve the same purpose. For

To use this parameter, you also need to specify at least one Integration.

The example below shows how to use the `skippable-integrations` parameter in a Deploy Button source URL:

```bash filename="skippable integrations"
https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&integrat
```

---
title: "Working with the Deploy Button"
description: "Deploy public Git projects with the Deploy Button, and set up new projects with Vercel and GitHub, GitLab, or Bitbucket"
last_updated: "2026-01-16T02:19:28.637Z"
source: "https://vercel.com/docs/deploy-button"

Working with the Deploy Button

The Deploy Button allows users to deploy a new project through the Vercel Project creation flow, while cloning the source Git repository

You can [create your Deploy Button with the generator below](#generate-your-own).

The Vercel Project creation flow allows users to deploy a Git repository, create a project with Vercel, and clone the source repository i

Snippets

With the Vercel Project creation flow, you can add various URL query parameters to control the experience a user will have, depending on

---
title: "Deploy Button Source"
description: "Learn how to use the Vercel Deploy Button source URL parameters.
  "
last_updated: "2026-01-16T02:19:28.661Z"
source: "https://vercel.com/docs/deploy-button/source"

Deploy Button Source

Repository URL

Parameter	Type	Value
`repository-url`	`string`	The source Git repository URL

The Repository URL parameter allows you to define a Git repository URL, optionally including the subdirectory within a repository, that u

The example below shows how to use the Repository URL parameter to set the repository URL to `hello-world`:

```bash filename="repository url"
https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world
```

The Repository URL parameter is required when sending a user to the Vercel Project creation flow to set up a project from a GitHub, GitLa

Project Name

Parameter	Type	Value
`project-name`	`string`	A default project name

The Project Name parameter allows you to define a default project name.

This parameter is useful for cases where already know what the user would like to name their project. For example, if you are sending the

If there is an existing project using the name passed with this parameter, the user will be required to define a new project name, and th

The example below shows how to use the Project Name parameter to set the project name to "my-awesome-project":

```bash filename="project name"
https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&project-
```

Repository Name

Parameter	Type	Value
`repository-name`	`string`	A default repository name (no spaces)

The Repository Name parameter allows you to define a default repository name.

Similar to the [Project Name](#project-name) parameter, this parameter is useful in cases where you already know what the user wants to n

The example below shows how to use the Repository Name parameter to set the repository name to "my-awesome-project":

```bash filename="repository name"
https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fnext.js%2Ftree%2Fcanary%2Fexamples%2Fhello-world&reposito
```

Store product integration

Parameter	Type	Value
`stores`	`string`	A default JSON object converted to a string

The JSON object parameter allows you to define a default store product.

The example below shows how to set the parameter to the following JSON value:

```json
{
  "type": "integration",
  "integrationSlug": "my-integration-slug",
  "productSlug": "my-product-slug",
  "protocol": "storage"
}
```

First, convert the parameter to a string:

```js
const jsonParam = encodeURIComponent(
  JSON.stringify([
    {
      type: 'integration',
      integrationSlug: 'my-integration-slug',
      productSlug: 'my-product-slug',
      protocol: 'storage',
    },
  ]),
);
```

Then, use it as follows:

```bash filename="stores"
https://vercel.com/new/clone?repository-url=https%3A%2F%2Fgithub.com%2Fvercel%2Fvercel%2Ftree%2Fmain%2Fexamples%2Fnextjs&project-name=my-
```

---
title: "Creating & Triggering Deploy Hooks"
description: "Learn how to create and trigger deploy hooks to integrate Vercel deployments with other systems."
last_updated: "2026-01-16T02:19:28.669Z"
source: "https://vercel.com/docs/deploy-hooks"

Creating & Triggering Deploy Hooks

Deploy Hooks allow you to create URLs that accept HTTP `POST` requests in order to trigger deployments and re-run the [Build Step](/docs/

This feature allows you to integrate Vercel deployments with other systems. For example, you can set up:

- Automatic deployments on content changes from a Headless CMS
- Scheduled deployments by configuring third-party cron job services to trigger the Deploy Hook
- Forced deployments from the command line

Creating a Deploy Hook

To create a Deploy Hook for your project, make sure your project is [connected to a Git repository](/docs/projects/overview#git).

Once your project is connected, navigate to its **Settings** page and then select the **Git** menu item.

In the "Deploy Hooks" section, choose a name for your Deploy Hook and select the branch that will be deployed when the generated URL is r

> **💡 Note:** We suggest you use a name that easily identifies the Deploy Hook so you will
> be able to understand when it triggers a deployment. We also suggest creating
> only one Deploy Hook per branch unless you're using multiple data sources.

After submitting the form, you will see a URL that you can copy and use.

Triggering a Deploy Hook

To trigger a Deploy Hook, send a GET or POST request to the provided URL.

> **💡 Note:** Deploy Hooks will not be triggered if you have the `github.enabled = false`
> [configuration](/docs/project-configuration/git-configuration#github.enabled)
> present in your `vercel.json` file.

Here's an example request and response you can use for testing:

Example Request

```bash filename="example-request"
curl -X POST https://api.vercel.com/v1/integrations/deploy/prj_98g22o5YUFVHlKOzj9vKPTyN2SDG/tKybBxqhQs
```

Example Response

```json filename="example-response"
{
  "job": {
    "id": "okzCd50AIap1O31g0gne",
    "state": "PENDING",
    "createdAt": 1662825789999
  }
}
```

> **💡 Note:** You do not need to add an authorization header. See [Security](#security) to
> learn more.

After sending a request, you can see that it triggered a deployment on your project dashboard.

Security

When you create a Deploy Hook, a unique identifier is generated in the URL. This allows anyone with the URL to deploy your project, so tr

If you believe your Deploy Hook URL has been compromised, you can revoke it and create a new one.

Build Cache

Builds triggered by a Deploy Hook are automatically provided with
an appropriate [Build Cache](/docs/deployments/troubleshoot-a-build#what-is-cached) by default, if it exists.

Caching helps speed up the [Build Step](/docs/deployments/configure-a-build), so we encourage you to keep the default behavior.
However, if you explicitly want to opt out of using a Build Cache, you can disable it by
appending `?buildCache=false` to the Deploy Hook URL.

Here is an example request that explicitly disables the Build Cache:

```bash filename="example-request"
curl -X POST https://api.vercel.com/v1/integrations/deploy/prj_98g22o5YUFVHlKOzj9vKPTyN2SDG/tKybBxqhQs?buildCache=false
```

> **💡 Note:** Deploy Hooks created before May 11th, 2021 do not have the Build Cache enabled
> by default. To change it, you can either explicitly append `?buildCache=true`
> to the Deploy Hook URL, or replace your existing Deploy Hook with a newly
> created one.

Other Optimizations

If you send multiple requests to deploy the same version of your project,
previous deployments for the same Deploy Hook will be canceled to reduce build times.

Limits

- Hobby and Pro accounts have a limit of 5 deploy hooks per project. Enterprise accounts have a limit of 10 deploy hooks per project.

Troubleshooting

If your deploy hook fails to create a deployment, check the status check on the commit associated with the deploy hook to identify any fa

---
title: "Deployment Checks"
description: "Set conditions that must be met before proceeding to the next phase of the deployment lifecycle."
last_updated: "2026-01-16T02:19:28.679Z"
source: "https://vercel.com/docs/deployment-checks"

Deployment Checks

Deployment Checks are conditions that must be met before promoting a production build to your production environment.

When a project is connected to GitHub using [Vercel for GitHub](/docs/git/vercel-for-github), Vercel can automatically read the statuses

Understanding Deployment Checks

Decoupling production builds and releases allows teams to move faster with higher confidence at scale.

- Feature branches are worked on in isolation and merged to the default branch once the code passes required checks for merging.
- Production deployments are created after new code is merged, but must pass a set of required checks before being released to end users.

By default, Vercel automatically promotes your most recent, successful production build to your custom production domains. This creates t

1. Push or merge code to your default branch.
2. Vercel creates a production build.
3. Once the build is ready, release the build to production.

At scale, this can mean the set of code that is tested **before merging** is not the same as the code that would be released to end users

With Deployment Checks, you introduce a new step that ensures the safety of the production deployment before it's released, with the foll

1. Push or merge code to your default branch.
2. Vercel creates a production deployment.
3. **Run safety checks to ensure that the build is safe for release.**
4. **Once Deployment Checks are passing**, release the build to production.

Enabling Deployment Checks

- ### Ensure prerequisites are enabled
  1. Link your project to a Github repository using [Vercel for GitHub](/docs/git/vercel-for-github). This can be verified by navigating
  2. Visit [your project's production environment settings](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Fsettings%2Fenvironmen

- ### Select your Deployment Checks
  Visit [your project's settings](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Fsettings%2Fdeployment-checks), and select *Add

- ### Update workflows (if necessary)
  If using GitHub Actions with a [`repository_dispatch`](https://docs.github.com/en/actions/writing-workflows/choosing-when-your-workflow
  ```yaml
  - name: 'Notify Vercel'
    uses: 'vercel/repository-dispatch/actions/status@v1'
    with:
      # The name of the check will be used to identify the check in the Deployment Checks settings and must be unique
      name: "Vercel - my-project: e2e-tests"
  ```
  If you are **not** using [`repository_dispatch`](https://docs.github.com/en/actions/writing-workflows/choosing-when-your-workflow-runs/

- ### Create a new production deployment
  Deployment Checks appear as part of a production deployment's lifecycle. Production deployments will still be created, but will not be

- ### Run GitHub Actions to fulfill all Deployment Checks
  To meet Deployment Checks, run their corresponding GitHub Actions.

  If you're using [`repository_dispatch`](https://docs.github.com/en/actions/writing-workflows/choosing-when-your-workflow-runs/events-th

- ### Promote to production once all Deployment Checks are met
  Once all of the Deployment Checks have passed, the deployment is aliased to your production domain(s) automatically.

  For additional release protection, enable [Rolling Releases](/docs/rolling-releases) to ensure your deployment is fractionally released

Bypassing Deployment Checks

You can bypass Deployment Checks by selecting [Force Promote](/docs/deployments/promoting-a-deployment) from the deployment details page.

Limitations

GitHub and GitHub Actions have edge cases with status reporting. These behaviors are matched in GitHub-backed Deployment Checks.

- To trigger a workflow in response to Vercel deployments using [`repository_dispatch`](https://docs.github.com/en/actions/writing-workfl
- GitHub uses the names of jobs to identify which checks are the same across instances. This means that:
  - Changing the name of a job requires updating your Deployment Checks to align with the names
  - Each run of a GitHub Workflow should result in only one commit status. For example, when using [`repository_dispatch`](https://docs.g
- Avoid using the same name for actions across multiple workflows. Due to GitHub's implementation of Check Runs, these will collide and i

---
title: "Deployment Protection Exception"
description: "Learn how to disable Deployment Protection for a list of preview domains."
last_updated: "2026-01-16T02:19:28.704Z"
source: "https://vercel.com/docs/deployment-protection/methods-to-bypass-deployment-protection/deployment-protection-exceptions"

Deployment Protection Exception

You can use [Deployment Protection Exceptions](/docs/security/deployment-protection/methods-to-bypass-deployment-protection/deployment-pr

When you add a domain to Deployment Protection Exceptions, it will automatically become publicly accessible and will no longer be covered

Deployment Protection Exceptions is designed for Preview Deployment domains, if you wish to make a Production Deployment domain public, s

Adding a Deployment Protection Exception

- ### Go to Project Deployment Protection Settings
  From your Vercel [dashboard](/dashboard):
  1. Select the project that you wish to enable Password Protection for
  2. Go to the **Settings** tab, and then to **Deployment Protection**

- ### Select Add Domain
  From the **Deployment Protection Exceptions** section, select **Add Domain**:

- ### Specify domain
  From the **Unprotect Domain** modal:
  1. Enter the domain that you wish to unprotect in the input
  2. Select **Continue**

- ### Confirm domain
  From the **Unprotect Domain** modal:
  1. Confirm the domain by entering it again in the first input
  2. Enter `unprotect my domain` in the second input
  3. Select **Confirm**
  All your existing and future deployments for that domain will be **unprotected**.

Removing a Deployment Protection Exception

- ### Go to Project Deployment Protection Settings
  From your Vercel [dashboard](/dashboard):
  1. Select the project that you wish to enable Password Protection for
  2. Go to the **Settings** tab, and then to **Deployment Protection**

- ### Select the Domain to Remove
  From the **Deployment Protection Exceptions** section:

  1. From the domain row in the **Unprotected Domains** table
  2. Select the dot menu at the end of the row
  3. From the context menu, select **Remove**

- ### Confirm the Domain to Remove
  From the **Reprotect Domain** modal:
  1. In the modal, type the domain in the first input
  2. Type `reprotect my domain` in the second input
  3. Select **Confirm**
  All your existing and future deployments for that domain will be **protected**.

Managing Deployment Protection Exceptions

You can view and manage all the existing Deployment Protection Exceptions for your team in the following way

1. From your [dashboard](/dashboard) go to the **Settings** tab
2. Select Deployment Protection and then choose the **Access** tab
3. Click the **All Access** button and select `Unprotected Previews`

---
title: "OPTIONS Allowlist"
description: "Learn how to disable Deployment Protection for CORS preflight requests for a list of paths."
last_updated: "2026-01-16T02:19:28.725Z"
source: "https://vercel.com/docs/deployment-protection/methods-to-bypass-deployment-protection/options-allowlist"

OPTIONS Allowlist

You can use OPTIONS Allowlist to disable Deployment Protection (including [Vercel Authentication](/docs/security/deployment-protection/me

When you add a path to OPTIONS Allowlist, any incoming request with the method `OPTIONS` that **starts with** the path will no longer be

For example, if you specify `/api`, all requests to paths that start with `/api` (such as `/api/v1/users` and `/api/v2/projects`) will be

Enabling OPTIONS Allowlist

- ### Go to Project Deployment Protection Settings
  From your Vercel [dashboard](/dashboard):
  1. Select the project that you wish to enable Password Protection for
  2. Go to the **Settings** tab, and then to **Deployment Protection**

- ### Enable OPTIONS Allowlist
  From the **OPTIONS Allowlist** section, enable the toggle labelled **Disabled**:

- ### Specify a path
  Specify a path to add to the **OPTIONS Allowlist**:

- ### Add more paths
  To add more paths, select **Add path**:

- ### Save changes
  Once all the paths are added, select **Save**

Disabling OPTIONS Allowlist

- ### Go to Project Deployment Protection Settings
  From your Vercel [dashboard](/dashboard):
  1. Select the project that you wish to enable Password Protection for
  2. Go to the **Settings** tab, and then to **Deployment Protection**

- ### Disable OPTIONS Allowlist
  From the **OPTIONS Allowlist** section, select the toggle labelled **Enabled**:

- ### Save changes
  Once all the paths are added, select **Save**

---
title: "Methods to bypass Deployment Protection"
description: "Learn how to bypass Deployment Protection for specific domains, or for all deployments in a project."
last_updated: "2026-01-16T02:19:28.740Z"
source: "https://vercel.com/docs/deployment-protection/methods-to-bypass-deployment-protection"

Methods to bypass Deployment Protection

To test, share, or exclude specific domains from Deployment Protection, you can use the following methods to allow specific access while r

- [**Shareable Links**](#sharable-links): Shareable links enable external users to access specific branch deployments by appending a secu
- [**Protection Bypass for Automation**](#protection-bypass-for-automation): Use a secret to bypass protection features for all deploymen
- [**Deployment Protection Exceptions**](#deployment-protection-exceptions): Specify preview domains that should be exempt from deploymen
- [**OPTIONS Allowlist**](#options-allowlist): Specify paths to be unprotected for CORS preflight `OPTIONS` requests

Sharable Links

Sharable Links allow external access to specific branch deployments through a secure query parameter. Users with this link can see the la

For example, if you generate a Sharable Link for the `feature-new-ui` branch. Users with this link can view the latest deployment and com

Learn more about [Sharable Links](/docs/security/deployment-protection/methods-to-bypass-deployment-protection/sharable-links), and how t

Protection bypass for Automation

For automated tasks like end-to-end (E2E) testing, you can use Protection bypass for Automation. When enabled, it generates a secret that

For example, you set up E2E tests that run on deployments. By using this feature and the generated secret, your tests can bypass the prot

Learn more about [Protection bypass for Automation](/docs/security/deployment-protection/methods-to-bypass-deployment-protection/protecti

Deployment Protection Exceptions

With Deployment Protection Exceptions you can specify preview domains that should be exempt from deployment protection. Adding a domain t

For example, if you add **`preview-branch-name.vercel.app`** to Deployment Protection Exceptions, this domain becomes publicly accessible

Learn more about [Deployment Protection Exceptions](/docs/security/deployment-protection/methods-to-bypass-deployment-protection/deployme

OPTIONS Allowlist

With OPTIONS Allowlist you can specify paths to be unprotected for preflight OPTIONS requests. This can be used to enable [CORS preflight

Incoming request paths will be compared with the paths in the allowlist, if a request path **starts with** one of the specified paths, an

For example, if you specify `/api`, all requests to paths that start with `/api` (such as `/api/v1/users` and `/api/v2/projects`) will be

Learn more about [OPTIONS Allowlist](/docs/security/deployment-protection/methods-to-bypass-deployment-protection/options-allowlist).

More resources

- [Understanding Deployment Protection by environment](/docs/deployment-protection#understanding-deployment-protection-by-environment)
- [Methods to protect deployments](/docs/deployment-protection/methods-to-protect-deployments)

---
title: "Protection Bypass for Automation"
description: "Learn how to bypass Vercel Deployment Protection for automated tooling (e.g. E2E testing)."
last_updated: "2026-01-16T02:19:28.749Z"
source: "https://vercel.com/docs/deployment-protection/methods-to-bypass-deployment-protection/protection-bypass-automation"

Protection Bypass for Automation

The Protection Bypass for Automation feature lets you bypass Vercel Deployment Protection ([Password Protection](/docs/security/deploymen

You can create multiple automation bypass secrets for a project, allowing you to manage access for different tools or environments indepe

Each generated secret can be used to bypass deployment protection on all deployments in a project until it is revoked. One bypass secret

The environment variable value is set when a deployment is built, so regenerating or deleting the secret in the project settings will inv

Who can manage protection bypass for automation?

- [Team members](/docs/rbac/access-roles#team-level-roles) with at least the [member](/docs/rbac/access-roles#member-role) role
- [Project members](/docs/rbac/access-roles#project-level-roles) with the [Project Administrator](/docs/rbac/access-roles#project-adminis

Using Protection Bypass for Automation

To use Protection Bypass for Automation, set an HTTP header (or query parameter) named `x-vercel-protection-bypass` with the value of the

Using a header is strongly recommended, however in cases where your automation tool is unable to specify a header, it is also possible to

Advanced Configuration

To bypass authorization on follow-up requests (e.g. for **in-browser testing**) you can set an additional header or query parameter named

This will set the authorization bypass as a cookie using a redirect with a `Set-Cookie` header.

If you are accessing the deployment through a non-direct way (e.g. in an `iframe`) then you may need to further configure `x-vercel-set-b

This will set `SameSite` to `None` on the `Set-Cookie` header, by default `SameSite` is set to `Lax`.

Examples

Playwright

---
title: "Sharable Links"
description: "Learn how to share your deployments with external users."
last_updated: "2026-01-16T02:19:28.761Z"
source: "https://vercel.com/docs/deployment-protection/methods-to-bypass-deployment-protection/sharable-links"

Sharable Links

Shareable links allow external users to securely access your deployments through a query string parameter.
Shareable links include the ability to leave [Comments](/docs/comments) on deployments which have them enabled.

Who can create Shareable Links?

- Non-Production Domains:
  - [Team members](/docs/rbac/access-roles#team-level-roles) with at least the [Developer](/docs/rbac/access-roles#developer-role) role
  - [Project members](/docs/rbac/access-roles#project-level-roles) with at least the [Project Developer](/docs/rbac/access-roles#project-
- Production Domains:
  - [Team members](/docs/rbac/access-roles#team-level-roles) with at least the [Member](/docs/rbac/access-roles#member-role) role
  - [Project members](/docs/rbac/access-roles#project-level-roles) with the [Project Administrator](/docs/rbac/access-roles#project-admin

Creating Sharable Links

Users with the Admin, Member, and Developer roles can create or revoke sharable links for their project's deployments. Personal accounts

> **💡 Note:** Developers on the hobby plan can only create one shareable link in total per
> account.

To manage Sharable Links, do the following:

- ### Select your project
  From your Vercel [dashboard](/dashboard):
  1. Select the project that you wish to enable Vercel Authentication for

  2. Go to the **Deployments** tab

- ### Select the deployment
  From the list of **Preview Deployments**, select the deployment you wish to share.

- ### Click Share button
  From the Deployment page, click **Share** to display the **Share** popover. From the popover, select **Anyone with the link** from the

- ### Revoking a Sharable Link
  To revoke access for users, switch the dropdown option to **Only people with access**.

  If you have also [shared the deployment](/docs/deployments/sharing-deployments) with individual users, you will need to remove them fro

Managing Shareable Links

You can view and manage all the existing Shareable Links for your team in the following way

1. From your [dashboard](/dashboard) go to the **Settings** tab
2. Select Deployment Protection and then choose the **Access** tab
3. Click the **All Access** button and select **Shareable Links**

---
title: "Methods to Protect Deployments"
description: "Learn about the different methods to protect your deployments on Vercel, including Vercel Authentication, Password Protecti
last_updated: "2026-01-16T02:19:28.769Z"
source: "https://vercel.com/docs/deployment-protection/methods-to-protect-deployments"

Methods to Protect Deployments

Vercel offers three methods for protecting your deployments. Depending on your use case, you can choose to protect a single environment,

You can see an overview of your projects' protections in the following way

1. From your [dashboard](/dashboard) go to the **Settings** tab
2. Select **Deployment Protection**

Vercel Authentication

With Vercel Authentication you can restrict access to all deployments (including non-public deployments), meaning only those with a Verce

When a Vercel user visits your protected deployment, but they do not have permission to access it, they have the option to [request acces

Learn more about [Vercel Authentication](/docs/security/deployment-protection/methods-to-protect-deployments/vercel-authentication) and h

Password Protection

Password Protection on Vercel lets you restrict access to both non-public, and public deployments depending on the type of [environment p

Learn more about [Password Protection](/docs/security/deployment-protection/methods-to-protect-deployments/password-protection) and how t

Trusted IPs

Trusted IPs restrict deployment access to specified IPv4 addresses and [CIDR ranges](https://www.ipaddressguide.com/cidr "What is are CID

Learn more about [Trusted IPs](/docs/security/deployment-protection/methods-to-protect-deployments/trusted-ips) and how to enable it.

More resources

- [Understanding Deployment Protection by environment](/docs/deployment-protection#understanding-deployment-protection-by-environment)
- [Methods to bypass deployment protection](/docs/deployment-protection/methods-to-bypass-deployment-protection)

---
title: "Password Protection"
description: "Learn how to protect your deployments with a password."
last_updated: "2026-01-16T02:19:28.784Z"
source: "https://vercel.com/docs/deployment-protection/methods-to-protect-deployments/password-protection"

Password Protection

With [Password Protection](#managing-password-protection) enabled, visitors to your deployment must enter the pre-defined password to gai
You can set the desired password from your project settings when enabling the feature, and update it any time

Password Protection security considerations

The table below outlines key considerations and security implications when using Password Protection for your deployments on Vercel.

Consideration	Description
Environment Configuration	Can be enabled for different environments. See [Understanding Deployment Protection by environment](/do
Compatibility	Compatible with [Vercel Authentication](/docs/security/deployment-protection/methods-to-protect-deploym
Bypass Methods	Can be bypassed using [Shareable Links](/docs/security/deployment-protection/methods-to-bypass-deployme
Password Persistence	Users only need to enter the password once per deployment, or when the password changes, due to cookie
Password Changes	Users must re-enter a new password if you change the existing one
Disabling Protection	All existing deployments become unprotected if you disable the feature
Token Scope	JWT tokens set as cookies are valid only for the URL they were set for and can't be reused for differen

Managing Password Protection

You can manage Password Protection through the dashboard, API, or Terraform:

- ### Go to Project Deployment Protection Settings
  From your Vercel [dashboard](/dashboard):
  1. Select the project that you wish to enable Password Protection for
  2. Go to **Settings** then **Deployment Protection**

- ### Manage Password Protection

  From the **Password Protection** section:
  1. Use the toggle to enable the feature
  2. Select the [deployment environment](/docs/security/deployment-protection#understanding-deployment-protection-by-environment) you wan
  3. **Enter a password** of your choice
  4. Finally, select **Save**
  All your existing and future deployments will be protected with a password for the project. Next time when you access a deployment, you

Manage using the API

You can manage Password Protection using the Vercel API endpoint to [update an existing project](/docs/rest-api/reference/endpoints/proje

- `deploymentType`
  - `prod_deployment_urls_and_all_previews`: Standard Protection
  - `all`: All Deployments
  - `preview`: Only Preview Deployments
- `password`: Password

```json
// enable / update password protection
{
  "passwordProtection": {
    "deploymentType": "prod_deployment_urls_and_all_previews" | "all" | "preview",
    "password": "<password>"
  },
}

// disable password protection
{
  "passwordProtection": null
}
```

Manage using Terraform

You can configure Password Protection using `password_protection` in the `vercel_project` data source in the [Vercel Terraform Provider](

---
title: "Trusted IPs"
description: "Learn how to restrict access to your deployments to a list of trusted IP addresses."
last_updated: "2026-01-16T02:19:28.797Z"
source: "https://vercel.com/docs/deployment-protection/methods-to-protect-deployments/trusted-ips"

Trusted IPs

With Trusted IPs [enabled](/docs/security/deployment-protection/methods-to-protect-deployments/trusted-ips#managing-trusted-ips) at the l

Trusted IPs is suitable for customers who access Vercel deployments through a specific IP address. For example, limiting preview deployme

Trusted IPs security considerations

The table below outlines key considerations and security implications when using Trusted IPs for your deployments on Vercel.

Consideration	Description
General Considerations	
Environment Configuration	Can be enabled for different environments. See [Understanding Deployment Protection by environment]((
Compatibility	Operates as a required layer on top of [Vercel Authentication](/docs/security/deployment-protection/
Bypass Methods	Can be bypassed using [Shareable Links](/docs/security/deployment-protection/methods-to-bypass-deplo
IP Address Support	Supports IPv4 addresses and IPv4 CIDR ranges
Prerequisites	
Preview Environment Requirements	Can only be enabled in preview when [Vercel Authentication](/docs/security/deployment-protection/met
External Proxy Configuration	Requires [rulesets](/kb/guide/can-i-use-a-proxy-on-top-of-my-vercel-deployment) configuration to avo
Security Considerations	
Firewall Precedence	[Vercel Firewall](/docs/vercel-firewall) takes precedence over Trusted IPs
IP Blocking	IPs or CIDRs listed in [IP Blocking](/docs/security/vercel-waf/ip-blocking) will be blocked even if
DDoS Mitigation	Trusted IPs do not bypass [DDoS Mitigation](/docs/security/ddos-mitigation) unless configured
Deployment Impact	Changing the Trusted IPs list affects all deployments
Disabling Trusted IPs	Disabling makes all existing deployments accessible from any IP

Managing Trusted IPs

You can manage Trusted IPs through the dashboard, API, or Terraform:

Manage using the dashboard

- ### Go to Project Deployment Protection Settings
  From your Vercel [dashboard](/dashboard):
  1. Select the project that you wish to enable Trusted IPs for
  2. Go to **Settings** then **Deployment Protection**

- ### Manage Vercel Authentication
  Ensure Vercel Authentication is enabled. See [Managing Vercel Authentication](/docs/security/deployment-protection/methods-to-protect-d

- ### Manage Trusted IPs
  From the **Trusted IPs** section:
  1. Use the toggle to enable the feature
  2. Select the [deployment environment](/docs/security/deployment-protection#understanding-deployment-protection-by-environment) you wan
  3. Enter your list of IPv4 addresses and IPv4 CIDR ranges with an optional note describing the address
  4. Finally, select **Save**
  All your existing and future deployments will be protected with Trusted IPs for that project. Visitors to your project deployments from

Manage using the API

You can manage Trusted IPs using the Vercel API endpoint to [update an existing project](/docs/rest-api/reference/endpoints/projects/upda

- `deploymentType`
  - `prod_deployment_urls_and_all_previews`: Standard Protection
  - `all`: All Deployments

  - `preview`: Only Preview Deployments
  - `production`: Only Production Deployments
- `addresses`: Array of addresses
  - `value`: The IPv4, or IPv4 CIDR address
  - `note`: Optional note about the address
  - `protectionMode`
    - `additional`: IP is required along with other enabled protection methods (recommended setting)
    - `additional`: IP is required along with other enabled protection methods

```json
// enable / update trusted ips
{
  "trustedIps": {
      "deploymentType": "all" | "preview" | "production" | "prod_deployment_urls_and_all_previews",
      "addresses": { "value": "<value>"; "note": "<note>" | undefined }[],
      "protectionMode": "additional"
  }
}
// disbale trusted ips
{
  "trustedIps": null
}
```

Manage using Terraform

You can configure Trusted IPs using `trusted_ips` in the `vercel_project` data source in the [Vercel Terraform Provider](https://registry

---
title: "Vercel Authentication"
description: "Learn how to use Vercel Authentication to restrict access to your deployments."
last_updated: "2026-01-16T02:19:28.820Z"
source: "https://vercel.com/docs/deployment-protection/methods-to-protect-deployments/vercel-authentication"

Vercel Authentication

Vercel Authentication lets you restrict access to your public and non-public deployments. It is the **recommended** approach to protectin

Users attempting to access the deployment will encounter a Vercel login redirect. If already logged into Vercel, Vercel will authenticate

After login, users are redirected and a cookie is set in the browser if they have view access. If the user does not have access to view t

Who can access protected deployments?

- Logged in [team members](/docs/rbac/access-roles#team-level-roles) with at least a viewer role ([Viewer Pro](/docs/rbac/access-roles#vi
- Logged in [project members](/docs/rbac/access-roles#project-level-roles) with at least the [project Viewer](/docs/rbac/access-roles#pro
- Logged in members of an [access group](/docs/rbac/access-groups) that has access to the project the deployment belongs to
- Logged in Vercel users who have been [granted access](#access-requests)
- Anyone who has been given a [Shareable Link](/docs/security/deployment-protection/methods-to-bypass-deployment-protection/sharable-link
- Tools using the [protection bypass for automation](/docs/security/deployment-protection/methods-to-bypass-deployment-protection/protect

Access requests

When a Vercel user visits your protected deployment, but they do not have permission to access it, they have the option to request access
This request triggers an email and Vercel notification to the branch authors.

The access request can be approved or declined. Additionally, granted access can be revoked for a user at any time.

Users granted access can view the latest deployment from a specific branch when logged in with their Vercel account.
They can also leave preview [Comments](/docs/comments) if these are enabled on your team.

Those on the Hobby plan can only have one external user per account. If you need more, you can upgrade to a [Pro plan](/docs/plans/pro-pl

You can manage access requests in the following way

1. From your [dashboard](/dashboard) go to the **Settings** tab
2. Select Deployment Protection and then choose the **Requests** tab to see pending requests
3. Choose **Access** to manage existing access

Access requests can also be managed using the share modal on the deployment page

Vercel Authentication security considerations

You can configure Vercel Authentication for different environments, as outlined in [Understanding Deployment Protection by environment](/

Disabling Vercel Authentication renders all existing deployments unprotected. However, re-enabling it allows previously authenticated use

Consideration	Description
Environment Configuration	Can be enabled for different environments. See [Understanding Deployment Protection by environment](/do
Compatibility	Compatible with [Password Protection](/docs/security/deployment-protection/methods-to-protect-deploymen
Bypass Methods	Can be bypassed using [Shareable Links](/docs/security/deployment-protection/methods-to-bypass-deployme
Disabling	All existing deployments become unprotected when Vercel Authentication is disabled
Re-enabling	Users who have logged in previously will still have access without re-authenticating
Token Scope	Tokens are valid for a single URL and are not reusable across different URLs

Managing Vercel Authentication

Admins and members can enable or disable Vercel Authentication for their team. Hobby teams can also enable or disable for their own proje

You can manage Vercel Authentication through the dashboard, API, or Terraform:

Manage using the dashboard

- ### Go to Project Deployment Protection Settings
  From your Vercel [dashboard](/dashboard):
  1. **Select the project** that you wish to enable Password Protection for
  2. Go to **Settings** then **Deployment Protection**

- ### Manage Vercel Authentication
  From the **Vercel Authentication** section:
  1. Use the toggle to enable the feature
  2. Select the [deployment environment](/docs/deployments/environments) you want to protect
  3. Finally, Select **Save**
  All your existing and future deployments will be protected with Vercel Authentication for the project. Next time when you access a depl

Manage using the API

You can manage Vercel Authentication using the Vercel API endpoint to [update an existing project](/docs/rest-api/reference/endpoints/pro

- `prod_deployment_urls_and_all_previews`: Standard Protection
- `all`: All Deployments
- `preview`: Only Preview Deployments

```json
// enable / update Vercel Authentication
{
  "ssoProtection": {
    "deploymentType": "prod_deployment_urls_and_all_previews" | "all" | "preview"
  }
}

// disable Vercel Authentication
{
  "ssoProtection": null
}
```

Manage using Terraform

You can configure Vercel Authentication using `vercel_authentication` in the `vercel_project` data source in the [Vercel Terraform Provid

---
title: "Deployment Protection on Vercel"
description: "Learn how to secure your Vercel project"
last_updated: "2026-01-16T02:19:28.856Z"
source: "https://vercel.com/docs/deployment-protection"

Deployment Protection on Vercel

Deployment Protection safeguards both your preview and production URLs across various environments. Configured at the project level throu

Vercel offers the following Deployment Protection features:

- [**Vercel Authentication**](/docs/security/deployment-protection/methods-to-protect-deployments/vercel-authentication): Restricts acces
- [**Password Protection**](/docs/security/deployment-protection/methods-to-protect-deployments/password-protection): Restricts access to
- [**Trusted IPs**](/docs/security/deployment-protection/methods-to-protect-deployments/trusted-ips): Restricts access to your deployment

> **💡 Note:** Deployment protection requires authentication for all requests, including
> those to Middleware.

Configuring Deployment Protection

Deployment Protection is managed through your project settings. To configure Deployment Protection:

1. From the [dashboard](/dashboard), select the project you wish to set Deployment Protection on
2. Once selected, navigate to the **Settings** tab
3. From the list, select the **Deployment Protection** tab

Team default settings

You can configure the default Deployment Protection for new projects in your team's Deployment Protection settings. This default can be o

When setting a team default, you can choose the protection level (All Deployments, Standard Protection, or None) and the protection metho

Understanding Deployment Protection by environment

You can configure the type of Deployment Protection for each environment in your project depending on your projects security needs. When

- [**Standard Protection**](#standard-protection): This option protects all domains except [Production Custom Domains](/docs/domains/work
- [**All Deployments**](#all-deployments): Protects all URLs. Protecting all deployments is **available on Pro and Enterprise plans**
- [**(Legacy) Standard Protection**](#legacy-standard-protection): This option protects all preview URLs and [deployment URLs](/docs/depl
- [**(Legacy) Only Preview Deployments**](#legacy-only-preview-deployments): This option protects only preview URLs. This does not protec

To protect [**only production URLs**](#only-production-deployments), you can use [Trusted IPs](/docs/security/deployment-protection/metho

Standard Protection

Standard Protection is the recommended way to secure your deployments, as it protects all domains except [Production Custom Domains](

Standard Protection can be configured with the following Deployment Protection features:

- [Vercel Authentication](/docs/security/deployment-protection/methods-to-protect-deployments/vercel-authentication)
- [Password Protection](/docs/security/deployment-protection/methods-to-protect-deployments/password-protection)
- [Trusted IPs](/docs/security/deployment-protection/methods-to-protect-deployments/trusted-ips)

Migrating to Standard Protection

Enabling Standard Protection restricts public access to the production [generated deployment URL](/docs/deployments/generated-urls). This

If you are using `VERCEL_URL` or `VERCEL_BRANCH_URL` to make fetch requests, you will need to update your requests to target the same dom

> **💡 Note:** The Framework Environment Variable `VERCEL_URL` is prefixed with the name of
> the framework. For example, `VERCEL_URL` for Next.js is
> `NEXT_PUBLIC_VERCEL_URL`, and `VERCEL_URL` for Nuxt is `NUXT_ENV_VERCEL_URL`.
> See the [Framework Environment

> Variables](/docs/environment-variables/framework-environment-variables)
> documentation for more information.

For client-side requests, use relative paths in the fetch call to target the current domain, automatically including the user's authentic

```ts
// Before
fetch(`${process.env.VERCEL_URL}/some/path`);

// After
fetch('/some/path');
// Note: For operations requiring fully qualified URLs, such as generating OG images,
// replace '/some/path' with the actual domain (e.g. 'https://yourdomain.com/some/path').
```

For server-side requests, use the origin from the incoming request and manually add request cookies to pass the user's authentication coo

```ts
const headers = { cookie: <incoming request header cookies> };
fetch('<incoming request origin>/some/path', { headers });
```

Bypassing protection using [Protection Bypass for Automation](/docs/security/deployment-protection/methods-to-bypass-deployment-protectio

All deployments

Selecting **All Deployments** secures all deployments, both preview and production, restricting public access entirely.

With this configuration, all URLs, including your production domain `example.com` and [generated URLs](/docs/deployments/generated-urls)

Protecting all deployments can be configured with the following Deployment Protection features:

- [Vercel Authentication](/docs/security/deployment-protection/methods-to-protect-deployments/vercel-authentication)
- [Password Protection](/docs/security/deployment-protection/methods-to-protect-deployments/password-protection)
- [Trusted IPs](/docs/security/deployment-protection/methods-to-protect-deployments/trusted-ips)

Only production deployments

Restrict access to protected deployments to a list of [Trusted IPs](/docs/deployment-protection/methods-to-protect-deployments/trusted-ip

Preview deployment URLs remain publicly accessible. This feature is **only available on the Enterprise plan**.

(Legacy) Standard Protection

(Legacy) Standard Protection is a legacy feature that protects all preview URLs and [deployment URLs](/docs/deployments/generated-url

(Legacy) Only Preview Deployments

Selecting **(Legacy) Only Preview Deployments** protects preview URLs, while the production environment remains publicly accessible.

For example, Vercel generates a preview URL such as `my-preview-5678.vercel.app`, which will be protected. In contrast, all production UR

Advanced Deployment Protection

Advanced Deployment Protection features are available to Enterprise customers by default. Customers on the Pro plan can access these feat

- [Password Protection](/docs/security/deployment-protection/methods-to-protect-deployments/password-protection)
- [Private Production Deployments](/docs/security/deployment-protection#configuring-deployment-protection)
- [Deployment Protection Exceptions](/docs/security/deployment-protection/methods-to-bypass-deployment-protection/deployment-protection-e

Enabling Advanced Deployment Protection

To opt-into Advanced Deployment Protection while on a Pro plan:

1. Navigate to your **Project Settings** and select the **Deployment Protection** tab
2. Then choose one of the above protection features
3. You will then be prompted to upgrade to the Advanced Deployment Protection add-on through an **Enable and Pay** button before you can

When you enable Advanced Deployment Protection, you will be charged $150 per month for the add-on, and will have access to *all* Advanced

Disabling Advanced Deployment Protection

To opt out of Advanced Deployment Protection:

1. Navigate to your **Team Settings**, then the **Billing** page
2. Press **Edit** on the feature you want to disable and follow the instructions to disable the add-on

In order to disable Advanced Deployment Protection, you must have been using the feature **for a minimum of thirty days**. After this tim

More resources

- [Methods to protect deployments](/docs/deployment-protection/methods-to-protect-deployments)
- [Methods to bypass deployment protection](/docs/deployment-protection/methods-to-bypass-deployment-protection)

---
title: "Deployment Retention"
description: "Learn how Deployment Retention policies affect a deployment"
last_updated: "2026-01-16T02:19:28.864Z"
source: "https://vercel.com/docs/deployment-retention"

Deployment Retention

Deployment retention refers to the configured policies that determine how long different types of deployments are kept before they are au

These configured retention policies allow you to control how long your deployment data is stored, providing:

- **Enhanced protection:** Remove outdated deployments with potential vulnerabilities or sensitive data
- **Compliance support:** Configure retention policies to align with your compliance requirements.

- **Efficient storage management:** Automatically clear out unnecessary deployments

Vercel provides unlimited deployment retention for all deployments, regardless of the plan that you are on.

You can configure retention durations for the following deployment states:

- Canceled deployments
- Errored deployments
- Preview deployments
- Production deployments

For example, imagine you created a production deployment with a 60-day retention period on 01/01/2024 and later replaced it with a newer

Once a policy is enabled on a project, deployments within the retention period will start to be automatically marked for deletion, within

Setting a deployment retention policy

To configure a retention policy, navigate to the **Settings** tab of your project and follow these steps:

1. Select **Security** on the side panel of the project settings page
2. Scroll down to the **Deployment Retention Policy** section
3. Select the drop down menu with the appropriate duration
4. Save the new retention policy for your project

Viewing deployment retention policy

You can view your deployments retention policy using [Vercel CLI](/docs/cli/list) and running the following command from your terminal:

```bash filename="terminal"
vercel list [project-name] [--policy errored=6m]
```

Exceptions to the retention policy

Deployments older than the configured retention interval are not always deleted. Deployments will be kept while any of the following is t

- The deployment is one of the last 10 deployments created in the project.
- The deployment is one of the last 20 production deployments in state Ready.
- The deployment is one of the last 20 non-production deployments in state Ready.
- The deployment has a production alias assigned to it.
- The deployment is the target of a [branch alias](/docs/domains/working-with-domains/assign-domain-to-a-git-branch) for a [custom enviro
- The deployment is a non-production deployment and has any custom alias assigned to it.

Restoring a deleted deployment

When a deployment is marked for deletion either accidentally or as part of the retention policy, you can restore it within the recovery p

To restore a deleted deployment, navigate to the **Settings** tab of your project:

1. Select **Security** on the side panel of the project settings page
2. Scroll down to the **Recently Deleted** section
3. Find the deployment that needs to be restored, and click on the dropdown menu item **Restore**
4. Complete the modal

More resources

- [View Deployment Retention Policies](/docs/cli/list#unique-options)
- [Troubleshoot Deployment Retention Errors](/docs/errors/DEPLOYMENT_DELETED)

---
title: "Claim Deployments"
description: "Learn how to take ownership of deployments on Vercel with the Claim Deployments feature."
last_updated: "2026-01-16T02:19:28.879Z"
source: "https://vercel.com/docs/deployments/claim-deployments"

Claim Deployments

The Claim Deployments feature enables users to take control of deployments by transferring them to their Vercel accounts. Users can gener

However, when transferring a project between two teams owned by the same user, it is recommended to use the [Project Transfer flow](/docs

Get started

- [Claim deployments template](https://github.com/vercel/claim-deployments-demo)
- [Demo](https://claim-deployments-demo.vercel.app)
- [Demo with resource claims](https://claim-deployments-demo-with-resource.vercel.app/)

Associated resources

When a user claims a deployment, Vercel also transfers any associated resources (limited to specific providers) to the new owner's accoun

The resource provider that currently supports resource transfer is [Prisma](https://vercel.com/marketplace/prisma).

For more details on the transfer process, see [Resources with Claim Deployments flows](/docs/integrations/create-integration/marketplace-

Important endpoints

- **Claim Deployments URL:** `https://vercel.com/claim-deployment?[...]`

- **Initiate a project transfer request:** [POST /projects/:idOrName/transfer-request](/docs/rest-api/reference/endpoints/projects/create

- **Complete a project transfer:** [PUT /projects/transfer-request/:code](/docs/rest-api/reference/endpoints/projects/accept-project-tran
  - *This endpoint is not needed if you are using the Claim Deployments URL*

Example use case: automated AI-generated deployment

1. **File upload:** The AI agent uploads the deployment files using the Vercel API: [POST /files](/docs/rest-api/reference/endpoints/depl

2. **Deployment creation:**
   - Create a new deployment using the [Vercel CLI](/docs/cli/deploying-from-cli)
   - Or create a deployment with the Vercel API: [POST /files](/docs/rest-api/reference/endpoints/deployments/upload-deployment-files) fo

3. **Project transfer request:**
   - The agent initiates a transfer request with: [POST /projects/:idOrName/transfer-request](/docs/rest-api/reference/endpoints/projects
   - This returns a `code` (valid for 24 hours) that allows the agent to transfer the project to another team, typically the end user who

4. **Generate claim URL:**
   - The agent generates a claim URL and shares it with the user:
     `https://vercel.com/claim-deployment?code=xxx&returnUrl=https://xxx`

5. **User claims the deployment:**
   - The user accesses the claim page using the URL and selects a team within their Vercel account to transfer the deployment.

6. **Project transfer completion:**
   - After the user clicks **Transfer**, the Vercel API ([PUT /projects/transfer-request/:code](/docs/rest-api/reference/endpoints/projec

Get started with [this template](https://github.com/vercel/claim-deployments-demo) of claiming deployments ([demo](https://claim-deployme

Team restructuring

When reorganizing teams, you can easily transfer ownership of projects to another team using the Claim Deployments feature.

Migrating personal projects to a company account

Freelancers or employees can move deployments from their personal accounts to a company's Vercel account by generating and sharing a clai

---
title: "Environments"
description: "Environments are for developing locally, testing changes in a pre-production environment, and serving end-users in producti
last_updated: "2026-01-16T02:19:28.901Z"
source: "https://vercel.com/docs/deployments/environments"

Environments

Vercel provides three default environments—**Local**, **Preview**, and **Production**:

1. **Local Development**: developing and testing code changes on your local machine
2. **Preview**: deploying for further testing, QA, or collaboration without impacting your live site
3. **Production**: deploying the final changes to your user-facing site with the production domain

Pro and Enterprise teams can create **Custom Environments** for more specialized workflows (e.g., `staging`, `QA`). Every environment can

Local Development Environment

This environment is where you develop new features and fix bugs on your local machine. When building with [frameworks](/docs/frameworks),

1. **Install the Vercel CLI**:

```bash filename="Terminal" package-manager="npm"
npm i -g vercel
```

```bash filename="Terminal" package-manager="bun"
bun i -g vercel
```

```bash filename="Terminal" package-manager="yarn"
yarn global add vercel
```

```bash filename="Terminal" package-manager="pnpm"
pnpm i -g vercel
```

2. **Link your Vercel project** with your local directory:

   ```bash
   vercel link
   ```

3. **Pull environment variables locally** for use with application development:

   ```bash
   vercel env pull
   ```

This will populate the `.env.local` file in your application directory.

Preview Environment (Pre-production)

Preview environments allow you to deploy and test changes in a live setting, without affecting your production site. By default, Verc

- Push a commit to a branch that is **not** your production branch (commonly `main`)
- Create a pull request (PR) on [GitHub, GitLab, or Bitbucket](/docs/git)
- Deploy using the CLI without the `-prod` flag, for example just `vercel`

Each deployment gets an automatically generated URL, and you'll typically see links appear in your Git provider's PR comments or in the V

There are two types of preview URLs:

- **Branch-specific URL** – Always points to the latest changes on that branch
- **Commit-specific URL** – Points to the exact deployment of that commit

Learn more about [generated URLs](/docs/deployments/generated-urls).

Production Environment

The **Production** environment is the live, user-facing version of your site or application.

By default, pushing or merging changes into your production branch (commonly `main`) triggers a production deployment. You can also expli

```bash
vercel --prod
```

When a production deployment succeeds, Vercel updates your production domains to point to the new deployment, ensuring your users see the

Custom Environments

Custom environments are useful for longer-running pre-production environments like `staging`, `QA`, or any other specialized workflow you

Team owners and project admins can create, update, or remove custom environments.

Creating a custom environment

\['Dashboard'

1. Go to your **Project Settings** in the Vercel Dashboard
2. Under **Environments**, click **Create Environment**
3. Provide a name (e.g., `staging`), and optionally:
   - **Branch Tracking** to automatically deploy whenever a matching branch is pushed
   - **Attach a Domain** to give a persistent URL to your environment
   - **Import variables** from another environment to seed this environment with existing environment variables

'cURL'

To create an Authorization Bearer token, see the [access token](/docs/rest-api/reference/welcome#creating-an-access-token) section of the

```bash filename="cURL"
curl --request POST \
  --url https://api.vercel.com/v9/projects/<project-id-or-name>/custom-environments \
  --header "Authorization: Bearer $VERCEL_TOKEN" \
  --header "Content-Type: application/json" \
  --data '{
    "slug": "<environment_name_slug>",
    "description": "<environment_description>",
  }'
```

'SDK']

To create an Authorization Bearer token, see the [access token](/docs/rest-api/reference/welcome#creating-an-access-token) section of the

```ts filename="createCustomEnvironment"
import { Vercel } from '@vercel/sdk';

const vercel = new Vercel({
  bearerToken: '<YOUR_BEARER_TOKEN_HERE>',
});

async function run() {
  const result = await vercel.environment.createCustomEnvironment({
    idOrName: '<project-id-or-name>',
    requestBody: {
      slug: '<environment_name_slug>',
      description: '<environment_description>',
    },
  });
  // Handle the result
  console.log(result);
}

run();
```

Using custom environments via the CLI

You can deploy, pull, and manage environment variables to your custom environment with the CLI:

```bash
# Deploy to a custom environment named "staging":
vercel deploy --target=staging

# Pull environment variables from "staging":
vercel pull --environment=staging

# Add environment variables to "staging":
vercel env add MY_KEY staging
```

Pricing and limits

Custom environments are available at no additional cost on the Pro and Enterprise plans. The number of custom environments you can create

- **Pro**: 1 custom environment per project
- **Enterprise**: 12 custom environments per project

More resources

- [Learn about the different environments on Vercel](https://www.youtube.com/watch?v=nZrAgov_-D8)

---
title: "Accessing Deployments through Generated URLs"
description: "When you create a new deployment, Vercel will automatically generate a unique URL which you can use to access that particul
last_updated: "2026-01-16T02:19:28.915Z"

source: "https://vercel.com/docs/deployments/generated-urls"

Accessing Deployments through Generated URLs

When you create a new [deployment](/docs/deployments) in either a preview or production [environment](/docs/deployments/environments), Ve

This URL is **publicly accessible by default**, but you can configure it to be private using [deployment protection](/docs/security/deplo

The make up of the URL depends on how it was created and if it relates to a branch of a specific commit. To learn more, see [URL Componen

Viewing generated URLs

You can access these automatically generated URLs in the following ways:

- On the command line when the build has completed.
- When using Git, you can access either a URL for the branch or for each commit. To learn more, see [Generated from Git](#generated-from-
- Under the Project's Overview and Deployments tabs, as shown below:

URL Components

Generated URLs are comprised of several different pieces of data associated with the underlying deployment. Varying combinations of the f

Value	Description
`<project-name>`	The name of the [Project](/docs/projects/overview) that contains the deployment
`<unique-hash>`	9 randomly generated numbers and letters
`<scope-slug>`	The slug (not the name) of the account or [team](/docs/accounts/create-a-team) that contains the project/deployment
`<branch-name>`	The name of the Git branch for which the deployment was created

Generated from Git

When are working with Git, Vercel will automatically generate a URL for the following:

- **The commit**: This URL will always show you a preview of changes from that specific commit. This is useful for sharing a specific ver
  ```bash filename="url-structure"
  <project-name>-<unique-hash>-<scope-slug>.vercel.app
  ```
- **The branch**: The URL generated from a Git branch will always show you the most recent changes for the branch and won't change if you
  ```bash filename="url-structure"
  <project-name>-git-<branch-name>-<scope-slug>.vercel.app
  ```

To access the commit URL, click the **View deployment** button from your pull request. To access the branch URL, click the **Visit Previe

Generated with Vercel CLI

To access the URL for a successful deployment from Vercel CLI, you can save the [standard output of the deploy command](/docs/cli/deploy#

```bash filename="url-structure"
<project-name>-<scope-slug>.vercel.app;
```

> **💡 Note:** Once you deploy to the production environment, the above URL will point to the
> production deployment.

If the deployment is created on a [Team](/docs/accounts/create-a-team), you can also use the URL specific to the deployment's author. It

```bash filename="url-structure"
<project-name>-<author-name>-<scope-slug>.vercel.app;
```

This allows you to stay on top of the latest change deployed by a particular [member](/docs/accounts/team-members-and-roles) of a team wi

Truncation

If more than 63 characters are present before the `.vercel.app` suffix (or the respective [Preview Deployment Suffix](#preview-deployment

Anti-phishing protection

If your `<project-name>` resembles a regular web domain, it may be shortened to avoid that resemblance. For example, `www-company-com` wo

Preview Deployment Suffix

Preview Deployment Suffixes allow you to customize the URL of a [preview deployment](/docs/deployments/environments#preview-environment-p

To learn more, see the [Preview Deployment Suffix](/docs/deployments/preview-deployment-suffix) documentation.

---
title: "Accessing Build Logs"
description: "Learn how to use Vercel"
last_updated: "2026-01-16T02:19:28.921Z"
source: "https://vercel.com/docs/deployments/logs"

Accessing Build Logs

When you deploy your website to Vercel, the platform generates build logs that show the deployment progress. The build logs contain infor

- The version of the build tools
- Warnings or errors encountered during the build process
- Details about the files and dependencies that were installed, compiled, or built during the deployment

Build logs are particularly useful for debugging issues that may arise during deployment. If a deployment fails, these can help you ident

To access build logs, click the **Build Logs** button from the production deployment tile in the projects overview page.

How build logs work?

Build logs are generated at build time for all [Deployments](/docs/deployments). The logs are similar to your framework's [Build Command]

In addition to the list of build actions, you can also find errors or warnings. These are highlighted with different colors, such as yell

> **💡 Note:** Build logs will automatically be truncated, if the total size reaches over
> 4MB.

Link to build logs

If you click on the timestamp to the left of the log entry, you get a link to that log entry. This will highlight the selected log and ap

You can select multiple lines by holding the `Shift` key and clicking the timestamps. This will create a link for the content between the

The log link is only accessible to [team members](/docs/rbac/managing-team-members). Anyone who is not a member or has a valid Vercel acc

> **💡 Note:** The link to build logs feature works for logs that are up to 2000 lines long.

Save logs

You can use [Drains](/docs/drains) to export, store, and analyze your build logs. Log Drains configuration can be accessed through the Ve

---
title: "Managing Deployments"
description: "Learn how to manage your current and previously deployed projects to Vercel through the dashboard. You can redeploy at any
last_updated: "2026-01-16T02:19:28.933Z"
source: "https://vercel.com/docs/deployments/managing-deployments"

Managing Deployments

You can manage all current and previous deployments regardless of environment, status, or branch from the [dashboard](/dashboard). To man

1. Ensure your team is selected from the scope selector
2. Select your project
3. From the top navigation, select the **Deployments** tab
4. You can then filter, redeploy, or manually promote your deployment to production

[Vercel CLI](https://vercel.com/cli) and [Vercel REST API](/docs/rest-api) also provide alternative ways to manage your deployments. You

Filter deployment

You can filter your deployments based on branch, status, and deployment environment:

1. Select your project from the [dashboard](/dashboard)
2. From the top navigation, select the **Deployments** tab
3. Use the dropdowns to search by **Branch**, **Date Range**, **All Environments**, or **Status**

Delete a deployment

\['Dashboard'

If you no longer need a specific deployment of your app, you can delete it from your project with the following steps:

1. From your [dashboard](/dashboard), select the project where the specific deployment is located.
2. Click on the **Deployments** tab.
3. From the list of deployments, click on the deployment that you want to delete
4. Click the ... button.
5. From the context menu, select **Delete**.

'cURL'

To create an Authorization Bearer token, see the [access token](/docs/rest-api/reference/welcome#creating-an-access-token) section of the

```bash filename="cURL"
curl --request DELETE \
  --url https://api.vercel.com/v13/deployments/<deployment-id> \
  --header "Authorization: Bearer $VERCEL_TOKEN"
```

'SDK']

To create an Authorization Bearer token, see the [access token](/docs/rest-api/reference/welcome#creating-an-access-token) section of the

```ts filename="deleteDeployment"
import { Vercel } from '@vercel/sdk';

const vercel = new Vercel({
  bearerToken: '<YOUR_BEARER_TOKEN_HERE>',
});

async function run() {
  const result = await vercel.deployments.deleteDeployment({
    id: 'deployment-id',
  });

  // Handle the result
  console.log(result);
}

run();
```

Deleting a deployment prevents you from using instant rollback on it and might break the links used in integrations, such as the ones in

You can also set a [deployment retention policy](#set-the-deployment-retention-policy) to automatically delete deployments after a certai

Set the deployment retention policy

You can set the retention policy for your deployments to automatically delete them after a certain period. To learn more, see [Deployment

Deployment protection

Vercel provides a way to protect your deployments from being accessed by unauthorized users. You can use [Vercel Authentication](/docs/se

In addition, Enterprise teams can use [Trusted IPs](/docs/security/deployment-protection/methods-to-protect-deployments/trusted-ips) and

To learn more, see [Deployment Protection](/docs/security/deployment-protection).

Redeploy a project

Vercel automatically redeploys your application when you make any commits. However, there can be situations such as bad cached data where

1. Ensure your team is selected from the scope selector
2. Select your project
3. From the top navigation, select the **Deployments** tab
4. Locate the deployment you wish to deploy. You may need to use the [filter](/docs/deployments/managing-deployments#filter-deployment) o
5. Click the ellipsis icon () and select **Redeploy**
6. In the **Redeploy to Production** window, decide if you want to use the existing [Build Cache](/docs/deployments/troubleshoot-a-build#

When to Redeploy

Other than your custom needs to redeploy, it's always recommended to redeploy your application to Vercel for the following use cases:

- Enabling the [Analytics](/docs/analytics/quickstart)
- Changing the [Environment Variables](/docs/environment-variables)
- [Outage Resiliency](/docs/regions#outage-resiliency)
- Making changes to **Build & Development Settings**
- **Redirect** or **Rewrites** from a subdomain to a subpath

---
title: "Inspecting your Open Graph metadata"
description: "Learn how to inspect and validate your Open Graph metadata through the Open Graph deployment tab."
last_updated: "2026-01-16T02:19:28.941Z"
source: "https://vercel.com/docs/deployments/og-preview"

Inspecting your Open Graph metadata

You can use the **Open Graph** tab on every deployment on Vercel to validate and view your [Open Graph (OG)](https://ogp.me/ "Open Graph

To view your data:

1. Choose your account or team from the [scope selector](/docs/dashboard-features#scope-selector)
2. Select your project and go to the **Deployments** tab
3. From the **Deployments** tab, select the deployment you wish to view the metadata for
4. Select the Open Graph tab:
5. From here, you can view the metadata and a preview for [Twitter](/docs/deployments/og-preview#twitter-specific-metadata), Slack, Faceb

Filter by pathname

You can use the **Path** dropdown to view the OG card for any page on that particular deployment.

Metadata

These properties set by the [Open Graph protocol](https://ogp.me/#metadata).

Property	Value	Description
`title`	The title tag used to name the page. 70 characters max.	This is used by default if no other values
`description`	The meta.description tag used to describe the page. 200 characters max.	This is used by default if no other values
`og:image`	Absolute URL to image.	Use the [OG Image Generation](/docs/og-ima
`og:title`	A title for link previews.	You can use this to override the meta titl
`og:description`	A one to two sentence description for link previews.	You can use this to override the meta desc
`og:url`	A canonical URL for link previews.	You should provide the absolute URL.

```html filename="index.js"
<div>
  <head>
    <meta name="og:title" content="Vercel CDN" />
    <meta name="og:description" content="Vercel CDN" />
    <meta name="og:image" content={ // Because OG images must have a absolute
    URL, we use the // `VERCEL_URL` environment variable to get the deployment’s
    URL. // More info: // https://vercel.com/docs/environment-variables
    `${ process.env.VERCEL_URL ? 'https://' + process.env.VERCEL_URL : ''
    }/api/vercel` } />
    <meta
      name="og:url"
      content="https://vercel.com/docs/cdn"
    />
  </head>
  <h1>A page with Open Graph Image.</h1>
</div>
```

Twitter-specific metadata

Property	`content` value	Additional info
`twitter:image`	A URL to an image file or to a dynamically generated image. `og:image` is used as a fallback.	`JPG`,`PNG`, `W
`twitter:card`	The type of card used for Twitter link previews	`summary`, `sum
`twitter:title`	A string that shows for Twitter link previews. `og:title` is used as a fallback.	70 characters m
`twitter:description`	A description for Twitter link previews. `og:description` is used as a fallback.	200 characters

```html filename="index.js"
<div>
  <head>
    <meta name="twitter:title" content="Vercel CDN for Twitter" />
    <meta name="twitter:description" content="Vercel CDN for Twitter" />
```

```
    <meta
      name="twitter:image"
      content="https://og-examples.vercel.sh/api/static"
    />
    <meta name="twitter:card" content="summary_large_image" />
  </head>
  <h1>A page with Open Graph Image.</h1>
</div>
```

---
title: "Deploying to Vercel"
description: "Learn how to create and manage deployments on Vercel."
last_updated: "2026-01-16T02:19:28.959Z"
source: "https://vercel.com/docs/deployments"

Deploying to Vercel

A **deployment** on Vercel is the result of a successful build of your project. Each time you deploy, Vercel generates a unique URL so yo

Vercel supports multiple ways to create a deployment:

- [Git](#git)
- [Vercel CLI](#vercel-cli)
- [Deploy Hooks](#deploy-hooks)
- [Vercel REST API](#vercel-rest-api)

Deployment Methods

Git

The most common way to create a deployment is by pushing code to a connected [Git repository](/docs/git). When you [import a Git reposito

Vercel supports the following providers:

- [GitHub](/docs/git/vercel-for-github)
- [GitLab](/docs/git/vercel-for-gitlab)
- [Bitbucket](/docs/git/vercel-for-bitbucket)
- [Azure DevOps](/docs/git/vercel-for-azure-pipelines)

You can also [create deployments from a Git reference](/docs/git#creating-a-deployment-from-a-git-reference) using the Vercel Dashboard i

Vercel CLI

You can deploy your Projects directly from the command line using [Vercel CLI](/docs/cli). This method works whether your project is conn

1. **Install Vercel CLI**:

```bash filename="Terminal" package-manager="npm"
npm i -g vercel
```

```bash filename="Terminal" package-manager="bun"
bun i -g vercel
```

```bash filename="Terminal" package-manager="yarn"
yarn global add vercel
```

```bash filename="Terminal" package-manager="pnpm"
pnpm i -g vercel
```

2. **Initial Deployment**:

In your project's root directory, run:

```bash
vercel --prod
```

This links your local directory to your Vercel Project and creates a [Production Deployment](/docs/deployments/environments#production-en

Vercel CLI can also integrate with custom CI/CD workflows or third-party pipelines. Learn more about the different [environments on Verce

Deploy Hooks

[Deploy Hooks](/docs/deploy-hooks) let you trigger deployments with a unique URL. You must have a connected Git repository to use this fe

1. From your Project settings, create a Deploy Hook
2. A unique URL is generated for each Project
3. Make an HTTP `GET` or `POST` request to this URL to trigger the deployment

Refer to the [Deploy Hooks documentation](/docs/deploy-hooks) for more information.

Vercel REST API

The [Vercel REST API](/docs/rest-api) lets you create deployments by making an HTTP `POST` request to the deployment endpoint. In this wo

1. Generate a SHA for each file you want to deploy
2. Upload those files to Vercel
3. Send a request to create a new deployment with those file references

This method is especially useful for custom workflows, multi-tenant applications, or integrating with third-party services not officially

Accessing Deployments

Vercel provides three default environments—**Local**, **Preview**, and **Production**:

1. **Local Development**: developing and testing code changes on your local machine
2. **Preview**: deploying for further testing, QA, or collaboration without impacting your live site
3. **Production**: deploying the final changes to your user-facing site with the production domain

Learn more about [environments](/docs/deployments/environments).

Using the Dashboard

Vercel's dashboard provides a centralized way to view, manage, and gain insights into your deployments.

Resources Tab and Deployment Summary

When you select a deployment from your **Project → Deployments** page, you can select the **Resources** tab to view and search:

- **Middleware**: Any configured [matchers](/docs/routing-middleware/api#match-paths-based-on-custom-matcher-config).
- **Static Assets**: Files (HTML, CSS, JS) and their sizes.
- **Functions**: The type, runtime, size, and regions.

You can use the three dot (…) menu for a given function to jump to that function in **Logs**, **Analytics**, **Speed Insights**, or the *

You can also see a summary of these resources by expanding the **Deployment Summary** section on a **Deployment Details** page. To visit

You'll also see your build time, detected framework, and any relevant logs or errors.

Project Overview

On your **Project Overview** page, you can see the latest production deployment, including the generated URL and commit details, and depl

Managing Deployments

From the **Deployments** tab, you can:

- **Redeploy**: Re-run the build for a specific commit or configuration.
- **Inspect**: View logs and build outputs.
- **Assign a Custom Domain**: Point custom domains to any deployment.
- **Promote to Production**: Convert a preview deployment to production (if needed).

For more information on interacting with your deployments, see [Managing Deployments](/docs/deployments/managing-deployments).

---
title: "Preview Deployment Suffix"
description: "When you create a new deployment, Vercel will automatically generate a unique URL which you can use to access that particul
last_updated: "2026-01-16T02:19:28.976Z"
source: "https://vercel.com/docs/deployments/preview-deployment-suffix"

Preview Deployment Suffix

Preview Deployment Suffixes allow you to customize the URL of a [preview deployment](/docs/deployments/environments#preview-environment-p

The entered custom domain must be:

- Available and active within the team that enabled the Preview Deployment Suffix
- Using Vercel's [Nameservers](/docs/domains/add-a-domain#vercel-nameservers)

Enabling the Preview Deployment Suffix

> **💡 Note:** Preview Deployment Suffix is included and enabled by default in Enterprise
> plans

To enable Preview Deployment Suffix, and customize the appearance of any of your generated URLs:

1. From your [dashboard](/dashboard), select the **Settings** tab
2. Select the **Billing** tab
3. Under **Add-Ons**, set the toggle for **Preview Deployment Suffix** to **Enabled**
4. Navigate to the **Settings** tab on the team dashboard
5. Select the **General** tab and scroll down to the **Preview Deployment Suffix** section
6. Enter the custom domain of your choice in the input, and push **Save**

If you are not able to use Vercel's Nameservers, see our guide on [how to use a custom domain without Vercel's Nameservers](/kb/guide/pre

See the [plans add-ons](/docs/pricing#pro-plan-add-ons) documentation for information on pricing.

Disabling the Preview Deployment Suffix

To disable Preview Deployment Suffix:

1. From your [dashboard](/dashboard), select the **Settings** tab
2. Select the **Billing** tab
3. Under **Add-Ons**, set the toggle for **Preview Deployment Suffix** to **Disabled**

The next preview deployment generated will revert back to the default `vercel.app` suffix.

Broken Preview Deployment Suffix error

You may encounter this error if you are using the [Preview Deployment Suffix](#preview-deployment-suffix) in your team. Make sure that th

- Active (not deleted)
- Available within the team that enabled the [Preview Deployment Suffix](#preview-deployment-suffix)
- Backed by an [active wildcard certificate](https://knowledge.digicert.com/generalinformation/INFO900.html)

The best way to satisfy all of these constraints is to ensure the domain is also added to a project located in the same team. In this pro

---
title: "Promoting Deployments"
description: "Learn how to promote deployments to production on Vercel."

last_updated: "2026-01-16T02:19:29.044Z"
source: "https://vercel.com/docs/deployments/promoting-a-deployment"

Promoting Deployments

By default, when you merge to or make commits to your production branch (often `main`), Vercel will automatically promote the changes to

- **[Instant rollback](#instant-rollback)**: You can use this as a way to instantly revert to an earlier [deployment](/docs/instant-rollb
- **[Promote preview to production](#promote-a-deployment-from-preview-to-production)**: You can use this as a way to promote a preview d
- **[Promote a staged production build](#staging-and-promoting-a-production-deployment)**: You can use this option to promote a productio

Instant Rollback

Use this when you want to replace the **current** production deployment with another deployment that has already been serving as current

For more information on how and when to use it, see the [Instant Rollback docs](/docs/instant-rollback).

Promote a deployment from preview to production

There may be times when you need to promote an existing preview deployment to production, such as when you need to temporarily use a bran

To promote an existing preview deployment to production on Vercel, do the following:

1. Go to your project's **Deployments** tab. This tab lists all the previously deployed builds
2. Click the ellipsis (), and from the context menu select **Promote to Production**
3. The popup dialog informs you of which domain(s) will be linked to the build once promoted. To confirm, select **Promote to Production*

> **💡 Note:** If you have different [Environment
> Variables](/docs/environment-variables#environments) set for preview and
> production deployments then the variables used will change from preview to
> those you have linked to the production environment. You **cannot use your
> preview environment variables in a production deployment**

Staging and promoting a production deployment

In some cases you may want to create a production-like deployment to use as a staging environment before promoting it to production.

In this scenario, you can turn off the auto-assignment of domains for your production build, as described below. Turning off the auto-ass

CLI

For steps on using this workflow in the CLI, see [Deploying a staged production build](/docs/cli/deploying-from-cli#deploying-a-staged-pr

Dashboard

1. On your [dashboard](/dashboard), select your project
2. Select the **Settings** tab and go to your **Environments** settings
3. Click on **Production** and go to the **Branch Tracking** section
4. Disable the **Auto-assign Custom Production Domains** toggle
5. When you are ready to promote this staged deployment to production, select the ellipses (…) next to the deployment
6. From the menu, select the **Promote** option
7. From the **Promote** dialog, confirm the deployment, and select the **Promote** button:

Vercel will instantly promote the deployment; it doesn't require a rebuild. Once promoted, the deployment is marked as [**Current**](#pro

Production deployment state

Your production deployments could be in one of three states:

- **Staged** – This means that a commit has been pushed to `main`, but a domain has not been auto-assigned to the deployment. This type o
- **Promoted** – This production deployment has been [promoted](#staging-and-promoting-a-production-deployment) from staging. If a deploy
- **Current** – This is the production deployment that is aliased to your domain and the one that is currently being served to your users

---
title: "Sharing a Preview Deployment"
description: "Learn how to share a preview deployment with your team and external collaborators."
last_updated: "2026-01-16T02:19:28.989Z"
source: "https://vercel.com/docs/deployments/sharing-deployments"

Sharing a Preview Deployment

Sharing with members of your team

By default, members of your [Vercel team](/docs/accounts/create-a-team) that have [access to your project](/docs/rbac/access-roles/projec

To share a preview deployment with a member of your team you can do any of the following:

- Send an invite to individual users (external or team members), or groups of people
- Copy the URL from the address bar of your browser and send it to them
- Select **Share** in the [Vercel Toolbar](/docs/vercel-toolbar) menu, copy the URL and send it to them

They will also be able to find it by using the [generated URL](/docs/deployments/generated-urls) from any deployment in the [Vercel dashb

Sharing a preview deployment with external collaborators

To share a deployment with anyone, you can do any of the following:

- **Recommended**: [Invite users](#invite-users) to view your deployment
- [Set access to anyone](#sharing-with-sharable-links-and-managing-permissions) (or anyone with the link if deployment protection is enab
- [Accept an access request](/docs/deployments/sharing-deployments#request-access)

When you share a preview deployment with an external user, **they will not be added to your Vercel team**. The collaborator does not need

Note that you can share two types of links: branch links, which reflect the latest commit, and commit links, which reflect changes up to

When sharing from the **Share** button next to the deployment in the Vercel dashboard, the share modal defaults to the branch link. You c

When sharing from **Share** in the toolbar menu, you'll share the current link. If it's a commit-specific link, you can switch to the br

Invite users

Users on Pro and Enterprise teams can use this method to add one or more collaborators. Hobby users are limited to one collaborator at an

1. Select **Share** in the toolbar menu or select the **Share** button next to the deployment in the [Vercel dashboard](/dashboard)
2. In the **Share** modal that appears, enter the email(s), or names of people on your Vercel team you want to invite. You can also add a
3. The invited user can now view the preview deployment. If Deployment Protection is enabled or if they want to add a comment, they will
4. You can revoke access at any time by returning to the **Share** dialog and choosing the **Revoke** icon next to the user's email

This is the **recommended method for sharing a deployment with external collaborators**, as it allows you to control who has access to yo

Sharing with sharable links and managing permissions

1. Select **Share** in the toolbar menu or select the **Share** button on the deployment page in the [Vercel dashboard](/dashboard)
2. In the **Share** modal that appears, you can manage who can view and comment on deployments:
   - **Team members with access**: This is the default setting. Only team members who have access to this project and external users gran
   - **Anyone (Without deployment protection)**: If you don't have [deployment protection](/docs/security/deployment-protection) enabled,
   - **Anyone with the link (With deployment protection)**: If you have deployment protection on, you can select **Anyone with the link**

3) After setting the chosen permission, use the **Copy Link** button to copy the link to your clipboard. This specific URL should be used

To learn more, see [sharable links](/docs/security/deployment-protection/methods-to-bypass-deployment-protection/sharable-links) in Deplo

Request access

If someone without access to comment attempts to log into the toolbar on a deployment, they will see a screen with the option to **Reques

To respond to the request:

1. Select **Share** in the toolbar menu or select the **Share** button next to the deployment in the [Vercel dashboard](/dashboard)
2. In the popup modal that appears, review the list under **Access Requests**
3. Respond to the request by either allowing or denying access

Sharing with deployment protection enabled

It is important to ensure the security of your preview deployments, which you can enable through [deployment protection](/docs/security/d

Deployment protection allows you to secure your preview deployments, with [Authentication](/docs/security/deployment-protection/methods-t

- If you don't have deployment protection enabled, anyone with the link can view your deployment
- If you have **Authentication** enabled, only team members can view your deployment, unless you have added the user individually or they
- If you have **Password Protection** enabled, only users with the password can view your deployment, unless you have added the user indi

---
title: "Troubleshooting Build Errors"
description: "Learn how to resolve common scenarios you may encounter during the Build step, including build errors that cancel a deploym
last_updated: "2026-01-16T02:19:29.026Z"
source: "https://vercel.com/docs/deployments/troubleshoot-a-build"

Troubleshooting Build Errors

You can troubleshoot build errors that occur during the Build step of your deployment to Vercel. This guide will help you understand how

Troubleshooting views

You can use the following views on your dashboard to troubleshoot a build:

- **Build** logs - the console output when your deployment is building which can be found under the Deployment Status section of the Proj
- **Resources** tab - the functions, middleware, and assets from your deployment's build.
- **Source** tab - the output of the build after the deployment is successful. This can also be accessed by appending `/_src` to the Depl

You can navigate to these views from the Deployment page by clicking on the **Source** tab, the **Resources** tab or the **Build Logs** a

Troubleshoot Build failures

If your build fails, Vercel will report the error message on the **Deployments** page so that you can investigate and fix the underlying

In the following we show you how to look up the error message of your failed build.

Investigating Build logs

[Build logs](/docs/deployments/logs) provide you with an insight into what happened during the build of a deployment and can be accessed

1. From your Vercel dashboard, select the project and then the [**Deployments**](/docs/projects/project-dashboard#deployments) tab
2. Select the deployment. When there are build issues you will notice an error status next to deployment name

3) On the errored deployment's page, you will see a summary of the error in the preview section. In the **Deployment Details** section, e
   There are situations where [build logs are not available](#build-logs-not-available), in this scenario the error will be presented in

4) Scroll down in the build logs until you find a red section where the keyword **"Error"** is mentioned.
   It can be mentioned once or multiple times.
   In many cases, the last mention is not indicative like in the example below where it says **`yarn run build exited with 1`**.
   If you look a few lines above, you will see an additional error which in this case indicates where the problem is with a link for more

It is recommended to build your project on your local machine first (the build command varies per project) before deploying on Vercel. Th

Build Logs not available

Builds can fail without providing any build logs when Vercel detects a missing precondition that prevents a build from starting. For exam

- An invalid [`vercel.json` configuration](/docs/project-configuration) was committed
- When using [Ignored Build Steps](/kb/guide/how-do-i-use-the-ignored-build-step-field-on-vercel)
- Commits were made from a contributor that is not a [team member](/docs/accounts/team-members-and-roles)

In this case, you cannot access the **Building** accordion described above, and instead, Vercel will present an overlay that contains the

Cancelled Builds due to limits

Sometimes, your Deployment Build can hit platform limits so that the build will be cancelled and throw a corresponding error that will be

Build container resources

Every Build is provided with the following resources:

	Hobby	Pro	Enterprise
Memory	8192 MB	8192 MB	Custom
Disk space	23 GB	23 GB	Custom
CPUs	2	4	Custom

For Hobby customers, these limits are fixed. Pro and Enterprise customers can purchase [Enhanced or Turbo build machines](/docs/builds/ma

By default, the system generates this report only when it detects a problem. To receive a report for every deployment, set `VERCEL_BUILD_

This report helps you detect hidden Out of Memory (OOM) events, and provides insights into disk usage by breaking down the sizes of your

Resource	Allocation	Description
Memory	8192 MB	Fixed memory allocation. Builds exceeding this limit will be cancelled
CPUs	4	Number of CPUs allocated for build processing
Disk Space	23 GB	Fixed disk space allocation. Builds exceeding this limit will be cancelled
System Report	Conditional	Generated in [Build logs](/docs/deployments/logs) when memory or disk space limits are reached. Availa
Forced Reporting	Optional	Set `VERCEL_BUILD_SYSTEM_REPORT=1` as an [environment variable](/docs/environment-variables#creating-e

Review the below steps to help navigate this situation:

- Review what package the error is related to and explore the package's documentation to see if the memory allocation can be customized w
- If no package is specified, look into reducing the amount of JavaScript that your Project might be using or find a more efficient JavaS
- Review what you are importing in your code such as third-party services, icons and media files

Build duration

The total build duration is shown on the Vercel Dashboard and includes all three steps: building, checking, and assigning domains. Each s

A Build can last for a maximum of 45 minutes. If the build exceeds this time, the deployment will be canceled and the error will be shown

Caching

The maximum size of the Build's cache is 1 GB. It is retained for one month and it applies at the level of each [Build cache key](#cachin

It is not possible to manually configure which files are cached at this time.

Other Build errors

You may come across the following Build-specific errors when deploying your Project. The link for each error provides possible causes of

- [Missing Build script](/docs/errors/error-list#missing-build-script)
- [Recursive invocation of commands](/docs/errors/error-list#recursive-invocation-of-commands)
- [Pnpm engine unsupported](/docs/errors/error-list#pnpm-engine-unsupported)

A 'module not found' error is a syntax error that will appear at build time. This error appears when the static import statement cannot f

Troubleshoot Build time

Understanding Build cache

The first Build in a Project will take longer as the Build cache is initially empty. Subsequent builds that have the same [Build cache ke

What is cached

Vercel caches files based on the [Framework Preset](/docs/deployments/configure-a-build#framework-preset) selected in your [Project setti

Note that the framework detection is dependent on the preset selection made in the [Build settings](/docs/deployments/configure-a-build#b

Caching process

At the beginning of each build, the previous Build's cache is restored prior to the [Install Command](/docs/deployments/configure-a-builds

- [Personal Account](/docs/teams-and-accounts#creating-a-personal-account) or [Team](/docs/teams-and-accounts)
- [Project](/docs/projects/overview)
- [Framework Preset](/docs/deployments/configure-a-build#framework-preset)
- [Root Directory](/docs/deployments/configure-a-build#root-directory)
- [Node.js Version](/docs/functions/runtimes/node-js#node.js-version)
- [Package Manager](/docs/deployments/configure-a-build#install-command)
- [Git branch](/docs/git)

Let's say that under your account `MyTeam`, you have a project `MySite` that is connected to your Git repository `MyCode` on the `main` b

If you create a new Git branch in `MyCode` and make a commit to it, there is no cache for that specific branch. In this case, the last [p

If you use [Vercel functions](/docs/functions) to process HTTP requests in your project, each Vercel Function is built separately in the

At the end of each Build step, successful builds will update the cache and failed builds will **not modify** the existing cache.

Excluding development dependencies

Since development dependencies (for example, packages such as `webpack` or `Babel`) are not needed in production, you may want to prevent

Managing Build cache

Sometimes, you may not want to use the Build cache for a specific deployment. You can invalidate or delete the existing Build cache in th

- Use the **Redeploy** button for the specific deployment in the Project's [Deployments](/docs/deployments/managing-deployments) page. In
- Use [`vercel --force`](/docs/cli/deploy#force) with [Vercel CLI](/docs/cli) to build and deploy the project **without** the Build cache
- Use an Environment Variable `VERCEL_FORCE_NO_BUILD_CACHE` with a value of `1` on your project to skip the Build cache
- Use an Environment Variable `TURBO_FORCE` with a value of `true` on your project to skip Turborepo [Remote Cache](/docs/monorepos/remot

- Use the `forceNew` optional query parameter with a value of `1` when [creating a new deployment with the Vercel API](/docs/rest-api/ref

> **💡 Note:** When redeploying **without** the existing Build Cache, the Remote Cache from
> [Turborepo](https://turborepo.com/docs/core-concepts/remote-caching) and
> [Nx](https://nx.app/) are automatically excluded.

---
title: "Troubleshoot project collaboration"
description: "Learn about common reasons for deployment issues related to team member requirements and how to resolve them."
last_updated: "2026-01-16T02:19:29.103Z"
source: "https://vercel.com/docs/deployments/troubleshoot-project-collaboration"

Troubleshoot project collaboration

This guide will help you understand how to troubleshoot deployment failures related to project collaboration.

For private repositories, if we cannot identify the Vercel user associated with a commit, any deployment associated with that commit will

> **💡 Note:** Collaboration is free for public repositories.

Team configuration

Hobby teams

The [Hobby Plan](/docs/plans/hobby) does not support collaboration for private repositories. If you need collaboration, upgrade to the [P

To deploy commits under a Hobby team, the commit author must be the owner of the Hobby team containing the Vercel project connected to th

To make sure we can verify your commits:

1. Make sure all commits are authored by the git user associated with your account.
2. Link your git provider to your Vercel account in [Account Settings](/docs/accounts#sign-up-with-a-git-provider)

> **💡 Note:** If your account is not connected to your git provider, make sure you've properly configured your [Vercel email address](/d

For more information, see [Using Hobby teams](/docs/git#using-hobby-teams)

Pro teams

The [Pro Plan](/docs/plans/pro-plan) allows for collaboration through team membership. Each person committing to your codebase should be

To deploy commits under a Vercel Pro team, the commit author must be a member of the team containing the Vercel project connected to the

To make sure we can verify commits associated with your team:

1. Each person committing code can be [added as a team member](/docs/rbac/managing-team-members).
2. Make sure the commit author is a confirmed [member of your team](/docs/accounts#team-membership).
3. Team members should link their git provider to their Vercel account in [Account Settings](/docs/accounts#sign-up-with-a-git-provider)

For more information, see [Using Pro teams](/docs/git#using-pro-teams)

Bot access

Ensure your bots are properly configured and that their commits are clearly identified as automated.

Account configuration

Connecting Git provider accounts

Each team member must connect their git provider account to their Vercel account:

1. Visit [Account Settings](https://vercel.com/account/settings/authentication)
2. Navigate to the [**Login Connections**](/docs/accounts#login-methods-and-connections) section
3. Connect your GitHub, GitLab, or Bitbucket account

Managing multiple email addresses

If you use multiple email addresses for git commits, you will need to configure a secondary email address with either your git provider o

To add secondary email addresses to your Vercel account:

1. Go to your [Account Settings](https://vercel.com/account/settings#email)
2. Add any email addresses you use for git commits
3. Verify each email address

---
title: "Exclude Files from Deployments with .vercelignore"
description: "The .vercelignore file allows you to define which files and directories should be ignored when uploading your project to Ve
last_updated: "2026-01-16T02:19:29.061Z"
source: "https://vercel.com/docs/deployments/vercel-ignore"

Exclude Files from Deployments with .vercelignore

The `.vercelignore` file can be used to specify files and directories that should be excluded from the deployment process when using Verc

The `.vercelignore` file should be placed in the root directory of your project and should contain a list of files and directories, one p

```bash filename=".vercelignore"
image
private.html
```

Allowlist

A typical `.vercelignore` file assumes all files are allowed and each entry is a pattern to ignore. Alternatively, you can ignore all fil

Add a wildcard `/*` as the first line in `.vercelignore` to ensure all directories and files in the **project root** are ignored. The fol

```bash filename=".vercelignore"
# Ignore everything (folders and files) on root only
/*
!api
!vercel.json
!*.html
```

Uploaded Files

Aside from the [default exclusions](/docs/deployments/build-features#ignored-files-and-folders), all files within your project are upload

The complete list of files and directories excluded by default can be found in the [ignored files and folders](/docs/deployments/build-fe

Served Files

The use of a `.vercelignore` configuration file allows you to keep private files safe and also makes your deployment faster by uploading

Monorepos

If you have a monorepo, a `.vercelignore` in the project root directory always takes precedence over one that is defined at the root leve

---
title: "Using the Directory Listing"
description: "The Directory Listing is served when a particular path is a directory and does not contain an index file. Learn how to togg
last_updated: "2026-01-16T02:19:29.095Z"
source: "https://vercel.com/docs/directory-listing"

Using the Directory Listing

The Directory Listing setting enables you to display the contents of a directory when a user visits its path. For example, if you create

You can enable or disable Directory Listing from the **Advanced** tab in your project settings.

When enabled, the Directory Listing will be displayed. When disabled, a "Not Found" error will be displayed with status code `404`.

> **💡 Note:** If Directory Listing isn't working, navigate to your deployment in the
> dashboard and select the tab to view the contents of
> your project. Ensure the expected directory and files are listed.

Disabling Directory Listing on a specific directory

To prevent Directory Listing for a specific path, you can either:

- Add an index file with any extension except `.css`, such as `index.html`. This file will be displayed instead of the Directory Listing
- Or, [set up a custom 404 error](/kb/guide/custom-404-page)

---
title: "Directory Sync"
description: "Learn how to configure Directory Sync for your Vercel Team."
last_updated: "2026-01-16T02:19:29.085Z"
source: "https://vercel.com/docs/directory-sync"

Directory Sync

Directory Sync helps teams manage their organization membership from a third-party identity provider like Google Directory or Okta. Direc

When Directory Sync is configured, changes to your Directory Provider will automatically be synced with your [team members](/docs/rbac/ma

> **💡 Note:** Make sure that you still have the right permissions/role after configuring
> Directory Sync, [otherwise you might lock yourself
> out.](#preventing-account-lockout)

All team members will receive an email detailing the change. For example, if a new user is added to your Okta directory, that user will a

You can configure a mapping between your Directory Provider's groups and a Vercel Team role. For example, your **Engineers** group on Okt

Configuring Directory Sync

To configure directory sync for your team:

1. Ensure your team is selected in the [scope selector](/docs/dashboard-features)
2. From your team's dashboard, select the **Settings** tab, and then **Security & Privacy**
3. Under SAML Single Sign-On, select the **Configure** button. This opens a dialog to guide you through configuring Directory Sync for yo
4. Once you have completed the configuration walkthrough, configure how Directory Groups should map to Vercel Team roles:
5. Finally, an overview of all synced members is shown. Click **Confirm and Sync** to complete the syncing:
6. Once confirmed, Directory Sync will be successfully configured for your Vercel Team.

> **💡 Note:** SAML Single Sign-On is optionally available on the Enterprise plan, or as a paid add-on for the Pro plan. To enable,
> Enterprise teams can contact [sales](https://vercel.com/contact/sales), and Pro teams can purchase the add-on
> from their team's [Billing settings](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fsettings%2Fbilling%23paid-add-ons).

Supported providers

See [SAML Single Sign-On](/docs/saml#saml-providers) for a list of all the SAML providers that Vercel supports.

Preventing account lockout

To prevent account lockout ensure that at least one person in your team has the owner role, and that they are not removed from the team.

If access is lost due to removal of team owners, use the following group names to automatically allocate the corresponding roles to indiv

Group name	Role

| `vercel-role-owner` | [Owner](/docs/rbac/access-roles#owner-role)
| `vercel-role-member` | [Member](/docs/rbac/access-roles#member-role)
| `vercel-role-developer` | [Developer](/docs/rbac/access-roles#developer-role)
| `vercel-role-billing` | [Billing](/docs/rbac/access-roles#billing-role)
| `vercel-role-viewer` | [Viewer Pro](/docs/rbac/access-roles#pro-viewer-role) or [Viewer Enterprise](/docs/rbac/access-roles#enterp
| `vercel-role-contributor` | [Contributor](/docs/rbac/access-roles#contributor-role)

---
title: "Uploading Custom SSL Certificates"
description: "By default, Vercel provides all domains with a custom SSL certificates. However, Enterprise teams can upload their own cust
last_updated: "2026-01-16T02:19:29.090Z"
source: "https://vercel.com/docs/domains/custom-SSL-certificate"

Uploading Custom SSL Certificates

By default, Vercel provides all domains with custom SSL certificates. However, Enterprise teams can upload a custom SSL certificate. This

Custom SSL certificates can be uploaded through the [**Domains** tab on your team's dashboard](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7

Uploading a custom certificate follows a three step process:

1. Providing the private key for the certificate
2. Providing the certificate itself
3. Providing the Certificate Authority root certificate such as one of [Let's Encrypt's ISRG root certificates](https://letsencrypt.org/c

The content of each element must be copied and pasted into the input box directly. The certificate and private key can be extracted from

```bash filename="certificate.pem"
-----BEGIN CERTIFICATE-----
<Certificate body will be here>
-----END CERTIFICATE-----
```

```bash filename="private-key.pem"
-----BEGIN PRIVATE KEY-----
<Private key body will be here>
-----END PRIVATE KEY-----
```

SSL best practices

When uploading your SSL certificate, you should note the following:

1. The automatically generated certificate will remain in place, but a custom certificate is prioritized over the existing certificate. T
2. You can include canonical names CN's (CN's) for other subdomains on the certificate without needing to add these domains to Vercel. Th
3. Wildcards certificates can be uploaded.
4. Certificates with an explicitly defined subdomain are prioritized over a wildcard certificate when both are valid for a given subdomai
5. Vercel cannot automatically renew custom certificates. If a custom certificate is within 5 days of expiration, an automatically genera

---
title: "Managing DNS Records"
description: "Learn how to add, verify, and remove DNS records for your domains on Vercel with this guide."
last_updated: "2026-01-16T02:19:29.131Z"
source: "https://vercel.com/docs/domains/managing-dns-records"

Managing DNS Records

Once you've added a domain and it's using Vercel's nameservers, you can view its DNS records from your team's [**Domains** page](https://

> **💡 Note:** To make sure DNS records are applied, and to allow you to manage them, your
> domain needs to use [Vercel's nameservers](/docs/domains/managing-nameservers)
> . If you are using a third-party domain, you will be provided with the Vercel
> nameservers to copy and use with your registrar.

Adding DNS Records

- ### Selecting your Domain
  On your team's [dashboard](/dashboard), select the **Domains** tab. From the Domains page, click on a domain of your choice to view its

- ### Add DNS Record
  Once on the Advanced Settings page of your domain, select the **Enable Vercel DNS** button to fill out the DNS Record form. Once comple

  You can then create a new DNS record with the following data:
  - **Name:** The prefix or location of the record. For www.example.com, the name argument would be www.
  - **Type:** Types can be `A`, `AAAA`, `ALIAS`, `CAA`, `CNAME`, `HTTPS`, `MX`, `NS`, `SRV`, or `TXT`.
  - **Value:** The value of the record.
  - **TTL:** Default is 60 seconds. For advanced users, this value can be customized.
  - **Comment:** An optional comment to provide context on what this record is for.
  - **More:** Some records will require more data. MX records, for example, will request "priority".

    > **💡 Note:** Once a DNS record has been added, it can take up to 24 hours to the DNS
    > records to fully update and any local caches to be cleared.

Verifying DNS Records

Once DNS records have been changed, you may wish to check that these have been set correctly. There are many third-party tools that do th

You can also use the `dig` command to check the DNS record for your domain:

```bash filename="terminal"
$ dig A api.example.com +short
```

```bash filename="terminal"
```

```
$ dig MX example.com +short
```

Removing DNS Records

To remove DNS records:

1. On your team's [dashboard](/dashboard), select the [**Domains** tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fdomains\&title=Go+t
2. Select the ellipsis (⋯) to access the context menu and select **Delete DNS Zone**. Follow the prompts to delete the record.

Default records can't be removed. However, new records can override them if required.

DNS Presets

Vercel does not provide an email service. To be able to receive emails or add specific DNS configurations through a domain that you've ad

Vercel streamlines this process for common third-party services by allowing you to add missing DNS Records using **DNS Presets** on your

1. From your [dashboard](/dashboard), navigate to the **Domains** tab.

2. Select the domain you wish to add a preset to and click the **Add DNS Preset** dropdown on the right:

3. You will be presented with a list of commonly used third-party providers. If your provider is listed, select it, and the necessary DNS

If your provider is not listed, please refer to their documentation to find out which DNS Records you need to add.

Migrating DNS records from an external registrar

Once you have added a [domain to your Vercel project](https://vercel.com/docs/concepts/projects/custom-domains) and also verified the cer

If you decide to change the Nameservers of your domain, you can follow the below instructions which will help you migrate your DNS config

Clone the Current DNS Configuration

To locate the current DNS provider of your domain, you can run the following command:

```bash filename="terminal"
$ dig NS example.com +short
```

The result will show the current DNS authority. Next, you'll need to locate your DNS records from the provider's dashboard.

After you've successfully located all records associated with your domain, you may now add them to Vercel. You can either do this manuall

Importing a zone file

If you have downloaded a zone file from your existing file, you may use the following comand to upload that to Vercel:

```bash
vercel dns import [your-domain] [zonefile]
```

If you do not apply a custom zone file, transferring in a domain automatically applies the default Vercel DNS settings.

Verify the Records

To verify the records, you can now query the DNS configuration that will be served by Vercel:

```bash filename="terminal"
$ dig A api.example.com +short @ns1.vercel-dns.com
```

Then, check the DNS records from the existing provider to make sure they match. If you were moving your DNS from [Cloudflare](https://ver

```bash filename="terminal"
$ dig A api.example.com +short @example.ns.cloudflare.com
```

Before proceeding, we recommend checking every record you moved. For more insight into the DNS resolution, remove the `+short` flag.

Switch the Nameservers

In your registrar's dashboard (where you bought the domain), change the Nameservers to your new provider.
Nameserver changes can take up to 48 hours to propagate. If you bought the domain from Vercel, you can
[manage nameservers](https://vercel.com/docs/concepts/projects/domains/managing-nameservers) from the [domains page](https://vercel.com/d

---
title: "Managing Nameservers"
description: "Learn how to add custom nameservers and restore original nameservers for your domains on Vercel with this guide."
last_updated: "2026-01-16T02:19:29.139Z"
source: "https://vercel.com/docs/domains/managing-nameservers"

Managing Nameservers

[Nameservers](/docs/domains/working-with-nameservers) are used to resolve domain names to IP addresses. For domains with Vercel as the re

Sometimes, however, you may need to delegate nameserver management to another host. For domains registered with Vercel, you can [add cust

For domains that are not registered with Vercel, you can change the nameservers directly from the domain registrar's dashboard.

Nameserver changes can take up to 48 hours to complete due to [DNS propagation](https://ns1.com/resources/dns-propagation).

Add custom nameservers

1. Ensure your account or team is selected in the scope selector

2. Navigate to the **Domains** tab and select the domain

3. On your domain's settings page, under **Nameservers**, click the **Edit** button:

4. In the **Edit Nameservers** modal, add the new nameservers:

Add Vercel's nameservers

> **💡 Note:** Before using Vercel's nameservers, you should ensure that you own the domain.

1. Ensure your account or team is selected in the scope selector
2. Navigate to the **Domains** tab and select the domain
3. On your domain's settings page, under **DNS Records**, click the **Enable Vercel DNS** button to opt in
4. You then must configure the following nameservers from the domain registrar's dashboard

- `ns1.vercel-dns.com`
- `ns2.vercel-dns.com`

Restore original nameservers

1. Ensure your account or team is selected in the scope selector
2. Navigate to the **Domains** tab and select the domain
3. Under **Nameservers**, select the **Restore Original Nameservers** button
4. On the **Restore Original Nameservers** modal confirm the nameservers that will be present after the change

Vercel will present a message when you have successfully submitted the nameserver change.

---
title: "Domains Overview"
description: "Learn the fundamentals of how domains, DNS, and nameservers work on Vercel."
last_updated: "2026-01-16T02:19:29.148Z"
source: "https://vercel.com/docs/domains"

Domains Overview

A **domain** is a user-friendly way of referring to the address access a website on the internet. For example, the domain you're reading

The system that manages the details about where a site is located on the internet, is known as **DNS or the Domain Name System**. At its r

1. You enter `vercel.com` in your browser. Your browser will first check its local DNS cache to see if it knows the IP address of `vercel

2. Your browser initiates a DNS query through a server known as a **recursive resolver**, usually provided by your [ISP](# "ISP or Intern

3. There is a network of DNS servers, in a hierarchy, located all around the world. The recursive resolver will query in the following pa
   - At the entrance to the network are 13 **root nameservers**. These are the servers that will be contacted first. The root server will
   - The TLD nameservers store information about domain names that belong to the same TLD. For example, when searching for `vercel.com`,
   - This TLD server will then respond resolver with details about the **authoritative nameserver** that has the IP address mapping for `

4. Once your browser has the IP address, an HTTP request is made by the browser to the web server located at that IP address.

> **💡 Note:** This list is just a general overview and doesn't happen every time. Most of us
> tend to visit the same sites over and over. Therefore, the request will first
> check the cache from your browser and then from the recursive resolver,
> allowing for quicker load times. In addition, this example describes a basic
> unicast DNS network. In reality, when using Vercel, you're using anycast
> servers on the Vercel CDN.

This overview shows a point of view of a user visiting your site. **But what does this look like when you're the developer creating a sit

When you've created a Project and deployed it on Vercel, your site lives on Vercel's web servers, which we know to be at the IP address `

This is where, as a developer, you may have to configure the DNS settings to tell the authoritative server exactly where your site lives.

- **DNS records**: DNS records are an entry in a database that maps the domain with the IP address, which is then stored on the authorita
- **Nameserver**: Nameservers are an important part of the DNS. They refer to the *actual* server that maintains and manages the DNS reco
- **SSL Certificates**: SSL Certificates are a way to show that there is a secure connection from your domain to your website. These are

More resources

- [Working with domains](/docs/domains/working-with-domains)
- [Working with DNS](/docs/domains/working-with-dns)
- [Working with nameservers](/docs/domains/working-with-nameservers)
- [Working with SSL](/docs/domains/working-with-ssl)
- [Troubleshooting domains](/docs/domains/troubleshooting)

---
title: "Pre-Generate SSL Certificates"
description: "test"
last_updated: "2026-01-16T02:19:29.160Z"
source: "https://vercel.com/docs/domains/pre-generating-ssl-certs"

Pre-Generate SSL Certificates

> **💡 Note:** This page is part the domains transfer experience. See [this
> page](/docs/domains/working-with-domains/transfer-your-domain#transfer-a-domain-to-vercel)
> for the full set of steps to transfer a domain to Vercel.

This article guides you through all the steps necessary to set up SSL certificates for a domain
being migrated to Vercel without downtime. Your domain should be serving content from 3rd party
servers that are unrelated to Vercel, and you need to be prepared to make the necessary

DNS changes.

You can do this using either the Vercel Domains dashboard, or the [Vercel CLI](/docs/cli).

Generating a Certificate

In order to issue certificates through the dashboard for a domain, first ensure the domain belongs to a team. You can then click into the
scroll down to "SSL Certificates" and click "Pre-generate SSL certificates". Please note this option is only available if you do not alre
have any SSL certificates issued for the domain.

If you choose to do this through the terminal, you can run the following command to get the challenge records for your domain:

```bash filename="terminal"
vercel certs issue "*.example.com" example.com --challenge-only
```

Setting your DNS records and finalizing

In order to verify ownership of your domain, copy the TXT records into your DNS on the registrar you are using.

Click "Verify" to verify that the records have been set and issue the certificate. DNS records can take time to propagate,
so if it doesn't work immediately, it's worth waiting for the records to propagate before taking further action.

To check whether the TXT records have propagated, you can use the following command in a terminal of your choice:

```bash filename="terminal"
nslookup -type=TXT example.com
```

Once TXT records have propagated, you can click "Verify" to issue the SSL certificates.

If you choose to issue SSL certificates through the terminal, you can run the command previously used without the
`--challenge-only` flag:

```bash filename="terminal"
vercel certs issue "*.example.com" example.com
```

Verifying the Certificate

Before you change the DNS records of your domain, you can verify if the certificate is correct and will be accepted by browsers. Run the

```bash filename="terminal"
curl https://example.com --resolve example.com:443:76.76.21.21 -I
```

If the request is successful, the certificate is working and you can proceed with the migration.

Finishing connecting your domain to Vercel

To migrate your deployment to Vercel, add the provided A or CNAME record from your project's Domain Settings page to your DNS configurati
See [this detailed guide](/kb/guide/a-record-and-caa-with-vercel) on using domains with **A** records for more information.

For more details on performing a migration, see [this guide](/docs/domains/managing-dns-records#migrating-dns-records-from-an-external-re

---
title: "Programmatic Domain Management"
description: "Programmatically search, price, purchase, renew, and manage domains with Vercel"
last_updated: "2026-01-16T02:19:29.176Z"
source: "https://vercel.com/docs/domains/registrar-api"

Programmatic Domain Management

The domains registrar API enables you to programmatically manage your domain lifecycle from search to renewal.

Getting started with the API

You can start using the REST API by:

1. [Creating a token](https://vercel.com/docs/rest-api/reference/welcome#creating-an-access-token)
2. Using the token in either of the following ways:

   - Use the [Vercel SDK](https://vercel.com/docs/rest-api/reference/sdk)

   In the following example, use the Vercel SDK to get the supported TLDs.

   ```ts filename="index.ts"
   import { Vercel } from '@vercel/sdk';

   const vercel = new Vercel({
     bearerToken: '<YOUR_BEARER_TOKEN_HERE>',
   });

   const result = await vercel.domainsRegistrar.getSupportedTlds();
   ```

   - Use the language of your choice by calling the endpoints directly and providing your token.

   In the following example, we use `cURL` to get the supported TLDs.

   ```bash filename="terminal"
   curl --request GET \
     --url https://api.vercel.com/v1/registrar/tlds/supported \
     --header 'Authorization: Bearer <token>'
   ```

You can use the domains registrar API to do the following:

Catalog & pricing

- [List all supported top-level domains (TLDs)](https://vercel.com/docs/rest-api/reference/endpoints/domains-registrar/get-supported-tlds
- [Get pricing for specific TLDs](https://vercel.com/docs/rest-api/reference/endpoints/domains-registrar/get-tld-price-data)
- [Retrieve per-domain pricing information](https://vercel.com/docs/rest-api/reference/endpoints/domains-registrar/get-price-data-for-a-d

Availability

- [Check single domain availability](https://vercel.com/docs/rest-api/reference/endpoints/domains-registrar/get-availability-for-a-domain
- [Perform bulk availability checks for multiple domains](https://vercel.com/docs/rest-api/reference/endpoints/domains-registrar/get-avai

Orders & purchases

- [Purchase a domain](https://vercel.com/docs/rest-api/reference/endpoints/domains-registrar/buy-a-domain)
- [Execute bulk domain purchases](https://vercel.com/docs/rest-api/reference/endpoints/domains-registrar/buy-multiple-domains)
- [Fetch order status by ID](https://vercel.com/docs/rest-api/reference/endpoints/domains-registrar/get-a-domain-order)

Transfers

- [Retrieve authorization codes for domain transfers](https://vercel.com/docs/rest-api/reference/endpoints/domains-registrar/get-the-auth
- [Initiate domain transfers to Vercel](https://vercel.com/docs/rest-api/reference/endpoints/domains-registrar/transfer-in-a-domain)
- [Track transfer status and completion](https://vercel.com/docs/rest-api/reference/endpoints/domains-registrar/get-a-domains-transfer-st

Management

- [Renew domains before expiration](https://vercel.com/docs/rest-api/reference/endpoints/domains-registrar/renew-a-domain)
- [Enable or disable automatic renewal](https://vercel.com/docs/rest-api/reference/endpoints/domains-registrar/update-auto-renew-for-a-do
- [Update nameserver configurations](https://vercel.com/docs/rest-api/reference/endpoints/domains-registrar/update-nameservers-for-a-doma
- [Fetch TLD-specific contact information schemas](https://vercel.com/docs/rest-api/reference/endpoints/domains-registrar/get-contact-inf

Deprecations and migration

The following legacy domains API endpoints are now deprecated and will be sunset on November 8, 2025:

- [Purchase a domain](https://vercel.com/docs/rest-api/reference/endpoints/domains/purchase-a-domain-deprecated)
- [Check the price for a domain](https://vercel.com/docs/rest-api/reference/endpoints/domains/check-the-price-for-a-domain-deprecated)
- [Check a Domain Availability](https://vercel.com/docs/rest-api/reference/endpoints/domains/check-a-domain-availability-deprecated)
- [Get domain transfer info](https://vercel.com/docs/rest-api/reference/endpoints/domains/get-domain-transfer-info-deprecated)

If you are currently using the Vercel CLI for domain purchases, pricing, or availability, upgrade to CLI version `48.2.8` or later.

---
title: "Supported domains"

last_updated: "2026-01-16T02:19:29.195Z"
source: "https://vercel.com/docs/domains/supported-domains"

Supported domains

Vercel supports the following top-level domains (TLDs) for [purchase](/docs/domains/working-with-domains#buying-a-domain-through-vercel)

---
title: "Troubleshooting domains"
description: "Learn about common reasons for domain misconfigurations and how to troubleshoot your domain on Vercel."
last_updated: "2026-01-16T02:19:29.216Z"
source: "https://vercel.com/docs/domains/troubleshooting"

Troubleshooting domains

There are many common reasons why your domain configuration may not be working. Check the following:

- Is your domain [added](/docs/domains/add-a-domain#add-and-configure-domain) to your Vercel project?
- Is your custom domain pointed to the provided Vercel `CNAME`/`A` record correctly? You can check it by using `dig [example.com]` in you
- If you use the [nameservers method](/docs/domains/troubleshooting#configuring-nameservers-for-wildcard-domains) on your apex domain, pl
- Is the issue only local to you? Try to clear your browser cache, and flush DNS caches on your machine/network if possible.

Misconfigured domain issues

When you add a domain to Vercel that you have purchased from a third-party DNS provider, you may see an **Invalid Configuration** alert.

- You need to configure the [DNS](#common-dns-issues) records of your domain with your DNS provider so they can be used with your project
- If your domain is in use by another Vercel account, you may be prompted to [verify access to the domain](/docs/domains/add-a-domain#ver
- There was an issue generating the SSL certificate for your domain. The most common reason for this is [missing CAA records](#missing-ca
- You have configured [wildcard subdomains](/docs/domains/add-a-domain#using-wildcard-domain) on your project, but their nameservers aren

Common DNS issues

Vercel is expecting either an `A` record or a `CNAME` record. In your Project Settings under the Domain page, you'll find the precise `CN

- `dig ns [domain]` to get a domain's nameservers
- `dig a [apex domain e.g. example.com]` to get a domain's `A` record
- `dig cname [subdomain e.g. www.example.com]` to get a domain's `CNAME` record

If you prefer a non-command-line interface, you can use a free online tool, such as [Google Public DNS](https://dns.google/). If any of t

Why are my DNS records taking so long to update?

DNS changes can take a while to propagate across the globe, depending on the previous DNS record TTL length. This may mean that certain r
Changes to standard DNS records (A, CNAME, TXT, etc.) typically propagate quicker, but changing a domain's nameservers can take up to **2

For more information on [propagation times](/docs/domains/working-with-dns#dns-propagation) for nameservers and other DNS records, see "[

> **💡 Note:** Before changing your DNS records to point to Vercel, we recommend updating
> your existing DNS record to "lower" the TTL (for example 60 seconds) and

> waiting for the old TTL to expire. Lowering the current TTL and changing a DNS
> record after its TTL expiration period can ensure that you can quickly roll
> back the change if you encounter an issue. You can then increase the DNS
> record TTL to its original value once you confirm everything is working as
> expected.

IPv6 support

While we allow the [creation](/docs/domains/managing-dns-records#adding-dns-records) of AAAA records when using Vercel's nameservers, **w

Syntax errors debugging

When working with DNS records, you may make minor errors in the syntax. These errors can be difficult to debug. Below is a list of common

Using the domain as part of the **Name** argument

When you add a new DNS record to a domain, the **Name** field should use the prefix or location of the record. For `www.example.com`, the

If you have already added a record with this, [remove the record](/docs/domains/managing-dns-records#removing-dns-records) from the **DNS

Absolute CNAME records

When you add a custom domain with a subdomain to your project, we'll prompt you to add a CNAME DNS record in order to configure the domai

This means that when you add a new CNAME record to your DNS provider, you **must** copy the value exactly as it appears, **including** th

Common Nameserver issues

Configuring nameservers for wildcard domains

When you add any custom domain to your Vercel project you must [configure](/docs/domains/add-a-domain#configure-the-domain) the DNS recor

This is because Vercel needs to be able to set DNS records in order to generate the wildcard certificates. The service that Vercel uses t

For more information, see [Why must we use the Domain Nameservers method for Wildcard Domains on Vercel?](/kb/guide/why-use-domain-namese

Common domain issues

Domains and emails

When you buy a new domain, you may want to also set up an email address with this domain. Vercel **does not provide a mail service for do

When you add your custom domain to a project and use Vercel's nameservers, you will need to add `MX` records to continue receiving email.
[Why am I no longer receiving email after adding my domain to Vercel?](/kb/guide/why-has-email-stopped-working)

Purchasing a domain through Vercel

All domain purchases and renewals through Vercel are final and cannot be refunded once processed. For more information, see [Can I get a

Pending domain purchases

When a domain purchase does not go through immediately, your payment method may show a **temporary authorization** — this is a pending ho

If the purchase is processing, your domain will appear in the [Domains tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fdomains\&title=

Pending verification

If verification is needed, you will receive an email with instructions from Vercel. You will also see an alert on your team's domain page

Emoji and ASCII support

You will need to convert the domain to [punycode](https://www.punycoder.com) in order to add it to your project. For example, a user look

Unable to transfer-in a domain

[ICANN](https://www.icann.org/) forces domain registrars to wait **60 days**:

- between transfers
- between a new registration and a subsequent transfer

If you transfer before this time, the transfer will fail. Besides this restriction, some DNS providers may further restrict domain transf

Working with Apex domain

When you add an [apex domain](/docs/domains/working-with-domains#subdomains-wildcard-domains-and-apex-domains) (e.g. `example.com`) to yo

The main reason for that is the DNS [RFC1034](https://www.ietf.org/rfc/rfc1034.txt) (section 3.6.2) states that `If a CNAME RR is present

Does my domain IP address on Vercel resolve to a geographic region?

When you configure an apex domain (example.com) as a custom domain for your project on Vercel, Vercel will be give you an IP address to a

This is because Vercel uses [Anycast](https://en.wikipedia.org/wiki/Anycast) IP addresses, which are shared across all regions. That mean

Domain ownership errors

When you add a domain to your project, Vercel checks if it is already associated with a [Personal Account or Team](/docs/accounts). A dom

The following table shows errors that can be encountered when adding a domain to your project:

Error Text	Description
`This team has already registered this domain`	The domain you are trying to add is alrea
`You have already registered this domain`	The domain you are trying to add is alrea
`The domain mydomain.com is not available` or `Another Vercel account is using this domain`	This error indicates that the domain is a

Common SSL certificate issues

There are many reasons why a certificate may not be generated. As the first starting point, we recommend testing your domain with:

1. **[Let's Debug](https://letsdebug.net)**: Let's Debug is a diagnostic tool/website to help figure out why you might not be able to iss
2. **[DNSViz](https://dnsviz.net/)**: DNSViz is a tool suite for analysis and visualization of Domain Name System (DNS) behavior, includi

For non-wildcard domains, we use [HTTP-01](https://letsencrypt.org/docs/challenge-types/#http-01-challenge) challenge by default, which V

For wildcard domains, only [DNS-01](https://letsencrypt.org/docs/challenge-types/#dns-01-challenge) challenge is supported, which Vercel

Missing `CAA` records

Since we use Let's Encrypt for our automatic SSL certificates, you must add a `CAA` record with the value `0 issue "letsencrypt.org"` if

You can check if your domain currently has any `CAA` records by running the `dig -t CAA +noall +ans example.com` command on your terminal

For more information, see [Why is my domain not automatically generating an SSL certificate?](/kb/guide/domain-not-generating-ssl-certifi

Existing `_acme-challenge` record

An `_acme-challenge` record allows Let's Encrypt to verify the domain ownership using [DNS-01](https://letsencrypt.org/docs/challenge-typ

If the domain was previously hosted on a different provider, and if the `_acme-challenge` record resolves to something, please consider [

Rewriting or redirecting `/.well-known`

The [/.well-known](# "The /.well-known directory") path is reserved and cannot be redirected or rewritten. Only Enterprise teams can conf

---
title: "Working with DNS"
description: "Learn how DNS works in order to properly configure your domain."
last_updated: "2026-01-16T02:19:29.225Z"
source: "https://vercel.com/docs/domains/working-with-dns"

Working with DNS

DNS is the system used to connect domain names to IP addresses. When you make a request for a website, the browser performs a DNS query.

DNS records

There are a number of different types of DNS records that can be used together to create a DNS configuration. Some of the common informat

- **Host Name**: The hostname of `www`
- **IP Address or URL**: The IP address (or domain or in the case of a CNAME record), for example, `76.76.21.21` or `cname.vercel-dns-0.c
- **TTL (Time to live)**: The length of time the recursive server should keep a particular record in its cache. You should set this time
- **Record Type**: For example, `CNAME`. There are many different types of records, some of the most common are listed below.

To learn more about adding, verifying, and removing DNS records, see "[Managing DNS records](/docs/domains/managing-dns-records)".

Type	Description
A	This is used to translate domain names into IPv4 addresses. It is the most common type of DNS record.
AAAA	Similar to `A`, but this is used to translate domain names into IPv6 addresses. **IPv6 is not supported on Vercel**. See [IPv6
ALIAS	This is used to map a domain name to another domain name. It is similar to a `CNAME` record, but can only be used at the zone a
CAA	This is used to specify which certificate authorities are allowed to issue certificates for a domain. Vercel automatically adds
CNAME	This is used to specify that the domain name is an alias for another domain name. It cannot be used at the zone apex. See [Work
HTTPS	This is used to achieve a CNAME-like functionality, but can be used at the zone apex. This is designed specifically for HTTP pr
MX	This is used to specify the mail server for the domain.
NS	This is used to specify the authoritative name server for the domain.
TXT	This is used to provide information about the domain in text format. Commonly used for verification purposes.
SRV	This is used to specify the location of the service. The record contains priority, weight, port, target, and other information.

DNS propagation

When you're configuring or making changes to your DNS settings, you should be aware that it doesn't happen instantaneously. There's a who

As we described earlier, when you set a record, you normally set a **TTL** value, or **Time to Live**, on a DNS record. This value, set i

When you set the TTL value in your DNS record, you need to find the balance between serving your users the site quickly, and ensuring the

DNS best practices

When you are transferring an existing (in-use) domain to Vercel, it's a good practice to check the existing DNS record and its TTL before
Ideally, about 24 hours in advance of changes, you should shorten the DNS TTL to 60s. Once it's propagated, you can then change the DNS r

You can use tools such as <https://www.whatsmydns.net> to determine if your DNS settings have been fully propagated.

Troubleshooting

To learn more about common DNS issues, see the [troubleshooting](/docs/domains/troubleshooting#common-dns-issues) doc.

Related

---
title: "Adding & Configuring a Custom Domain"
description: "Learn how to add a custom domain to your Vercel project, verify it, and correctly set the DNS or Nameserver values."
last_updated: "2026-01-16T02:19:29.245Z"
source: "https://vercel.com/docs/domains/working-with-domains/add-a-domain"

Adding & Configuring a Custom Domain

Vercel provides all deployments with a `vercel.app` URL, which enables you to share Deployments with your Team for collaboration. However

You can manage all domain settings related to a project in the **Domains** section of the **Settings** tab of the project, regardless of

Hobby teams have a limit of 50 custom domains per project.

Add and configure domain

The following steps provide an overview of how to add and configure a custom domain in Vercel:

- ### Navigate to Domain Settings
  On the [dashboard](/dashboard), pick the project to which you would like to assign your domain.

  Once you have selected your project, click on the **Settings** tab and then select the **Domains** menu item:

- ### Add your domain
  From the **Domains** page, click the **Add Domain** button:

  Input the domain you wish to include in the project:

  If you add an apex domain (e.g. `example.com`) to the project, Vercel will prompt you to add the `www` subdomain prefix. For more infor

- ### Using wildcard domain
  You can also use your **custom domain** as a **wildcard domain** by prefixing it with `*.`.
  > **💡 Note:** If using your custom domain as a wildcard domain, you **must use the
  > nameservers method for verification**.
  To add a **wildcard domain**, use the prefix `*`, for example `*.acme.com`.

- ### Configure the domain
  Once you have added your custom domain, you will need to configure the DNS records of your domain with your registrar so it can be used
  - **If the domain is in use by another Vercel account**, you will need to [verify access to the domain](#verify-domain-access), with a
  - If you're using an [**Apex domain**](#apex-domains) (e.g. example.com), you will need to configure it with an **A** record
  - If you're using a [**Subdomain**](#subdomains) (e.g. docs.example.com), you will need to configure it with a **CNAME** record
  Both **apex domains** and **subdomains** can also be configured using the [**Nameservers**](#vercel-nameservers) method.
  #### Apex domains
  You can configure apex domains with an **A** record.
  #### Subdomains
  You can configure **subdomains** with a **CNAME** record. Each project has a unique CNAME record e.g. `d1d4fc829fe7bc7c.vercel-dns-017.
  #### Vercel Nameservers
  If you choose to use a wildcard domain Vercel's nameservers will be automatically enabled for you on saving the domain settings. You wi

- ### Verify domain access
  If the domain is in use by another Vercel account, you may be prompted to verify access to the domain. Note that this will not move the

Once the domain has been configured and Vercel has verified it, the status of the domain will be updated within the UI to confirm that it

> **💡 Note:** If a someone visits your domain with or without the "www" subdomain prefix,
> Vercel will attempt to redirect them to your domain. For more robust
> protection, you should explicitly add this domain and [redirect
> it](/docs/domains/deploying-and-redirecting#redirecting-domains).

---
title: "Assigning a custom domain to an environment"
description: "Learn how to add a custom domain to your Vercel project, verify it, and correctly set the DNS or Nameserver values."
last_updated: "2026-01-16T02:19:29.249Z"
source: "https://vercel.com/docs/domains/working-with-domains/add-a-domain-to-environment"

Assigning a custom domain to an environment

1. From the [dashboard](/dashboard), pick the project to which you would like to assign your domain and select the **Settings** tab.
2. Click on the **Environments** menu item.
3. Select the environment to which you would like to assign your domain. Users on Pro and Enterprise plans can create [custom environment
4. Once you've added your domain, you will need to configure the DNS records of your domain with your registrar so it can be used with yo
   - **If the domain is in use by another Vercel account**, you will need to [verify access to the domain](/docs/domains/add-a-domain#ver

- If you're using an [**Apex domain**](/docs/domains/add-a-domain#apex-domains) (e.g. example.com), you will need to configure it with an
- If you're using a [**Subdomain**](/docs/domains/add-a-domain#subdomains) (e.g. docs.example.com), you will need to configure it with a

---
title: "Assigning a domain to a Git branch"
description: "Learn how to assign a domain to a different Git branch with this guide."
last_updated: "2026-01-16T02:19:29.260Z"
source: "https://vercel.com/docs/domains/working-with-domains/assign-domain-to-a-git-branch"

Assigning a domain to a Git branch

Every commit pushed to the [Production Branch](/docs/git#production-branch) of your [connected Git repository](/docs/git) will be assigne

To automatically assign a domain to a different branch:

1. From the [dashboard](/dashboard), pick the project to which you would like to assign your domain and select the **Settings** tab.
2. Click on the **Domains** menu item.
3. Select the **Edit** dropdown item for the domain to which you would like to assign your branch.
4. Select **Preview** from the **Connect to an environment** section
5. In the **Git Branch** field, enter the branch name to which you would like to assign the domain:

Pro and Enterprise teams can also set branch tracking for their [custom environments](/docs/deployments/environments#custom-environments)

> **💡 Note:** If you prefer to do this using the Vercel REST API instead, you can use the
> ["Update a project
> domain"](/docs/rest-api/reference/endpoints/projects/update-a-project-domain)
> PATCH endpoint.

---
title: "Deploying & Redirecting Domains"
description: "Learn how to deploy your domains and set up domain redirects with this guide."
last_updated: "2026-01-16T02:19:29.289Z"
source: "https://vercel.com/docs/domains/working-with-domains/deploying-and-redirecting"

Deploying & Redirecting Domains

Deploying your Domain

Once the domain has been added to your project and configured, it is **automatically applied to your latest production deployment**.

> **💡 Note:** The first deployment of a new project will be marked as production and
> subsequently assigned with your custom domain automatically.

When you assign a custom domain to a project that's using [Git](/docs/git), each push (including merges) that you make to the [production

When you assign a domain to a *different* branch, you'll need to make a new deployment to the desired branch for the domain to resolve co

Reverts take effect immediately, assigning the **Custom Domain** to the deployment made prior to the point the revert is effective from.

Redirecting domains

You can add domain redirects from the **Domains** tab when more than one domain is present in the project. This provides a way to, for ex

> **💡 Note:** If a user visits your domain with or without the "www" subdomain prefix, we
> will attempt to redirect automatically. You might still want to add this
> redirect explicitly.

To add a redirect, navigate to the **Domains** tab within **Project Settings**, then select **Edit** on the domain you want to redirect f

Redirecting `www` domains

Adding an [apex domain](/docs/domains/working-with-domains#apex-domain) to a [Project](/docs/projects/overview) on Vercel will automatica

We recommend using the `www` subdomain as your primary domain, with a redirect from the non-`www` domain to it. This allows the [Vercel C

Some browsers like Google Chrome automatically hide the `www` subdomain from the address bar, so this redirect may not affect your URL ap

Choosing to redirect the `www` domain to the non-`www` also works but provides Vercel less control over incoming traffic. Alternatively, y

Additional technical information about Domain redirects

The DNS spec forbids using CNAME records on apex domains like `example.com`. They are, however, allowed for subdomains like `www.example.

Using CNAME instead of A records ensures that domains on Vercel are fast, reliable, and fault-tolerant. Unlike A records, CNAME records a

While we recommend using `www` as described above, Vercel maximizes the reliability and performance of your apex domain if you choose to

Programmatic redirects

You can also add redirects programmatically using frameworks and Vercel Functions. [Learn more](/docs/redirects).

---
title: "Working with domains"
description: "Learn how domains work and the options Vercel provides for managing them."
last_updated: "2026-01-16T02:19:29.358Z"
source: "https://vercel.com/docs/domains/working-with-domains"

Working with domains

You can [buy a domain through Vercel](#buying-a-domain-through-vercel) by going to the [Vercel.com domains page](https://vercel.com/domai

Buying a domain name

When you create a deployment on Vercel, we automatically assign it a domain based on your project name and ending in `.vercel.app`. Your

More often than not, you will want to assign a domain to a project that reflects its nature better. You can buy a domain name either [thr

Buying a domain through Vercel

When you buy a domain through Vercel, we configure and set the nameservers, which means you do not need to set any DNS records or make an

> **💡 Note:** For the ICANN registrant information:

Buying a domain through a third-party

When you buy a custom domain through a third-party, you can use the [add a custom domain](/docs/domains/add-a-domain) workflow to configu

Domain ownership and Project assignment

When you are using domains with Vercel, there are two areas of the dashboard that you may need to go to in order to configure them correc

- **Domain ownership**: Domains are owned by a specific team and can be accessed from the [**Domains** tab on your team's dashboard](http

- **Project assignment**: This is accessed by selecting the project that you wish to assign the domain to and navigating to **Settings >

> **💡 Note:** When you add a domain to Vercel for the first time, it will appear as an in your team's tab. If you add that domain (for

Subdomains, wildcard domains, and apex domains

Apex Domain

The **apex domain** is the root-level domain, such as `acme.com`. When you add an apex domain, Vercel will recommend that you add a [redi

Subdomain

A **subdomain** is a more specific part of that domain that can be assigned to a particular part of your site, for example, `blog.acme.co

Wildcard domain

You can also configure **wildcard domains**. Using a wildcard domain, such as `*.acme.com`, is a way to scale and customize your project

To add a wildcard domain, follow the steps in [Adding a domain](/docs/domains/add-a-domain#using-wildcard-domain).

Wildcard domains **must** be configured with the [nameservers method](/docs/domains/add-a-domain#vercel-nameservers). This is because in

Using email with domains

When you create a domain, you may want to also set up a way for users to contact you through an email address that is pointed at that dom

Because many domain providers do not offer a mail service, several third-party services specifically offer this type of functionality and

Troubleshooting

[Invalid domain configurations](/docs/domains/troubleshooting#misconfigured-domain-issues) are one of the most common types of domain iss

More resources

- [Domains overview: Learn the concepts behind how domains work](/docs/domains)
- [Learn how DNS works in order to properly configure your domain](/docs/domains/working-with-dns)
- [Learn about nameservers and the benefits Vercel nameservers provide](/docs/domains/working-with-nameservers)
- [Learn how Vercel uses SSL certificates to keep your site secure](/docs/domains/working-with-ssl)
- [Learn how to troubleshoot your domain on Vercel](/docs/domains/troubleshooting)
- [What is a Domain Name?](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/What_is_a_domain_name)

---
title: "Removing a Domain from a Project"
description: "Learn how to remove a domain from a Project and from your account completely with this guide."
last_updated: "2026-01-16T02:19:29.266Z"
source: "https://vercel.com/docs/domains/working-with-domains/remove-a-domain"

Removing a Domain from a Project

When you add a domain to any project, it will be connected to your account until you choose to delete it. This guide demonstrates how to

- ### Navigate to the Domains tab
  To remove a domain that is assigned to a project, navigate to the **Domains** tab from the **Project Overview** and click the **More Op

- ### Click remove button
  Once the **• • •** menu button has been clicked, you will be presented with further options. Click the **Delete** menu button to remove

- ### Remove domain from your account
  Optionally, if you wish to remove a domain from all Projects, as well as your Account, navigate to the **Domains** section of your dash
  > **💡 Note:** If the domain was purchased through Vercel, you must first wait for the domain
  > to expire before you can remove it from your account.

Using cURL

To remove a domain from a project using cURL, you can use the following command. To create an Authorization Bearer token, see the [access

```bash filename="cURL"
curl --request DELETE \
  --url https://api.vercel.com/v9/projects/<project-id-or-name>/domains/<domain-name> \
  --header "Authorization: Bearer $VERCEL_TOKEN"
```

---
title: "Managing Domain Renewals and Redemptions"
description: "Learn how to manage automatic and manual renewals for custom domains purchased through or registered with Vercel, and how t
last_updated: "2026-01-16T02:19:29.345Z"
source: "https://vercel.com/docs/domains/working-with-domains/renew-a-domain"

Managing Domain Renewals and Redemptions

Custom domains purchased through or registered with Vercel are [automatically renewed](#auto-renewal) by default with the option to [manu

You can see the expiration or [renewal date](#filter-on-renewal-status) of your Vercel-managed domains in the list of domains on the [**D

Auto renewal

To enable automatic renewal, follow these steps:

- ### Select the Domains tab
  You can choose to prevent the automatic renewal of a Domain from the [**Domains** tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fdo

- ### View the auto renewal status
  From the list of domains, find the domain you want to enable automatic renewal for. You can use the search bar or filter button to find
  You'll see the auto-renewal or expiry status of the domain in the domain's row.

- ### Toggle the auto renewal status
  Click on the hamburger menu icon to the right of the domain and toggle the Auto Renewal to on or off.

Auto renewal off

If auto renewal is off, Vercel will not try to re-register the Domain when it expires at the end of the registration period. You will not

If the Domain enters the redemption period, you can attempt to manually recover it by selecting **Renew** in the [Domains tab](https://ve

Vercel will send you three emails regarding the Domain before this happens. 24 and 14 days before the Domain is set to expire, you will b

Auto renewal on

> **💡 Note:** Vercel can only renew your domain if your payment method is valid at the time of renewal. If your card fails, the domain m

If auto renewal is on, Vercel will use the following process to renew the domain:

1. 60 days before expiration, Vercel will send you a warning email that the domain will expire and that we will try to renew it
2. 30 days before expiration, Vercel will try to renew the domain
3. Starting at 29 days before expiration, Vercel will check for any failed renewals and try to renew them again

Manual renewal

- ### Select the Domains Tab
  Navigate to the [**Domains** tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fdomains\&title=Go+to+Domains) on the **Vercel Dashboard

- ### Find your domain from the list
  From the list of domains, find the domain you want to renew. You can use the search bar or filter button to find it if you have many do
  You'll see the auto renewal or expiry status of the domain in the domain's row.

- ### Click the Renew button
  Click on the hamburger menu icon to the right of the domain and click the **Renew** button.
  > **💡 Note:** Your domain must be within 1 year of expiration to be eligible for renewal.

- ### Confirm your renewal

Domain redemptions

For expired domains with a redemption period (typically 30 days), you can now recover them directly in the dashboard:

A redemption fee will be applied, depending on the domain registry.

> **💡 Note:** Not all top-level domains (TLDs) support redemptions.

Filter on renewal status

You can filter your Vercel owned domains by their renewal status by clicking the filter icon in the top right of the Domains table:

Renewing third-party domains

Third-Party Domains (ones not purchased with or transferred into Vercel) are not subject to auto-renewal. Please refer to your Domain nam

---
title: "Transferring Domains to Another Team or Project"
description: "Domains can be transferred to another team or project within Vercel, or to and from a third-party registrar. Learn how to t
last_updated: "2026-01-16T02:19:29.315Z"
source: "https://vercel.com/docs/domains/working-with-domains/transfer-your-domain"

Transferring Domains to Another Team or Project

Transfer a domain to another Vercel user or Team

- ### Select the Domains tab
  You can move domains to another team using the [**Domains** tab of your team's dashboard](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2

- ### Select the domain
  Once on the **Domains** tab, select the context menu next to the domain you wish to move, and click **Move**. You can also use checkbox

- ### Select the team
  After selecting the domain(s) and clicking **Move**, you will be asked to confirm which profile or team you wish to move them to.

  When selecting the input field, you will be provided with a list of teams you belong to. If the profile or team you wish to move the do
  > **💡 Note:** When moving domains to another team or user, all existing project domains
  > associated with them will remain and not be moved to prevent service
  > disruption. However, any [custom aliases](/docs/cli/alias) that are not part
  > of project domains will be removed immediately.

- ### Confirm the change
  To confirm the change, select **Move**. The domains will be transferred to the new profile of team immediately.

Transferring domains between projects

You can use the Dashboard to remove a domain from a project and then re-add it to another. However, this could potentially end up with so

Transferring domains out of Vercel

- ### Verifying Transfer Eligibility
  Due to [ICANN rules](https://www.icann.org/resources/pages/text-2012-02-25-en#:~:text=Please%20note%20that%20you%20may,60%20days%20afte

  You can verify that your domain has been registered with Vercel for at least 60 days by visiting the team's [Domains Dashboard](https:/

- ### Select the tab
  For domains that are registered with Vercel, you can retrieve an authorization code for transferring out to another registrar from the

- ### Select the "Transfer out" option
  Once on the tab, click on the triple-dot menu button for the relevant domain. A menu-item button to transfer the domain out will be pr
  > **💡 Note:** If under a Team scope, only [Team Owners](/docs/rbac/access-roles#owner-role)
  > will see the menu-item button.

- ### Use the authorization code with the new registrar
  After clicking the menu-item button, a modal will open up with the authorization code required to transfer the domain. Use this authori

  If you encounter problems with the transfer code, ensure you've entered it correctly without typos or extra spaces. If the code seems c

Transfer a domain to Vercel

By transferring your domain into Vercel, you allow Vercel to manage the DNS records for the domain and can use it with any Projects liste

- ### Verifying Transfer Eligibility
  Due to [ICANN rules](https://www.icann.org/resources/pages/text-2012-02-25-en#:~:text=Please%20note%20that%20you%20may,60%20days%20afte

  If the domain has not been registered with the current registrar for at least 60 days, the domain transfer will fail.

  NOTE: To find further information on ICANN rules, visit the [ICANN website](https://www.icann.org/resources/pages/text-2012-02-25-en#:~

- ### Unlock the Domain
  Once you have verified your domain's eligibility to transfer, proceed with unlocking your domain in your registrar's domain settings. M

  The domain lock feature appears in different forms across registrars. Sign into the host where your domain is registered and look for a

- ### Obtain Authorization Code
  After unlocking the domain, you will need to obtain an authorization code. The code will be sent to the email address associated with y

- ### Transferring to Vercel
  When transferring a domain, you will have two options to choose from. Either using the Vercel Dashboard or Vercel CLI.

  Option 1: Using Vercel Dashboard

  After obtaining the authorization code, click on the Transfer in button in the Vercel Domains Dashboard and enter in your domain and re

  Option 2: Using Vercel CLI

  With Vercel CLI, you can run the following command from your terminal.
  ```bash
  vercel domains transfer-in [your-domain]
  ```
  You will be requested to provide an authorization code from your registrar after running this command. Once you get the authorization c
  > **💡 Note:** In a case where your domain cannot be transferred, check that it has been over
  > 60 days since the domain has been registered or previously transferred. If it
  > still does not work, contact your registrar.

- ### Configure domain
  Follow these steps to ensure that there is no downtime while the domain is transferred to Vercel.

  Pre-generate SSL certificates

  If you are migrating a deployment to Vercel, require zero downtime, and aren't using Vercel's nameservers, you can pre-generate and iss
  If you have enabled Vercel DNS by pointing your domain's nameserver to Vercel and have generated an SSL certificate, you can ignore thi

  Follow the [detailed guide](/docs/domains/pre-generating-ssl-certs) to set up SSL certificates before finalizing the domain transfer.

  Set DNS records in your registrar

  Once you have pre-generated the SSL certificates, you need to add the new TXT records to your DNS records in your domain registrar dash

- ### Deploy the domain
  You can deploy your app with Vercel once the domain has been successfully added to your account.

  By setting a production domain from your projects' Domains dashboard, you will be able to use the following command with Vercel CLI:
  ```bash
  vercel --prod
  ```
  This command will deploy your project and make it accessible at the production domain that you have setup.

---
title: "Viewing & Searching Domains"
description: "Learn how to view and search all registered domains that are assigned to Vercel Projects through the Vercel dashboard."
last_updated: "2026-01-16T02:19:29.297Z"
source: "https://vercel.com/docs/domains/working-with-domains/view-and-search-domains"

Viewing & Searching Domains

Viewing domains

To view all your registered domains, go to the **Domains** tab in your Vercel dashboard.

The domains list will show you all domains that are currently active on your account, and display the following information:

- **Domain** - The domain name
- **Registrar and status** - The domain registrar (Vercel or Third Party). If the registrar is Vercel, you will see the renewal or expiry
- **Creator** - The person who created the domain, indicated by their avatar and username and include the creation date

Searching domains

You can search for a specific domain by using the search bar above the domains list.

It is not possible to search a multi-level wildcard subdomain. It is only possible to search a subdomain at one level down.

---
title: "Working with nameservers"
description: "Learn about nameservers and the benefits Vercel nameservers provide."
last_updated: "2026-01-16T02:19:29.321Z"
source: "https://vercel.com/docs/domains/working-with-nameservers"

Working with nameservers

> **⚠️ Warning:** Before moving your domain to use Vercel's nameservers, you should ensure that
> you own the domain listed on the [Domains](/domains) page of your account."

Nameservers are the actual servers on the network that are responsible for resolving domain names to the IP addresses where your site is

- `ns1.vercel-dns.com`
- `ns2.vercel-dns.com`

When you purchase your domain through Vercel, we can set all the DNS records, including nameserver records, that tell anyone looking for

Benefits of using Vercel nameservers

- **Automatic DNS Records**: Domains with nameservers pointed to Vercel don't need explicit DNS records created for the apex domain or fi
- **Wildcard Domains**: When using Vercel's nameservers you can add [wildcard domains](/docs/domains/working-with-domains#subdomains-wild
- **Custom nameservers**: For domains registered with Vercel, you can add custom nameservers to your Vercel-hosted domain directly from t

For domains that are not registered with Vercel, you can change the nameservers directly from the domain registrar's dashboard. For more

> **💡 Note:** Before using Vercel's nameservers, you should ensure that you own the domain.

Troubleshooting

To learn more about common nameserver issues, see the [troubleshooting](/docs/domains/troubleshooting#common-nameserver-issues) doc.

Related

---
title: "Working with SSL Certificates"
description: "Learn how Vercel uses SSL certification to keep your site secure."
last_updated: "2026-01-16T02:19:29.362Z"
source: "https://vercel.com/docs/domains/working-with-ssl"

Working with SSL Certificates

An SSL certificate enables encrypted communication between user's browser and your web server to be encrypted. The certificate is install

SSL certificates are issued from a [certificate authority (CA)](# "certificate authority (CA)") for each domain. While it is possible to

Vercel uses LetsEncrypt for certificates. For all non-wildcard domains, we use the [HTTP-01 challenge method](https://letsencrypt.org/doc
For wildcard requests, we use the [DNS-01 challenge method](https://letsencrypt.org/docs/challenge-types/#dns-01-challenge). This is why

Issuing a certificate happens in the following way:

1. Vercel asks LetsEncrypt for a certificate for that domain and asks how it can prove control of the domain
2. Let's Encrypt reviews the domain and issues Vercel with a [challenge](https://letsencrypt.org/docs/challenge-types/) in order to autho
3. Vercel creates that file with the code on the HTTP-01 or DNS-01 validation path and tells LetsEncrypt it's done
4. LetsEncrypt then check to see if the file is there and if they can see the file, they send us the certificate
5. Vercel then adds the certificate to our infrastructure and it then starts working on HTTPS

For information about when SSL certificate renewals happen, see [When is the SSL Certificate on my Vercel Domain renewed?](/kb/guide/renew

The [/.well-known](# "The /.well-known directory") path is reserved and cannot be redirected or rewritten. Only
Enterprise teams can configure custom SSL. [Contact sales](/contact/sales) to
learn more.

Troubleshooting

To learn more about common SSL issues, see the [troubleshooting](/docs/domains/troubleshooting#common-ssl-certificate-issues) doc.

Related

---
title: "Draft Mode"
description: "Vercel"
last_updated: "2026-01-16T02:19:29.370Z"
source: "https://vercel.com/docs/draft-mode"

Draft Mode

Draft Mode lets you view your unpublished headless CMS content on your website rendered with all the normal styling and layout that you w

Both [Next.js](/docs/frameworks/nextjs#draft-mode) and [SvelteKit](/docs/frameworks/sveltekit#draft-mode) support Draft Mode. Any framewo

> **💡 Note:** Draft Mode was called Preview Mode before the release of Next.js
> [13.4](https://nextjs.org/blog/next-13-4). The name was changed to avoid
> confusion with [preview
> deployments](/docs/deployments/environments#preview-environment-pre-production),
> which is a different product.

You can use Draft Mode if you:

1. Use [Incremental Static Regeneration (ISR)](/docs/incremental-static-regeneration) to fetch and render data from a headless CMS
2. Want to view your unpublished headless CMS content on your site without rebuilding your pages when you make changes
3. Want to protect your unpublished content from being viewed publicly

How Draft Mode works

Draft Mode allows you to bypass [ISR](/docs/incremental-static-regeneration) caching to fetch the latest CMS content at request time. Thi

The process works like this:

1. Each ISR route has a `bypassToken` configuration option, which is assigned a generated, cryptographically-secure value at build time
2. When someone visits an ISR route with a `bypassToken` configured, the page will check for a `__prerender_bypass` cookie
3. If the `__prerender_bypass` cookie exists and has the same value as the `bypassToken` your project is using, the visitor will view the

> **💡 Note:** Only team members will be able to view pages in Draft Mode.

Getting started

> For \['nextjs-app', 'nextjs']:

To use Draft Mode with Next.js on Vercel, you must:

1. [Enable ISR](/docs/incremental-static-regeneration) on pages that fetch content. Using ISR is required on pages that you want to view
2. Add code to your ISR pages to detect when Draft Mode is enabled and render the draft content
3. Toggle Draft Mode in the Vercel Toolbar by selecting Draft Mode in [the toolbar menu](/docs/vercel-toolbar#using-the-toolbar-menu) to

   ```tsx filename="app/page.tsx" framework=nextjs-app
   import { draftMode } from 'next/headers';

   async function getContent() {
     const { isEnabled } = await draftMode();

     const contentUrl = isEnabled
       ? 'https://draft.example.com'
   ```

```
      : 'https://production.example.com';

    // This line enables ISR, required for draft mode
    const res = await fetch(contentUrl, { next: { revalidate: 120 } });

    return res.json();
  }

  export default async function Page() {
    const { title, desc } = await getContent();

    return (
      <main>
        <h1>{title}</h1>
        <p>{desc}</p>
      </main>
    );
  }
  ```

  ```jsx filename="app/page.jsx" framework=nextjs-app
  import { draftMode } from 'next/headers';

  async function getContent() {
    const { isEnabled } = await draftMode();

    const contentUrl = isEnabled
      ? 'https://draft.example.com'
      : 'https://production.example.com';

    // This line enables ISR, required for draft mode
    const res = await fetch(contentUrl, { next: { revalidate: 120 } });

    return res.json();
  }

  export default async function Page() {
    const { title, desc } = await getContent();

    return (
      <main>
        <h1>{title}</h1>
        <p>{desc}</p>
      </main>
    );
  }
  ```

  ```tsx filename="pages/example.tsx" framework=nextjs
  export async function getStaticProps(context) {
    const url = context.draftMode
      ? 'https://draft.example.com'
      : 'https://production.example.com';
    const res = await fetch(url);
    return {
      props: {
        posts: await res.json(),
      },
      revalidate: 120,
    };
  }
  ```

  ```jsx filename="pages/example.jsx" framework=nextjs
  export async function getStaticProps(context) {
    const url = context.draftMode
      ? 'https://draft.example.com'
      : 'https://production.example.com';
    const res = await fetch(url);
    return {
      props: {
        posts: await res.json(),
      },
      revalidate: 120,
    };
  }
  ```

See the Next.js docs to learn how to use Draft Mode with self-hosted Next.js projects:

- [App Router](https://nextjs.org/docs/app/building-your-application/configuring/draft-mode)
- [Pages Router](https://nextjs.org/docs/pages/building-your-application/configuring/draft-mode)

> For \['sveltekit']:

To use a SvelteKit route in Draft Mode, you must:

1. Export a `config` object [that enables Incremental Static Regeneration](https://kit.svelte.dev/docs/adapter-vercel#incremental-static-

```ts filename="blog/[slug]/+page.server.ts" framework=sveltekit
import { BYPASS_TOKEN } from '$env/static/private';

export const config = {
  isr: {
    // Random token that can be provided to bypass the cached version of the page with a __prerender_bypass=<token> cookie. Allows render
    bypassToken: BYPASS_TOKEN,

    // Expiration time (in seconds) before the cached asset will be re-generated by invoking the Vercel Function.
    // Setting the value to `false` means it will never expire.
    expiration: 60,
```

  },
};
```

```js filename="blog/[slug]/+page.server.js" framework=sveltekit
import { BYPASS_TOKEN } from '$env/static/private';

export const config = {
  isr: {
    // Random token that can be provided to bypass the cached version of the page with a __prerender_bypass=<token> cookie. Allows render
    bypassToken: BYPASS_TOKEN,

    // Expiration time (in seconds) before the cached asset will be re-generated by invoking the Vercel Function.
    // Setting the value to `false` means it will never expire.
    expiration: 60,
  },
};
```

2. Send a `__prerender_bypass` cookie with the same value as `bypassToken` in your config.

To render the draft content, SvelteKit will check for `__prerender_bypass`. If its value matches the value of `bypassToken`, it will rend

Once implemented, team members can access Draft Mode from the Vercel Toolbar by selecting the eye icon . Once selected, the toolbar will

Sharing drafts

To share a draft URL, it must have the `?__vercel_draft=1` query parameter. For example:

```bash
https://my-site.com/blog/post-01?__vercel_draft=1
```

Viewers outside your Vercel team cannot enable Draft Mode or see your draft content, even with a draft URL.

---
title: "Working with Drains"
description: "Drains collect logs, traces, speed insights, and analytics from your applications. Forward observability data to custom end
last_updated: "2026-01-16T02:19:29.379Z"
source: "https://vercel.com/docs/drains"

Working with Drains

Drains let you forward observability data from your applications to external services for debugging, performance optimization, analysis,

- Store observability data persistently in your preferred external services
- Process large volumes of telemetry data using your own tools
- Set up alerts based on application behavior patterns
- Build custom metrics and dashboards from your data

Getting started with Drains

You can add Drains in two ways:

- Custom endpoints: [Configure](/docs/drains/using-drains#configuring-drains) any data type to send to a [custom HTTP endpoint](/docs/dra
- Native integrations: [Configure](/docs/drains/using-drains#configuring-drains) logs and trace data types to send to popular services li

Learn how to [manage your active drains](/docs/drains/using-drains#managing-your-active-drains).

Data types

You can drain four types of data:

- **Logs**: Runtime, build, and static logs from your deployments (supports custom endpoints and native integrations)
- **Traces**: Distributed tracing data in OpenTelemetry format (supports custom endpoints and native integrations)
- **Speed Insights**: Performance metrics and web vitals (custom endpoints only)
- **Web Analytics**: Page views and custom events (custom endpoints only)

Data type references

Each drain data type has specific formats, fields, and schemas. Review the reference documentation for [logs](/docs/drains/reference/logs

Security

You can secure your drains by checking for valid signatures and hiding IP addresses. Learn how to [add security to your drains](/docs/dra

Usage and pricing

Drains are available to all users on the [Pro](/docs/plans/pro-plan) and [Enterprise](/docs/plans/enterprise) plans. If you are on the [H

Drains usage is billed based on the pricing table below. Pricing is the same regardless of data type:

See [Optimizing Drains](/docs/pricing/observability#optimizing-drains-usage) for information on how to manage costs associated with Drain

More resources

For more information on Drains, check out the following resources:

- [Configure Drains](/docs/drains/using-drains)
- [Log Drains reference](/docs/drains/reference/logs)
- [Traces reference](/docs/drains/reference/traces)
- [Speed Insights reference](/docs/drains/reference/speed-insights)
- [Analytics reference](/docs/drains/reference/analytics)

---
title: "Web Analytics Drains Reference"
description: "Learn about Web Analytics Drains - data formats and custom events configuration."
```

last_updated: "2026-01-16T02:19:29.388Z"
source: "https://vercel.com/docs/drains/reference/analytics"

Web Analytics Drains Reference

If a Web Analytics Drains has been configured, Vercel will send page views and custom events from your applications to external endpoints

Web Analytics Schema

The following table describes the possible fields that are sent via Web Analytics Drains:

Name	Type	Description	Example
`schema`	string	Schema version identifier	`vercel.analytics.v1`
`eventType`	string	Type of analytics event	`pageview` or `event`
`eventName`	string	Name of the custom event	`button_click`
`eventData`	string	Additional data associated with the event	`{"button": "signup"}`
`timestamp`	number	Unix timestamp when the event was recorded	1694723400000
`projectId`	string	Identifier for the Vercel project	`Qmc52npNy86S8VV4Mt8a8dP1LEkRNbgosW3pBCQytkcgf2`
`ownerId`	string	Identifier for the project owner	`team_nLlpyC6REAqxydlFKbrMDlud`
`dataSourceName`	string	Name of the data source	`web-analytics`
`sessionId`	number	Unique session identifier	12345
`deviceId`	number	Unique device identifier	67890
`origin`	string	Origin URL where the event was recorded	`https://example.com`
`path`	string	URL path where the event was recorded	`/dashboard`
`referrer`	string	Referrer URL	`https://google.com`
`queryParams`	string	Query parameters from the URL	`utm_source=google&utm_medium=cpc`
`route`	string	Route pattern for the page	`/dashboard/[id]`
`country`	string	Country code of the user	`US`
`region`	string	Region code of the user	`CA`
`city`	string	City of the user	`San Francisco`
`osName`	string	Operating system name	`macOS`
`osVersion`	string	Operating system version	`13.4`
`clientName`	string	Client browser name	`Chrome`
`clientType`	string	Type of client	`browser`
`clientVersion`	string	Client browser version	`114.0.5735.90`
`deviceType`	string	Type of device	`desktop`
`deviceBrand`	string	Device brand	`Apple`
`deviceModel`	string	Device model	`MacBook Pro`
`browserEngine`	string	Browser engine name	`Blink`
`browserEngineVersion`	string	Browser engine version	`114.0.5735.90`
`sdkVersion`	string	SDK version used to track events	`2.1.0`
`sdkName`	string	SDK name used to track events	`@vercel/analytics`
`sdkVersionFull`	string	Full SDK version string	`2.1.0-beta.1`
`vercelEnvironment`	string	Vercel environment	`production`
`vercelUrl`	string	Vercel deployment URL	`*.vercel.app`
`flags`	string	Feature flags information	`{"feature_a": true}`
`deployment`	string	Identifier for the Vercel deployment	`dpl_2YZzo1cJAjijSf1hwDFK5ayu2Pid`

Format

Vercel supports the following formats for Web Analytics Drains, which you can configure when [setting the Drain destination](/docs/drains

JSON

Vercel sends Web Analytics data as JSON arrays containing event objects:

```json
[
  { "schema": "vercel.analytics.v1", "eventType": "pageview", "timestamp": 1694723400000, "projectId": "Qmc52npNy86S8VV4Mt8a8dP1LEkRNbgosW
  { "schema": "vercel.analytics.v1", "eventType": "event", "eventName": "button_click", "eventData": "{\"button\": \"signup\"}", "timestam
]
```

NDJSON

Vercel sends Web Analytics data as newline-delimited JSON objects:

```json
{"schema": "vercel.analytics.v1","eventType": "pageview","timestamp": 1694723400000,"projectId": "Qmc52npNy86S8VV4Mt8a8dP1LEkRNbgosW3pBCQ
{"schema": "vercel.analytics.v1","eventType": "event","eventName": "button_click","eventData": "{\"button\": \"signup\"}","timestamp": 16
```

Sampling Rate

When you configure a Web Analytics Drain in the Vercel UI, you can set the sampling rate to control the volume of data sent. This helps m

More resources

For more information on Web Analytics Drains and how to use them, refer to the following resources:

- [Drains overview](/docs/drains)
- [Configure Drains](/docs/drains/using-drains)

---
title: "Log Drains Reference"
description: "Learn about Log Drains - data formats, sources, environments, and security configuration."
last_updated: "2026-01-16T02:19:29.491Z"
source: "https://vercel.com/docs/drains/reference/logs"

Log Drains Reference

Log Drains forward logs from your deployments to external endpoints for storage and analysis. You can configure Log Drains in two ways:

- **[Custom endpoint](/docs/drains/using-drains#custom-endpoint)**: Send logs to any HTTP endpoint you configure
- **[Native integration](/docs/drains/using-drains#native-integrations)**: Use integrations from the Vercel Marketplace like [Dash0](http

Vercel sends logs to endpoints over HTTPS every time your deployments generate logs. Multiple logs may be batched into a single request w

Logs Schema

The following table describes the possible fields that are sent via Log Drains:

Name	Type	Required	Description	Example
`id`	string	Yes	Unique identifier for the log entry	`15738171873303
`deploymentId`	string	Yes	Identifier for the Vercel deployment	`dpl_233NRGRjVZ
`source`	enum	Yes	Origin of the log	`build`, `edge`
`host`	string	Yes	Hostname of the request	`test.vercel.ap
`timestamp`	number	Yes	Unix timestamp when the log was generated	1573817187330
`projectId`	string	Yes	Identifier for the Vercel project	`gdufoJxB6b9b1f
`level`	enum	Yes	Log severity level	`info`, `warnin
`message`	string	No	Log message content (may be truncated if over 256 KB)	`Adding custome
`buildId`	string	No	Identifier for the Vercel build	`bld_cotnkcr76`
`entrypoint`	string	No	Entrypoint for the request	`api/index.js`
`destination`	string	No	Origin of the external content	`https://vitals
`path`	string	No	Function or dynamic path of the request	`/dynamic/[rout
`type`	string	No	Log output type	`command`, `std
`statusCode`	number	No	HTTP status code of the request	200 (`-1` means
`requestId`	string	No	Identifier of the request	`643af4e3-975a-
`environment`	enum	No	Deployment environment	`production` or
`branch`	string	No	Git branch name	`main`
`ja3Digest`	string	No	JA3 fingerprint digest	`769c83e5b...`
`ja4Digest`	string	No	JA4 fingerprint digest	`t13d1516h2...`
`edgeType`	enum	No	Type of edge runtime	`edge-function`
`projectName`	string	No	Name of the Vercel project	`my-app`
`executionRegion`	string	No	Region where the request is executed	`sfo1`
`traceId`	string	No	Trace identifier for distributed tracing	`1b02cd14bb8642
`spanId`	string	No	Span identifier for distributed tracing	`f24e8631bd11fa
`trace.id`	string	No	Trace	`1b02cd14bb8642
`span.id`	string	No	Span	`f24e8631bd11fa
`proxy`	object	No	Contains information about proxy requests	See proxy field
`proxy.timestamp`	number	Yes*	Unix timestamp when the proxy request was made	1573817250172
`proxy.method`	string	Yes*	HTTP method of the request	`GET`
`proxy.host`	string	Yes*	Hostname of the request	`test.vercel.ap
`proxy.path`	string	Yes*	Request path with query parameters	`/dynamic/some-
`proxy.userAgent`	array	Yes*	User agent strings of the request	`["Mozilla/5.0.
`proxy.region`	string	Yes*	Region where the request is processed	`sfo1`
`proxy.referer`	string	No	Referer of the request	`*.vercel.app`
`proxy.statusCode`	number	No	HTTP status code of the proxy request	200 (`-1` means
`proxy.clientIp`	string	No	Client IP address	`120.75.16.101`
`proxy.scheme`	string	No	Protocol of the request	`https`
`proxy.responseByteSize`	number	No	Size of the response in bytes	1024
`proxy.cacheId`	string	No	Original request ID when request is served from cache	`pdx1::v8g4b-17
`proxy.pathType`	string	No	How the request was served based on its path and project configuration	`func`, `preren
`proxy.pathTypeVariant`	string	No	Variant of the path type	`api`
`proxy.vercelId`	string	No	Vercel-specific identifier	`sfo1::abc123`
`proxy.vercelCache`	enum	No	Cache status sent to the browser	`MISS`, `HIT`,
`proxy.lambdaRegion`	string	No	Region where lambda function executed	`sfo1`
`proxy.wafAction`	enum	No	Action taken by firewall rules	`log`, `challen
`proxy.wafRuleId`	string	No	ID of the firewall rule that matched	`rule_gAHz8jtSB

*Required when `proxy` object is present

Format

Vercel supports the following formats for Log Drains. You can configure the format when [configuring the Drain destination](/docs/drains/

JSON

Vercel sends logs as JSON arrays containing log objects:

```json
{ "id": "157381718733037706171730000O", "deploymentId": "dpl_233NRGRjVZX1caZrXWtz5g1TAksD", "source": "build", "host": "my-app-abc123.ver
{ "id": "157381725028325465109720207O", "deploymentId": "dpl_233NRGRjVZX1caZrXWtz5g1TAksD", "source": "lambda", "host": "my-app-abc123.ve
```

NDJSON

Vercel sends logs as newline-delimited JSON objects:

```json
{"id": "157381718733037706171730000O","deploymentId": "dpl_233NRGRjVZX1caZrXWtz5g1TAksD","source": "build","host": "my-app-abc123.vercel.
{"id": "157381725028325465109720207O","deploymentId": "dpl_233NRGRjVZX1caZrXWtz5g1TAksD","source": "lambda","host": "my-app-abc123.vercel
```

Log Sources

When you configure a Log Drain, select which sources to collect in **Additional configuration for logs**:

value	Details
`static`	Requests to static assets like HTML and CSS files
`lambda`	Output from Vercel Functions like [API Routes](/docs/functions)
`edge`	Output from Vercel Functions using Edge runtime
`build`	Output from the [Build Step](/docs/deployments/configure-a-build)
`external`	External [rewrites](/docs/project-configuration#rewrites) to a different domain. Includes [cached external rewrites](/docs
`firewall`	Outputs log data from requests denied by [Vercel Firewall](/docs/vercel-firewall) rules
`redirect`	Requests that are redirected by [redirect rules](/docs/project-configuration#redirects)

Log Environments

Use the same panel to choose which environments send logs to your drain:

value	Details
`production`	Logs from production deployments with assigned domain(s)

| `preview` | Logs from deployments accessed through the [generated deployment URL](/docs/deployments/generated-urls) |

Sampling Rate

Sampling rules let you control how much log data each drain receives. Use them to send the right volume of data for observability and cos

1. If no rules exist, click **Add sampling rule**.
2. Choose the environment you want to sample from.
3. Set the sampling percentage.
4. (Optional) Specify a request path prefix. Leave it blank to apply the rule to every path.

Example workflows:

- Launch-day monitoring: sample **100%** of production traffic when you launch a new feature, then decrease to **10%** once traffic stabi
- Static coverage: always collect **5%** from `/docs` so you can spot regressions on a static documentation site.

Rules run from top to bottom. Requests that match a rule use that rule's sampling rate, and any other requests are dropped. If you do not

More resources

For more information on Log Drains and how to use them, check out the following resources:

- [Drains overview](/docs/drains)
- [Configure Drains](/docs/drains/using-drains)

---
title: "Speed Insights Drains Reference"
description: "Learn about Speed Insights Drains - data formats and performance metrics configuration."
last_updated: "2026-01-16T02:19:29.409Z"
source: "https://vercel.com/docs/drains/reference/speed-insights"

Speed Insights Drains Reference

Speed Insights Drains send performance metrics and web vitals from your applications to external endpoints for storage and analysis. To e

Vercel sends Speed Insights data to endpoint URLs over HTTPS when your application collects performance metrics.

Speed Insights Schema

The following table describes the possible fields that are sent via Speed Insights Drains:

Name	Type	Description	Example
`schema`	string	Schema version identifier	`vercel.speed_insights.v1`
`timestamp`	string	ISO timestamp when the metric was collected	`2023-09-14T15:30:00.000Z`
`projectId`	string	Identifier for the Vercel project	`Qmc52npNy86S8VV4Mt8a8dP1LEkRNbgosW3pBCQytkcgf2`
`ownerId`	string	Identifier for the project owner	`team_nLlpyC6REAqxydlFKbrMDlud`
`deviceId`	number	Unique device identifier	12345
`metricType`	string	Type of performance metric	`CLS`, `LCP`, `FID`, `FCP`, `TTFB`, `INP`
`value`	number	Metric value	0.1
`origin`	string	Origin URL where the metric was collected	`https://example.com`
`path`	string	URL path where the metric was collected	`/dashboard`
`route`	string	Route pattern for the page	`/dashboard/[id]`
`country`	string	Country code of the user	`US`
`region`	string	Region code of the user	`CA`
`city`	string	City of the user	`San Francisco`
`osName`	string	Operating system name	`macOS`
`osVersion`	string	Operating system version	`13.4`
`clientName`	string	Client browser name	`Chrome`
`clientType`	string	Type of client	`browser`
`clientVersion`	string	Client browser version	`114.0.5735.90`
`deviceType`	string	Type of device	`desktop`
`deviceBrand`	string	Device brand	`Apple`
`connectionSpeed`	string	Network connection speed	`4g`
`browserEngine`	string	Browser engine name	`Blink`
`browserEngineVersion`	string	Browser engine version	`114.0.5735.90`
`scriptVersion`	string	Speed Insights script version	`1.0.0`
`sdkVersion`	string	SDK version used to collect metrics	`2.1.0`
`sdkName`	string	SDK name used to collect metrics	`@vercel/speed-insights`
`vercelEnvironment`	string	Vercel environment	`production`
`vercelUrl`	string	Vercel deployment URL	`*.vercel.app`
`deploymentId`	string	Identifier for the Vercel deployment	`dpl_2YZzo1cJAjijSf1hwDFK5ayu2Pid`
`attribution`	string	Attribution information for the metric	`attribution-data`

Format

Vercel supports the following formats for Speed Insights Drains. You can configure the format when [configuring the Drain destination](/d

JSON

Vercel sends Speed Insights data as JSON arrays containing metric objects:

```json
[
  { "schema": "vercel.speed_insights.v1", "timestamp": "2023-09-14T15:30:00.000Z", "projectId": "Qmc52npNy86S8VV4Mt8a8dP1LEkRNbgosW3pBCQy
  { "schema": "vercel.speed_insights.v1", "timestamp": "2023-09-14T15:30:05.000Z", "projectId": "Qmc52npNy86S8VV4Mt8a8dP1LEkRNbgosW3pBCQy
]
```

NDJSON

Vercel sends Speed Insights data as newline-delimited JSON objects:

```json
{"schema": "vercel.speed_insights.v1","timestamp": "2023-09-14T15:30:00.000Z","projectId": "Qmc52npNy86S8VV4Mt8a8dP1LEkRNbgosW3pBCQytkcgf
{"schema": "vercel.speed_insights.v1","timestamp": "2023-09-14T15:30:05.000Z","projectId": "Qmc52npNy86S8VV4Mt8a8dP1LEkRNbgosW3pBCQytkcgf
```

Sampling Rate

When you configure a Speed Insights Drain in the Vercel UI, you can set the sampling rate to control the volume of data sent. This helps r

More resources

For more information on Speed Insights Drains and how to use them, check out the following resources:

- [Drains overview](/docs/drains)
- [Configure Drains](/docs/drains/using-drains)

---
title: "Trace Drains Reference"
description: "Learn about Trace Drains - OpenTelemetry-compliant distributed tracing data formats and configuration."
last_updated: "2026-01-16T02:19:29.496Z"
source: "https://vercel.com/docs/drains/reference/traces"

Trace Drains Reference

Trace Drains forward distributed tracing data from your deployments to external endpoints for storage and analysis. You can configure Tra

- **[Custom endpoint](/docs/drains/using-drains#custom-endpoint)**: Send traces to any HTTP endpoint you configure
- **[Native integration](/docs/drains/using-drains#native-integrations)**: Use integrations from the Vercel Marketplace like [Braintrust]

Vercel sends traces to endpoints over HTTPS following the [OpenTelemetry Protocol (OTLP)](https://opentelemetry.io/docs/concepts/signals/

Traces Schema

Trace Drains follow the [OpenTelemetry traces specification](https://opentelemetry.io/docs/concepts/signals/traces/). Vercel automaticall

Name	Type	Description	Example
`vercel.projectId`	string	Identifier for the Vercel project	`"Qmc52npNy86S8VV4Mt8a8dP1LEkRNbgosW3pBCQytkcgf2"`
`vercel.deploymentId`	string	Identifier for the Vercel deployment	`"dpl_2YZzo1cJAjijSf1hwDFK5ayu2Pid"`

Format

Vercel supports the following formats for Trace Drains. You can configure the format when [configuring the Drain destination](/docs/drain

JSON

Vercel sends traces as JSON objects following the OpenTelemetry specification:

```json
{ "resourceSpans": [{ "resource": { "attributes": [{ "key": "service.name", "value": { "stringValue": "vercel-function" } }] }, "scopeSpa
```

Protobuf

Vercel sends traces in binary protobuf format following the OTLP/HTTP specification. This format is more efficient for high-volume trace

Sampling Rate

Sampling rules control how much trace data each drain forwards so you can manage observability depth and spend. Add sampling rules to def

1. If no rules exist, click **Add sampling rule**.
2. Choose the environment you want to sample from.
3. Set the sampling percentage.
4. (Optional) Specify a request path prefix. Leave it blank to apply the rule to every path.

Example workflows:

- Launch-day monitoring: sample **100%** of production traffic when you launch a new feature, then decrease to **10%** once traffic stabi
- Static coverage: always collect **5%** from `/docs` so you can spot regressions on a static documentation site.

Rules run from top to bottom. Requests that match a rule use that rule's sampling rate, and any other requests are dropped. If you do not

Limitations

Custom spans from functions using the [Edge runtime](/docs/functions/runtimes/edge) are not forwarded via the Trace Drain.

More resources

For more information on Trace Drains and how to use them, check out the following resources:

- [Drains overview](/docs/drains)
- [Configure Drains](/docs/drains/using-drains)
- [OpenTelemetry traces specification](https://opentelemetry.io/docs/concepts/signals/traces/)

---
title: "Drains Security"
description: "Learn how to secure your Drains endpoints with authentication and signature verification."
last_updated: "2026-01-16T02:19:29.421Z"
source: "https://vercel.com/docs/drains/security"

Drains Security

All Drains support transport-level encryption using HTTPS protocol.

When your server starts receiving payloads, a third party could send data to your server if it knows the URL. Therefore, you should verif

Secure Drains

Vercel sends an `x-vercel-signature` header with each drain, which is a hash of the payload body created using your Drain signature secre

To verify the request is coming from Vercel, you can generate the hash and compare it with the header value as shown below:

```js filename="server.js" framework=nextjs-app
import crypto from 'crypto';

export async function POST(request) {
  // Store the signature secret in environment variables
  const signatureSecret = '<Drain signature secret>';

  const rawBody = await request.text();
  const rawBodyBuffer = Buffer.from(rawBody, 'utf-8');
  const bodySignature = sha1(rawBodyBuffer, signatureSecret);

  if (bodySignature !== request.headers.get('x-vercel-signature')) {
    return Response.json(
      {
        code: 'invalid_signature',
        error: "signature didn't match",
      },
      { status: 403 },
    );
  }

  console.log(rawBody);

  return Response.json({ success: true });
}

function sha1(data, secret) {
  return crypto.createHmac('sha1', secret).update(data).digest('hex');
}
```

```ts filename="server.ts" framework=nextjs-app
import crypto from 'crypto';

export async function POST(request: Request) {
  // Store the signature secret in environment variables
  const signatureSecret = '<Drain signature secret>';

  const rawBody = await request.text();
  const rawBodyBuffer = Buffer.from(rawBody, 'utf-8');
  const bodySignature = sha1(rawBodyBuffer, signatureSecret);

  if (bodySignature !== request.headers.get('x-vercel-signature')) {
    return Response.json(
      {
        code: 'invalid_signature',
        error: "signature didn't match",
      },
      { status: 403 },
    );
  }

  console.log(rawBody);

  return Response.json({ success: true });
}

function sha1(data: Buffer, secret: string): string {
  return crypto.createHmac('sha1', secret).update(data).digest('hex');
}
```

```js filename="server.js" framework=nextjs
import crypto from 'crypto';
import getRawBody from 'raw-body';

export default async function handler(
  request,
  response,
) {
  if (request.method !== 'POST') {
    return response.status(405).json({ error: 'Method not allowed' });
  }

  // Store the signature secret in environment variables
  const signatureSecret = '<Drain signature secret>';

  const rawBody = await getRawBody(request);
  const bodySignature = sha1(rawBody, signatureSecret);

  if (bodySignature !== request.headers['x-vercel-signature']) {
    return response.status(403).json({
      code: 'invalid_signature',
      error: "signature didn't match",
    });
  }

  console.log(rawBody);

  response.status(200).json({ success: true });
}

function sha1(data: Buffer, secret: string): string {
  return crypto.createHmac('sha1', secret).update(data).digest('hex');
}

export const config = {
  api: {
    bodyParser: false,
```

```
  },
};
```

```ts filename="server.ts" framework=nextjs
import type { NextApiRequest, NextApiResponse } from 'next';
import crypto from 'crypto';
import getRawBody from 'raw-body';

export default async function handler(
  request: NextApiRequest,
  response: NextApiResponse,
) {
  if (request.method !== 'POST') {
    return response.status(405).json({ error: 'Method not allowed' });
  }

  // Store the signature secret in environment variables
  const signatureSecret = '<Drain signature secret>';

  const rawBody = await getRawBody(request);
  const bodySignature = sha1(rawBody, signatureSecret);

  if (bodySignature !== request.headers['x-vercel-signature']) {
    return response.status(403).json({
      code: 'invalid_signature',
      error: "signature didn't match",
    });
  }

  console.log(rawBody);

  response.status(200).json({ success: true });
}

async function sha1(data: Buffer, secret: string): string {
  return crypto.createHmac('sha1', secret).update(data).digest('hex');
}

export const config = {
  api: {
    bodyParser: false,
  },
};
```

```js filename="server.js" framework=other
import crypto from 'crypto';
import getRawBody from 'raw-body';

export default async function handler(
  request,
  response,
) {
  if (request.method !== 'POST') {
    return response.status(405).json({ error: 'Method not allowed' });
  }

  // Store the signature secret in environment variables
  const signatureSecret = '<Drain signature secret>';

  const rawBody = await getRawBody(request);
  const bodySignature = sha1(rawBody, signatureSecret);

  if (bodySignature !== request.headers['x-vercel-signature']) {
    return response.status(403).json({
      code: 'invalid_signature',
      error: "signature didn't match",
    });
  }

  console.log(rawBody);

  response.status(200).json({ success: true });
}

function sha1(data: Buffer, secret: string): string {
  return crypto.createHmac('sha1', secret).update(data).digest('hex');
}

export const config = {
  api: {
    bodyParser: false,
  },
};
```

```ts filename="server.ts" framework=other
import type { VercelRequest, VercelResponse } from '@vercel/node';
import crypto from 'crypto';
import getRawBody from 'raw-body';

export default async function handler(
  request: VercelRequest,
  response: VercelResponse,
) {
  if (request.method !== 'POST') {
    return response.status(405).json({ error: 'Method not allowed' });
  }
```

```
  // Store the signature secret in environment variables
  const signatureSecret = '<Drain signature secret>';

  const rawBody = await getRawBody(request);
  const bodySignature = sha1(rawBody, signatureSecret);

  if (bodySignature !== request.headers['x-vercel-signature']) {
    return response.status(403).json({
      code: 'invalid_signature',
      error: "signature didn't match",
    });
  }

  console.log(rawBody);

  response.status(200).json({ success: true });
}

function sha1(data: Buffer, secret: string): string {
  return crypto.createHmac('sha1', secret).update(data).digest('hex');
}

export const config = {
  api: {
    bodyParser: false,
  },
};
```

> **💡 Note:** For enhanced security against timing attacks, use constant-time comparison when verifying the `x-vercel-signature` header.

For additional authentication or identification purposes, you can also add custom headers when [configuring the Drain destination](/docs/

IP Address Visibility

Drains can include public IP addresses in the data, which may be considered personal information under certain data protection laws. To h

1. Go to the Vercel [dashboard](/dashboard) and ensure your team is selected in the scope selector
2. Go to the **Settings** tab and navigate to **Security & Privacy**
3. Under **IP Address Visibility**, toggle the switch off so the text reads **IP addresses are hidden in your Drains**

This setting is applied team-wide across all projects and drains.

More resources

For more information on Drains security and how to use them, check out the following resources:

- [Drains overview](/docs/drains)
- [Configure Drains](/docs/drains/using-drains)
- [Log Drains reference](/docs/drains/reference/logs)

---
title: "Using Drains"
description: "Learn how to configure drains to forward observability data to custom HTTP endpoints and add integrations."
last_updated: "2026-01-16T02:19:29.509Z"
source: "https://vercel.com/docs/drains/using-drains"

Using Drains

You can add drains to your project by following the configuration steps below. When you configure the destination, choose between sending

Configuring Drains

Teams on [Pro](/docs/plans/pro-plan) and [Enterprise](/docs/plans/enterprise) plans can configure drains to forward observability data. Y

- ### Add a drain
  From the Vercel dashboard, go to **Team Settings > Drains** and click **Add Drain**.

- ### Choose data type
  Select the type of observability data you want to drain:
  - **Logs**: Runtime, build and static logs from your deployments
  - **Traces**: Distributed tracing data using OpenTelemetry Protocol
  - **Speed Insights**: Performance metrics and web vitals
  - **Web Analytics**: Page views and custom events
  At any time, you can also add an [external integration](#external-integrations) to available [connectable account](/docs/integrations#c

- ### Configure the drain
  Provide a name for your drain and select which projects should send data to your endpoint. You can choose all projects or select specif

  The drain type determines which configuration options you can set:
  - **Log**: Configure the [log source](/docs/drains/reference/logs#log-sources), [environment](/docs/drains/reference/logs#log-environme
  - **Trace**: Configure the project and advanced sampling rate controls.
  - **Speed Insights**: Configure the project and a basic sampling rate.
  - **Web Analytics**: Configure the project and a basic sampling rate.
  Configure the sampling rate to control the volume of data sent to your drain. This can help manage costs when you have high traffic vol

- ### Configure the sampling rules (optional)
  For **Log** and **Trace** drains, add sampling rules to define how much data reaches your destination:
  1. If no rules exist, click **Add sampling rule**.
  2. Choose the environment you want to sample from.
  3. Set the sampling percentage.
  4. (Optional) Specify a request path prefix. Leave it blank to apply the rule to every path.
  Example workflows:
  - Launch-day monitoring: sample **100%** of production traffic when you launch a new feature, then decrease to **10%** once traffic sta
  - Static coverage: always collect **5%** from `/docs` so you can spot regressions on a static documentation site.
  Rules run from top to bottom. Requests that match a rule use that rule's sampling rate, and any other requests are dropped. If you do n

- ### Configure destination

Choose how you want to receive your drain data by selecting either the [**Custom Endpoint**](#custom-endpoint) or [**Native Integration
Custom endpoint
Configure a custom HTTP endpoint to receive drain data for any data type.

Endpoint URL

This is the URL of the endpoint we will send your data to. The request will be sent over HTTPS using the POST method. Make sure your en

Format

Choose the delivery format based on your data type:
- **Logs**: JSON or NDJSON (see [Logs reference](/docs/drains/reference/logs))
- **Traces**: JSON or Protobuf (see [Traces reference](/docs/drains/reference/traces))
- **Speed Insights**: JSON or NDJSON (see [Speed Insights reference](/docs/drains/reference/speed-insights))
- **Web Analytics**: JSON or NDJSON (see [Analytics reference](/docs/drains/reference/analytics))
Signature Verification Secret (Optional)

You can secure your endpoint by comparing the `x-vercel-signature` header with this secret. See [Securing your Drains](/docs/drains/sec

A secret will be automatically generated for you, and you can change it and provide your own secret at any time.

Custom Headers (Optional)

Add custom headers for authentication, identification, or routing purposes. Common use cases include:
- **Authentication**: Bearer tokens, API keys, or custom auth headers
- **Routing**: Headers to route requests to specific services or regions
- **Identification**: Custom headers to identify the source or type of data
- **Content negotiation**: Headers to specify preferred response formats
Format headers as `Header-Name: Header-Value` with one header per line.
Native integrations
Native integrations are available for log and traces data. You have 2 options:
1. Installed Products

   If you've already installed a marketplace integration product that supports drains, you can select it here. The integration will han

2. Available Products

   Browse and install available product integrations for this drain type:

   - Click **Install** on any available product. This opens the Marketplace integration creation page in a new window.
   - Update the default product name if needed and select a subscription plan
   - Click **Create** and click **Done** once the integration has been created
   - Go back to the Project's settings Drains page and select your newly created integration
You can also add a Drain from your team's **Integrations** tab
- Select the installed integration from the list. (Trace data shows under "Observability" and log data shows under "Logging")
- Click **Manage** and select your installed product
- Under **Drains**, click **Add Drain**
- Configure which project you would like to send data from and click **Create Drain**

- ### Create the drain
Once you have configured all settings, click **Create Drain**. If you configured a custom endpoint, it will be tested automatically whe

You can test your endpoint anytime by clicking the **Test** button to ensure it receives the data correctly.

Logs and traces correlation

Vercel automatically correlates logs with distributed traces when you setup [Tracing](/docs/tracing). Any logs generated during traced re

- `traceId` - The trace identifier
- `spanId` - The span identifier

This correlation happens automatically without code changes. For example, this log:

```js
console.log('User logged in', { userId: 123 });
```

Will be automatically enriched with [trace and span identifiers](/docs/drains/reference/logs#json-format-fields).

Limitations: Only applies to user code logs during traced requests, not build-time logs.

Drain integrations

You can create Drains with native integrations for the following data types by using [native integrations](#native-integrations) during t

- **Log drains**: [Logging services](https://vercel.com/marketplace/category/logging) like [Dash0](https://vercel.com/marketplace/dash0)
- **Trace drains**: [Observability services](https://vercel.com/marketplace/category/observability) like [Braintrust](https://vercel.com/

External integrations

1. From the Vercel dashboard, go to **Team Settings > Drains** and click **Add Drain**.
2. Click the **External Integrations** link on the top right of the **Add Drain** side modal.
3. From the **External Integration Log Drains** modal, select the installed or available external integration you would like to use and f

Learn more about [native integrations](/docs/integrations#native-integrations) and [external (connectable accounts) integrations](/docs/i

Errored Drains

Occasionally your drain endpoints can return an error. If more than 80% of drain deliveries fail or the number of failures exceed 50 for

Managing your active Drains

1. From your team's [dashboard](/dashboard), select the **Settings** tab
2. Select **Drains** from the sidebar
3. In available Drains, click on the menu on the right and click **Pause** to pause a drain or **Resume** to resume it
4. In available Drains, click on the menu on the right and click **Delete** to delete a drain

More resources

For more information on Drains and how to use them, check out the following resources:

- [Drains overview](/docs/drains)
- [Log Drains reference](/docs/drains/reference/logs)
- [Traces reference](/docs/drains/reference/traces)
- [Speed Insights reference](/docs/drains/reference/speed-insights)
- [Analytics reference](/docs/drains/reference/analytics)

---
title: "Managing Edge Configs with the Dashboard"
description: "Learn how to create, view and update your Edge Configs and the data inside them in your Vercel Dashboard at the Hobby team,
last_updated: "2026-01-16T02:19:29.455Z"
source: "https://vercel.com/docs/edge-config/edge-config-dashboard"

Managing Edge Configs with the Dashboard

You can create, view and update your [Edge Configs](/edge-config), and the data inside them, in your Vercel Dashboard at both the [accoun

Creating an Edge Config

At the account level

To add an Edge Config at the Hobby team or team level, follow these steps:

1. Make sure that you are in the [right Hobby team or team](/docs/dashboard-features)

2. Click on the **Storage** tab

3. Click the **Create Store** button

4. Type a name for your Edge Config in the dialog and click **Create**. The name shouldn't exceed 32 characters and can only contain alph

5. On creation, you are taken to the `my_edge_config_id` config page. By default, a key-value pair of `"greeting": "hello world"` is crea
   - View and manage stored items
   - Connect projects to and disconnect projects from this Edge Config
   - Generate, copy, and delete tokens associated with this Edge Config

Your Edge Config is now ready to be used. You can also [create an Edge Config at the project level](/docs/edge-config/edge-config-dashboa

> **💡 Note:** If you're creating a project at the account-level, we won't automatically
> create a token, connection string, and environment variable until a project
> has been connected.

At the project level

1. Navigate to your [project](/docs/projects/overview) page and click on the **Edge Config** tab

2. Click **Create Project Store** and type a name slug for your Edge Config in the dialog that opens. The name shouldn't exceed 32 charac

3. Click **Create**.

4. Once created, you can click on the Edge Config to [manage it](#managing-edge-configs). The following items are automatically created:
   - An environment variable `EDGE_CONFIG` that holds a [connection string](/docs/edge-config/using-edge-config#using-a-connection-string
   - A [read access token](/docs/edge-config/using-edge-config#creating-a-read-access-token). This token, along with your **EDGE CONFIG I

Managing Edge Configs

To view a list of all Edge Configs available in your Hobby team or team, go to **Storage** then select **Edge Config** from the drop-down

In the **Used by** column, you can see in which project(s) the Edge Config is used. The right column shows the size of the data contained

To rename the Edge Config, select the **Settings** item in the sidebar, update the Edge Config Name, and select **Save**.

To delete the Edge Config, select the **Settings** item in the sidebar, then select **Delete Edge Config**.

Managing items in the store

The default view of the Edge Config's detail page shows the list of all items in the store. You will see an open accordion titled **Learn

To add, edit or delete any item in your store, edit the `json` object in the right panel and click **Save Items**.

Restoring Edge Config backups

Backups of your Edge Config are automatically created when you make changes, and are stored for a [length of time](/docs/edge-config/edge

1. From your dashboard, select the **Storage** tab and then select your Edge Config
2. From the left section, select the **Backups** tab
3. From the list, select the backup that you would like to view. You'll be taken to the **Items** tab to view a comparison of the backup
4. To restore the backup, select the **Restore** button and confirm the action

To learn more about backups, see [Edge Config backups](/docs/edge-config/using-edge-config#edge-config-backups).

> **💡 Note:** When protected by a JSON schema, the backup must pass schema validation to be
> restored.

Schema validation

You can protect your Edge Config by adding a JSON schema to it. Vercel uses this schema to validate the data that is added to the store a

1. From your dashboard, select the **Storage** tab and then select your Edge Config
2. Toggle the **Schema** button to open the schema editing tab
3. Enter your [JSON schema](https://json-schema.org/) into the editor. Vercel will use the schema to validate your data in real-time
4. Click **Save**. This will save changes to both the schema and data

The following snippet is an example of a schema that allows you to set boolean flags and a list of redirects.

```json filename="schema.json"
{
  "$schema": "http://json-schema.org/draft-07/schema",
  "type": "object",
  "additionalProperties": false,
  "required": ["flags", "redirects"],
  "properties": {
    "flags": {
      "type": "object",
      "patternProperties": {
        "^.*$": {
          "type": "boolean"
        }
      }
    },
    "redirects": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "from": { "type": "string" },
          "to": { "type": "string" }
        }
      }
    }
  }
}
```

Managing connected projects

Click on **Projects** in the left panel of the Edge Config detail page to open a view that shows the projects connected with this Edge Co

To delete a connection, click on the vertical ellipsis icon on the right hand side of the row and click **Delete environment variable**

Deleting a connection does not delete the underlying token used by that Connection String. To learn how to delete tokens, review [Managing

To connect the Edge Config with another project, click **Connect Project**, find the project from the drop-down in the dialog and click *

Managing read access tokens

To delete a token, click on the vertical ellipsis icon on the right hand side of the token's row and click **Delete Token** and confirm

You can copy the connection string to be used in your code by clicking on **Copy Connection String** from the same pop up from the vertic

You can generate a new token by clicking the **Generate Token** button, typing a name slug in the dialog that opens and clicking **Create

Up Next

---
title: "Using Edge Config with DevCycle"
description: "Learn how to use Edge Config with Vercel"
last_updated: "2026-01-16T02:19:29.465Z"
source: "https://vercel.com/docs/edge-config/edge-config-integrations/devcycle-edge-config"

Using Edge Config with DevCycle

This guide will help you get started with using Vercel's DevCycle integration with Edge Config. This integration allows you to use Edge C

DevCycle is a feature management platform designed for developers. DevCycle allows you to work with feature flags more naturally, where y

With DevCycle and Vercel Edge Config the decision logic for your features lives with your hosted site, so you can run your feature rollou

Prerequisites

Before using this integration, you should have:

1. The latest version of Vercel CLI. To check your version, use `vercel --version`. To [install](/docs/cli#installing-vercel-cli) or upda
   <CodeBlock>
     <Code tab="pnpm">
       ```bash
       pnpm i vercel
       ```
     </Code>
     <Code tab="yarn">
       ```bash
       yarn i vercel
       ```
     </Code>
     <Code tab="npm">
       ```bash
       npm i vercel
       ```
     </Code>
     <Code tab="bun">
       ```bash
       bun i vercel
       ```
     </Code>
   </CodeBlock>
2. A project. If you don't have one, you can run the following terminal commands to create a Next.js project:

<CodeBlock>

  <Code tab="pnpm">
    ```bash
    pnpm i
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i
    ```
  </Code>
</CodeBlock>

1. A Vercel project. If you don't have one, see [Creating a Project](/docs/projects/overview#creating-a-project)
2. An Edge Config. If you don't have one, follow [the Edge Config quickstart](/docs/edge-config/get-started)
3. The Edge Config SDK:
   <CodeBlock>
     <Code tab="pnpm">
       ```bash
       pnpm i @vercel/edge-config
       ```
     </Code>
     <Code tab="yarn">
       ```bash
       yarn i @vercel/edge-config
       ```
     </Code>
     <Code tab="npm">
       ```bash
       npm i @vercel/edge-config
       ```
     </Code>
     <Code tab="bun">
       ```bash
       bun i @vercel/edge-config
       ```
     </Code>
   </CodeBlock>

- ### Set up the DevCycle integration
  Visit [the DevCycle page in the Integration Marketplace](/integrations/devcycle) and select the **Add Integration** button. From the mo
  1. Select your Vercel team and project.
  2. Continue and log into DevCycle.
  3. Select the DevCycle Organization and Project you want to use with Vercel Edge Config.
  4. Connect your DevCycle project to an existing or new Edge Config store.
  5. Click **Finish Setup**.

- ### Install the DevCycle Edge Config package
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i @devcycle/vercel-edge-config @vercel/edge-config
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i @devcycle/vercel-edge-config @vercel/edge-config
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i @devcycle/vercel-edge-config @vercel/edge-config
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i @devcycle/vercel-edge-config @vercel/edge-config
      ```
    </Code>
  </CodeBlock>

- ### Use the DevCycle integration in your code
  > For \['node']:
  For more information on DevCycle Node SDK usage, see [DevCycle docs](https://docs.devcycle.com/sdk/server-side-sdks/node).
  > For \['nextjs', 'nextjs-app']:
  For more information on DevCycle Next.js SDK usage, see the [DevCycle docs](https://docs.devcycle.com/sdk/client-side-sdks/nextjs).
  ```ts filename="app/index.tsx" framework=nextjs-app
  import { createClient } from '@vercel/edge-config';
  import { EdgeConfigSource } from '@devcycle/vercel-edge-config';
  import { setupDevCycle } from '@devcycle/nextjs-sdk/server';

  const edgeClient = createClient(process.env.EDGE_CONFIG ?? '');
  const edgeConfigSource = new EdgeConfigSource(edgeClient);

  export const { getVariableValue, getClientContext } = setupDevCycle({
    serverSDKKey: process.env.DEVCYCLE_SERVER_SDK_KEY ?? '',
    clientSDKKey: process.env.NEXT_PUBLIC_DEVCYCLE_CLIENT_SDK_KEY ?? '',
    userGetter: () => ({ user_id: 'test_user' }),
    options: {
      // pass the configSource option with the instance of EdgeConfigSource
      configSource: edgeConfigSource,
  ```

```
  },
});
```

```js filename="app/index.jsx" framework=nextjs-app
import { createClient } from '@vercel/edge-config';
import { EdgeConfigSource } from '@devcycle/vercel-edge-config';
import { setupDevCycle } from '@devcycle/nextjs-sdk/server';

const edgeClient = createClient(process.env.EDGE_CONFIG ?? '');
const edgeConfigSource = new EdgeConfigSource(edgeClient);

export const { getVariableValue, getClientContext } = setupDevCycle({
  serverSDKKey: process.env.DEVCYCLE_SERVER_SDK_KEY ?? '',
  clientSDKKey: process.env.NEXT_PUBLIC_DEVCYCLE_CLIENT_SDK_KEY ?? '',
  userGetter: () => ({ user_id: 'test_user' }),
  options: {
    // pass the configSource option with the instance of EdgeConfigSource
    configSource: edgeConfigSource,
  },
});
```

```ts filename="pages/index.tsx" framework=nextjs
import type { GetServerSideProps } from 'next';
import { createClient } from '@vercel/edge-config';
import { EdgeConfigSource } from '@devcycle/vercel-edge-config';
import { getServerSideDevCycle } from '@devcycle/nextjs-sdk/pages';

const edgeClient = createClient(process.env.EDGE_CONFIG ?? '');
const edgeConfigSource = new EdgeConfigSource(edgeClient);

export const getServerSideProps: GetServerSideProps = async (context) => {
  const user = {
    user_id: 'server-user',
  };

  return {
    props: {
      ...(await getServerSideDevCycle({
        serverSDKKey: process.env.DEVCYCLE_SERVER_SDK_KEY ?? '',
        clientSDKKey: process.env.NEXT_PUBLIC_DEVCYCLE_CLIENT_SDK_KEY ?? '',
        user,
        context,
        options: {
          // pass the configSource option with the instance of EdgeConfigSource
          configSource: edgeConfigSource,
        },
      })),
    },
  };
};
```

```js filename="pages/index.jsx" framework=nextjs
import { createClient } from '@vercel/edge-config';
import { EdgeConfigSource } from '@devcycle/vercel-edge-config';
import { getServerSideDevCycle } from '@devcycle/nextjs-sdk/pages';

const edgeClient = createClient(process.env.EDGE_CONFIG ?? '');
const edgeConfigSource = new EdgeConfigSource(edgeClient);

export const getServerSideProps = async (context) => {
  const user = {
    user_id: 'server-user',
  };

  return {
    props: {
      ...(await getServerSideDevCycle({
        serverSDKKey: process.env.DEVCYCLE_SERVER_SDK_KEY ?? '',
        clientSDKKey: process.env.NEXT_PUBLIC_DEVCYCLE_CLIENT_SDK_KEY ?? '',
        user,
        context,
        options: {
          // pass the configSource option with the instance of EdgeConfigSource
          configSource: edgeConfigSource,
        },
      })),
    },
  };
};
```

```ts filename="index.tsx" framework=node
import { createClient } from '@vercel/edge-config';
import { EdgeConfigSource } from '@devcycle/vercel-edge-config';
import { initializeDevCycle } from '@devcycle/nodejs-server-sdk';

// the EDGE_CONFIG environment variable contains a connection string for a particular edge config. It is set automatically
// when you connect an edge config to a project in Vercel.
const edgeClient = createClient(process.env.EDGE_CONFIG);
const edgeConfigSource = new EdgeConfigSource(edgeClient);

const devcycleClient = initializeDevCycle(
  process.env.DEVCYCLE_SERVER_SDK_KEY,
  // pass the edgeConfigSource as the "configSource" option during SDK intialization to tell the SDK to use Edge Config
  // for retrieving its configuration
  { configSource: edgeConfigSource },
);
```

```js filename="index.jsx" framework=node
import { createClient } from '@vercel/edge-config';
import { EdgeConfigSource } from '@devcycle/vercel-edge-config';
```

```
  import { initializeDevCycle } from '@devcycle/nodejs-server-sdk';

  // the EDGE_CONFIG environment variable contains a connection string for a particular edge config. It is set automatically
  // when you connect an edge config to a project in Vercel.
  const edgeClient = createClient(process.env.EDGE_CONFIG);
  const edgeConfigSource = new EdgeConfigSource(edgeClient);

  const devcycleClient = initializeDevCycle(
    process.env.DEVCYCLE_SERVER_SDK_KEY,
    // pass the edgeConfigSource as the "configSource" option during SDK intialization to tell the SDK to use Edge Config
    // for retrieving its configuration
    { configSource: edgeConfigSource },
  );
  ```

Next steps

Now that you have the DevCycle Edge Config integration set up, you can explore the following topics to learn more:

- [Get started with Edge Config](/docs/edge-config/get-started)
- [Read with the SDK](/docs/edge-config/edge-config-sdk)
- [Use the dashboard](/docs/edge-config/edge-config-dashboard)
- [Edge Config limits](/docs/edge-config/edge-config-limits)

---
title: "Using Edge Config with Hypertune"
description: "Learn how to use Hypertune"
last_updated: "2026-01-16T02:19:29.478Z"
source: "https://vercel.com/docs/edge-config/edge-config-integrations/hypertune-edge-config"

Using Edge Config with Hypertune

Hypertune is a feature flag, A/B testing and app configuration platform with full type-safety and Git version control.

The Hypertune Edge Config integration synchronizes with your Functions for low latency retrieval without fetch requests.

Prerequisites

Before using this integration, you should have the latest version of Vercel CLI.

To check your version, use `vercel --version`. To install or update Vercel CLI, use:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i vercel
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i vercel
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i vercel
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i vercel
    ```
  </Code>
</CodeBlock>

Get Started

> **💡 Note:** If you deploy a template like the [Hypertune Flags SDK
> Example](https://vercel.com/templates/next.js/flags-sdk-hypertune-nextjs), it
> will guide you through most of these steps.

Navigate to your **Project** and click the **Flags** tab.

Install a flag provider, select **Hypertune** and click **continue**, then toggle **Enable Edge Config Syncing** on.

- ### Set up your local environment
  Open your project in your development environment and link it to Vercel.
  ```bash
  vercel link
  ```
  Once linked, you can pull the environment variables that were added to your project.
  ```bash
  vercel env pull
  ```
  You should have a `.env.local` file with the following environment variables:
  ```bash
  EXPERIMENTATION_CONFIG="..."
  EXPERIMENTATION_CONFIG_ITEM_KEY="..."
  NEXT_PUBLIC_HYPERTUNE_TOKEN="..."
  ```
  > **💡 Note:** If you don't see these environment variables, ensure your project is linked to
  > the Hypertune integration in the Flags tab.

- ### Manage your flags in Hypertune
  From the Flags tab, click **Open in Hypertune** to make changes in your Hypertune project.

  When you click **save**, changes will be synchronized to your Edge Config and ready for use.
```
```

- ### Generate a type-safe client
  Run code generation to produce the type-safe client for use with the Hypertune SDK.
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>
  You should now have a `generated` directory with generated code reflecting your saved changes.

- ### Declare flags in your code
  You can declare server side flags using the Flags SDK with Hypertune as follows:
  ```ts filename="flags.ts" framework=all
  import {
    createSource,
    vercelFlagDefinitions as flagDefinitions,
    flagFallbacks,
    type FlagValues,
    type Context,
  } from '@/generated/hypertune';
  import { flag } from 'flags/next';
  import { createHypertuneAdapter } from '@flags-sdk/hypertune';
  import { identify } from './lib/identify';

  // Generate a Flags SDK adapter from generated Hypertune code
  const hypertuneAdapter = createHypertuneAdapter<FlagValues, Context>({
    createSource,
    flagDefinitions,
    flagFallbacks,
    identify,
  });

  // Use generated definitions to declare flags in your framework
  export const exampleFlag = flag(hypertuneAdapter.declarations.exampleFlag);
  ```
  > **💡 Note:** See the [more resources](#more-resources) section for more information about
  > the Hypertune and Flags SDK.

- ### Use flags in your app
  ```ts filename="app/page.tsx" framework=all
  import { exampleFlag } from '@/flags';

  export default async function Home() {
    const isExampleFlagEnabled = await exampleFlag();
    return <div>Example Flag is {isExampleFlagEnabled ? 'enabled' : 'disabled'}</div>;
  }
  ```

Next steps

Learn more about Edge Config:

- [Get started with Edge Config](/docs/edge-config/get-started)
- [Manage Edge Config on the dashboard](/docs/edge-config/edge-config-dashboard)
- [View the Edge Config SDK reference](/docs/edge-config/edge-config-sdk)
- [View Edge Config limits](/docs/edge-config/edge-config-limits)

More resources

Learn more about Hypertune and the Flags SDK adapter:

- [Hypertune App Router Quickstart](https://docs.hypertune.com/getting-started/next.js-app-router-quickstart)
- [Flags SDK Hypertune Provider](https://flags-sdk.dev/providers/hypertune)

---
title: "Using Edge Config with LaunchDarkly"
description: "Learn how to use Edge Config with Vercel"
last_updated: "2026-01-16T02:19:29.537Z"
source: "https://vercel.com/docs/edge-config/edge-config-integrations/launchdarkly-edge-config"

Using Edge Config with LaunchDarkly

This guide will help you get started with using Vercel's LaunchDarkly integration with Edge Config. This integration allows you to use Ed

[LaunchDarkly](https://docs.launchdarkly.com/home) allows you to enable and disable feature flags dynamically, decoupling feature rollout

> **💡 Note:** The LaunchDarkly Edge Config integration is only available to **Enterprise**
> LaunchDarkly customers. However, you **do not** need to have a Vercel
> [Enterprise](/docs/plans/enterprise) account.

Prerequisites

Before using this integration, you should have:

1. The latest version of Vercel CLI. To check your version, use `vercel --version`. To [install](/docs/cli#installing-vercel-cli) or upda
   <CodeBlock>
     <Code tab="pnpm">
       ```bash
       pnpm i vercel
       ```
     </Code>
     <Code tab="yarn">
       ```bash
       yarn i vercel
       ```
     </Code>
     <Code tab="npm">
       ```bash
       npm i vercel
       ```
     </Code>
     <Code tab="bun">
       ```bash
       bun i vercel
       ```
     </Code>
   </CodeBlock>
2. A project. If you don't have one, you can run the following terminal commands to create a Next project:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i
    ```
  </Code>
</CodeBlock>

1. A Vercel project. If you don't have one, see [Creating a Project](/docs/projects/overview#creating-a-project)
2. An Edge Config. If you don't have one, follow [the Edge Config quickstart](/docs/edge-config/get-started)
3. The Edge Config SDK:
   <CodeBlock>
     <Code tab="pnpm">
       ```bash
       pnpm i @vercel/edge-config
       ```
     </Code>
     <Code tab="yarn">
       ```bash
       yarn i @vercel/edge-config
       ```
     </Code>
     <Code tab="npm">
       ```bash
       npm i @vercel/edge-config
       ```
     </Code>
     <Code tab="bun">
       ```bash
       bun i @vercel/edge-config
       ```
     </Code>
   </CodeBlock>

- ### Set up the LaunchDarkly integration
  Visit [the LaunchDarkly page in the Integration Marketplace](/integrations/launchdarkly) and select the **Add Integration** button. Fro
  1. Select a Vercel team and project to connect the integration to
  2. Log into LaunchDarkly
  3. Select the **Authorize** button to allow the integration to access your LaunchDarkly account data
  4. Name the integration, and select an existing Edge Config or create a new one

- ### Get your client-side ID
  To use the integration, you'll need your client-side ID from LaunchDarkly. Here's how to add it to your project:
  1. [Go to the settings page of your LaunchDarkly dashboard](https://app.launchdarkly.com/settings/projects).
  2. Select the LaunchDarkly project your integration is connected to
  3. On the next page, copy the Client-side ID under the environment your integration is connected to (for example, Test or Production)
  Now, you must add the value to your project as an Environment Variable:
  1. Navigate to [your Vercel dashboard](/dashboard) and select the project you want to use LaunchDarkly with
  2. Under the **Settings** tab, navigate to **Environment Variables**, and create an `LD_CLIENT_SIDE_ID` variable with the value of your
  [See our Environment Variables docs to learn more](/docs/environment-variables#creating-environment-variables).

- ### Use the LaunchDarkly integration in your code
  Open your project's code on your local machine and do the following:
  1. Install LaunchDarkly's Vercel Server SDK:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i @launchdarkly/vercel-server-sdk
      ```

````
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i @launchdarkly/vercel-server-sdk
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i @launchdarkly/vercel-server-sdk
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i @launchdarkly/vercel-server-sdk
      ```
    </Code>
  </CodeBlock>
  2. Use [Vercel CLI](/docs/cli#installing-vercel-cli) to pull your Vercel project's environment variables:

     ```bash
     vercel env pull
     ```

  3. Finally, create a  file at the root of your project. This file will configure a Middleware that redirects your site visitors from `/

     ```ts filename="middleware.ts" framework=all
     import { init } from '@launchdarkly/vercel-server-sdk';
     import { createClient } from '@vercel/edge-config';

     const edgeConfigClient = createClient(process.env.EDGE_CONFIG!);
     const launchDarklyClient = init('YOUR CLIENT-SIDE ID', edgeConfigClient);

     export const config = {
       // Only run the middleware on the dashboard route
       matcher: '/homepage',
     };

     export default function middleware(request: Request): Response {
       await launchDarklyClient.initFromServerIfNeeded();
       const launchDarklyContext = { kind: 'org', key: 'my-org-key' };
       const showExperimentalHomepage = await launchDarklyClient.variation(
         'experimental-homepage',
         launchDarklyContext,
         true,
       );

       if (showExperimentalHomepage) {
         const url = new URL(request.url);
         url.pathname = '/new-homepage';
         return Response.redirect(url);
       }
     }
     ```

     ```js filename="middleware.js" framework=all
     import { init } from '@launchdarkly/vercel-server-sdk'
     import { createClient } from '@vercel/edge-config'

     const edgeConfigClient = createClient(process.env.EDGE_CONFIG);
     const launchDarklyClient = init("YOUR CLIENT-SIDE ID", edgeConfigClient);

     export const config = {
       // Only run the middleware on the dashboard route
       matcher: '/homepage',
     };

     export default function middleware(request) {
       await launchDarklyClient.initFromServerIfNeeded();
       const launchDarklyContext = { kind: 'org', key: 'my-org-key' };
       const showExperimentalHomepage = await launchDarklyClient.variation(
         'experimental-homepage',
         launchDarklyContext,
         true
       );

       if(showExperimentalHomepage) {
         const url = new URL(request.url);
         url.pathname = '/new-homepage';
         return Response.redirect(url);
       }
     }
     ```

## Next steps

Now that you have set up the LaunchDarkly Edge Config integration, you can explore the following topics to learn more:

- [Get started with Edge Config](/docs/edge-config/get-started)
- [Read with the SDK](/docs/edge-config/edge-config-sdk)
- [Use the dashboard](/docs/edge-config/edge-config-dashboard)
- [Edge Config limits](/docs/edge-config/edge-config-limits)


---
title: "Using Edge Config with an integration"
description: "Learn how to use Edge Config with popular A/B testing and feature flag service integrations."
last_updated: "2026-01-16T02:19:29.543Z"
source: "https://vercel.com/docs/edge-config/edge-config-integrations"
---
````
````

Using Edge Config with an integration

Vercel has partnered with A/B testing and feature flag services such as LaunchDarkly and Statsig to make it easier to integrate Edge Conf

To see these integrations in action, explore a template:

You can get started with any of these Edge Config integrations by following the quickstart:

- **[LaunchDarkly](/docs/edge-config/integrations/launchdarkly-edge-config)**
- **[Statsig](/docs/edge-config/integrations/statsig-edge-config)**
- **[Hypertune](/docs/edge-config/integrations/hypertune-edge-config)**
- **[Split](/docs/edge-config/integrations/split-edge-config)**
- **[DevCycle](/docs/edge-config/integrations/devcycle-edge-config)**

More resources

- [Quickstart](/docs/edge-config/get-started)
- [Read with the SDK](/docs/edge-config/edge-config-sdk)
- [Use the Dashboard](/docs/edge-config/edge-config-dashboard)
- [Manage with the API](/docs/edge-config/vercel-api)
- [Edge Config Limits](/docs/edge-config/edge-config-limits)

---
title: "Using Edge Config with Split"
description: "Learn how to use Edge Config with Vercel"
last_updated: "2026-01-16T02:19:29.603Z"
source: "https://vercel.com/docs/edge-config/edge-config-integrations/split-edge-config"

Using Edge Config with Split

This guide will help you get started with using Vercel's Split integration with Edge Config. This integration allows you to use Edge Conf

Split is a feature flag provider that tracks event data, enabling you to release features, target them to audiences, and measure their im

The Split Edge Config integration enables you to write your [Split rollout plan](https://help.split.io/hc/en-us/articles/9805284145549-Cr

Prerequisites

Before using this integration, you should have:

1. The latest version of Vercel CLI. To check your version, use `vercel --version`. To [install](/docs/cli#installing-vercel-cli) or upda
   <CodeBlock>
     <Code tab="pnpm">
       ```bash
       pnpm i vercel
       ```
     </Code>
     <Code tab="yarn">
       ```bash
       yarn i vercel
       ```
     </Code>
     <Code tab="npm">
       ```bash
       npm i vercel
       ```
     </Code>
     <Code tab="bun">
       ```bash
       bun i vercel
       ```
     </Code>
   </CodeBlock>
2. A project. If you don't have one, you can run the following terminal commands to create a Next project:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i
    ```
  </Code>
</CodeBlock>

1. A Vercel project. If you don't have one, see [Creating a Project](/docs/projects/overview#creating-a-project)
2. An Edge Config. If you don't have one, follow [the Edge Config quickstart](/docs/edge-config/get-started)
3. The Edge Config SDK:
   <CodeBlock>
     <Code tab="pnpm">
       ```bash
       pnpm i @vercel/edge-config
       ```
     </Code>

```
    <Code tab="yarn">
      ```bash
      yarn i @vercel/edge-config
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i @vercel/edge-config
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i @vercel/edge-config
      ```
    </Code>
  </CodeBlock>

> **💡 Note:** To configure this integration, Split Admin access (Split Admin users can add
> feature flags and segments, and edit them at will) is required.

- ### Set up the Split integration
  Visit [the Split page in the Vercel Integration Marketplace](/integrations/split) and select the **Add Integration** button. From the I
  1. Select a Vercel team and project to connect the integration to
  2. Log into Split
  3. Select the **[Split Environment](https://help.split.io/hc/en-us/articles/360019915771-Environments)** you want to use
  4. Select an existing Edge Config or create a new one
  5. Copy the Edge Config item key provided on this page. You'll need it to add it to your Environment Variables
  > **💡 Note:** You can also find your Edge Config Split item key in [your dashboard on
  > Vercel](/dashboard/integrations). In the tab,
  > select , then select on the
  > integration page. You should see the item key on the page that opens.

- ### Create your feature flags
  If you already have existing feature flags, you can skip this step and use those. In this example, we'll create one called `New_Marketi

  To create a feature flag in Split:
  1. Log into your [Split management console](https://app.split.io/login) and select the workspace icon near the top-left of the page
  2. In the sidebar, under **Target**, select **Feature flags**. Add the name `New_Marketing_Page`, and set the traffic type to `user`. S
  3. With your feature flag created, select the feature flag and select the **Definition** tab. Select **Initiate Environment** to config
  4. Add valid users to the feature flag
  5. Scroll down to **Targeting** and select **Add new individual target**
  6. Under **To user**, add any username you want to test. This example uses `Joe`.
  7. Select **Add new individual target**, then set the **Description** option to `off`. Add another username under **To user**. This exa
  8. Select **Review Changes**, then **Create** to finish
  Next, you need to add your credentials to your project's local environment to use the Split integration in your code.

- ### Get your credentials
  Next, you'll add the following credentials to your Vercel project:
  - `SPLIT_SDK_CLIENT_API_KEY`
  - `EDGE_CONFIG_SPLIT_ITEM_KEY`
  - `EDGE_CONFIG`
  To add environment variables to your project, visit [your Vercel dashboard](/dashboard) and select the project you want to use the Spli

  To get your Split client-side API keys:
  1. Log into your [Split management console](https://app.split.io/login) and select the workspace icon near the top-left of the page
  2. In the list of options that appears, select **Admin Settings**, then navigate to **API Keys** -> **SDK API Keys**
  3. Copy the client-side keys associated with the workspace and environment you're using
  To add your Edge Config Split item key, if you didn't copy it after setting up the integration on Vercel:
  1. Visit [your dashboard on
     Vercel](/dashboard/integrations)
  2. In the **Integrations** tab, select **Manage**
  3. On the integration page, select **Configure**
  4. You should see the item key on the page that opens. Copy it
  To add your Edge Config's connection string to your project:
  1. Visit your project's page in [the dashboard](/dashboard)
  2. Select the **Storage** tab. Select **Connect Store** and select the Edge Config associated with your Split integration. The `EDGE_CO
  Now you're ready to use the Split Edge Config integration in your code.

- ### Use the Split integration in your code
  Open your project's code on your local machine and do the following:
  1. Install Split's Browser SDK, Vercel integration utilities, and Vercel's Edge Config SDK:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i @splitsoftware/splitio-browserjs @splitsoftware/vercel-integration-utils @vercel/edge-config
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i @splitsoftware/splitio-browserjs @splitsoftware/vercel-integration-utils @vercel/edge-config
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i @splitsoftware/splitio-browserjs @splitsoftware/vercel-integration-utils @vercel/edge-config
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i @splitsoftware/splitio-browserjs @splitsoftware/vercel-integration-utils @vercel/edge-config
      ```
    </Code>
  </CodeBlock>
  2. Create an API route in your project. The following example fetches a treatement based on which user is visiting. You can specify the
  ```ts filename="app/api/marketing-example/route.ts" framework=nextjs-app
  import {
    SplitFactory,
    PluggableStorage,
    ErrorLogger,
  } from '@splitsoftware/splitio-browserjs';
```
```

```
import { EdgeConfigWrapper } from '@splitsoftware/vercel-integration-utils';
import { createClient } from '@vercel/edge-config';

export async function GET(request: Request) {
  const { EDGE_CONFIG_SPLIT_ITEM_KEY, SPLIT_SDK_CLIENT_API_KEY } = process.env;

  if (!SPLIT_SDK_CLIENT_API_KEY || !EDGE_CONFIG_SPLIT_ITEM_KEY)
    return new Response(
      `Failed to find your SDK Key (${SPLIT_SDK_CLIENT_API_KEY})
      or item key ${EDGE_CONFIG_SPLIT_ITEM_KEY}`,
    );

  const edgeConfigClient = createClient(process.env.EDGE_CONFIG);
  const { searchParams } = new URL(request.url);
  const userKey = searchParams.get('userKey') || 'anonymous';
  const client = SplitFactory({
    core: {
      authorizationKey: SPLIT_SDK_CLIENT_API_KEY,
      key: userKey,
    },
    mode: 'consumer_partial',
    storage: PluggableStorage({
      wrapper: EdgeConfigWrapper({
        // The Edge Config item key where Split stores
        // feature flag definitions
        edgeConfigItemKey: EDGE_CONFIG_SPLIT_ITEM_KEY,
        // The Edge Config client
        edgeConfig: edgeConfigClient,
      }),
    }),
    // Disable or keep only ERROR log level in production,
    // to minimize performance impact
    debug: ErrorLogger(),
  }).client();

  await new Promise((resolve) => {
    client.on(client.Event.SDK_READY, () => resolve);
    client.on(client.Event.SDK_READY_TIMED_OUT, () => resolve);
  });

  // Replace this with the feature flag you want
  const FEATURE_FLAG = 'New_Marketing_Page';
  const treatment = await client.getTreatment(FEATURE_FLAG);

  // Must await in app-router; waitUntil() is not
  // yet supported
  await client.destroy();

  // treatment will be 'control' if the SDK timed out
  if (treatment == 'control') return new Response('Control marketing page');

  return treatment === 'on'
    ? new Response('New marketing page')
    : new Response('Old marketing page');
}
```
```js filename="app/api/marketing-example/route.js" framework=nextjs-app
import {
  SplitFactory,
  PluggableStorage,
  ErrorLogger,
} from '@splitsoftware/splitio-browserjs';
import { EdgeConfigWrapper } from '@splitsoftware/vercel-integration-utils';
import { createClient } from '@vercel/edge-config';

export async function GET(request) {
  const { EDGE_CONFIG_SPLIT_ITEM_KEY, SPLIT_SDK_CLIENT_API_KEY } = process.env;

  if (!SPLIT_SDK_CLIENT_API_KEY || !EDGE_CONFIG_SPLIT_ITEM_KEY)
    return new Response(
      `Failed to find your SDK Key (${SPLIT_SDK_CLIENT_API_KEY})
      or item key ${EDGE_CONFIG_SPLIT_ITEM_KEY}`,
    );

  const edgeConfigClient = createClient(process.env.EDGE_CONFIG);
  const { searchParams } = new URL(request.url);
  const userKey = searchParams.get('userKey') || 'anonymous';
  const client = SplitFactory({
    core: {
      authorizationKey: SPLIT_SDK_CLIENT_API_KEY,
      key: userKey,
    },
    mode: 'consumer_partial',
    storage: PluggableStorage({
      wrapper: EdgeConfigWrapper({
        // The Edge Config item key where Split stores
        // feature flag definitions
        edgeConfigItemKey: EDGE_CONFIG_SPLIT_ITEM_KEY,
        // The Edge Config client
        edgeConfig: edgeConfigClient,
      }),
    }),
    // Disable or keep only ERROR log level in production,
    // to minimize performance impact
    debug: ErrorLogger(),
  }).client();

  await new Promise((resolve) => {
    client.on(client.Event.SDK_READY, () => resolve);
    client.on(client.Event.SDK_READY_TIMED_OUT, () => resolve);
```

```
  });

  // Replace this with the feature flag you want
  const FEATURE_FLAG = 'New_Marketing_Page';
  const treatment = await client.getTreatment(FEATURE_FLAG);

  // Must await in app-router; waitUntil() is not
  // yet supported
  await client.destroy();

  // treatment will be 'control' if the SDK timed out
  if (treatment == 'control') return new Response('Control marketing page');

  return treatment === 'on'
    ? new Response('New marketing page')
    : new Response('Old marketing page');
}
```
```ts filename="pages/api/marketing-example.ts" framework=nextjs
import {
  SplitFactory,
  PluggableStorage,
  ErrorLogger,
} from '@splitsoftware/splitio-browserjs';
import { EdgeConfigWrapper } from '@splitsoftware/vercel-integration-utils';
import { createClient } from '@vercel/edge-config';
import { NextFetchEvent } from 'next/server';

export default async function handler(
  request: Request,
  context: NextFetchEvent,
) {
  const { EDGE_CONFIG_SPLIT_ITEM_KEY, SPLIT_SDK_CLIENT_API_KEY } = process.env;

  if (!SPLIT_SDK_CLIENT_API_KEY || !EDGE_CONFIG_SPLIT_ITEM_KEY)
    return new Response(
      `Failed to find your SDK Key (${SPLIT_SDK_CLIENT_API_KEY})
      or item key ${EDGE_CONFIG_SPLIT_ITEM_KEY}`,
    );

  const edgeConfigClient = createClient(process.env.EDGE_CONFIG);
  const { searchParams } = new URL(request.url);
  const userKey = searchParams.get('userKey') || 'anonymous';
  const client = SplitFactory({
    core: {
      authorizationKey: SPLIT_SDK_CLIENT_API_KEY,
      key: userKey,
    },
    mode: 'consumer_partial',
    storage: PluggableStorage({
      wrapper: EdgeConfigWrapper({
        // The Edge Config item key where Split stores
        // feature flag definitions
        edgeConfigItemKey: EDGE_CONFIG_SPLIT_ITEM_KEY,
        // The Edge Config client
        edgeConfig: edgeConfigClient,
      }),
    }),
    // Disable or keep only ERROR log level in production,
    // to minimize performance impact
    debug: ErrorLogger(),
  }).client();

  // Wait until
  await new Promise((resolve) => {
    client.on(client.Event.SDK_READY, () => resolve);
    client.on(client.Event.SDK_READY_TIMED_OUT, () => resolve);
  });

  // Replace this with the feature flag you want
  const FEATURE_FLAG = 'New_Marketing_Page';
  const treatment = await client.getTreatment(FEATURE_FLAG);

  // Must await in app-router; waitUntil() is not
  // yet supported

  context.waitUntil(client.destroy());
  // treatment will be 'control' if the SDK timed out
  if (treatment == 'control') return new Response('Control marketing page');

  return treatment === 'on'
    ? new Response('New marketing page')
    : new Response('Old marketing page');
}
```
```js filename="pages/api/marketing-example.js" framework=nextjs
import {
  SplitFactory,
  PluggableStorage,
  ErrorLogger,
} from '@splitsoftware/splitio-browserjs';
import { EdgeConfigWrapper } from '@splitsoftware/vercel-integration-utils';
import { createClient } from '@vercel/edge-config';

export default async function handler(request, context) {
  const { EDGE_CONFIG_SPLIT_ITEM_KEY, SPLIT_SDK_CLIENT_API_KEY } = process.env;

  if (!SPLIT_SDK_CLIENT_API_KEY || !EDGE_CONFIG_SPLIT_ITEM_KEY)
    return new Response(
      `Failed to find your SDK Key (${SPLIT_SDK_CLIENT_API_KEY})
```

```
      or item key ${EDGE_CONFIG_SPLIT_ITEM_KEY}`,
    );

  const edgeConfigClient = createClient(process.env.EDGE_CONFIG);
  const { searchParams } = new URL(request.url);
  const userKey = searchParams.get('userKey') || 'anonymous';
  const client = SplitFactory({
    core: {
      authorizationKey: SPLIT_SDK_CLIENT_API_KEY,
      key: userKey,
    },
    mode: 'consumer_partial',
    storage: PluggableStorage({
      wrapper: EdgeConfigWrapper({
        // The Edge Config item key where Split stores
        // feature flag definitions
        edgeConfigItemKey: EDGE_CONFIG_SPLIT_ITEM_KEY,
        // The Edge Config client
        edgeConfig: edgeConfigClient,
      }),
    }),
    // Disable or keep only ERROR log level in production,
    // to minimize performance impact
    debug: ErrorLogger(),
  }).client();

  // Wait until
  await new Promise((resolve) => {
    client.on(client.Event.SDK_READY, () => resolve);
    client.on(client.Event.SDK_READY_TIMED_OUT, () => resolve);
  });

  // Replace this with the feature flag you want
  const FEATURE_FLAG = 'New_Marketing_Page';
  const treatment = await client.getTreatment(FEATURE_FLAG);

  // Must await in app-router; waitUntil() is not
  // yet supported

  context.waitUntil(client.destroy());
  // treatment will be 'control' if the SDK timed out
  if (treatment == 'control') return new Response('Control marketing page');

  return treatment === 'on'
    ? new Response('New marketing page')
    : new Response('Old marketing page');
}
```
```ts filename="/api/marketing-example.ts" framework=other
import {
  SplitFactory,
  PluggableStorage,
  ErrorLogger,
} from '@splitsoftware/splitio-browserjs';
import { EdgeConfigWrapper } from '@splitsoftware/vercel-integration-utils';
import { RequestContext } from '@vercel/edge';
import { createClient } from '@vercel/edge-config';

export default async function handler(
  request: Request,
  context: RequestContext,
) {
  const { EDGE_CONFIG_SPLIT_ITEM_KEY, SPLIT_SDK_CLIENT_API_KEY } = process.env;

  if (!SPLIT_SDK_CLIENT_API_KEY || !EDGE_CONFIG_SPLIT_ITEM_KEY)
    return new Response(
      `Failed to find your SDK Key (${SPLIT_SDK_CLIENT_API_KEY})
      or item key ${EDGE_CONFIG_SPLIT_ITEM_KEY}`,
    );

  const edgeConfigClient = createClient(process.env.EDGE_CONFIG);
  const { searchParams } = new URL(request.url);
  const userKey = searchParams.get('userKey') || 'anonymous';
  const client = SplitFactory({
    core: {
      authorizationKey: SPLIT_SDK_CLIENT_API_KEY,
      key: userKey,
    },
    mode: 'consumer_partial',
    storage: PluggableStorage({
      wrapper: EdgeConfigWrapper({
        // The Edge Config item key where Split stores
        // feature flag definitions
        edgeConfigItemKey: EDGE_CONFIG_SPLIT_ITEM_KEY,
        // The Edge Config client
        edgeConfig: edgeConfigClient,
      }),
    }),
    // Disable or keep only ERROR log level in production,
    // to minimize performance impact
    debug: ErrorLogger(),
  }).client();

  // Wait until
  await new Promise((resolve) => {
    client.on(client.Event.SDK_READY, () => resolve);
    client.on(client.Event.SDK_READY_TIMED_OUT, () => resolve);
  });

  // Replace this with the feature flag you want
```

```
    const FEATURE_FLAG = 'New_Marketing_Page';
    const treatment = await client.getTreatment(FEATURE_FLAG);

    // Must await in app-router; waitUntil() is not
    // yet supported

    context.waitUntil(client.destroy());
    // treatment will be 'control' if the SDK timed out
    if (treatment == 'control') return new Response('Control marketing page');

    return treatment === 'on'
      ? new Response('New marketing page')
      : new Response('Old marketing page');
  }
  ```
  ```js filename="pages/api/marketing-example.js" framework=other
    SplitFactory,
  import {
    PluggableStorage,
    ErrorLogger,
  } from '@splitsoftware/splitio-browserjs';
  import { EdgeConfigWrapper } from '@splitsoftware/vercel-integration-utils';
  import { createClient } from '@vercel/edge-config';

  export default async function handler(request, context) {
    const { EDGE_CONFIG_SPLIT_ITEM_KEY, SPLIT_SDK_CLIENT_API_KEY } = process.env;

    if (!SPLIT_SDK_CLIENT_API_KEY || !EDGE_CONFIG_SPLIT_ITEM_KEY)
      return new Response(
        `Failed to find your SDK Key (${SPLIT_SDK_CLIENT_API_KEY})
        or item key ${EDGE_CONFIG_SPLIT_ITEM_KEY}`,
      );

    const edgeConfigClient = createClient(process.env.EDGE_CONFIG);
    const { searchParams } = new URL(request.url);
    const userKey = searchParams.get('userKey') || 'anonymous';
    const client = SplitFactory({
      core: {
        authorizationKey: SPLIT_SDK_CLIENT_API_KEY,
        key: userKey,
      },
      mode: 'consumer_partial',
      storage: PluggableStorage({
        wrapper: EdgeConfigWrapper({
          // The Edge Config item key where Split stores
          // feature flag definitions
          edgeConfigItemKey: EDGE_CONFIG_SPLIT_ITEM_KEY,
          // The Edge Config client
          edgeConfig: edgeConfigClient,
        }),
      }),
      // Disable or keep only ERROR log level in production,
      // to minimize performance impact
      debug: ErrorLogger(),
    }).client();

    // Wait until
    await new Promise((resolve) => {
      client.on(client.Event.SDK_READY, () => resolve);
      client.on(client.Event.SDK_READY_TIMED_OUT, () => resolve);
    });

    // Replace this with the feature flag you want
    const FEATURE_FLAG = 'New_Marketing_Page';
    const treatment = await client.getTreatment(FEATURE_FLAG);

    // Must await in app-router; waitUntil() is not
    // yet supported

    context.waitUntil(client.destroy());
    // treatment will be 'control' if the SDK timed out
    if (treatment == 'control') return new Response('Control marketing page');

    return treatment === 'on'
      ? new Response('New marketing page')
      : new Response('Old marketing page');
  }
  ```

- ### Test your code
  1. Start a local development server. If you're using Vercel CLI, enter the following command in the terminal:
  ```bash filename="terminal"
  vercel dev
  ```
  1. Navigate to <http://localhost:3000/api/split-example?userKey=Joe>. You should see either `New marketing page` or `Old marketing page
     - Try changing the `userKey` search param's value to `Bobby`, or deleting it altogether, to see different responses when you visit t

Next steps

Now that you have set up the Split Edge Config integration, you can explore the following topics to learn more:

- [Get started with Edge Config](/docs/edge-config/get-started)
- [Read with the SDK](/docs/edge-config/edge-config-sdk)
- [Use the dashboard](/docs/edge-config/edge-config-dashboard)
- [Edge Config limits](/docs/edge-config/edge-config-limits)

---
title: "Using Edge Config with Statsig"
description: "Learn how to use Edge Config with Vercel"
```
```

last_updated: "2026-01-16T02:19:29.663Z"
source: "https://vercel.com/docs/edge-config/edge-config-integrations/statsig-edge-config"

Using Edge Config with Statsig

This guide will help you get started with using Vercel's Statsig integration with Edge Config. This integration allows you to use Edge Co

Statsig is a statistics engine that enables you to automate A/B testing and make data-driven decisions at scale. The Statsig integration

Prerequisites

Before using this integration, you should have:

1. The latest version of Vercel CLI. To check your version, use `vercel --version`. To [install](/docs/cli#installing-vercel-cli) or upda
   <CodeBlock>
     <Code tab="pnpm">
       ```bash
       pnpm i vercel
       ```
     </Code>
     <Code tab="yarn">
       ```bash
       yarn i vercel
       ```
     </Code>
     <Code tab="npm">
       ```bash
       npm i vercel
       ```
     </Code>
     <Code tab="bun">
       ```bash
       bun i vercel
       ```
     </Code>
   </CodeBlock>
2. A project. If you don't have one, you can run the following terminal commands to create a Next project:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i
    ```
  </Code>
</CodeBlock>

1. A Vercel project. If you don't have one, see [Creating a Project](/docs/projects/overview#creating-a-project)
2. An Edge Config. If you don't have one, follow [the Edge Config quickstart](/docs/edge-config/get-started)
3. The Edge Config SDK:
   <CodeBlock>
     <Code tab="pnpm">
       ```bash
       pnpm i @vercel/edge-config
       ```
     </Code>
     <Code tab="yarn">
       ```bash
       yarn i @vercel/edge-config
       ```
     </Code>
     <Code tab="npm">
       ```bash
       npm i @vercel/edge-config
       ```
     </Code>
     <Code tab="bun">
       ```bash
       bun i @vercel/edge-config
       ```
     </Code>
   </CodeBlock>

- ### Set up the Statsig integration
  Visit [the Statsig page in the Integration Marketplace](/integrations/statsig) and select the **Add Integration** button. Then:
  1. Select a Vercel team and Vercel project for your integration to be applied to
  2. Log into Statsig
  3. Select or create a new Edge Config to connect to Statsig
  4. Statsig will provide you with a **[Connection String](/docs/edge-config/using-edge-config#using-a-connection-string "Connection Stri

- ### Add your Environment Variables
  Navigate to [your Vercel dashboard](/dashboard), and select the project you want to use the Statsig integration with.

  Under the **Settings** tab, navigate to **Environment Variables**, and add the following variables:
  1. `EDGE_CONFIG`: Set this to the value of your Connection String

2. `EDGE_CONFIG_ITEM_KEY`: Set this to the value of your Edge Config Item Key
See [our Environment Variables documentation](/docs/environment-variables#creating-environment-variables) to learn more.

- ### Use the Statsig integration in your code
Statsig's [`statsig-node-vercel`](https://www.npmjs.com/package/statsig-node-vercel) package offers an `EdgeConfigDataAdapter` class, w

The following example sets up a Statsig experiment with Edge Config in an [Middleware](/docs/routing-middleware) file, using the `EDGE_
```ts filename="middleware.ts" framework=other
import type { VercelRequest } from '@vercel/node';
import Statsig from 'statsig-node';
import { createClient } from '@vercel/edge-config';
import { EdgeConfigDataAdapter } from 'statsig-node-vercel';

export const config = {
  matcher: '/',
};

const edgeConfigClient = createClient(process.env.EDGE_CONFIG!);
const dataAdapter = new EdgeConfigDataAdapter({
  edgeConfigClient: edgeConfigClient,
  edgeConfigItemKey: process.env.EDGE_CONFIG_ITEM_KEY!,
});

export async function middleware(request: VercelRequest) {
  await Statsig.initialize('statsig-server-api-key-here', { dataAdapter });

  const experiment = await Statsig.getExperiment(
    { userID: 'exampleId' },
    'statsig_example_experiment',
  );
}
```
```ts filename="middleware.ts" framework=nextjs
import { NextResponse } from 'next/server';
import type { NextRequest, NextFetchEvent } from 'next/server';
import Statsig from 'statsig-node';
import { createClient } from '@vercel/edge-config';
import { EdgeConfigDataAdapter } from 'statsig-node-vercel';

export const config = {
  matcher: '/',
};

const edgeConfigClient = createClient(process.env.EDGE_CONFIG!);
const dataAdapter = new EdgeConfigDataAdapter({
  edgeConfigClient: edgeConfigClient,
  edgeConfigItemKey: process.env.EDGE_CONFIG_ITEM_KEY!,
});

export async function middleware(request: NextRequest, event: NextFetchEvent) {
  await Statsig.initialize('statsig-server-api-key-here', { dataAdapter });

  const experiment = await Statsig.getExperiment(
    { userID: 'exampleId' },
    'statsig_example_experiment',
  );

  // Do any other experiment actions here

  // Ensure that all logged events are flushed to Statsig servers before the middleware exits
  event.waitUntil(Statsig.flush());

  return NextResponse.next();
}
```
```ts filename="middleware.ts" framework=nextjs-app
import { NextResponse } from 'next/server';
import type { NextRequest, NextFetchEvent } from 'next/server';
import Statsig from 'statsig-node';
import { createClient } from '@vercel/edge-config';
import { EdgeConfigDataAdapter } from 'statsig-node-vercel';

export const config = {
  matcher: '/',
};

const edgeConfigClient = createClient(process.env.EDGE_CONFIG!);
const dataAdapter = new EdgeConfigDataAdapter({
  edgeConfigClient: edgeConfigClient,
  edgeConfigItemKey: process.env.EDGE_CONFIG_ITEM_KEY!,
});

export async function middleware(request: NextRequest, event: NextFetchEvent) {
  await Statsig.initialize('statsig-server-api-key-here', { dataAdapter });

  const experiment = await Statsig.getExperiment(
    { userID: 'exampleId' },
    'statsig_example_experiment',
  );

  // Do any other experiment actions here

  // Ensure that all logged events are flushed to Statsig servers before the middleware exits
  event.waitUntil(Statsig.flush());

  return NextResponse.next();
}
```
```js filename="middleware.js" framework=other
import Statsig from 'statsig-node';
```

```
  import { createClient } from '@vercel/edge-config';
  import { EdgeConfigDataAdapter } from 'statsig-node-vercel';

  export const config = {
    matcher: '/',
  };

  const edgeConfigClient = createClient(process.env.EDGE_CONFIG);
  const dataAdapter = new EdgeConfigDataAdapter({
    edgeConfigClient: edgeConfigClient,
    edgeConfigItemKey: process.env.EDGE_CONFIG_ITEM_KEY,
  });

  export async function middleware() {
    await Statsig.initialize('statsig-server-api-key-here', { dataAdapter });

    const experiment = await Statsig.getExperiment(
      { userID: 'exampleId' },
      'statsig_example_experiment',
    );
  }
```

```js filename="middleware.js" framework=nextjs
  import Statsig from 'statsig-node';
  import { createClient } from '@vercel/edge-config';
  import { EdgeConfigDataAdapter } from 'statsig-node-vercel';

  export const config = {
    matcher: '/',
  };

  const edgeConfigClient = createClient(process.env.EDGE_CONFIG);
  const dataAdapter = new EdgeConfigDataAdapter({
    edgeConfigClient: edgeConfigClient,
    edgeConfigItemKey: process.env.EDGE_CONFIG_ITEM_KEY,
  });

  export async function middleware() {
    await Statsig.initialize('statsig-server-api-key-here', { dataAdapter });

    const experiment = await Statsig.getExperiment(
      { userID: 'exampleId' },
      'statsig_example_experiment',
    );
  }
```

```js filename="middleware.js" framework=nextjs-app
  import Statsig from 'statsig-node';
  import { createClient } from '@vercel/edge-config';
  import { EdgeConfigDataAdapter } from 'statsig-node-vercel';

  export const config = {
    matcher: '/',
  };

  const edgeConfigClient = createClient(process.env.EDGE_CONFIG);
  const dataAdapter = new EdgeConfigDataAdapter({
    edgeConfigClient: edgeConfigClient,
    edgeConfigItemKey: process.env.EDGE_CONFIG_ITEM_KEY,
  });

  export async function middleware() {
    await Statsig.initialize('statsig-server-api-key-here', { dataAdapter });

    const experiment = await Statsig.getExperiment(
      { userID: 'exampleId' },
      'statsig_example_experiment',
    );
  }
```

Next steps

Now that you have set up the Statsig Edge Config integration, you can explore the following topics to learn more:

- [Get started with Edge Config](/docs/edge-config/get-started)
- [Read with the SDK](/docs/edge-config/edge-config-sdk)
- [Use the dashboard](/docs/edge-config/edge-config-dashboard)
- [Edge Config limits](/docs/edge-config/edge-config-limits)

---
title: "Edge Config Limits and pricing"
description: "Learn about the Edge Configs limits and pricing based on account plans."
last_updated: "2026-01-16T02:19:29.576Z"
source: "https://vercel.com/docs/edge-config/edge-config-limits"

Edge Config Limits and pricing

An [Edge Config](/edge-config) is a global data store that [enables experimentation with feature flags, A/B testing, critical redirects,

Keep the number of stores to a minimum. Fewer large stores improve your overall latency.

Pricing

The following table outlines the price for each resource according to the plan you are on:

Limits by plan

The following table outlines the limits for each resource according to the plan you are on:

	Hobby	P
[**Maximum store size**](#maximum-store-size)	8 KB	6
[**Maximum number of stores (total)**](#maximum-number-of-stores)	1	3
[**Maximum number of stores connected to a project**](#maximum-number-of-stores-connected-to-a-project)	1	3
[**Maximum item key name length**](#maximum-item-key-name-length)	256 characters	2
[**Write propagation**](#write-propagation)	Up to 10 seconds globally	Up to 10 seconds g
[**Backup retention**](#backup-retention)	7 days	9

Usage

The table below shows the metrics for the [**Edge Config**](/docs/pricing/edge-config) section of the **Usage** dashboard.

To view information on managing each resource, select the resource link in the **Metric** column. To jump straight to guidance on optimiz

See the [manage and optimize Edge Config usage](/docs/pricing/edge-config) section for more information on how to optimize your usage.

Reads

Reads indicate how often your project has requested access to Edge Config to retrieve data through the SDK or the REST API. Vercel counts

Writes

Writes represent how often you updated your Edge Config through the SDK or the REST API.

Maximum store size

The maximum store size represents the total size limit of **each** Edge Config store, including all keys and values of the document.

Maximum number of stores

The maximum number of stores represents the total number of Edge Config stores that you can create for your account or team.

Maximum number of stores connected to a project

The maximum number of stores connected to a project represents the total number of Edge Config stores that you can connect to a single pr

Maximum item key name length

Each key name in your Edge Config can be up to 256 characters long. The key name must adhere to the regex pattern `^[\w-]+$`, which is eq

Write propagation

When updating an item in your Edge Config, it may take up to 10 seconds for the update to be globally propagated. You should avoid using

Backup retention

Backups are automatically saved when you make any changes, allowing you to [restore](/docs/edge-config/edge-config-dashboard#restoring-ed

To learn more about backups, see [Edge Config backups](/docs/edge-config/using-edge-config#edge-config-backups)

Reviewing Edge Config reads

The **Reads** chart shows the number of times your [Edge Config](/docs/functions/edge-config) has been read. You can filter the data by *

Optimizing Edge Config reads

- Select the **Project** tab to identify which project has the most Edge Config reads
- Review how you access the stores through both the [REST API](/docs/edge-config/vercel-api) and the [SDK](/docs/edge-config/edge-config-
- Where possible, use [`getAll()`](/docs/edge-config/edge-config-sdk#read-multiple-values) instead of separate [`get(key)`](/docs/edge-co

Managing Edge Config writes

The **Writes** chart shows the number of times your [Edge Configs](/docs/functions/edge-config) were updated. You can filter the data by

Optimizing Edge Config writes

- Select the **Edge Configs** tab to identify which Edge Config has the most Edge Config writes
- Review your points of updating the stores through the [REST API](/docs/edge-config/vercel-api) as they count towards your writes

Troubleshooting

If reading from your Edge Config seems slower than expected, ensure that the following are true:

- You've set [the connection string](/docs/edge-config/using-edge-config#using-a-connection-string) as an environment variable
- You are using the [SDK](/docs/edge-config/edge-config-sdk) to read from your Edge Config
- You see the Edge Config icon on the row for the connected environment variable
  on the Environment Variables page of your project settings
- You are testing on your Vercel deployment, as the optimizations happen only
  when you deploy to Vercel

- You are testing your Vercel deployment. The optimizations happen only
  when you deploy to Vercel

Edge Config Limit reached

Error: `Tried to attach 4 Edge Configs. Only 3 can be attached to one Deployment at a time.`

Depending on your plan, you can have up to 10 Edge Config stores created for your account. However, you are limited to a maximum of 3 Edg

If you get this error, review your storage by visiting [the Vercel Dashboard](/dashboard), selecting your project, and selecting the **St

To learn how to prevent this error, see [best practices](#edge-config-best-practices).

Edge Config update rejected

Updates to items in your Edge Config will be rejected if the resulting size of your Edge Config would exceed your account plan's limits. U

To resolve this issue, you can:

- Delete unused entries from your Edge Config to free up space
- [Upgrade your plan](/pricing)
- [Contact sales](/contact/sales) to unlock larger Edge Config store sizes

Edge Config best practices

- Where possible, having fewer large stores is better than having multiple small stores, as having fewer Edge Config stores requested mor

Security

If you are developing locally or self-hosting, your Edge Config is loaded through the public internet network. In this case, you may wond

- **It is safe to have the token as a parameter in the connection string**, because the SDK parses the passed string, then sends the toke

---
title: "@vercel/edge-config"
description: "The Edge Config client SDK is the most ergonomic way to read data from Edge Configs. Learn how to set up the SDK so you can
last_updated: "2026-01-16T02:19:29.723Z"
source: "https://vercel.com/docs/edge-config/edge-config-sdk"

@vercel/edge-config

The [Edge Config](/edge-config) client SDK is the most ergonomic way to read data from Edge Configs. It provides several helper methods f

It does not have functionality for *creating* new Edge Configs and *writing* to existing Edge Configs, which can be done [using the Verce

You can also [read Edge Config data with the Vercel REST API](/docs/edge-config/vercel-api#read-all-items). Review [Reading from an Edge

Requirements

Before you can start using the SDK, you need to have done the following:

- Created an Edge Config, which can be done using the [API](/docs/edge-config/vercel-api#create-an-edge-config) or the [Dashboard](/docs/
- Added [an Edge Config read access token](/docs/edge-config/using-edge-config#creating-a-read-access-token) to access your Edge Config
- Defined [a connection string](/docs/edge-config/using-edge-config#using-a-connection-string) with the Edge Config read access token and

Setting up the SDK

To get started, install the SDK:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i @vercel/edge-config
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @vercel/edge-config
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @vercel/edge-config
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i @vercel/edge-config
    ```
  </Code>
</CodeBlock>

Use connection strings

Use connection strings to connect your Edge Config to one or more projects. This allows Vercel to optimize your reads when you read the E

By default, the SDK will run all helper methods using the connection string stored in the `EDGE_CONFIG` environment variable. That means,

```typescript filename="example.ts"
import { NextResponse } from 'next/server';
import { getAll } from '@vercel/edge-config';

const configItems = await getAll();
```

However, you can store your connection string as **any** environment variable, and even connect to multiple Edge Configs by storing more

To do so, you must use the `createClient` helper.

The `createClient` helper method takes a connection string and returns an object that lets you use helper methods on the associated Edge

```typescript filename="example.ts
import { createClient } from '@vercel/edge-config';

// Fetch a single value from one config
const firstConfig = createClient(process.env.FIRST_EDGE_CONFIG);
const firstExampleValue1 = await firstConfig.get('other_example_key_1');

// Fetch all values from another config
const secondConfig = createClient(process.env.SECOND_EDGE_CONFIG);
const allValues = await secondConfig.getAll();
```

The following sections will teach you how to use all of the SDK's helper methods.

Read a single value

The `get` helper method allows you to fetch a value at a given key in your Edge Config.

```ts filename="app/api/get-example/route.ts" framework=nextjs
import { NextResponse } from 'next/server';
import { get } from '@vercel/edge-config';

export async function GET() {
  const val = await get('key');

  return NextResponse.json({
    label: `Value of "key" in my Edge Config.`,
    value: val,
  });
}
```

```js filename="app/api/get-example/route.js" framework=nextjs
import { NextResponse } from 'next/server';
import { get } from '@vercel/edge-config';

export async function GET() {
  const val = await get('key');

  return NextResponse.json({
    label: `Value of "key" in my Edge Config.`,
    value: val,
  });
}
```

```ts filename="api/get-example.ts" framework=other
import { get } from '@vercel/edge-config';

export async function GET() {
  const val = await get('key');

  return Response.json({
    label: `Value of "key" in my Edge Config.`,
    value: val,
  });
}
```

```js filename="api/get-example.js" framework=other
import { get } from '@vercel/edge-config';

export async function GET() {
  const val = await get('key');

  return Response.json({
    label: `Value of "key" in my Edge Config.`,
    value: val,
  });
}
```

```ts filename="app/api/get-example/route.ts" framework=nextjs-app
import { NextResponse } from 'next/server';
import { get } from '@vercel/edge-config';

export async function GET() {
  const val = await get('key');

  return NextResponse.json({
    label: `Value of "key" in my Edge Config.`,
    value: val,
  });
}
```

```js filename="app/api/get-example/route.js" framework=nextjs-app
import { NextResponse } from 'next/server';
import { get } from '@vercel/edge-config';

export async function GET() {
  const val = await get('key');

  return NextResponse.json({
    label: `Value of "key" in my Edge Config.`,
    value: val,
  });
}
```

Read multiple values

The `getAll` helper method returns all of your Edge Config's items.

```ts filename="app/api/getall-example/route.ts" framework=nextjs
import { NextResponse } from 'next/server';
import { getAll } from '@vercel/edge-config';

export async function GET() {
  const configItems = await getAll();

  return NextResponse.json({
    label: `These are all the values in my Edge Config.`,
```

```
    value: configItems,
  });
}
```

```js filename="app/api/getall-example/route.ts" framework=nextjs
import { NextResponse } from 'next/server';
import { getAll } from '@vercel/edge-config';

export async function GET() {
  const configItems = await getAll();

  return NextResponse.json({
    label: `These are all the values in my Edge Config.`,
    value: configItems,
  });
}
```

```ts filename="app/api/getall-example/route.ts" framework=nextjs-app
import { NextResponse } from 'next/server';
import { getAll } from '@vercel/edge-config';

export async function GET() {
  const configItems = await getAll();

  return NextResponse.json({
    label: `These are all the values in my Edge Config.`,
    value: configItems,
  });
}
```

```js filename="app/api/getall-example/route.ts" framework=nextjs-app
import { NextResponse } from 'next/server';
import { getAll } from '@vercel/edge-config';

export async function GET() {
  const configItems = await getAll();

  return NextResponse.json({
    label: `These are all the values in my Edge Config.`,
    value: configItems,
  });
}
```

```ts filename="api/getall-example.ts" framework=other
import { getAll } from '@vercel/edge-config';

export async function GET() {
  const configItems = await getAll();

  return Response.json({
    label: `These are all the values in my Edge Config.`,
    value: configItems,
  });
}
```

```js filename="api/getall-example.js" framework=other
import { getAll } from '@vercel/edge-config';

export async function GET() {
  const configItems = await getAll();

  return Response.json({
    label: `These are all the values in my Edge Config.`,
    value: configItems,
  });
}
```

Passing an array of key names causes `getAll` to return only the specified keys.

```ts filename="app/api/getall-keys-example/route.ts" framework=nextjs
import { NextResponse } from 'next/server';
import { getAll } from '@vercel/edge-config';

export async function GET() {
  const someItems = await getAll(['keyA', 'keyB']);

  return NextResponse.json({
    label: `These are a few values in my Edge Config.`,
    value: someItems,
  });
}
```

```js filename="app/api/getall-keys-example/route.js" framework=nextjs
import { NextResponse } from 'next/server';
import { getAll } from '@vercel/edge-config';

export async function GET() {
  const someItems = await getAll(['keyA', 'keyB']);

  return NextResponse.json({
    label: `These are a few values in my Edge Config.`,
    value: someItems,
  });
```

```
}
```

```ts filename="app/api/getall-keys-example/route.ts" framework=nextjs-app
import { NextResponse } from 'next/server';
import { getAll } from '@vercel/edge-config';

export async function GET() {
  const someItems = await getAll(['keyA', 'keyB']);

  return NextResponse.json({
    label: `These are a few values in my Edge Config.`,
    value: someItems,
  });
}
```

```js filename="app/api/getall-keys-example/route.js" framework=nextjs-app
import { NextResponse } from 'next/server';
import { getAll } from '@vercel/edge-config';

export async function GET() {
  const someItems = await getAll(['keyA', 'keyB']);

  return NextResponse.json({
    label: `These are a few values in my Edge Config.`,
    value: someItems,
  });
}
```

```ts filename="api/getall-keys-example.ts" framework=other
import { getAll } from '@vercel/edge-config';

export async function GET() {
  const someItems = await getAll(['keyA', 'keyB']);

  return Response.json({
    label: `These are a few values in my Edge Config.`,
    value: someItems,
  });
}
```

```js filename="api/getall-keys-example.js" framework=other
import { getAll } from '@vercel/edge-config';

export async function GET() {
  const someItems = await getAll(['keyA', 'keyB']);

  return Response.json({
    label: `These are a few values in my Edge Config.`,
    value: someItems,
  });
}
```

Check if a key exists

The `has` helper method lets you verify if a key exists in your Edge Config. It returns `true` if the key does, and `false` if it doesn't

```ts filename="app/api/has-example/route.ts" framework=nextjs
import { NextResponse } from 'next/server';
import { has } from '@vercel/edge-config';

export async function GET() {
  const exists = await has('key');

  return NextResponse.json({
    keyExists: exists ? `The key exists!` : `The key doesn't exist!`,
  });
}
```

```js filename="app/api/has-example/route.js" framework=nextjs
import { NextResponse } from 'next/server';
import { has } from '@vercel/edge-config';

export async function GET() {
  const exists = await has('key');

  return NextResponse.json({
    keyExists: exists ? `The key exists!` : `The key doesn't exist!`,
  });
}
```

```ts filename="app/api/has-example/route.ts" framework=nextjs-app
import { NextResponse } from 'next/server';
import { has } from '@vercel/edge-config';

export async function GET() {
  const exists = await has('key');

  return NextResponse.json({
    keyExists: exists ? `The key exists!` : `The key doesn't exist!`,
  });
}
```

```js filename="app/api/has-example/route.js" framework=nextjs-app
import { NextResponse } from 'next/server';
import { has } from '@vercel/edge-config';

export async function GET() {
  const exists = await has('key');

  return NextResponse.json({
    keyExists: exists ? `The key exists!` : `The key doesn't exist!`,
  });
}
```

```ts filename="api/has-example.ts" framework=other
import { has } from '@vercel/edge-config';

export async function GET() {
  const exists = await has('key');

  return Response.json({
    keyExists: exists ? `The key exists!` : `The key doesn't exist!`,
  });
}
```

```js filename="api/has-example.js" framework=other
import { has } from '@vercel/edge-config';

export async function GET() {
  const exists = await has('key');

  return Response.json({
    keyExists: exists ? `The key exists!` : `The key doesn't exist!`,
  });
}
```

Check the Edge Config version

Every Edge Config has a hash string associated with it, which is updated whenever the Config is updated. Checking this digest can help yo

The `digest` helper method lets you check the version of the Edge Config you're reading.

```ts filename="app/api/digest-example/route.ts" framework=nextjs
import { NextResponse } from 'next/server';
import { digest } from '@vercel/edge-config';

export async function GET() {
  const version = await digest();

  return NextResponse.json({
    digest: version,
  });
}
```

```js filename="app/api/digest-example/route.js" framework=nextjs
import { NextResponse } from 'next/server';
import { digest } from '@vercel/edge-config';

export async function GET() {
  const version = await digest();

  return NextResponse.json({
    digest: version,
  });
}
```

```ts filename="app/api/digest-example/route.ts" framework=nextjs-app
import { NextResponse } from 'next/server';
import { digest } from '@vercel/edge-config';

export async function GET() {
  const version = await digest();

  return NextResponse.json({
    digest: version,
  });
}
```

```js filename="app/api/digest-example/route.js" framework=nextjs-app
import { NextResponse } from 'next/server';
import { digest } from '@vercel/edge-config';

export async function GET() {
  const version = await digest();

  return NextResponse.json({
    digest: version,
  });
}
```

```ts filename="api/digest-example.ts" framework=other
import { digest } from '@vercel/edge-config';

export async function GET() {
  const version = await digest();
```

````
  return Response.json({
    digest: version,
  });
}
```

```js filename="api/digest-example.js" framework=other
import { digest } from '@vercel/edge-config';

export async function GET() {
  const version = await digest();

  return Response.json({
    digest: version,
  });
}
```

The digest's creation may change, so it is not documented. A matching digest indicates that the Edge Config content remains unchanged, wh

## Writing Edge Config Items

You cannot write to Edge Config items using the Edge Config SDK. Instead, you can programmatically write using the [Vercel REST API](/doc

The Edge Config SDK is designed to read from our `edge-config.vercel.com` endpoint using read only tokens to authenticate reads, while wr

If your project requires frequent writes, you should consider using a [Redis database from the Vercel Marketplace](/marketplace?category=

## Errors

All helper methods throw errors when:

- Your Edge Config read access token is invalid
- The Edge Config you're reading from doesn't exists
- A network error occurs

## Up Next


---
title: "Getting started with Edge Config"
description: "Learn how to create an Edge Config store and read from it in your project."
last_updated: "2026-01-16T02:19:29.747Z"
source: "https://vercel.com/docs/edge-config/get-started"
---

# Getting started with Edge Config

Edge Config is a distributed key-value store that allows you to store and retrieve data on Vercel's global network, close to your users.

This guide will help you will create an Edge Config called `hello_world_store` at the project-level, through the Vercel [dashboard](/dash

## Prerequisites

- Install the Edge Config SDK:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i @vercel/edge-config
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i @vercel/edge-config
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i @vercel/edge-config
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i @vercel/edge-config
      ```
    </Code>
  </CodeBlock>
- An existing project. This quickstart uses Next.js, but you can use any supported framework with Edge Config storage
- [Install](/docs/cli#installing-vercel-cli) or [update](/docs/cli#updating-vercel-cli) to the latest version of Vercel CLI

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i vercel
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i vercel
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i vercel
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i vercel
    ```
````
````

  </Code>
</CodeBlock>

- ### Create an Edge Config store
  Navigate to the [Project](/docs/projects/overview) you'd like to add an Edge Config store to. Click on the **Storage** tab, then click

  Create a new store by typing `hello_world_store` under **Edge Config** in the dialog that opens, and click **Create**.
  > **💡 Note:** The name can only contain alphanumeric letters, "_" and "-". It cannot exceed
  > 32 characters.

- ### Review what was created
  Once created, select `hello_world_store` to see a summary of what was created for you. Notice the following:
  - If you select **Project**, you'll see that your project was connected to the Edge Config by using an environment variable. If you go
  - If you select **Tokens**, you'll see a [read access token](/docs/edge-config/using-edge-config#creating-a-read-access-token). This to
  > **💡 Note:** If you're creating a project at the account-level, we won't automatically
  > create a token, connection string, and environment variable until a project
  > has been connected.

- ### Add a key-value pair
  Under **Items**, add the following key-value pair and click **Save Items**:
  ```json
  {
    "greeting": "hello world"
  }
  ```
  You can see more information about what can be stored in an Edge Config in the [limits](/docs/edge-config/edge-config-limits) documenta

- ### Connect your Vercel project
  Once you've created the store, you need to set up your project to read the contents of the store. This is detailed under **Learn how to

  On your local machine, connect your Vercel Project. If you haven't already, install the Edge Config SDK, as mentioned in [prerequisites

- ### Pull the latest environment variables
  Using Vercel CLI, pull the latest environment variables, specifically `EDGE_CONFIG`, so that it's available to your project locally:
  ```bash filename="terminal"
  vercel env pull
  ```

- ### Create a Middleware
  Create a [Middleware](/docs/routing-middleware) for your project by creating a new file called `middleware.js` at the root of the proje
  ```ts filename="middleware.ts" framework=all
  import { NextResponse } from 'next/server';
  import { get } from '@vercel/edge-config';

  export const config = { matcher: '/welcome' };

  export async function middleware() {
    const greeting = await get('greeting');
    return NextResponse.json(greeting);
  }
  ```
  ```js filename="middleware.js" framework=all
  import { NextResponse } from 'next/server';
  import { get } from '@vercel/edge-config';

  export const config = { matcher: '/welcome' };

  export async function middleware() {
    const greeting = await get('greeting');
    return NextResponse.json(greeting);
  }
  ```
  > **💡 Note:** `NextResponse.json` requires at least Next v13.1 or enabling
  > `experimental.allowMiddlewareResponseBody` in `next.config.js`.

- ### Run your application locally
  Run your application locally and visit `localhost:3000/welcome` to see your greeting. The middleware intercepts requests to `localhost:

Your project is now ready to read more key-value data pairs from the `hello_world_store` Edge Config using the [SDK](/docs/edge-config/ed

> **💡 Note:** Your Edge Config uses the public internet for reads when you develop locally.
> Therefore, you will see higher response times. However, when you deploy your
> application to Vercel, the reads are optimized to happen at ultra low latency
> without any network requests.

Next steps

Now that you've created an Edge Config store and read from it, you can explore the following:

- [Creating the Edge Config at the account level](/docs/edge-config/edge-config-dashboard#at-the-account-level)
- [Creating a read access token](/docs/edge-config/using-edge-config#creating-a-read-access-token)
- [Setting up a connection string](/docs/edge-config/using-edge-config#using-a-connection-string)
- [Learn about the `@vercel/edge-config` package](https://github.com/vercel/storage/tree/main/packages/edge-config#readme)
- [Explore the SDK](/docs/edge-config/edge-config-sdk)

---
title: "Vercel Edge Config"
description: "An Edge Config is a global data store that enables experimentation with feature flags, A/B testing, critical redirects, and
last_updated: "2026-01-16T02:19:29.780Z"
source: "https://vercel.com/docs/edge-config"

Vercel Edge Config

An [Edge Config](/edge-config) is a global data store that [enables experimentation with feature flags, A/B testing, critical redirects,

With Vercel's optimizations, you can read Edge Config data at negligible latency. The vast majority of your reads will complete within 15

You can use an Edge Config in [Middleware](/docs/routing-middleware) and [Vercel Functions](/docs/functions).

> **💡 Note:** Vercel's Edge Config read optimizations are **only available on the Edge and
> Node.js runtimes**. Optimizations can be enabled for other runtimes, [such as
> Ruby, Go, and Python](/docs/functions/runtimes) upon request. See [our Edge
> Config limits docs](/docs/edge-config/edge-config-limits) to learn more.

Use cases

Edge Configs are great for data that is accessed frequently and updated infrequently. Here are some examples of storage data suitable for

Getting started

You can create and manage your Edge Config from either [Vercel REST API](/docs/edge-config/vercel-api) or [Dashboard](/docs/edge-config/e

To get started, see [our quickstart](/docs/edge-config/get-started).

Using Edge Config in your workflow

If you'd like to know whether or not Edge Config can be integrated into your workflow, it's worth knowing the following:

- You can have one or more Edge Configs per Vercel account, depending on your plan as explained in [Limits](/docs/edge-config/edge-config
- You can use multiple Edge Configs in one Vercel project
- Each Edge Config can be accessed by multiple Vercel projects
- Edge Configs can be scoped to different environments within projects using environment variables
- **Edge Config access is secure by default**. A [read access token](/docs/edge-config/using-edge-config#creating-a-read-access-token) is

See [our Edge Config limits docs to learn more](/docs/edge-config/edge-config-limits)

Why use Edge Config instead of alternatives?

There are alternative solutions to Edge Config for handling A/B testing, feature flags, and IP blocking. The following table lays out how

Edge Config vs alternatives	**Read latency**
Edge Config	**Ultra-low**	**Varies**	**No**	**No**
Remote JSON files	Varies	Varies
Embedded JSON files	Lowest
Environment Variables	Lowest

Limits

To learn about Edge Config limits and pricing, see [our Edge Config limits docs](/docs/edge-config/edge-config-limits).

More resources

- [Quickstart](/docs/edge-config/get-started)
- [Read with the SDK](/docs/edge-config/edge-config-sdk)
- [Use the Dashboard](/docs/edge-config/edge-config-dashboard)
- [Manage with the API](/docs/edge-config/vercel-api)
- [Edge Config Limits](/docs/edge-config/edge-config-limits)

---
title: "Using Edge Config"
description: "Learn how to use Edge Configs in your projects."
last_updated: "2026-01-16T02:19:29.806Z"
source: "https://vercel.com/docs/edge-config/using-edge-config"

Using Edge Config

[Edge Config](/docs/edge-config) is a global data store that offers ultra-low latency read speeds from anywhere in the world thanks to [V

We recommend using [the Edge Config client SDK](/docs/edge-config/edge-config-sdk) to read data from your Edge Configs. To write data to

This page outlines all the ways you can interact with your Edge Configs, and our recommended best approaches.

Reading data from Edge Configs

There are multiple ways to read data from your Edge Configs, but **we recommend using [our Edge Config client SDK](/docs/edge-config/edge

If you prefer making direct API requests to your Edge Config, we recommend sending them to your [Edge Config endpoint](#understanding-edg

Edge Config is optimized to work with Vercel's CDN. As a result, Edge Configs accessed from local development environments cannot benefit

Understanding Edge Config endpoints

Edge Config is available at two separate REST APIs which are built for distinct use cases:

- `api.vercel.com`: [Vercel REST API](/docs/rest-api) built for managing Edge Config
- `edge-config.vercel.com`: [Edge Config endpoint](/docs/edge-config/using-edge-config#querying-edge-config-endpoints) intended for readi

`api.vercel.com`

- This endpoint is part of the [Vercel REST API](/docs/rest-api)
- It is intended to [manage Edge Configs](/docs/edge-config/vercel-api)
- You can use this endpoint to create, update, and delete Edge Configs
- This endpoint is served from a single region and we do not apply any of our read optimizations
- This endpoint is rate limited to 20 Edge Config Item reads per minute
- Reading Edge Config from this endpoint will always return the latest version of an Edge Config
- This endpoint uses the [Vercel REST API authentication](/docs/rest-api#authentication)

`edge-config.vercel.com`

- This is a highly optimized, globally distributed, actively replicated endpoint built for global, low latency, high volume reads
- This endpoint has no rate limits
- This is the endpoint [`@vercel/edge-config`](/docs/edge-config/edge-config-sdk) reads from
- This endpoint uses the Edge Config's own [Read Access tokens](/docs/edge-config/using-edge-config#creating-a-read-access-token)

Querying Edge Config endpoints

You can use the following routes when querying your Edge Config endpoint:

- `/<edgeConfigId>/items`
- `/<edgeConfigId>/item/<itemKey>`
- `/<edgeConfigId>/digest`

You can authenticate with a [Read Access token](/docs/edge-config/using-edge-config#creating-a-read-access-token), which you can add to t

Finding your Edge Config ID

You can find your Edge Config ID with one of the following methods:

- In your dashboard, under the **Storage** tab. Select your Edge Config, and you'll see the ID under the **Edge Config ID** label near th

* Send a `GET` request to the `/edge-config` endpoint of Vercel REST API, as outlined in [our API reference](/docs/rest-api/reference/end

```bash
https://api.vercel.com/v1/edge-config?teamId=<teamId>
```

Creating a read access token

A read access token is automatically generated when you connect an Edge Config to a project.

There are multiple ways to create a Read Access token for your Edge Config manually:

- In the **Storage** tab of your project dashboard. See [our Edge Config dashboard docs](/docs/edge-config/edge-config-dashboard#managing
- Through a `POST` request to Vercel REST API

Using Vercel API

First, you'll need an access token for Vercel REST API, which you must add to an `Authorization` header with the `Bearer <token>` pattern

Then you can send a `POST` request to the [`/edge-config/<edgeConfigId>/token`](/docs/rest-api/reference/endpoints/edge-config/create-an-

\['cURL'

```bash filename="cURL"
curl -X 'POST' 'https://api.vercel.com/v1/edge-config/my_edge_config_id/token' \
     -H 'Authorization: Bearer your_vercel_api_token_here' \
     -H 'Content-Type: application/json; charset=utf-8' \
     -d $'{ "label": "my edge config token label" }'

```

'fetch']

```javascript filename="fetch"
try {
  const createReadAccessToken = await fetch(
    'https://api.vercel.com/v1/edge-config/my_edge_config_id/token',
    {
      method: 'POST',
      headers: {
        Authorization: `Bearer ${your_vercel_api_token_here}`,
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({
        label: 'my edge config token label',
      }),
    },
  );
  const result = await createReadAccessToken.json();
  console.log(result);
} catch (error) {
  console.log(error);
}
```

> **💡 Note:** Append the query parameter to if the config is
> scoped to a Vercel team.

The response from the API will be a JSON object with a `"token"` key that contains the value for the Edge Config read access token:

```bash filename="response"
{ "token": "your_edge_config_read_access_token_here" }
```

Using a connection string

A connection string is a URL that connects a project to an Edge Config.

To find and copy the connection string:

1. Navigate to the **Tokens** tab of your project's **Storage** dashboard
2. Select the three dots icon displayed in the list of tokens
3. Select **Copy Connection String** from the dropdown menu

> **💡 Note:** A token is not created when you create an Edge Config at the account level,
> until you connect a project.

Vercel will optimize your reads to be faster if you set the connection string as an environment variable. Hard-coding your connection

The variable can be called anything, but [our Edge Config client SDK](/docs/edge-config/edge-config-sdk) will search for `process.env.EDG

Writing data to Edge Configs

Edge Config is optimized for **many reads** and **few writes**. To write data to your Edge Configs, see [our docs on doing so with Vercel

Edge Config backups

Edge Config backups are a backup and restore functionality that allows you to access and roll back to a previous point in time.

Restoring a backup will immediately update the live data, and the data that was live before the restore will become available as a new ba

Backups are taken when you make any changes either through the dashboard or API. They do not contribute to your storage size. The length

---
title: "Managing Edge Configs with Vercel REST API"
description: "Learn how to use the Vercel REST API to create and update Edge Configs. You can also read data stored in Edge Configs with
last_updated: "2026-01-16T02:19:29.863Z"
source: "https://vercel.com/docs/edge-config/vercel-api"

Managing Edge Configs with Vercel REST API

We recommend you use the Vercel REST API only for creating and updating an [Edge Config](/edge-config). For reading data (which you shoul

Updates to your Edge Config can take up to a few seconds to propagate globally, and therefore might not be available from the Edge Config

You can also request metadata about your Edge Configs through the API.

This section will show you how to update, read metadata about, and read the contents of your Edge Configs with the Vercel REST API. To le

Create an Edge Config

To create an Edge Config with the [Vercel REST API](/docs/rest-api), make a `POST` request to the `edge-config` path of the API endpoint.

```javascript filename="endpoint"
'https://api.vercel.com/v1/edge-config';
```

The request body should be a JSON object containing a `"slug"` with the name you would like to call your Edge Config as its value.

> **💡 Note:** The name can only contain alphanumeric letters, "_" and "-". It cannot exceed
> 32 characters.

See the example below:

\['cURL'

```bash filename="cURL"
curl  -X 'POST' 'https://api.vercel.com/v1/edge-config' \
      -H 'Authorization: Bearer your_vercel_api_token_here' \
      -H 'Content-Type: application/json; charset=utf-8' \
      -d $'{ "slug": "your_edge_config_name_here" }'
```

'fetch']

```javascript filename="fetch"
try {
  const createEdgeConfig = await fetch(
    'https://api.vercel.com/v1/edge-config',
    {
      method: 'POST',
      headers: {
        Authorization: `Bearer ${your_vercel_api_token_here}`,
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({
        slug: 'your_edge_config_name_here',
      }),
    },
  );
  const result = await createEdgeConfig.json();
  console.log(result);
} catch (error) {
  console.log(error);
}
```

Upon success, you should receive a JSON response similar to the following:

```json filename="response"
{
  "createdAt": 1234567890123,
  "updatedAt": 1234567890123,
  "slug": "your_edge_config_slug_here",
  "id": "your_edge_config_id_here",
  "digest": "abc123efg456hij789",
  "sizeInBytes": 2,
  "itemCount": 0,
  "ownerId": "your_id_here"
}
```

The above example will create an Edge Config scoped to your Hobby team. To scope your Edge Config to a Vercel Team:

- [Generate a Vercel REST API access token](/docs/rest-api/vercel-api-integrations#create-an-access-token) that is scoped to the appropri
- Add the `?teamId` query parameter to your `POST` request. Set its value to [the Team's ID](/docs/accounts#find-your-team-id), which you

> **💡 Note:** The key's value will be your
> if you created
> the Edge Config using the query parameter.

Update your Edge Config items

To add an item to or update an item in your Edge Config, send a `PATCH` request to the `edge-config` endpoint, appending `/your_edge_conf

If you're requesting an Edge Config scoped to a team, add `?teamId=` to the end of the endpoint, pasting [the Vercel Team's ID](/docs/acc

Your URL should look like this:

```javascript filename="endpoint"
'https://api.vercel.com/v1/edge-config/your_edge_config_id_here/items?teamId=your_team_id_here';
```

Your request body should be a JSON object containing an `"items"` array. The `"items"` array must contain objects that describe the chang

Property	Description	Valid values
`"operation"`	The change you want to make to your Edge Config.	`"create"`, `"update"`, `"upsert"`, `"d
`"key"`	The name of the key you want to add to or update within your Edge Config.	String of alphanumeric characters, "_"
`"value"`	The value you want to assign to the key.	Strings, JSON objects, `null` objects,

The following example demonstrates a request body that creates an `"example_key_1"` key with a value of `"example_value_1"`, then updates

```json filename="body"
{
  "items": [
    {
      "operation": "create",
      "key": "example_key_1",
      "value": "example_value_1"
    },
    {
      "operation": "update",
      "key": "example_key_2",
      "value": "new_value"
    }
  ]
}
```

The following is an API call that sends the above request body to your Edge Config:

\['cURL'

```bash filename="cURL"
curl -X 'PATCH' 'https://api.vercel.com/v1/edge-config/your_edge_config_id_here/items' \
     -H 'Authorization: Bearer your_vercel_api_token_here' \
     -H 'Content-Type: application/json' \
     -d $'{ "items": [ { "operation": "create", "key": "example_key_1", "value": "example_value_1" }, { "operation": "update", "key": "ex
```

'fetch']

```javascript filename="fetch"
try {
  const updateEdgeConfig = await fetch(
    'https://api.vercel.com/v1/edge-config/your_edge_config_id_here/items',
    {
      method: 'PATCH',
      headers: {
        Authorization: `Bearer ${your_vercel_api_token_here}`,
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({
        items: [
          {
            operation: 'create',
            key: 'example_key_1',
            value: 'example_value_1',
          },
          {
            operation: 'update',
            key: 'example_key_2',
            value: 'new_value',
          },
        ],
      }),
    },
  );
  const result = await updateEdgeConfig.json();
  console.log(result);
} catch (error) {
  console.log(error);
}
```

Successful requests will receive a response of `{"status":"ok"}`.

Failing Edge Config `PATCH` requests

If only one of the operations in the `"items"` array of your `PATCH` request body fails, the entire request will fail. Failed requests wi

For example:

```json filename="error"
{
  "error": {
    "code": "forbidden",
    "message": "The request is missing an authentication token",
    "missingToken": true
  }
}
```

Read all items

Reading items from your Edge Configs with the Vercel REST API is not recommended. Instead, you should [use the SDK](/docs/edge-config

However, if you must read your Edge Config with the API, you can do so by making a `GET` request to the `edge-config` endpoint.

Your URL should look like this:

```javascript filename="endpoint"
'https://api.vercel.com/v1/edge-config/your_edge_config_id_here/items?teamId=your_team_id_here';
```

The following is an example of a request that fetches an Edge Config's items with the Vercel REST API:

\['cURL'

```bash filename="request"
curl "https://api.vercel.com/v1/edge-config/your_edge_config_id_here/items?teamId=your_team_id_here" \
     -H 'Authorization: Bearer your_vercel_api_token_here'
```

'fetch']

```javascript filename="fetch"
try {
  const readItems = await fetch(
    'https://api.vercel.com/v1/edge-config/your_edge_config_id_here/items?teamId=your_team_id_here',
    {
      method: 'GET',
      headers: {
        Authorization: `Bearer ${your_vercel_api_token_here}`,
      },
    },
  );
  const result = await readItems.json();
  console.log(result);
} catch (error) {
  console.log(error);
}
```

Read metadata

You can read your Edge Config's metadata (but not its key-value pair contents) by making a `GET` request to the `edge-config` API endpoin

The following is an example `GET` request that fetches metadata about an Edge Config associated with a Vercel Team.

\['cURL'

```bash filename="request"
curl "https://api.vercel.com/v1/edge-config/your_edge_config_id_here?teamId=your_team_id_here" \
     -H 'Authorization: Bearer your_vercel_api_token_here'
```

'fetch']

```javascript filename="fetch"
try {
  const readMetadata = await fetch(
    'https://api.vercel.com/v1/edge-config/your_edge_config_id_here?teamId=your_team_id_here',
    {
      method: 'GET',
      headers: {
        Authorization: `Bearer ${your_vercel_api_token_here}`,
      },
    },
  );
  const result = await readMetadata.json();
  console.log(result);
} catch (error) {
  console.log(error);
}
```

If the Edge Config exists, the response will be the same JSON object you receive when [creating your Edge Config with the Vercel REST API

```json filename="response"
{
  "createdAt": 1234567890123,
  "updatedAt": 1234567890123,
  "slug": "your_edge_config_slug_here",
  "id": "your_edge_config_id_here",
  "digest": "abc123efg456hij789",
  "sizeInBytes": 2,
  "itemCount": 0,
  "ownerId": "your_id_here"
}
```

List all Edge Configs

You can list all of your Edge Configs in a specific Hobby team or team with a `GET` request to the `edge-config` API endpoint. For exampl

\['cURL'

```bash filename="request"
curl "https://api.vercel.com/v1/edge-config?teamId=your_team_id_here" \
     -H 'Authorization: Bearer your_vercel_api_token_here'
```

'fetch']

```javascript filename="fetch"
try {
  const listItems = await fetch(
    'https://api.vercel.com/v1/edge-config?teamId=your_team_id_here',
    {
      method: 'GET',
      headers: {
        Authorization: `Bearer ${your_vercel_api_token_here}`,
      },
    },
  );
  const result = await listItems.json();
  console.log(result);
} catch (error) {
  console.log(error);
}
```

The response should be similar to this:

```json filename="response"
[
  {
    "slug": "example_config_1",
    "itemCount": 0,
    "createdAt": 1234567890123,
    "updatedAt": 1234567890123,
    "id": "your_edge_config_id_here",
    "digest": "abc123efg456hij789",
    "sizeInBytes": 2,
    "ownerId": "your_id_here"
  },
  {
    "slug": "example_config_2",
    "itemCount": 0,
    "createdAt": 0123456789012,
    "updatedAt": 0123456789012,
    "id": "your_edge_config_id_here",
    "digest": "123efg456hij789abc",
    "sizeInBytes": 2,
    "ownerId": "your_id_here"
  }
]
```

Make requests to the Edge Config endpoint

We recommend storing your [connection string](/docs/edge-config/using-edge-config#using-a-connection-string) as an environment variable i

To do so, create an [Edge Config read access token](/docs/edge-config/using-edge-config#creating-a-read-access-token), which will be used

The Edge Config endpoint used in the connection string is distinct from a Vercel REST API endpoint. Its root is `https://edge-config.verc

Request all items

To read all of your Edge Config's items, send a `GET` request to the appropriate edge config endpoint by adding your Edge Config's ID and

\['cURL'

```bash filename="cURL"
curl 'https://edge-config.vercel.com/your_edge_config_id_here/items?token=your_edge_config_read_access_token_here'
```

'fetch']

```javascript filename="fetch"
try {
  const readAllEdgeConfigItems = await fetch(
    'https://edge-config.vercel.com/your_edge_config_id_here/items?token=your_edge_config_read_access_token_here',
  );
  const result = await readAllEdgeConfigItems.json();
  console.log(result);
} catch (error) {
  console.log(error);
}
```

You can also send your Edge Config read access token in an Authorization header rather than as a query param.

\['cURL'

```bash filename="request"
curl "https://edge-config.vercel.com/your_edge_config_id_here/items" \
     -H 'Authorization: Bearer your_edge_config_read_access_token_here'
```

'fetch']

```javascript filename="fetch"
try {
  const readAllWithAuth = await fetch(
    'https://edge-config.vercel.com/your_edge_config_id_here/items',
    {
      method: 'GET',
      headers: {
        Authorization: `Bearer ${your_edge_config_read_access_token_here}`,
      },
```

```
    },
  );
  const result = await readAllWithAuth.json();
  console.log(result);
} catch (error) {
  console.log(error);
}
```

The response will be a JSON object containing all key-value pairs in the Edge Config. For example:

```json filename="response"
{
  "example_key_1": "example_value_1",
  "example_key_2": "example_value_2",
  "example_key_3": "example_value_3"
}
```

Request a single item

To request a single item, you can use the `/item` path instead of `/items`, then add the key of the item you want as the final path as sh

\['cURL'

```bash filename="request"
curl "https://edge-config.vercel.com/your_edge_config_id_here/item/example_key_1?token=your_edge_config_read_access_token_here" \
```

'fetch']

```javascript filename="fetch"
try {
  const readSingle = await fetch(
    'https://edge-config.vercel.com/your_edge_config_id_here/item/example_key_1?token=your_edge_config_read_access_token_here',
  );
  const result = await readSingle.json();
  console.log(result);
} catch (error) {
  console.log(error);
}
```

You can also send your Edge Config read access token in an Authorization header rather than as a query param.

\['cURL'

```bash filename="request"
curl -X 'https://edge-config.vercel.com/your_edge_config_id_here/item/example_key_1' \
     -H 'Authorization: Bearer your_edge_config_read_access_token_here'
```

'fetch']

```javascript filename="fetch"
try {
  const readSingleWithAuth = await fetch(
    'https://edge-config.vercel.com/your_edge_config_id_here/item/example_key_1',
    {
      method: 'GET',
      headers: {
        Authorization: `Bearer ${your_edge_config_read_access_token_here}`,
      },
    },
  );
  const result = await readSingleWithAuth.json();
  console.log(result);
} catch (error) {
  console.log(error);
}
```

The response will be the raw value at the specified key. For example, if `example_key_1` has a string value of `"example_value"`, the res

```bash filename="response"
"example_value"
```

Request the digest

When you create an Edge Config, a hash string called a digest is generated and attached to it. This digest is replaced with a new hash st

To fetch an Edge Config's digest, send a `GET` request to your edge config endpoint, as shown below:

\['cURL'

```bash filename="request"
curl "https://edge-config.vercel.com/your_edge_config_id_here/digest?teamId=your_team_id_here&token=your_edge_config_read_access_token_he
```

'fetch']

```javascript filename="fetch"
try {
  const readDigest = await fetch(
    'https://edge-config.vercel.com/your_edge_config_id_here/digest?teamId=your_team_id_here&token=your_edge_config_read_access_token_her
  );
  const result = await readDigest.json();
  console.log(result);
} catch (error) {
```

```
  console.log(error);
}
```

You can also send the Edge Config read access token in the `Authorization` header of your request using the `Bearer token` format:

\['cURL'

```bash filename="request"
curl  -X 'GET' 'https://edge-config.vercel.com/your_edge_config_id_here/digest?teamId=your_team_id_here' \
      -H 'Authorization: Bearer your_edge_config_read_access_token_here
```

'fetch']

```javascript filename="fetch"
try {
  const readDigestWithAuth = await fetch(
    'https://edge-config.vercel.com/your_edge_config_id_here/digest?teamId=your_team_id_here',
    {
      method: 'GET',
      headers: {
        Authorization: `Bearer ${your_edge_config_read_access_token_here}`,
      },
    },
  );
  const result = await readDigestWithAuth.json();
  console.log(result);
} catch (error) {
  console.log(error);
}
```

Up Next

---
title: "Edge Middleware"
description: "Learn how you can use Edge Middleware, code that executes before a request is processed on a site, to provide speed and per
last_updated: "2026-01-16T02:19:29.873Z"
source: "https://vercel.com/docs/edge-middleware"

Edge Middleware

Edge Middleware is **code that executes *before* a request is processed on a site**. Based on the request, you can modify the response. B

Middleware uses the [Edge runtime](/docs/functions/runtimes/edge), which exposes and extends a subset of Web Standard APIs such `FetchEve

> **💡 Note:** You can use the Node.js runtime (as an experimental feature) to run
> middleware. For more details, see the [Node.js
> runtime](/docs/functions/runtimes/node-js#using-node.js-with-middleware)
> documentation.

Create Edge Middleware

You can use Edge Middleware with **any framework**. To add Middleware to your app, you need to create a file at your project's root dire

> For \['nextjs', 'nextjs-app']:

Logging

Edge Middleware has full support for the [`console`](https://developer.mozilla.org/docs/Web/API/Console) API, including `time`, `debug`,

Using a database with Edge Middleware

If your Edge Middleware depends on a database far away from one of [our regions](/docs/regions), the overall latency of API requests coul

---
title: "Edit Mode"
description: "Discover how Vercel"
last_updated: "2026-01-16T02:19:30.001Z"
source: "https://vercel.com/docs/edit-mode"

Edit Mode

Content editing in CMSs usually occurs separately from the website's layout and design. This separation makes it hard for authors to visu

Accessing Edit Mode

To access Edit Mode:

1. Ensure you're logged into the [Vercel Toolbar](/docs/vercel-toolbar) with your Vercel account.
2. Navigate to a page with editable content. The **Edit Mode** option will only appear in the [Vercel Toolbar](/docs/vercel-toolbar) men
3. Select the **Edit Mode** option in the toolbar menu. This will highlight the editable fields as [Content Links](/docs/edit-mode#conte

Content Link

Content Link enables you to edit content on websites using headless CMSs by providing links on elements that match a content model in the

You can enable Content Link on a preview deployment by selecting **Edit Mode** in the [Vercel Toolbar](/docs/vercel-toolbar) menu.

The corresponding model in the CMS determines an editable field. You can hover over an element to display a link in the top-right corner

You don't need any additional configuration or code changes on the page to use this feature.

The following CMS integrations support Content Link:

-
-
-
-
-
-
-
-
-

See the [CMS integration documentation](/docs/integrations/cms) for information on how to use Content Link with your chosen CMS.

---
title: "Encryption"
description: "Learn how Vercel encrypts data in transit and at rest."
last_updated: "2026-01-16T02:19:29.895Z"
source: "https://vercel.com/docs/encryption"

Encryption

Out of the box, every **deployment** on Vercel is served over an HTTPS connection. The [SSL](https://en.wikipedia.org/wiki/Transport_Laye

Any HTTP requests to your **deployment** are automatically forwarded to HTTPS using the `308` status code:

```bash
HTTP/1.1 308 Moved Permanently
Content-Type: text/plain
Location: https://<your-deployment-host>
```

Enabling HTTPS redirection for deployments is considered an industry standard, and therefore it is not possible to disable it. This ensur

> **💡 Note:** If the client that is issuing requests to your **deployment** wants to
> establish a WebSocket connection, please ensure it is connecting using HTTPS.
> directly, as the WSS protocol does not support redirections.

Supported TLS versions

Vercel supports TLS version [1.2](https://en.wikipedia.org/wiki/Transport_Layer_Security#TLS_1.2) and TLS version [1.3](https://en.wikipe

TLS resumption

Vercel supports both Session Identifiers and Session Tickets as methods for [resuming a TLS connection](https://hpbn.co/transport-layer-s

OCSP stapling

To ensure clients can validate TLS certificates as quickly as possible, we [staple an OCSP response](https://en.wikipedia.org/wiki/OCSP_s

Supported ciphers

In order to ensure the integrity of the data received and sent by any **deployment** running on the **Vercel** platform, we only support

The following cipher algorithms are supported:

- `TLS_AES_128_GCM_SHA256` (TLS 1.3)
- `TLS_AES_256_GCM_SHA384` (TLS 1.3)
- `TLS_CHACHA20_POLY1305_SHA256` (TLS 1.3)
- `ECDHE-ECDSA-AES128-GCM-SHA256` (TLS 1.2)
- `ECDHE-RSA-AES128-GCM-SHA256` (TLS 1.2)
- `ECDHE-ECDSA-AES256-GCM-SHA384` (TLS 1.2)
- `ECDHE-RSA-AES256-GCM-SHA384` (TLS 1.2)
- `ECDHE-ECDSA-CHACHA20-POLY1305` (TLS 1.2)
- `ECDHE-RSA-CHACHA20-POLY1305` (TLS 1.2)
- `DHE-RSA-AES256-GCM-SHA384` (TLS 1.2)

This is the [recommended configuration from Mozilla](https://wiki.mozilla.org/Security/Server_Side_TLS#Intermediate_compatibility_.28reco

Post-quantum cryptography

Vercel offers the `X25519MLKEM768` key exchange mechanism during TLS handshakes, which protects your deployments against future quantum c

- Chrome 131 and above
- Firefox 132 and above
- Safari 26 and above

Support for HSTS

The `.vercel.app` domain (and therefore all of its sub domains, which are the unique URLs set when creating a deployment) support [HSTS](

```bash
Strict-Transport-Security: max-age=63072000; includeSubDomains; preload;
```

Custom domains use HSTS, but only for the particular subdomain.

```bash
Strict-Transport-Security: max-age=63072000;
```

You can modify the `Strict-Transport-Security` header by configuring [custom response headers](/docs/headers/cache-control-headers#custom

Theoretically, you could set the `max-age` parameter to a different value (it indicates how long the client should remember that your sit

> **💡 Note:** You can test whether your site qualifies for HSTS Preloading
> [here](https://hstspreload.org/). It also allows submitting the domain to
> Google Chrome's hardcoded HSTS list. Making it onto that list means your site
> will become even faster, as it is always accessed over HTTPS right away,
> instead of the browser following the redirection issued by our **Network**
> layer.

How certificates are handled

The unique URLs generated when creating a deployment are handled using a wildcard certificate issued for the `.vercel.app` domain. The **

When custom certificates are generated using `vercel certs issue`, however, their keys are placed in our database and [encrypted at rest]

Then, once a hostname is requested, the certificate and key are read from the database and used for establishing the secure connection. I

Full specification

Any features of the encryption mechanism that were left uncovered are documented on [SSL Labs](https://www.ssllabs.com/ssltest/analyze.ht

---
title: "Framework environment variables"
description: "Framework environment variables are automatically populated by the Vercel, based on your project"
last_updated: "2026-01-16T02:19:29.899Z"
source: "https://vercel.com/docs/environment-variables/framework-environment-variables"

Framework environment variables

Frameworks typically use a prefix in order to expose environment variables to the browser.

The following prefixed environment variables will be available during the **build step**, based on the project's selected [framework pres

Using prefixed framework environment variables locally

Many frontend frameworks require prefixes on environment variable names to make them available to the client, such as `NEXT_PUBLIC_` for

Framework environment variables are not prefixed when pulled into your local development environment with `vercel env pull`. For exam

To use framework-prefixed environment variables locally:

1. [Define them in your project settings](/docs/environment-variables#creating-environment-variables) with the appropriate prefix
2. Scope them to `Development`
3. Pull them into your local environment with Vercel CLI using the `vercel env pull` command

Framework environment variables

---
title: "Managing environment variables"
description: "Learn how to create and manage environment variables for Vercel."
last_updated: "2026-01-16T02:19:29.916Z"
source: "https://vercel.com/docs/environment-variables/managing-environment-variables"

Managing environment variables

Environment variables are key-value pairs configured outside your source code so that each value can change depending on the [Environment

Changes to environment variables are not applied to previous deployments, they only apply to new deployments. You must redeploy your proj

Declare an Environment Variable

To declare an Environment Variable for your deployment:

1. From your [dashboard](/dashboard), select your project. If necessary, you can also set environment variables team-wide so that they wi

2. Select the **Settings** tab.

3. Go to the **Environment Variables** section of your **Project Settings**.

4. Enter the desired **Name** for your Environment Variable. For example, if you are using Node.js and you create an Environment Variable

   #### \['Node.js'

   ```js
   process.env.API_URL;
   ```

   #### 'Go'

   ```go
   os.Getenv("API_URL")
   ```

   #### 'Python'

   ```py
   os.environ.get('API_URL')
   ```

   #### 'Ruby']

   ```ruby
   ENV['API_URL']
   ```

5. Then, enter the **Value** for your Environment Variable. The value is encrypted at rest so it is safe to add sensitive data like authe

6. Configure which [deployment environment(s)](/docs/deployments/environments) this variable should apply to.

7. Click **Save**.

8. To ensure that the new Environment Variable is applied to your deployment, you must [redeploy](/docs/deployments/managing-deployments#

Viewing, editing, or deleting an Environment Variable

To find and view all environment variables.

1. From your [dashboard](/dashboard), select your project. You can also view all team-wide environment variables through the Team Setting
2. Select the **Settings** tab.
3. Go to the **Environment Variables** section of your **Project Settings**.
4. Below the *Add New* form is a list of all the environment variables for the Project.
5. You can search for an existing Environment Variable by name using the search input and/or filter by [Environment](/docs/deployments/en
6. To edit or delete the Environment Variable, click the three dots to the right of the Environment Variable name.

---
title: "Environment variables"
description: "Learn more about environment variables on Vercel."
last_updated: "2026-01-16T02:19:30.010Z"
source: "https://vercel.com/docs/environment-variables"

Environment variables

Environment variables are key-value pairs configured outside your source code so that each value can change depending on the [Environment

Your source code can read these values to change behavior during the [Build Step](/docs/deployments/configure-a-build) or during [Functio

Any change you make to environment variables are not applied to previous deployments, they only apply to new deployments.

Creating environment variables

Environment variables can either be declared at the team or project level. When declared at the team level, they are available to all pro

To learn how to create and manage environment variables, see [Managing environment variables](/docs/environment-variables/managing-enviro

Environment Variable size

Developers on all plans using the runtimes stated below can use a total of **64 KB** in Environments Variables **per-Deployment** on Verc

With support for 64 KB of environment variables, you can add large values for authentication tokens, JWTs, or certificates.

Deployments using the following runtimes can support environment variables up to 64 KB:

- Node.js
- Python
- Ruby
- Go
- [PHP Community Runtime](https://github.com/vercel-community/php)

Vercel also provides support for custom runtimes, through the Build Output API. For information on creating custom runtime support, see t

- [Guides for runtime builders](https://github.com/vercel/vercel/blob/main/DEVELOPING_A_RUNTIME.md#supporting-large-environment)
- [Build Output API documentation](/docs/build-output-api/v3/primitives#base-config)

> **💡 Note:** While Vercel allows environment variables up to a total of 64KB in size, Edge
> Functions and Middleware using the `edge` runtime are limited to 5KB per
> Environment Variable.

Environments

For each Environment Variable, you can select one or more Environments to apply the Variable to:

Environment	Description
[**Production**](/docs/deployments/environments#production-environment)	When selected, the Environment Variable will be applied
[**Preview**](#preview-environment-variables)	The Environment Variable is applied to your next Previe
[**Custom environments**](/docs/deployments/environments#custom-environments)	With custom environments you can choose to [import envi
[Development](#development-environment-variables)	The Environment Variable is used when running your proj

Preview environment variables

> **💡 Note:** You need Vercel CLI version 22.0.0 or higher to use the features described in
> this section.

Preview environment variables are applied to deployments from any Git branch that does not match the [Production Branch](/docs/git#produc

Any branch-specific variables will override other preview environment variables with the same name. This means you don't need to replicat

Development environment variables

> **💡 Note:** You need Vercel CLI version 21.0.1 or higher to use the features described in
> this section.

Environment variables for local development are defined in the `.env.local` file. This is a plain text file that contains `key=value` pai

You can use the `vercel env pull` command to automatically create and populate the `.env` file (which serves the same purpose as `.env.lo

This command creates a `.env` file in your project's current directory with the environment variables from your Vercel project's **Develo

If you're using [`vercel dev`](/docs/cli/dev), there's no need to run `vercel env pull`, as `vercel dev` automatically downloads the Deve

For more information, see [Environment variables for local development](/docs/deployments/local-env#environment-variables-for-local-develo

Integration environment variables

[Integrations](/docs/integrations) can automatically add environment variables to your Project Settings.
In that case, the Integration that added the Variable will be displayed in your project settings:

---
title: "Reserved environment variables"
description: "Reserved environment variables are reserved by Vercel Vercel Function runtimes."
last_updated: "2026-01-16T02:19:29.942Z"

source: "https://vercel.com/docs/environment-variables/reserved-environment-variables"

Reserved environment variables

The following [environment variable](/docs/environment-variables) names are [reserved](https://docs.aws.amazon.com/lambda/latest/dg/confi

- `AWS_SECRET_KEY`
- `AWS_EXECUTION_ENV`
- `AWS_LAMBDA_LOG_GROUP_NAME`
- `AWS_LAMBDA_LOG_STREAM_NAME`
- `AWS_LAMBDA_FUNCTION_NAME`
- `AWS_LAMBDA_FUNCTION_MEMORY_SIZE`
- `AWS_LAMBDA_FUNCTION_VERSION`
- `NOW_REGION`
- `TZ`
- `LAMBDA_TASK_ROOT`
- `LAMBDA_RUNTIME_DIR`

Allowed environment variables

The following [environment variable](/docs/environment-variables) names are [allowed](/kb/guide/how-can-i-use-aws-sdk-environment-variabl

- `AWS_ACCESS_KEY_ID`
- `AWS_SECRET_ACCESS_KEY`
- `AWS_SESSION_TOKEN`
- `AWS_REGION`
- `AWS_DEFAULT_REGION`

> **💡 Note:** These variables may appear in your Vercel Functions even if you don't set them in your project explicitly. These values do

---
title: "Rotating environment variables"
description: "Safely rotate API keys, tokens, and other secrets in your Vercel environment variables."
last_updated: "2026-01-16T02:19:29.953Z"
source: "https://vercel.com/docs/environment-variables/rotating-secrets"

Rotating environment variables

> **💡 Note:** Find guides for rotating secrets for our Marketplace providers in [Vercel's Knowledge Base](https://vercel.com/kb/integrat

When you need to rotate API keys, tokens, or other credentials stored in your [environment variables](/docs/environment-variables), you'l

Secret rotation is a security best practice that limits the exposure window if a credential is compromised. You might rotate secrets when

- A team member with access to credentials leaves
- You suspect a credential has been exposed
- Your security policy requires periodic rotation
- A third-party service forces a credential update

Rotating secrets safely

The key to safe rotation is updating Vercel *before* you invalidate the old credential. This prevents your deployments from breaking when

For project-level environment variables

If your secret is configured at the [project level](/docs/environment-variables), only that project uses it. You'll only need to redeploy

1. Go to your third-party service (like a database provider, API service, or integration) and generate a new credential. Don't delete or
2. From your Vercel [dashboard](/dashboard), select the project that uses this credential.
3. Go to **Settings** > **Environment Variables**.
4. Find the environment variable that stores the credential. Click the three dots to the right and select **Edit**.
5. Replace the old value with your new credential.
6. Make sure the correct [environments](/docs/deployments/environments) are selected (Production, Preview, and/or Development).
7. Click **Save**.
8. [Redeploy your project](/docs/deployments/managing-deployments#redeploy-a-project) to apply the new credential:
   - Go to the **Deployments** tab.
   - Find your latest production deployment.
   - Click the three dots and select **Redeploy**.
   - Once the new deployment succeeds and you've verified it works, go back to your third-party service and invalidate the old credential

> **💡 Note:** If you're using the [Preview environment](/docs/deployments/environments#preview-environment-pre-production), redeploy you

For team-level environment variables

If your secret is configured at the [team level](/docs/environment-variables/shared-environment-variables), multiple projects might use i

1. Go to your third-party service and generate a new credential. Keep the old one active for now.
2. From your Vercel [dashboard](/dashboard), select your team from the scope selector.
3. Go to **Settings** > **Environment Variables**.
4. Find the environment variable that stores the credential. Click the three dots to the right and select **Edit**.
5. Replace the old value with your new credential.
6. Make sure the correct [environments](/docs/deployments/environments) are selected (Production, Preview, and/or Development).
7. Click **Save**.
8. Identify which projects use this credential. You can check the **Link to Projects** field in the environment variable form to see whic
9. Redeploy each project that uses this credential:
   - Go to each project's **Deployments** tab.
   - Find the latest production deployment.
   - Click the three dots and select **Redeploy**.
10. Once all deployments succeed and you've verified they work, go back to your third-party service and invalidate the old credential.

> **⚠️ Warning:** If you invalidate the old credential before all projects are redeployed, any project still using the old value will fai

Rotating credentials for integrations

> **💡 Note:** Find guides for rotating secrets for our Marketplace providers in [Vercel's Knowledge Base](https://vercel.com/kb/integrat

When rotating credentials for [integrations](/docs/integrations) (like database providers or third-party services installed from the Verc

1. Navigate to the integration from the **Integrations** tab and then by selecting the provider you want to rotate credentials for
2. Click "Log into provider" (if there are multiple resources for the integration, you will need to do this for each resource)
3. Manually reset the credentials in the provider's dashboard (this will synchronize the new secrets to your Vercel team and stage them t
4. Redeploy the projects that use this integration or resource.
5. Test your application to confirm the new credential works.
6. Return to your integration's dashboard and revoke or delete the old credential (if the provider supports this).

> **💡 Note:** If the integration allows for provisioning resources, you will have to repeat this process for each resource that you've p

Troubleshooting

Deployment fails after rotating a secret

If your deployment fails after updating a credential:

- Check that you copied the new credential correctly (no extra spaces or line breaks).
- Verify that the new credential is active in your third-party service.
- Make sure you selected the correct environment (Production, Preview, or Development) when updating the variable.
- Check your application logs in the Vercel dashboard for specific error messages.

Old deployments still use the old credential

Environment variable changes only apply to new deployments. If you visit an old deployment URL, it will still use the old credential. Thi

Once you invalidate the old credential, old deployments that relied on it will fail if they make API calls using that credential. Redeplo

Multiple projects broke after rotation

If you rotated a team-level environment variable and multiple projects broke, you may have missed redeploying some projects:

1. Go to your team's **Environment Variables** settings.
2. Find the variable you rotated and check which projects have access to it.
3. Redeploy any projects you missed.

---
title: "Sensitive environment variables"
description: "Environment variables that cannot be decrypted once created."
last_updated: "2026-01-16T02:19:29.964Z"
source: "https://vercel.com/docs/environment-variables/sensitive-environment-variables"

Sensitive environment variables

Sensitive environment variables are [environment variables](/docs/environment-variables "Environment variables") whose values are non-rea

To mark an existing environment variable as sensitive, remove and re-add it with the **Sensitive** option enabled. Once you mark it as se

Both [project environment variables](/docs/environment-variables) and [shared environment variables](/docs/environment-variables/shared-e

Creating sensitive environment variables

> **💡 Note:** You can only create sensitive environment variables in the preview and
> production environments.

\['Dashboard'

Sensitive environment variables can be created at the project or team level:

1. Go to the Vercel [dashboard](/dashboard) and select your team from the scope selector. Click on the **Settings** tab and then select *
2. At the top of the form, toggle the **Sensitive** switch to **Enabled**. If the **Development** environment is selected, you will be un
3. Fill in the details to create a new environment variable.
4. In the environment variable table, sensitive environment variables are marked with a "Sensitive" tag:

'cURL'

To create an Authorization Bearer token, see the [access token](/docs/rest-api/reference/welcome#creating-an-access-token) section of the

```bash filename="cURL"
curl --request POST \
  --url https://api.vercel.com/v10/projects/<project-id-or-name>/env \
  --header "Authorization: Bearer $VERCEL_TOKEN" \
  --header "Content-Type: application/json" \
  --data '[
    {
      "key": "<env-key-1>",
      "value": "<env-value-1>",
      "type": "sensitive",
      "target": ["<target-environment>"],
      "gitBranch": "<git-branch>",
      "comment": "<comment>",
      "customEnvironmentIds": ["<custom-env-id>"]
    }
  ]'
```

'SDK']

To create an Authorization Bearer token, see the [access token](/docs/rest-api/reference/welcome#creating-an-access-token) section of the

```ts filename="createProjectEnv"
import { Vercel } from '@vercel/sdk';

const vercel = new Vercel({
  bearerToken: '<YOUR_BEARER_TOKEN_HERE>',
});

async function run() {
  const result = await vercel.projects.createProjectEnv({
```

```
    idOrName: '<project-id-or-name>',
    requestBody: {
      key: '<env-key-1>',
      value: '<env-value-1>',
      type: 'sensitive',
      target: ['<target-environment>'],
      gitBranch: '<git-branch>',
      comment: '<comment>',
      customEnvironmentIds: ['<custom-env-id>'],
    },
  });

  // Handle the result
  console.log(result);
}

run();
```

Edit sensitive environment variables

You can edit the value and [environment](/docs/environment-variables#environments) for a sensitive environment variable. You cannot edit

1. From your dashboard, go to the team or project's **Settings** tab and select **Environment Variables** from the left navigation. Find
2. Click **Edit** from the three-dot menu in the environment variables list
3. Provide a new value for the sensitive environment variable. The current value is hidden.
4. Select the environment(s) for the sensitive environment variable.
5. After making the change, click the **Save** button.

Environment variables policy

Users with the [owner](/docs/rbac/access-roles#owner-role) role can set a team-wide environment variable policy for creating environment

1. From the dashboard, ensure your team is selected in the scope selector and select the **Settings** tab.
2. From the left navigation, click **Security & Privacy**.
3. From the **Environment Variable Policies** section, toggle the **Enforce Sensitive Environment Variables** switch to **Enabled**:

---
title: "Shared environment variables"
description: "Learn how to use Shared environment variables, which are environment variables that you define at the Team level and can li
last_updated: "2026-01-16T02:19:29.984Z"
source: "https://vercel.com/docs/environment-variables/shared-environment-variables"

Shared environment variables

Shared Environment Variables are [environment variables](/docs/environment-variables "Environment variables") that you define at the

> **💡 Note:** When a project-level and a Shared Environment Variable share the same key and
> environment, the project-level environment variable always overrides the
> Shared Environment Variable.

Creating shared environment variables

Shared Environment Variables are created on the [Team Settings page](/docs/accounts/create-a-team). To create a new Shared Environment Va

1. Go to the Vercel [dashboard](/dashboard) and select your team from the scope selector. Click on the **Settings** tab and then select *
2. Populate the form with your environment variable details or paste or import an `.env` file:
   - **Key**: Fill in the key of the environment variable. - **Value**: Fill in
     the value of the environment variable. - **Environment**: Select the
     [Environments](/docs/environment-variables#environments) where you want to
     include it. The environment(s) chosen for the Shared Environment Variable is
     used when linked to a project. - **Link to Projects**: Select one or more
     [projects](/docs/projects/overview) in succession to link the new Shared
     Environment Variable by using the searchable drop-down. You can keep this
     empty and [link to projects](#linking-to-projects) later.
3. Click **Save** to save your new Shared Environment Variable.

Linking to projects

A Shared Environment Variable is activated once it is linked to at least one project.

You can link an existing Shared Environment Variable to a project either at the project-level or the team-level.

Project level linking

For project-level linking:

1. From your [dashboard](/dashboard), select the Project you would like to link the Shared Environment Variable to and click the **Settin
2. Select **Environment Variables** from the list, and click on the **Link Shared Environment Variables** tab.
3. Select one or more Shared Environment Variables using the search box:
4. Click the **Link** button

Team level linking

1. From your [dashboard](/dashboard), click the **Settings** tab and go to **Environment Variables**.

2. Scroll down below the Shared Environment Variable creation form.

3. Find the variable you would like to link. You can use the **Search** box, the **Environments** drop-down filter and sort by **last upd

4. Open the context menu for the specific Shared Environment Variable using the vertical ellipsis icon on the right hand side of the row

5. From the Environment Variable form, you can link additional projects using the **Link to Projects** field

6. Click **Save** when you are done

Removing shared environment variables

There are two ways to remove a Shared Environment Variable from a project:

- **Unlinking**: It is disassociated from the selected project(s) but continues to exist at the level of the team
- **Deleting**: It is **permanently** removed from the team and disconnected from all projects it was previously linked to.

Unlinking at the project level

1. From your [dashboard](/dashboard), select the project you would like to unlink the Shared Environment Variable from and click the **Se
2. Select **Environment Variables**, and scroll down to the **Shared Environment Variables** section.
3. Open the context menu for the specific shared environment variable you would like to unlink using the vertical ellipsis icon on the r
4. Click **Unlink from this Project**:

Unlinking at the team level

1. From your [dashboard](/dashboard), click the **Settings** tab and go to **Environment Variables**.
2. Scroll down below the Environment Variable creation form.
3. Find the variable you would like to link. You can use the **Search** box, the **Environments** drop-down filter and sort by **last upd
4. Open the context menu for the specific shared environment variable using the vertical ellipsis icon on the right hand side of the row
5. From the Environment Variable form, click the minus icon to unlink existing projects
6. When you are done, click the **Save** button.

Deleting environment variables from a team

1. From your [dashboard](/dashboard), click the **Settings** tab and go to **Environment Variables**.
2. Scroll down below the Environment Variable creation form
3. Use the context menu on the specific Shared Environment Variable by clicking the vertical ellipsis icon on the right side of the row
4. Click the **Delete** button

Known limitations

[Branch-specific variables](/docs/environment-variables#preview-environment-variables) are not currently supported with Shared Environmen

---
title: "System environment variables"
description: "System environment variables are automatically populated by Vercel, such as the URL of the deployment or the name of the Gi
last_updated: "2026-01-16T02:19:30.016Z"
source: "https://vercel.com/docs/environment-variables/system-environment-variables"

System environment variables

Vercel provides a set of environment variables that are automatically populated by the system, such as the URL of the deployment or the n

Automatically expose system environment variables

To expose these environment variables to your deployments:

1. Navigate to your project on your [dashboard](/dashboard).
2. Go to the **Settings** tab and click on the **Environment Variables** section.
3. Select the **Automatically expose System Environment Variables** checkbox.

> **⚠️ Warning:** If you disable this setting, no deployment ID will be made available for supported frameworks (like Next.js) to use, wh

System environment variables

If you are using a framework for your project, Vercel provides the following prefixed environment variables:

> **💡 Note:** When you choose to automatically expose system environment variables, some
> React warnings, such as those in a `create-react-app` will display as build
> errors. For more information on this error, see [How do I resolve a
> `process.env.CI = true`
> error?](/kb/guide/how-do-i-resolve-a-process-env-ci-true-error)

---
title: "BODY_NOT_A_STRING_FROM_FUNCTION"
description: "The function returned a non-string body. This is a function error."
last_updated: "2026-01-16T02:19:30.085Z"
source: "https://vercel.com/docs/errors/BODY_NOT_A_STRING_FROM_FUNCTION"

BODY_NOT_A_STRING_FROM_FUNCTION

The `BODY_NOT_A_STRING_FROM_FUNCTION` error occurs when a function returns a body that is not a string. It's essential that functions ret

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check function return type:** Ensure that the function is structured to return a string. If the function is returning a different da
2. **Review function code:** Inspect the function code for any logic that might cause a non-string value to be returned
3. **Check data types:** If the function is processing input data or retrieving data from external sources, ensure that the data is being
4. **Review function logs:** Check the [function logs](/docs/runtime-logs#type) for any errors or warnings that might indicate why a non-

---
title: "DEPLOYMENT_BLOCKED"
description: "The deployment was blocked due to certain conditions. This is a deployment error."
last_updated: "2026-01-16T02:19:30.098Z"
source: "https://vercel.com/docs/errors/DEPLOYMENT_BLOCKED"

DEPLOYMENT_BLOCKED

The `DEPLOYMENT_BLOCKED` error occurs when a deployment is blocked due to certain conditions that prevent it from proceeding. This could

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check configuration:** Ensure that your deployment configuration is correct and complies with the platform's requirements
2. **Check your account plan**: If you have recently downgraded to the [Hobby plan](/docs/plans/hobby), you may need to redeploy your pro
3. **Review email notifications**: If you receive an email from Vercel about the pause, it may contain more details about the issue and n
4. **Verify account status:** Ensure your account is in good standing and hasn't exceeded any [limits or quotas](/docs/limits)
5. **Review policies:** Ensure that your deployment complies with all platform [policies](/legal/privacy-policy) and isn't in violation o
6. **Check for platform outages:** Sometimes, platform-wide outages or issues can cause deployments to be blocked. Check the [status page
7. **Contact support:** If you've verified the above and are still experiencing the issue, [contact support](/help#issues) for further as

---
title: "DEPLOYMENT_DELETED"
description: "The deployment has been removed"
last_updated: "2026-01-16T02:19:30.020Z"
source: "https://vercel.com/docs/errors/DEPLOYMENT_DELETED"

DEPLOYMENT_DELETED

The `DEPLOYMENT_DELETED` error occurs when a request is made for a deployment that has been removed based on the projects deployment rete

Troubleshoot

Recently deleted deployments can be restored within 30 days of deletion.

To restore a deleted deployment, navigate to the **Settings** tab of your project:

1. Select **Security** on the side panel of the project settings page
2. Scroll down to the **Recently Deleted** section
3. Find the deployment that needs to be restored, and click on the dropdown menu item **Restore**
4. Complete the modal

---
title: "DEPLOYMENT_DISABLED"
description: "The deployment is disabled. This is a deployment error."
last_updated: "2026-01-16T02:19:30.037Z"
source: "https://vercel.com/docs/errors/DEPLOYMENT_DISABLED"

DEPLOYMENT_DISABLED

The `DEPLOYMENT_DISABLED` error occurs when a deployment is disabled due to certain conditions or configurations. This might happen if th

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check your usage**: Check the specific [usage limits](/dashboard/usage) you've exceeded in the [Vercel dashboard](/dashboard/usage)
2. **Check your account plan**: If you have recently downgraded to the [Hobby plan](/docs/plans/hobby), you may need to redeploy your pro
3. **Review email notifications**: If you receive an email from Vercel about the pause, it may contain more details about the issue and n
4. **Restore your site**: The fastest solution is to [upgrade to the Pro plan](/docs/plans/pro-plan). This plan offers more generous usag
5. **Contact support:** If you've checked the above and are still unable to resolve the issue, [contact support](/help#issues) for furthe

---
title: "DEPLOYMENT_NOT_FOUND"
description: "The deployment was not found. This is a deployment error."
last_updated: "2026-01-16T02:19:30.060Z"
source: "https://vercel.com/docs/errors/DEPLOYMENT_NOT_FOUND"

DEPLOYMENT_NOT_FOUND

The `DEPLOYMENT_NOT_FOUND` error occurs when a request is made for a deployment that doesn't exist. This could happen if the deployment I

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check the deployment URL**: Ensure that the deployment URL you are using is correct and does not contain any typos or incorrect path
2. **Check deployment existence:** Verify that the [deployment exists](/docs/projects/project-dashboard#deployments) and has not been del
3. **Review deployment logs:** If the deployment exists, review the [deployment logs](/docs/deployments/logs) to identify any issues that
4. **Verify permissions:** Ensure you have the necessary [permissions](/docs/accounts/team-members-and-roles) to access the deployment
5. **Consult community resources:** Visit our [community post on debugging 404 errors](https://community.vercel.com/t/debugging-404-error
6. **Contact support:** If you've checked the above and are still unable to resolve the issue, [contact support](/help#issues) for furthe

---
title: "DEPLOYMENT_NOT_READY_REDIRECTING"
description: "The deployment is not ready and is redirecting to another location. This is a deployment error."
last_updated: "2026-01-16T02:19:30.042Z"
source: "https://vercel.com/docs/errors/DEPLOYMENT_NOT_READY_REDIRECTING"

DEPLOYMENT_NOT_READY_REDIRECTING

The `DEPLOYMENT_NOT_READY_REDIRECTING` error occurs when the requested deployment is not yet ready, and the request is redirected to the

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check deployment status:** Ensure that the [deployment process](/docs/projects/project-dashboard#deployments) has completed successf
2. **Inspect deployment logs:** Review the [deployment logs](/docs/deployments/logs) for any indications as to why the deployment is not
3. **Verify Configuration:** Check the configuration settings to ensure they are correct and that there are no misconfigurations
4. **Wait and refresh**: If you encounter this error, wait for a few seconds and then refresh the page. In some cases, the deployment may

---

title: "DEPLOYMENT_PAUSED"
description: "The deployment was paused. This is a deployment error."
last_updated: "2026-01-16T02:19:30.055Z"
source: "https://vercel.com/docs/errors/DEPLOYMENT_PAUSED"

DEPLOYMENT_PAUSED

The `DEPLOYMENT_PAUSED` error occurs when a deployment is paused due to certain conditions or configurations. This might happen if there'

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check configuration:** Ensure that your deployment configuration is correct and complies with the platform's requirements
2. **Review your spend management**: You may have configured your deployments to pause once your spend amount is reached. Review your [sp
3. **Verify account status:** Ensure your account is in good standing and hasn't exceeded any [limits or quotas](/docs/limits)
4. **Review email notifications**: If your account or deployment has been paused, Vercel will email you to share more details about the p
5. **Check for terms of service violations**: If the pause is due to a breach of the [terms of service](/legal/terms) or [fair use guidel
6. **Check for platform outages:** Sometimes, platform-wide outages or issues can cause deployments to be blocked. Check the [status page
7. **Contact support:** If you've verified the above and are still experiencing the issue, [contact support](/help#issues) for further as

---
title: "DNS_HOSTNAME_EMPTY"
description: "An empty DNS record was received as part of the DNS response. This is a DNS error."
last_updated: "2026-01-16T02:19:30.076Z"
source: "https://vercel.com/docs/errors/DNS_HOSTNAME_EMPTY"

DNS_HOSTNAME_EMPTY

The `DNS_HOSTNAME_EMPTY` error occurs when an empty DNS record is received as part of the DNS response while attempting to connect to a p

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Review DNS configuration:** Check the [DNS configuration](/docs/domains/working-with-dns) to ensure that it's correctly set up and d
2. **Check for private IP addresses:** Ensure that the request isn't attempting to connect to a private IP address from an external sourc
3. **Review application logs:** Inspect the [application logs](/docs/deployments/logs) for any warnings or errors related to DNS or the a

---
title: "DNS_HOSTNAME_NOT_FOUND"
description: "The domain does not exist, resulting in an NXDOMAIN error during DNS resolution. This is a DNS error."
last_updated: "2026-01-16T02:19:30.104Z"
source: "https://vercel.com/docs/errors/DNS_HOSTNAME_NOT_FOUND"

DNS_HOSTNAME_NOT_FOUND

The `DNS_HOSTNAME_NOT_FOUND` error occurs when there's an `NXDOMAIN` error during the DNS resolution while attempting to connect to a pri

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Review DNS configuration:** Check the [DNS configuration](/docs/domains/working-with-dns) to ensure that the domain being requested
2. **Verify domain registration:** Ensure that the domain has been [registered](/docs/domains/working-with-domains/view-and-search-domain
3. **Check for private IP addresses:** Ensure that the request isn't attempting to connect to a private IP address from an external sourc
4. **Review application logs:** Inspect the [application logs](/docs/deployments/logs) for any warnings or errors related to DNS or the a

---
title: "DNS_HOSTNAME_RESOLVED_PRIVATE"
description: "The DNS hostname resolved to a private IP address or an IPv6 address during an external rewrite. This is a DNS error."
last_updated: "2026-01-16T02:19:30.110Z"
source: "https://vercel.com/docs/errors/DNS_HOSTNAME_RESOLVED_PRIVATE"

DNS_HOSTNAME_RESOLVED_PRIVATE

The `DNS_HOSTNAME_RESOLVED_PRIVATE` error occurs when attempting to connect to a private IP from an external rewrite, or when trying to c

Examples of such IPs would be:

- `192.0.0.1`
- `168.0.0.1`

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check the IP address:** Ensure that the IP address you are trying to connect to is publicly accessible and not a private or reserved
2. **Inspect network connectivity:** Ensure that there are no network issues that could be affecting the DNS resolution
3. **Review application logs:** Inspect the [application logs](/docs/deployments/logs) for any warnings or errors related to DNS or the a

---
title: "DNS_HOSTNAME_RESOLVE_FAILED"
description: "No error with the DNS resolution but no IP was returned. This is a DNS error."
last_updated: "2026-01-16T02:19:30.114Z"
source: "https://vercel.com/docs/errors/DNS_HOSTNAME_RESOLVE_FAILED"

DNS_HOSTNAME_RESOLVE_FAILED

The `DNS_HOSTNAME_RESOLVE_FAILED` error occurs when attempting to connect to a private IP from an external rewrite. Although there's no e

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check the domain name:** Ensure that the [domain name](/docs/domains/working-with-domains/view-and-search-domains) you are trying to
2. **Check DNS configuration:** Verify the [configuration](/docs/domains/working-with-dns) of the DNS server and ensure it is set up corr
3. **Firewall and security software:** Check if any firewall or security software on your system is blocking DNS requests. Ensure that th
4. **Inspect network connectivity:** Ensure that there are no network issues that could be affecting the DNS resolution

---
title: "DNS_HOSTNAME_SERVER_ERROR"
description: "The DNS server was unable to fulfill the DNS request due to an internal issue or misconfiguration. This is a DNS error."
last_updated: "2026-01-16T02:19:30.118Z"
source: "https://vercel.com/docs/errors/DNS_HOSTNAME_SERVER_ERROR"

DNS_HOSTNAME_SERVER_ERROR

The `DNS_HOSTNAME_SERVER_ERROR` error occurs when attempting to connect to a private IP from an external rewrite. This error typically me

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Review DNS configuration:** Check the [DNS configuration](/docs/domains/working-with-dns) to ensure it's correctly set up and doesn'
2. **Inspect network connectivity:** Ensure that there are no network issues that could be affecting the DNS resolution
3. **Check DNS server logs:** Review the logs of the DNS server for any warnings or errors that might indicate what's causing the issue
4. **Verify domain registration:** Ensure that the domain has been [registered](/docs/domains/working-with-domains/view-and-search-domain

---
title: "EDGE_FUNCTION_INVOCATION_FAILED"
description: "The request for a Edge Function was not completed successfully. This is an application error."
last_updated: "2026-01-16T02:19:30.122Z"
source: "https://vercel.com/docs/errors/EDGE_FUNCTION_INVOCATION_FAILED"

EDGE_FUNCTION_INVOCATION_FAILED

The `EDGE_FUNCTION_INVOCATION_FAILED` error occurs when there is an issue with the Edge Function being invoked on the CDN. This error can

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check application logs**: Review the application logs to identify any specific errors related to the Edge Function being invoked. Th
2. **Review deployment configuration**: Double-check the deployment configuration to ensure that the Edge Function is being deployed corr
3. **Investigate build errors**: If the error occurs during the build process, troubleshoot any build errors that might be preventing the
4. **Check function code**: Ensure that the code for the Edge Function is correct and does not contain any errors or infinite loops
5. **Use Vercel's status page**: If you have tried the steps above and are still experiencing the error, check Vercel's [status page](htt

---
title: "EDGE_FUNCTION_INVOCATION_TIMEOUT"
description: "The Edge Function invocation timed out. This is an application error."
last_updated: "2026-01-16T02:19:30.127Z"
source: "https://vercel.com/docs/errors/EDGE_FUNCTION_INVOCATION_TIMEOUT"

EDGE_FUNCTION_INVOCATION_TIMEOUT

The `EDGE_FUNCTION_INVOCATION_TIMEOUT` error occurs when an Edge Function takes longer than the allowed execution time to complete or doe

If your backend API takes time to respond, we recommend [streaming the response](/docs/functions/streaming-functions) to avoid the idle t

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check application logs**: Review the application logs to identify any specific errors related to the Edge Function being invoked. Th
2. **Review function code:** Inspect the Edge Function code for any long-running operations or infinite loops that could cause a timeout
3. **Verify return value:** Ensure the function returns a response within the specified time limit of [25 seconds](/docs/functions/limita
4. **Optimize external calls:** If the function makes calls to external services or APIs, ensure they are optimized and responding quickl
5. **Consider streaming data**: If the function is processing large amounts of data, consider using a [streaming approach](/docs/function
6. **Implement error handling:** Add error handling in the function to manage timeouts and other exceptions effectively

---
title: "FALLBACK_BODY_TOO_LARGE"
description: "The fallback body is too large for the cache. This is a cache error."
last_updated: "2026-01-16T02:19:30.134Z"
source: "https://vercel.com/docs/errors/FALLBACK_BODY_TOO_LARGE"

FALLBACK_BODY_TOO_LARGE

The `FALLBACK_BODY_TOO_LARGE` error indicates that the size of the fallback body exceeds the maximum cache limit. This error typically oc

Troubleshoot

To resolve this error, consider the following steps:

1. **Review response size:** Examine the size of the response body for the affected page. If it's too large, try to reduce the content si
2. **Optimize content:** Minimize HTML, CSS, and JavaScript on the fallback page Remove unnecessary assets or data to reduce the page siz
3. **Implement pagination:** If the large response body is due to extensive datasets, consider using pagination. This divides the data in
4. **Dynamic data loading:** Where possible, load data dynamically on the client-side instead of sending all data in the initial server r

To prevent this error, ensure that the size of the fallback page is less than 10 MB.

---
title: "FUNCTION_INVOCATION_FAILED"
description: "The invocation of a function failed. This is a function error."
last_updated: "2026-01-16T02:19:30.158Z"
source: "https://vercel.com/docs/errors/FUNCTION_INVOCATION_FAILED"

FUNCTION_INVOCATION_FAILED

The `FUNCTION_INVOCATION_FAILED` error occurs when a function invocation fails. This could be due to an error within the function itself,

Possible causes

- The runtime process (Node.js, Bun, Python, etc.) has crashed.
- Node.js or Bun threw an unhandled rejection/uncaught exception.

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check application logs:** Review the application logs to identify any specific errors related to the function invocation. They can b
2. **Review function code:** Ensure that the code for the function is correct and does not contain any errors or infinite loops. Use a `t
3. **Check for unhandled exceptions:** Look for any unhandled exceptions within the function code that might be causing the invocation to
4. **Verify function configuration:** Double-check the function configuration to ensure that it's set up correctly, including any environ

---
title: "FUNCTION_INVOCATION_TIMEOUT"
description: "The request for a Vercel Function reached the timeout threshold. This is an application error."
last_updated: "2026-01-16T02:19:30.165Z"
source: "https://vercel.com/docs/errors/FUNCTION_INVOCATION_TIMEOUT"

FUNCTION_INVOCATION_TIMEOUT

The `FUNCTION_INVOCATION_TIMEOUT` error occurs when a function invocation takes longer than the [allowed execution time](/docs/functions/

Troubleshoot

To troubleshoot this error, follow these steps:

1. **The function is taking too long to process a request**: Ensure that any API or database requests you make in your function respond w
2. **The function isn't returning a response**: The function must return an HTTP response, even if that response is an error. If no respo
3. **You have an infinite loop within your function**: Check that your function is not making an infinite loop at any stage of execution
4. **Upstream errors**: Check that any external API or database that you are attempting to call doesn't have any errors
5. A common cause for this issue is when the application contains an unhandled exception. Check the application logs, which can be found

```javascript filename="logs-url"
https://my-deployment-my-username.vercel.app/_logs
```

For more information on Vercel Functions timeouts, see [What can I do about Vercel Serverless Functions timing out?](/kb/guide/what-can-i

---
title: "FUNCTION_PAYLOAD_TOO_LARGE"
description: "The payload sent to the function is too large. This is a function error."
last_updated: "2026-01-16T02:19:30.177Z"
source: "https://vercel.com/docs/errors/FUNCTION_PAYLOAD_TOO_LARGE"

FUNCTION_PAYLOAD_TOO_LARGE

The `FUNCTION_PAYLOAD_TOO_LARGE` error occurs when the payload sent to a function exceeds the maximum allowed size. This typically happen

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Review payload size:** Check the size of the payload being sent to the function to ensure it's within the allowed limits, and does n
2. **Reduce payload size:** If possible, reduce the size of the payload being sent to the function. This might include sending less data
3. **Client-side uploads**: For large file uploads, consider using client-side uploads directly to [Vercel Blob](/docs/storage/vercel-blo
4. **Split into multiple requests:** If the payload data is too large to be sent in a single request, consider splitting the data into sm
5. **Use external storage:** If the data is very large, consider using external storage solutions to handle the data instead of sending i

---
title: "FUNCTION_RESPONSE_PAYLOAD_TOO_LARGE"
description: "The function returned a response that is too large. This is a function error."
last_updated: "2026-01-16T02:19:30.181Z"
source: "https://vercel.com/docs/errors/FUNCTION_RESPONSE_PAYLOAD_TOO_LARGE"

FUNCTION_RESPONSE_PAYLOAD_TOO_LARGE

The `FUNCTION_RESPONSE_PAYLOAD_TOO_LARGE` error occurs when the function returned a response that exceeds the maximum allowed size of 4.5

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Review response payload size:** Check the size of the response payload being returned by the function to ensure it's within the allo
2. **Reduce response payload size:** If possible, reduce the size of the response payload being returned by the function

---
title: "FUNCTION_THROTTLED"
description: "The function you are trying to call has exceeded the rate limit."
last_updated: "2026-01-16T02:19:30.186Z"

source: "https://vercel.com/docs/errors/FUNCTION_THROTTLED"

FUNCTION_THROTTLED

The `FUNCTION_THROTTLED` error occurs when your Vercel Functions exceed the concurrent execution limit, often due to a sudden request spi

Although this is a rare scenario, this error can also occur when Vercel's infrastructure encounters an abnormal system load and tries to r

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check application logs**: Review the application logs to identify any specific errors related to the Vercel Function being invoked.
2. **Handle request spikes**: If you're experiencing a sudden spike in requests, consider using the [Vercel Firewall](/docs/vercel-firewa
3. **Optimize your function**: Review your function code to ensure it's optimized for performance. For example, you can use [Vercel's CDN

---
title: "INFINITE_LOOP_DETECTED"
description: "An infinite loop was detected within the application."
last_updated: "2026-01-16T02:19:30.195Z"
source: "https://vercel.com/docs/errors/INFINITE_LOOP_DETECTED"

INFINITE_LOOP_DETECTED

The `INFINITE_LOOP_DETECTED` error occurs when an infinite loop is detected within the application. This error can occur when the applica

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check the application's source code:** Look for any code that might cause an infinite loop, such as a looping fetch or an unconditio
2. **Check the application's configuration:** Review any [configuration](/docs/redirects#configuration-redirects) files, such as `next.co
3. **Review external API or database calls:** Ensure that any external API or database calls your application is making do not have error
4. **Handle unhandled exceptions:** Check the application logs for any unhandled exceptions that might be causing the infinite loop
5. **Use Vercel's status page:** If you have tried the steps above and are still experiencing the error, check Vercel's [status page](htt

---
title: "INTERNAL_CACHE_ERROR"
description: "An unexpected error happened when CDN is fetching data from the Vercel CDN cache."
last_updated: "2026-01-16T02:19:30.198Z"
source: "https://vercel.com/docs/errors/INTERNAL_CACHE_ERROR"

INTERNAL_CACHE_ERROR

The `INTERNAL_CACHE_ERROR` error occurs during an unexpected issue in the CDN while retrieving data from the Vercel CDN cache.

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Contact support:** If the error persists, [contact support](/help#issues) for further assistance

---
title: "INTERNAL_CACHE_KEY_TOO_LONG"
description: "The CDN is failing to fetch from the internal cache due to a cache key being too long. This error can be caused by a reques
last_updated: "2026-01-16T02:19:30.202Z"
source: "https://vercel.com/docs/errors/INTERNAL_CACHE_KEY_TOO_LONG"

INTERNAL_CACHE_KEY_TOO_LONG

The `INTERNAL_CACHE_KEY_TOO_LONG` error occurs when the CDN is unable to fetch from the internal cache due to a cache key being too long.

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Contact support:** If the error persists, [contact support](/help#issues) for further assistance

---
title: "INTERNAL_CACHE_LOCK_FULL"
description: "An unexpected error happened when CDN is accessing internal cache."
last_updated: "2026-01-16T02:19:30.205Z"
source: "https://vercel.com/docs/errors/INTERNAL_CACHE_LOCK_FULL"

INTERNAL_CACHE_LOCK_FULL

The `INTERNAL_CACHE_LOCK_FULL` error occurs when CDN is accessing internal cache. This error is usually caused by a temporary issue with

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Contact support:** If the error persists, [contact support](/help#issues) for further assistance

---
title: "INTERNAL_CACHE_LOCK_TIMEOUT"
description: "An unexpected error happened when CDN is accessing internal cache."
last_updated: "2026-01-16T02:19:30.208Z"
source: "https://vercel.com/docs/errors/INTERNAL_CACHE_LOCK_TIMEOUT"

INTERNAL_CACHE_LOCK_TIMEOUT

The `INTERNAL_CACHE_LOCK_TIMEOUT` error occurs when CDN is accessing internal cache.

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Contact support:** If the error persists, [contact support](/help#issues) for further assistance

---
title: "INTERNAL_DEPLOYMENT_FETCH_FAILED"
description: "Failed to fetch the internal deployment. This is a deployment error."
last_updated: "2026-01-16T02:19:30.222Z"
source: "https://vercel.com/docs/errors/INTERNAL_DEPLOYMENT_FETCH_FAILED"

INTERNAL_DEPLOYMENT_FETCH_FAILED

The `INTERNAL_DEPLOYMENT_FETCH_FAILED` error occurs when the system is unable to fetch the deployment. This could happen due to network i

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check deployment status:** Ensure that the [deployment exists](/docs/projects/project-dashboard#deployments) and is in a stable stat
2. **Inspect deployment logs:** Review the [deployment logs](/docs/deployments/logs) to identify any specific errors or issues that might
3. **Review deployment history**: Check the deployment history to see if the deployment was deleted or [rolled back](/docs/instant-rollba

---
title: "INTERNAL_EDGE_FUNCTION_INVOCATION_FAILED"
description: "The request for a Edge Function was not completed successfully due to an internal error."
last_updated: "2026-01-16T02:19:30.225Z"
source: "https://vercel.com/docs/errors/INTERNAL_EDGE_FUNCTION_INVOCATION_FAILED"

INTERNAL_EDGE_FUNCTION_INVOCATION_FAILED

The `INTERNAL_EDGE_FUNCTION_INVOCATION_FAILED` error occurs when there is an issue with the Edge Function being invoked on the CDN. This

Troubleshoot

While this error can be caused by a variety of issues, it's transient and retrying the request will succeed. If the error persists, [**co

---
title: "INTERNAL_EDGE_FUNCTION_INVOCATION_TIMEOUT"
description: "The Edge Function invocation timed out unexpectedly."
last_updated: "2026-01-16T02:19:30.229Z"
source: "https://vercel.com/docs/errors/INTERNAL_EDGE_FUNCTION_INVOCATION_TIMEOUT"

INTERNAL_EDGE_FUNCTION_INVOCATION_TIMEOUT

The `INTERNAL_EDGE_FUNCTION_INVOCATION_TIMEOUT` error occurs when an Edge Function takes longer than the allowed execution time to comple

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check application logs**: Review the application logs to identify any specific errors related to the Edge Function being invoked. Th
2. **Review function code:** Inspect the Edge Function code for any long-running operations or infinite loops that could cause a timeout
3. **Verify return value:** Ensure the function begins responding within [25 seconds](/docs/functions/limitations#max-duration)
4. **Optimize external calls:** If the function makes calls to external services or APIs, ensure they are optimized and responding quickl
5. **Consider streaming data**: If the function is processing large amounts of data, consider using a [streaming approach](/docs/function
6. **Implement error handling:** Add error handling in the function to manage timeouts and other exceptions effectively

---
title: "INTERNAL_FUNCTION_INVOCATION_FAILED"
description: "The internal invocation of a function failed. This is Vercel"
last_updated: "2026-01-16T02:19:30.239Z"
source: "https://vercel.com/docs/errors/INTERNAL_FUNCTION_INVOCATION_FAILED"

INTERNAL_FUNCTION_INVOCATION_FAILED

The `INTERNAL_FUNCTION_INVOCATION_FAILED` error occurs when a function invocation fails. This could be due to an error within the functio

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check application logs:** Review the application logs to identify any specific errors related to the internal function invocation. T
2. **Review function code:** Ensure that the code for the function is correct and does not contain any errors or infinite loops
3. **Verify function configuration:** Double-check the function configuration to ensure that it's set up correctly, including any environ
4. **Check external dependencies:** If the function relies on external services or APIs, ensure they are responding in a timely manner

---
title: "INTERNAL_FUNCTION_INVOCATION_TIMEOUT"
description: "The internal invocation of a function timed out. This is Vercel"
last_updated: "2026-01-16T02:19:30.248Z"
source: "https://vercel.com/docs/errors/INTERNAL_FUNCTION_INVOCATION_TIMEOUT"

INTERNAL_FUNCTION_INVOCATION_TIMEOUT

The `INTERNAL_FUNCTION_INVOCATION_TIMEOUT` error occurs when a function invocation takes longer than the allowed execution time. This cou

Troubleshoot

To troubleshoot this error, follow these steps:

1. **The function is taking too long to process a request**: Ensure that any API or database requests you make in your function respond w
2. **The function isn't returning a response**: The function must return an HTTP response, even if that response is an error. If no respo
3. **You have an infinite loop within your function**: Check that your function is not making an infinite loop at any stage of execution
4. **Upstream errors**: Check that any external API or database that you are attempting to call doesn't have any errors
5. A common cause for this issue is when the application contains an unhandled exception. Check the application logs, which can be found

```javascript filename="logs-url"
https://my-deployment-my-username.vercel.app/_logs
```

For more information on Vercel Functions timeouts, see [What can I do about Vercel Serverless Functions timing out?](/kb/guide/what-can-i

---
title: "INTERNAL_FUNCTION_NOT_FOUND"
description: "The internal function could not be found. This is a Vercel"
last_updated: "2026-01-16T02:19:30.243Z"
source: "https://vercel.com/docs/errors/INTERNAL_FUNCTION_NOT_FOUND"

INTERNAL_FUNCTION_NOT_FOUND

The `INTERNAL_FUNCTION_NOT_FOUND` error occurs when an attempt to invoke a function fails because the function could not be found. This c

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Verify function deployment:** Ensure that the function has been successfully deployed and is available in the environment where it i
2. **Check function name:** Verify that the function name being used in the invocation matches the deployed function name
3. **Review configuration:** Check the function configuration in your project, including the function file name and the path where it is
4. **Check for typos:** Ensure that there are no typos or incorrect references in the function name or in the invocation command

---
title: "INTERNAL_FUNCTION_NOT_READY"
description: "The internal function is not ready to be invoked. This is a Vercel error."
last_updated: "2026-01-16T02:19:30.258Z"
source: "https://vercel.com/docs/errors/INTERNAL_FUNCTION_NOT_READY"

INTERNAL_FUNCTION_NOT_READY

The `INTERNAL_FUNCTION_NOT_READY` error occurs when an attempt is made to invoke a function before it is ready to accept requests. This m

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Verify deployment status:** Ensure that the function has been successfully deployed and the deployment process has completed
2. **Check initialization logs:** Review the function's initialization logs to identify any errors or warnings that might indicate why th
3. **Review configuration:** Ensure that the function and environment configurations are correct and that there are no misconfigurations
4. **Check dependencies:** Verify that all dependencies required by the function are available and correctly configured

---
title: "INTERNAL_MICROFRONTENDS_BUILD_ERROR"
description: "The microfrontend build did not include the required data as expected."
last_updated: "2026-01-16T02:19:30.282Z"
source: "https://vercel.com/docs/errors/INTERNAL_MICROFRONTENDS_BUILD_ERROR"

INTERNAL_MICROFRONTENDS_BUILD_ERROR

The `INTERNAL_MICROFRONTENDS_BUILD_ERROR` error occurs when the deployment is missing data that should have been included as part of the

This error should not occur because the build is designed to fail in such cases.

Troubleshoot

To troubleshoot this error, follow these steps:

1. We have been notified of this error. For more information, check the [Vercel status page](https://www.vercel-status.com/) or [contact '

---
title: "INTERNAL_MICROFRONTENDS_INVALID_CONFIGURATION_ERROR"
description: "The microfrontend configuration file is invalid."
last_updated: "2026-01-16T02:19:30.279Z"
source: "https://vercel.com/docs/errors/INTERNAL_MICROFRONTENDS_INVALID_CONFIGURATION_ERROR"

INTERNAL_MICROFRONTENDS_INVALID_CONFIGURATION_ERROR

The `INTERNAL_MICROFRONTENDS_INVALID_CONFIGURATION_ERROR` error occurs when the configuration file for the deployment is invalid.

This error indicates that an invalid configuration file has been deployed.

Troubleshoot

To troubleshoot this error, follow these steps:

1. Ensure the config in your `microfrontends.json` file is valid, see the [documentation](/docs/microfrontends/quickstart).

2. Ensure you are on the latest version of the [`@vercel/microfrontends`](https://www.npmjs.com/package/@vercel/microfrontends) package.
3. We have been notified of this error. For more information, check the [Vercel status page](https://www.vercel-status.com/) or [contact '

---
title: "INTERNAL_MICROFRONTENDS_UNEXPECTED_ERROR"
description: "An unexpected internal error occurred in the microfrontend."
last_updated: "2026-01-16T02:19:30.285Z"
source: "https://vercel.com/docs/errors/INTERNAL_MICROFRONTENDS_UNEXPECTED_ERROR"

INTERNAL_MICROFRONTENDS_UNEXPECTED_ERROR

The `INTERNAL_MICROFRONTENDS_UNEXPECTED_ERROR` occurs due to unspecified internal issues, such as system faults or unhandled exceptions.

Troubleshoot

To troubleshoot this error, follow these steps:

1. We have been notified of this error. For more information, check the [Vercel status page](https://www.vercel-status.com/) or [contact '

---
title: "INTERNAL_MISSING_RESPONSE_FROM_CACHE"
description: "This error indicates a missing response from the cache during a deployment or build process."
last_updated: "2026-01-16T02:19:30.261Z"
source: "https://vercel.com/docs/errors/INTERNAL_MISSING_RESPONSE_FROM_CACHE"

INTERNAL_MISSING_RESPONSE_FROM_CACHE

The `INTERNAL_MISSING_RESPONSE_FROM_CACHE` error occurs when an unexpected error happened during the CDN accessing the internal cache.

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Contact support:** If the error persists, [contact support](/help#issues) for further assistance

---
title: "INTERNAL_OPTIMIZED_IMAGE_REQUEST_FAILED"
description: "The request for an internally optimized image failed. This is a server error."
last_updated: "2026-01-16T02:19:30.265Z"
source: "https://vercel.com/docs/errors/INTERNAL_OPTIMIZED_IMAGE_REQUEST_FAILED"

INTERNAL_OPTIMIZED_IMAGE_REQUEST_FAILED

The `INTERNAL_OPTIMIZED_IMAGE_REQUEST_FAILED` error occurs when the request for an internally optimized image fails.

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Verify image path:** Ensure that the image path is correct and the server can access the image
2. **Check logs:** Review [logs](/docs/runtime-logs) for more details on the error
3. **Validate configuration:** Ensure that the configuration for image optimization is correct

---
title: "INTERNAL_ROUTER_CANNOT_PARSE_PATH"
description: "The CDN has failed to parse application-specified URL, such as rewrite/redirection URLs."
last_updated: "2026-01-16T02:19:30.268Z"
source: "https://vercel.com/docs/errors/INTERNAL_ROUTER_CANNOT_PARSE_PATH"

INTERNAL_ROUTER_CANNOT_PARSE_PATH

The `INTERNAL_ROUTER_CANNOT_PARSE_PATH` error occurs when the CDN has failed to parse application-specified URL, such as rewrite/redirect

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check configuration**: Check your configuration and make sure your app doesn't generate invalid URLs

---
title: "INTERNAL_STATIC_REQUEST_FAILED"
description: "This error occurs when a request for a static file in a project fails."
last_updated: "2026-01-16T02:19:30.275Z"
source: "https://vercel.com/docs/errors/INTERNAL_STATIC_REQUEST_FAILED"

INTERNAL_STATIC_REQUEST_FAILED

The `INTERNAL_STATIC_REQUEST_FAILED` error is encountered when a request for a static file within the project cannot be completed. This c

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check static files existence:** Ensure that all static files exist in your project and are correctly deployed. Confirm that they are
2. **Verify file paths:** Check that the paths to your static files are correct and reachable. Path errors or misconfigurations can lead
3. **Rollback changes:** If your project was working previously, consider reverting to a known working state. [Rollback](/docs/instant-ro

---
title: "INTERNAL_UNARCHIVE_FAILED"

description: "Unarchiving of the deployment or resource failed. This is an internal error."
last_updated: "2026-01-16T02:19:30.294Z"
source: "https://vercel.com/docs/errors/INTERNAL_UNARCHIVE_FAILED"

INTERNAL_UNARCHIVE_FAILED

The `INTERNAL_UNARCHIVE_FAILED` error typically occurs when the platform encounters a problem trying to extract your deployment's archive

- The structure of your project or the contents within it
- The size of your deployment bundle for Vercel functions exceeds the limit. For Vercel functions, the [maximum uncompressed size is 250 M

Troubleshoot

To troubleshoot this error, follow these steps:

- **Check your project files**: Check for any files or directories that have been unnecessarily included in the deployment. Removing unne
- **Check bundle size**: Looking into your `includeFiles` and `excludeFiles` configuration to specify items affecting the function size.

---
title: "INTERNAL_UNEXPECTED_ERROR"
description: "An unexpected internal error occurred. This is a general internal error."
last_updated: "2026-01-16T02:19:30.298Z"
source: "https://vercel.com/docs/errors/INTERNAL_UNEXPECTED_ERROR"

INTERNAL_UNEXPECTED_ERROR

The `INTERNAL_UNEXPECTED_ERROR` error occurs when an unexpected and unspecified internal error happens within the system. This type of er

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Contact support:** Since this error is general and could be due to a variety of reasons, [contact support](/help#issues) for further

---
title: "INVALID_IMAGE_OPTIMIZE_REQUEST"
description: "The query string is using an invalid value for q, w, or url parameters. This is a request error."
last_updated: "2026-01-16T02:19:30.304Z"
source: "https://vercel.com/docs/errors/INVALID_IMAGE_OPTIMIZE_REQUEST"

INVALID_IMAGE_OPTIMIZE_REQUEST

The `INVALID_IMAGE_OPTIMIZE_REQUEST` error occurs when the query string is using an invalid value for `q` (quality) or `w` (width), or `u

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check for typos:** Verify that there are no typos in the parameter names or values
2. **Review request format:** Ensure that the request URL is correctly formatted and includes the required parameters
   - The `q` parameter controls the quality of the image and must follow these rules:
     - The `q` parameter must be an integer
     - The `q` integer must be greater than or equal to 1
     - The `q` integer must be less than or equal to 100
     - The `q` integer must be the same as one specified in [`qualities`](https://nextjs.org/docs/app/api-reference/components/image#qual
   - The `w` parameter defines the width of the image and must follow these rules:
     - The `w` parameter must be an integer
     - The `w` integer must be the same as one specified in [`deviceSizes`](https://nextjs.org/docs/app/api-reference/components/image#de
   - The `url` parameter specifies the image location and must follow these rules:
     - The `url` parameter must start with `/`, `http://`, or `https://`
     - The `url` parameter must match one of the configured [`remotePatterns`](https://nextjs.org/docs/app/api-reference/components/image
     - The `url` parameter must have a `Content-Type` header that starts with `image/`
     - The `url` parameter must have a response body **less than 300 MB** (or **less than 100 MB for hobby**), otherwise the image won't

Run `next dev` locally to reproduce the error and get additional details.

---
title: "INVALID_REQUEST_METHOD"
description: "The request method used is invalid or not supported. This is a request error."
last_updated: "2026-01-16T02:19:30.316Z"
source: "https://vercel.com/docs/errors/INVALID_REQUEST_METHOD"

INVALID_REQUEST_METHOD

The `INVALID_REQUEST_METHOD` error occurs when a request is made with a method that is either invalid or not supported by the server. Thi

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Verify request method:** Ensure that the HTTP request method used is correct and supported by the endpoint. Common HTTP methods incl
2. **Review code:** Check the code where the request is being made to ensure the correct method is being used
3. **Test with different methods:** If possible, test the endpoint with different HTTP methods to determine if the issue is with the meth

---
title: "MALFORMED_REQUEST_HEADER"
description: "The MALFORMED_REQUEST_HEADER error occurs when a request contains an improperly formatted or invalid header. This is a requ
last_updated: "2026-01-16T02:19:30.320Z"
source: "https://vercel.com/docs/errors/MALFORMED_REQUEST_HEADER"

MALFORMED_REQUEST_HEADER

The `MALFORMED_REQUEST_HEADER` error signifies that a request made to the server includes a header that is incorrectly formatted or conta

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Inspect request headers**: Review the headers in your request. Ensure that they are correctly formatted and adhere to the [HTTP stan
2. **Validate UTF-8 encoding**: Confirm that all request headers, especially cookie values, are valid UTF-8 strings. Non-UTF-8 characters
3. **Examine Vercel Function behavior**: Since this error is specific to Vercel functions, verify the functionality and responses of your

---
title: "MICROFRONTENDS_MIDDLEWARE_ERROR"
description: "The microfrontend middleware returned an invalid application."
last_updated: "2026-01-16T02:19:30.324Z"
source: "https://vercel.com/docs/errors/MICROFRONTENDS_MIDDLEWARE_ERROR"

MICROFRONTENDS_MIDDLEWARE_ERROR

The `MICROFRONTENDS_MIDDLEWARE_ERROR` error occurs when the middleware returned a header `x-vercel-mfe-zone` with an invalid value. The v

Troubleshoot

To troubleshoot this error, follow these steps:

1. If you are setting the header, ensure that the value is a valid application name.
2. If you are not setting the header, this is an error caused by the [@vercel/microfrontends](https://www.npmjs.com/package/@vercel/micro

---
title: "MICROFRONTENDS_MISSING_FALLBACK_ERROR"
description: "The microfrontend request did not have a fallback for the environment."
last_updated: "2026-01-16T02:19:30.336Z"
source: "https://vercel.com/docs/errors/MICROFRONTENDS_MISSING_FALLBACK_ERROR"

MICROFRONTENDS_MISSING_FALLBACK_ERROR

The `MICROFRONTENDS_MISSING_FALLBACK_ERROR` error occurs when a microfrontends request did not match any other deployments in the same en

Troubleshoot

To troubleshoot this error, follow these steps:

In the [Production](/docs/deployments/environments#production-environment) environment, this error should not occur since every request i

In non-Production environments, the fallback is configured in the [Fallback Environment](/docs/microfrontends/managing-microfrontends#fal

If the issue persists after checking that every project has a deployment in the configured Fallback Environment setting, please contact V

---
title: "MIDDLEWARE_INVOCATION_FAILED"
description: "The request for an Routing Middleware was not completed successfully. This is an application error."
last_updated: "2026-01-16T02:19:30.340Z"
source: "https://vercel.com/docs/errors/MIDDLEWARE_INVOCATION_FAILED"

MIDDLEWARE_INVOCATION_FAILED

The `MIDDLEWARE_INVOCATION_FAILED` error occurs when there is an issue with the Routing Middleware being invoked on the CDN. This error c

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check application logs**: Review the application logs to identify any specific errors related to the Routing Middleware being invoke
2. **Use Vercel's status page**: If you have tried the steps above and are still experiencing the error, check Vercel's [status page](htt
3. **Check function code**: Ensure that the code for the Routing Middleware is correct and does not contain any errors or infinite loops

---
title: "MIDDLEWARE_INVOCATION_TIMEOUT"
description: "The Routing Middleware invocation timed out. This is an application error."
last_updated: "2026-01-16T02:19:30.345Z"
source: "https://vercel.com/docs/errors/MIDDLEWARE_INVOCATION_TIMEOUT"

MIDDLEWARE_INVOCATION_TIMEOUT

The `MIDDLEWARE_INVOCATION_TIMEOUT` error occurs when an Routing Middleware takes [longer than the allowed execution time](/docs/function

If your backend API takes time to respond, we recommend [streaming the response](/docs/functions/streaming-functions) to avoid the idle t

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check application logs**: Review the application logs to identify any specific errors related to the Routing Middleware being invoke
2. **Review function code:** Inspect the Routing Middleware code for any long-running operations or infinite loops that could cause a tim
3. **Verify return value:** Ensure the function returns a response within the specified time limit of [25 seconds](/docs/functions/limita
4. **Optimize external calls:** If the function makes calls to external services or APIs, ensure they are optimized and responding quickl
5. **Consider streaming data**: If the function is processing large amounts of data, consider using a [streaming approach](/docs/function
6. **Implement error handling:** Add error handling in the function to manage timeouts and other exceptions effectively

---
title: "MIDDLEWARE_RUNTIME_DEPRECATED"

description: "A middleware is using a deprecated runtime."
last_updated: "2026-01-16T02:19:30.358Z"
source: "https://vercel.com/docs/errors/MIDDLEWARE_RUNTIME_DEPRECATED"

MIDDLEWARE_RUNTIME_DEPRECATED

The `MIDDLEWARE_RUNTIME_DEPRECATED` error occurs when a middleware is using a deprecated runtime. This error can occur when a middleware

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Identify the affected project:** Use [Vercel Logs](/docs/observability/runtime-logs) to identify if your project is experiencing thi
2. **Locate the middleware:** Once you've identified the project, check if it has a `middleware.js` or `middleware.ts` file in the root d
3. **Redeploy the project:** Redeploy the project to automatically upgrade to the latest supported runtime version. However, if the redep
   - **Update your Node.js version:** Check your project's Node.js version setting in the Vercel dashboard or `package.json` and update i
   - **Update dependencies:** Outdated dependencies may not be compatible with newer Node.js versions. Update your `package.json` depende

---
title: "NOT_FOUND"
description: "The requested resource was not found. This is a deployment error."
last_updated: "2026-01-16T02:19:30.364Z"
source: "https://vercel.com/docs/errors/NOT_FOUND"

NOT_FOUND

The `NOT_FOUND` error occurs when a requested resource could not be found. This might happen if the resource has been moved, deleted, or

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check the deployment URL**: Ensure that the deployment URL you are using is correct and does not contain any typos or incorrect path
2. **Check deployment existence:** Verify that the [deployment exists](/docs/projects/project-dashboard#deployments) and has not been del
3. **Review deployment logs:** If the deployment exists, review the [deployment logs](/docs/deployments/logs) to identify any issues that
4. **Verify permissions:** Ensure you have the necessary [permissions](/docs/accounts/team-members-and-roles) to access the deployment
5. **Contact support:** If you've checked the above and are still unable to resolve the issue, [contact support](/help#issues) for furthe

---
title: "NO_RESPONSE_FROM_FUNCTION"
description: "The application did not respond correctly, this is likely due to an exception being thrown from the function handler."
last_updated: "2026-01-16T02:19:30.377Z"
source: "https://vercel.com/docs/errors/NO_RESPONSE_FROM_FUNCTION"

NO_RESPONSE_FROM_FUNCTION

The `NO_RESPONSE_FROM_FUNCTION` error occurs when a function invocation completes without returning a response. This might happen if the

Potential causes include:

- A global uncaught exception
- A global unhandled rejection
- A deployment that introduced incorrect syntax

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Verify return statements:** Ensure that the function has the necessary return statements to generate a response
2. **Check the function logs**: Open the [realtime request logs](/docs/logs#function-logs) for the application in a separate tab - this t
3. **Review realtime logs**: Repeat the application behavior that led to the error being thrown and review the realtime request logs wher
   - Use the information contained within the error logs to understand where the function is failing
4. **Use Log Drains**: Create a [Log Drain](/docs/drains) if you do not have one yet, to persist errors from Vercel functions
5. **Check external dependencies:** If the function relies on external services or APIs, ensure they are responding in a timely manner

---
title: "OPTIMIZED_EXTERNAL_IMAGE_REQUEST_FAILED"
description: "The request for an optimized external image failed. This is a server error."
last_updated: "2026-01-16T02:19:30.380Z"
source: "https://vercel.com/docs/errors/OPTIMIZED_EXTERNAL_IMAGE_REQUEST_FAILED"

OPTIMIZED_EXTERNAL_IMAGE_REQUEST_FAILED

The `OPTIMIZED_EXTERNAL_IMAGE_REQUEST_FAILED` error occurs when the request for an optimized external image fails.

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Verify external URL:** Ensure that the external image URL is correct and accessible
2. **Check query parameters:** Ensure that any query parameters are valid

---
title: "OPTIMIZED_EXTERNAL_IMAGE_REQUEST_INVALID"
description: "The external image request is invalid. This is a request error."
last_updated: "2026-01-16T02:19:30.384Z"
source: "https://vercel.com/docs/errors/OPTIMIZED_EXTERNAL_IMAGE_REQUEST_INVALID"

OPTIMIZED_EXTERNAL_IMAGE_REQUEST_INVALID

The `OPTIMIZED_EXTERNAL_IMAGE_REQUEST_INVALID` error occurs when the external image request is invalid.

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Verify external URL:** Ensure that the external image URL is absolute and correctly formatted
2. **Check query parameters:** Ensure that any query parameters are valid
3. **Validate source configuration:** Verify that the external source is configured correctly and accessible

---
title: "OPTIMIZED_EXTERNAL_IMAGE_REQUEST_UNAUTHORIZED"
description: "The external image request is unauthorized. This is a request error."
last_updated: "2026-01-16T02:19:30.387Z"
source: "https://vercel.com/docs/errors/OPTIMIZED_EXTERNAL_IMAGE_REQUEST_UNAUTHORIZED"

OPTIMIZED_EXTERNAL_IMAGE_REQUEST_UNAUTHORIZED

The `OPTIMIZED_EXTERNAL_IMAGE_REQUEST_UNAUTHORIZED` error occurs when the external image request is unauthorized.

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check permissions:** Ensure that you have the necessary permissions to access the external image
2. **Verify authentication:** Check for any authentication or authorization issues with the external source
3. **Update credentials:** Ensure that any required credentials or tokens are correctly set and not expired
4. **Remove filters**: Remove any filters that may be blocking the request, such as headers or IP restrictions

---
title: "OPTIMIZED_EXTERNAL_IMAGE_TOO_MANY_REDIRECTS"
description: "The external image request encountered too many redirects. This is a request error."
last_updated: "2026-01-16T02:19:30.423Z"
source: "https://vercel.com/docs/errors/OPTIMIZED_EXTERNAL_IMAGE_TOO_MANY_REDIRECTS"

OPTIMIZED_EXTERNAL_IMAGE_TOO_MANY_REDIRECTS

The `OPTIMIZED_EXTERNAL_IMAGE_TOO_MANY_REDIRECTS` error occurs when the external image request encounters too many redirects.

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check URL for redirects:** Verify the external image URL to ensure it does not cause an infinite redirect loop

---
title: "RANGE_END_NOT_VALID"
description: "The end value of the Range header in the request is invalid. This is a request error."
last_updated: "2026-01-16T02:19:30.395Z"
source: "https://vercel.com/docs/errors/RANGE_END_NOT_VALID"

RANGE_END_NOT_VALID

The `RANGE_END_NOT_VALID` error occurs when the end value of the `Range` header in a request is invalid. This header is used to request a

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Validate Range header values:** Ensure that the end value in the `Range` header is a valid integer. It should not be a letter, a dec
2. **Correct ordering:** Ensure the start value is less than the end value in the `Range` header
3. **Omit end value if necessary:** If you want to request all bytes from a certain start point to the end of the resource, you can omit
4. **Check configuration:** If the `Range` header values are being set automatically by some part of your system, check the configuration
5. **Debugging:** If the error persists, log the `Range` header values in your server logs to debug and understand what values are being

---
title: "RANGE_GROUP_NOT_VALID"
description: "The group value of the Range header in the request is invalid. This is a request error."
last_updated: "2026-01-16T02:19:30.399Z"
source: "https://vercel.com/docs/errors/RANGE_GROUP_NOT_VALID"

RANGE_GROUP_NOT_VALID

The `RANGE_GROUP_NOT_VALID` error occurs when the group value of the `Range` header in a request is invalid. This header is used to reque

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Validate Range header values:** Ensure that the group value in the `Range` header is a valid format. It should correctly specify the
2. **Correct grouping:** Ensure that the group value is correctly formatted and contains valid range specifications
3. **Check configuration:** If the `Range` header values are being set automatically by some part of your system, check the configuration
4. **Debugging:** If the error persists, log the `Range` header values in your server logs to debug and understand what values are being

---
title: "RANGE_MISSING_UNIT"
description: "The unit identifier of the Range header in the request is missing. This is a request error."
last_updated: "2026-01-16T02:19:30.415Z"
source: "https://vercel.com/docs/errors/RANGE_MISSING_UNIT"

RANGE_MISSING_UNIT

The `RANGE_MISSING_UNIT` error occurs when the unit identifier of the `Range` header in a request is missing. The `Range` header is used

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Specify unit identifier:** Ensure that the `Range` header in your request specifies a unit identifier like `bytes`
2. **Check configuration:** If the `Range` header values are being set automatically by some part of your system, check the configuration
3. **Verify syntax:** Verify that the syntax of the `Range` header is correct and follows the format `unit=range-start-range-end`, for ex
4. **Debugging:** If the error persists, log the `Range` header values in your server logs to debug and understand what values are being

---
title: "RANGE_START_NOT_VALID"
description: "The start value of the Range header in the request is invalid. This is a request error."
last_updated: "2026-01-16T02:19:30.420Z"
source: "https://vercel.com/docs/errors/RANGE_START_NOT_VALID"

RANGE_START_NOT_VALID

The `RANGE_START_NOT_VALID` error occurs when the start value of the `Range` header in a request is invalid. The `Range` header is used t

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Validate Range header values:** Ensure that the start value in the `Range` header is a valid integer. It should not be a letter, a d
2. **Correct ordering:** Ensure the start value is less than the end value in the `Range` header, if an end value is specified
3. **Check configuration:** If the `Range` header values are being set automatically by some part of your system, check the configuration
4. **Debugging:** If the error persists, log the `Range` header values in your server logs to debug and understand what values are being

---
title: "RANGE_UNIT_NOT_SUPPORTED"
description: "The unit identifier of the Range header in the request is not supported. This is a request error."
last_updated: "2026-01-16T02:19:30.429Z"
source: "https://vercel.com/docs/errors/RANGE_UNIT_NOT_SUPPORTED"

RANGE_UNIT_NOT_SUPPORTED

The `RANGE_UNIT_NOT_SUPPORTED` error occurs when the unit identifier of the `Range` header in a request is not supported by the server. T

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Verify supported Range units:** Check the documentation for the server or service you are interacting with to determine the supporte
2. **Correct Range unit:** If the `Range` header in your request specifies an unsupported unit, correct it to use a supported unit
3. **Check configuration:** If the `Range` header values are being set automatically by some part of your system, check the configuration
4. **Debugging:** If the error persists, log the `Range` header values in your server logs to debug and understand what values are being

---
title: "REQUEST_HEADER_TOO_LARGE"
description: "Request header size exceeds the permissible limit."
last_updated: "2026-01-16T02:19:30.436Z"
source: "https://vercel.com/docs/errors/REQUEST_HEADER_TOO_LARGE"

REQUEST_HEADER_TOO_LARGE

The `REQUEST_HEADER_TOO_LARGE` error occurs when the size of the request headers in your function and [Routing Middleware](/docs/routing-m

This issue often arises from excessively large headers in a request. On Vercel, applications may have custom headers, which, if overly la

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Limit header size:** Ensure that the size of each request header does not exceed 16 KB
2. **Manage total header size:** Monitor and control the combined size of all headers, keeping it under 32 KB
3. **Review cookies:** Since cookies are included in the header, it's crucial to limit their size as part of the overall header size

---
title: "RESOURCE_NOT_FOUND"
description: "This error signifies that a specified resource could not be located."
last_updated: "2026-01-16T02:19:30.441Z"
source: "https://vercel.com/docs/errors/RESOURCE_NOT_FOUND"

RESOURCE_NOT_FOUND

The `RESOURCE_NOT_FOUND` error indicates that a requested resource is not available or cannot be found. This error typically arises when

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Verify resource existence:** Confirm that the resource you're attempting to access exists. Check for any typos or errors in the reso
2. **Review access permissions:** Ensure that your application or function has the necessary permissions to access the resource
3. **Inspect resource path:** Double-check the path or URL to the resource. Ensure it is correctly formatted and corresponds to the inten
4. **Check application configuration:** Review your application's configuration settings to ensure they are correctly set up to locate an
5. **Review logs:** Consult your [application logs](/docs/runtime-logs) for more details or clues as to why the resource could not be fou

Additionally, the error can also occur in the context of the [Vercel REST API](/docs/rest-api), where it is similar to the [HTTP 500 Inte

---
title: "ROUTER_CANNOT_MATCH"
description: "The router cannot match the route to any of the known patterns. This is a routing error."
last_updated: "2026-01-16T02:19:30.444Z"
source: "https://vercel.com/docs/errors/ROUTER_CANNOT_MATCH"

ROUTER_CANNOT_MATCH

The `ROUTER_CANNOT_MATCH` error occurs when the router is unable to match the requested route to any of the known patterns. This could ha

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Review routing configuration:** Check the [routing configuration](/docs/redirects#configuration-redirects) to ensure that it is corr
2. **Verify request path:** Ensure that the request path is correct and adheres to the expected patterns defined in the routing configura
3. **Check for typos:** Look for any typos or misconfigurations in the routing setup that might be causing the mismatch
4. **Review application logs:** Inspect the [application logs](/docs/deployments/logs) for any warnings or errors related to routing

---
title: "ROUTER_EXTERNAL_TARGET_CONNECTION_ERROR"
description: "Connection error occurred while routing to an external target. This is a routing error."
last_updated: "2026-01-16T02:19:30.449Z"
source: "https://vercel.com/docs/errors/ROUTER_EXTERNAL_TARGET_CONNECTION_ERROR"

ROUTER_EXTERNAL_TARGET_CONNECTION_ERROR

The `ROUTER_EXTERNAL_TARGET_CONNECTION_ERROR` error occurs when there is a connection error while routing to an external target. This cou

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check network connectivity:** Ensure that the network connectivity between your deployment and the external target is stable
2. **Verify external target availability:** Make sure the external target is online and reachable
3. **Review routing configuration:** Check the [routing configuration](/docs/redirects#configuration-redirects) to ensure that it is corr
4. **Inspect firewall settings:** Verify that there are no firewall settings blocking the connection to the external target
5. **Review application logs:** Inspect the [application logs](/docs/deployments/logs) for any warnings or errors related to routing or n

---
title: "ROUTER_EXTERNAL_TARGET_ERROR"
description: "Error occurred while routing to an external target. This is a routing error."
last_updated: "2026-01-16T02:19:30.464Z"
source: "https://vercel.com/docs/errors/ROUTER_EXTERNAL_TARGET_ERROR"

ROUTER_EXTERNAL_TARGET_ERROR

The `ROUTER_EXTERNAL_TARGET_ERROR` error occurs when there is an error while routing to an external target. This could happen due to inco

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Review routing configuration:** Check the [routing configuration](/docs/redirects#configuration-redirects) to ensure that it is corr
2. **Verify external target availability:** Make sure the external target is online and reachable
3. **Check for errors in external target:** Investigate the external target for any errors that might be causing the routing issue
4. **Inspect firewall settings:** Verify that there are no firewall settings blocking the connection to the external target
5. **Review application logs:** Inspect the [application logs](/docs/deployments/logs) for any warnings or errors related to routing or t

---
title: "ROUTER_EXTERNAL_TARGET_HANDSHAKE_ERROR"
description: "Error in establishing a connection with an external target."
last_updated: "2026-01-16T02:19:30.453Z"
source: "https://vercel.com/docs/errors/ROUTER_EXTERNAL_TARGET_HANDSHAKE_ERROR"

ROUTER_EXTERNAL_TARGET_HANDSHAKE_ERROR

The `ROUTER_EXTERNAL_TARGET_HANDSHAKE_ERROR` error occurs when a connection cannot be successfully established with an external target. T

- **SSL handshake failure:** The SSL handshake may fail if the target has an invalid certificate or uses an unsupported Cipher Suite
- **Timeout:** The error could also be due to a timeout, which might be caused by issues connecting to the target. Note that proxied requ

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Check SSL configuration:** Ensure that the target's [SSL certificate](/docs/domains/custom-SSL-certificate) is valid and that it is
2. **Investigate connectivity issues:** Look into potential connectivity problems between your application and the external target
3. **Monitor response times:** Check if your application or the external target is experiencing unusual delays that might be contributing

---
title: "ROUTER_TOO_MANY_HAS_SELECTIONS"
description: "The router has too many selections. This is a routing error."
last_updated: "2026-01-16T02:19:30.468Z"
source: "https://vercel.com/docs/errors/ROUTER_TOO_MANY_HAS_SELECTIONS"

ROUTER_TOO_MANY_HAS_SELECTIONS

The `ROUTER_TOO_MANY_HAS_SELECTIONS` error occurs when the router encounters too many selections while processing the request. This could

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Review routing configuration:** Check the [routing configuration](/docs/redirects#configuration-redirects) to ensure it's correctly
2. **Simplify routing setup:** If possible, simplify the routing setup to reduce the number of selections the router has to process
3. **Check for recursive or looping logic:** Ensure there isn't any recursive or looping logic in the routing configuration that could le
4. **Review application logs:** Inspect the [application logs](/docs/deployments/logs) for any warnings or errors related to routing or s

---
title: "SANBDOX_NOT_FOUND"
description: "The Sandbox could not be found on Vercel. This is a platform error."
last_updated: "2026-01-16T02:19:30.456Z"
source: "https://vercel.com/docs/errors/SANDBOX_NOT_FOUND"

SANBDOX_NOT_FOUND

The `SANDBOX_NOT_FOUND` error occurs when you are trying to access a Sandbox that does not exist. This could happen if there is a typo in

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Verify the Sandbox URL:** Navigate to the [Sandboxes dashboard](/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Fobservability%2Fsandboxes\&tit
2. **Check for typos:** Ensure that there are no typos in the Sandbox URL you are trying to access

---
title: "SANBDOX_NOT_LISTENING"
description: "The Sandbox is not listening on the requested port. This is an application error."
last_updated: "2026-01-16T02:19:30.460Z"
source: "https://vercel.com/docs/errors/SANDBOX_NOT_LISTENING"

SANBDOX_NOT_LISTENING

The `SANDBOX_NOT_LISTENING` error occurs when you are trying to access a Sandbox that is not listening on the requested port. This could

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Verify the configured port:** Make sure that the `ports` field used in `Sandbox.create` matches the port your application is listeni
2. **Check the Sandbox history:** Navigate to the [Sandboxes dashboard](/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Fobservability%2Fsandboxes\&

---
title: "SANBDOX_STOPPED"
description: "The Sandbox was stopped and is no longer reachable. This is a platform error."
last_updated: "2026-01-16T02:19:30.480Z"
source: "https://vercel.com/docs/errors/SANDBOX_STOPPED"

SANBDOX_STOPPED

The `SANDBOX_STOPPED` error occurs when you are trying to access a Sandbox that has been stopped. This could happen if the Sandbox was ma

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Verify the Sandbox status:** Navigate to the [Sandboxes dashboard](/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Fobservability%2Fsandboxes\&
2. **Increase the timeout:** By default, Sandboxes have a timeout of 10 minutes. You can increase the timeout by passing the `timeout` pr

---
title: "TOO_MANY_FILESYSTEM_CHECKS"
description: "Too many filesystem checks occurred while processing the request. This is a routing error."
last_updated: "2026-01-16T02:19:30.497Z"
source: "https://vercel.com/docs/errors/TOO_MANY_FILESYSTEM_CHECKS"

TOO_MANY_FILESYSTEM_CHECKS

The `TOO_MANY_FILESYSTEM_CHECKS` error occurs when there are excessive filesystem checks while processing a request. This could happen du

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Review routing configuration**: Check the routing configuration to ensure that it is not causing excessive filesystem checks, especi
2. **Optimize routing configuration**: Reduce the number of has routes matched on a single path. You cannot have more than 5 has routes m
3. **Check for Loops**: Ensure there isn't any looping logic in the routing or filesystem access code that could lead to excessive filesy
4. **Review application logs**: Inspect the [application logs](/docs/deployments/logs) for any warnings or errors related to filesystem a

---
title: "TOO_MANY_FORKS"
description: "An error occurred in the application when matching too many conditional routes. You cannot have more than 5 \"
last_updated: "2026-01-16T02:19:30.485Z"
source: "https://vercel.com/docs/errors/TOO_MANY_FORKS"

TOO_MANY_FORKS

The `TOO_MANY_FORKS` error occurs when too many forks are generated while processing the request. This usually happens when matching too m

You cannot have more than 5 `has` routes matched on a single path.

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Review routing configuration**: Reduce the number of [rewrites](/docs/rewrites), [redirects](/docs/redirects#configuration-redirects
2. **Check for recursive logic**: Ensure there isn't any recursive logic in the routing configuration that could lead to excessive forkin
3. **Handle unhandled exceptions**: Check the [application logs](/docs/deployments/logs) for any unhandled exceptions that may be causing

---
title: "TOO_MANY_RANGES"
description: "Too many ranges have been specified in the Range header of the request. This is a request error."
last_updated: "2026-01-16T02:19:30.489Z"
source: "https://vercel.com/docs/errors/TOO_MANY_RANGES"

TOO_MANY_RANGES

The `TOO_MANY_RANGES` error occurs when too many ranges have been specified in the `Range` header of a request. The `Range` header is use

Troubleshoot

To troubleshoot this error, follow these steps:

To troubleshoot this error, follow these steps:

1. **Reduce number of Ranges:** Reduce the number of ranges specified in the `Range` header to a reasonable amount
2. **Check configuration:** If the `Range` header values are being set automatically by some part of your system, check the configuration
3. **Verify server capabilities:** Check the documentation for the server or service you are interacting with to determine the maximum nu
4. **Debugging:** If the error persists, log the `Range` header values in your server logs to debug and understand what values are being

---
title: "URL_TOO_LONG"
description: "The URL of the request is too long. This is a request error."
last_updated: "2026-01-16T02:19:30.501Z"
source: "https://vercel.com/docs/errors/URL_TOO_LONG"

URL_TOO_LONG

The `URL_TOO_LONG` error occurs when the URL of the request exceeds the maximum length allowed by the CDN (**14 KB**). Long URLs can be a

Troubleshoot

To troubleshoot this error, follow these steps:

1. **Shorten the URL:** Simplify the URL by reducing the length of the path segments and the query string
2. **Reduce query parameters:** If the URL has many query parameters, consider reducing the number of parameters or use `POST` method ins
3. **Use POST method:** If the long URL is a result of a form submission, consider changing the form method from `GET` to `POST`
4. **Check for unintended redirection:** Ensure there isn't a redirection loop or logic that is appending to the URL causing it to grow i

---
title: "Error List"
description: "You may encounter a variety of errors when you interact with the Vercel platform. This section focuses on errors that can h
last_updated: "2026-01-16T02:19:30.601Z"
source: "https://vercel.com/docs/errors/error-list"

Error List

Missing public directory

The [build step](/docs/deployments/configure-a-build) will result in an error if the output directory is missing, empty, or invalid (for

- Make sure the [output directory](/docs/deployments/configure-a-build#output-directory) is specified correctly in project settings
- If the output directory is correct, check the build command ([documentation](/docs/deployments/configure-a-build#build-command)) or the
- Try running the build command locally and make sure that the files are correctly generated in the specified output directory

Missing build script

> **💡 Note:** This is only relevant if you're using Vercel CLI 16.7.3 or older.

Suppose your project contains a `package.json` file, no `api` directory, and no `vercel.json` configuration. In that case, it is expected

When properly configured, your `package.json` file would look similar to this:

```json filename="package.json"
{
  "scripts": {
    "build": "[my-framework] build --output public"
  }
}
```

Once you have defined the `build` [script](https://docs.npmjs.com/misc/scripts), this error will disappear. Furthermore, it will not be d

Maximum team member requests

The maximum amount of open requests to join a team is 10. In order to allow for more requests, the existing requests need to be approved

This ensures the list always remains manageable and protected against spam.

Inviting users to team who requested access

If a user has already requested access to a team, it's impossible to invite them. Instead, their request must be approved by a [Team Owne

This ensures no team invites are accidentally accepted.

Request access with the required Git account

When the deployment for a commit fails with the message "Team access required to deploy.", the Git account of the commit author is not co

The link attached to the error allows someone to connect their Hobby team to the Git account of the commit author. If the Hobby team is c

Once the Git account is connected to the Hobby team on Vercel, it is possible to request access to the team. A [Team Owner](/docs/rbac/ac

Blocked scopes

A deployment, project, user, or team on Vercel can be blocked if it violates our [fair use guidelines](/docs/limits/fair-use-guidelines) o

Blocked deployments and projects will result in your website returning a 451 error. Blocked users and teams cannot create new deployments

Unused build and development settings

A [Project](/docs/projects/overview) has several settings that can be found in the dashboard. One of those sections, **Build & Developmen

However, the **Build & Development Settings** are only applied to [zero-configuration](/kb/guide/upgrade-to-zero-configuration) deploymen

If a deployment defines the [`builds`](/docs/project-configuration#builds) configuration property, the **Build & Development Settings** a

Unused Vercel Function region setting

A [project](/docs/projects/overview) has several settings that can be found in the dashboard. One of those settings, **Vercel Function Re

If a deployment defines the [`regions`](/docs/project-configuration#regions) configuration property in `vercel.json`, the **Vercel Functi

If a CLI Deployment defines the [`--regions`](/docs/cli/deploy#regions) option, the **Vercel Function Region** setting is ignored.

Invalid route source pattern

The `source` property follows the syntax from [path-to-regexp](https://github.com/pillarjs/path-to-regexp/tree/v6.1.0), not the `RegExp`

For example, negative lookaheads must be wrapped in a group.

Before

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "source": "/feedback/(?!general)",
  "destination": "/api/feedback/general"
}
```

After

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "source": "/feedback/((?!general).*)",
  "destination": "/api/feedback/general"
}
```

Invalid route destination segment

The `source` property follows the syntax from [path-to-regexp](https://github.com/pillarjs/path-to-regexp/tree/v6.1.0).

A colon (`:`) defines the start of a [named segment parameter](https://github.com/pillarjs/path-to-regexp/tree/v6.1.0#named-parameters).

A named segment parameter defined in the `destination` property must also be defined in the `source` property.

Before

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "source": "/feedback/:type",
  "destination": "/api/feedback/:id"
}
```

After

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "source": "/feedback/:id",
  "destination": "/api/feedback/:id"
}
```

Failed to install builder dependencies

When running the `vercel build` or `vercel dev` commands, `npm install` errors can be encountered if `npm` was invoked to install Builder

`npm install` may fail if:

- [`npm`](https://www.npmjs.com/get-npm) is not installed
- Your internet connection is unavailable
- The Builder that is defined in your configuration is not published to the npm registry

Double-check that the name and version of the Builder you are requesting is correct.

Mixed routing properties

If you have [`rewrites`](/docs/project-configuration#rewrites), [`redirects`](/docs/project-configuration#redirects), [`headers`](/docs/p

This is a necessary limitation because `routes` is a lower-level primitive that contains all of the other types. Therefore, it cannot be u

See the [Upgrading Routes](/docs/project-configuration#upgrading-legacy-routes) section for examples of `routes` compared to the new prop

Conflicting configuration files

For backward compatibility purposes, there are two naming conventions for configuration files used by Vercel CLI (for example `vercel.jso

These conflicting configuration errors occur if:

- Both `vercel.json` and `now.json` exist in your project.**Solution:** Delete the `now.json` file
- Both `.vercel` and `.now` directories exist in your project.**Solution:** Delete the `.now` directory
- Both `.vercelignore` and `.nowignore` files exist in your project.**Solution:** Delete the `.nowignore` file
- Environment Variables that begin with `VERCEL_` have a conflicting Environment Variable that begins with `NOW_`.**Solution:** Only defi

Conflicting functions and builds configuration

There are two ways to configure Vercel functions in your project: [functions](/docs/project-configuration#functions) *or* [`builds`](/doc

For most cases, it is recommended to use the `functions` property because it supports more features, such as:

- Allows configuration of the amount of memory that the Vercel Function is provided with
- More reliable because it requires a specific npm package version for the `runtime` property
- Supports "clean URLs" by default, which means that the Vercel functions are automatically accessible without their file extension in th

However, the [`builds`](/docs/project-configuration#builds) property will remain supported for backward compatibility purposes.

Unsupported functions configuration with Nextjs

When using Next.js, only `memory` and `maxDuration` can be configured within the [functions](/docs/project-configuration#functions) prope

Deploying Vercel functions to multiple regions

It's possible to deploy Vercel functions to multiple regions. This functionality is only available to [Enterprise teams](/docs/plans/ente

On the [Pro plan](/docs/plans/pro-plan), the limitation has existed since the [launch](/blog/simpler-pricing) of the current pricing mode

To select the region closest to you, read our [guide](/docs/functions/configuring-functions/region) on choosing deployment regions for Ve

Unmatched function pattern

The [functions](/docs/project-configuration#functions) property uses a glob pattern for each key. This pattern must match Vercel Function

If you are using Next.js, Vercel functions source files can be created in the following:

- `pages/api` directory
- `src/pages/api` directory
- `pages` directory when the module exports [getServerSideProps](https://nextjs.org/docs/basic-features/data-fetching/get-server-side-pro
- `src/pages` directory when the module exports [getServerSideProps](https://nextjs.org/docs/basic-features/data-fetching/get-server-side

Additionally, if you'd like to use a Vercel Function that isn't written with Node.js, and in combination with Next.js, you can place it i

Not Allowed

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "users/**/*.js": {
      "maxDuration": 30
    }
  }
}
```

Allowed

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "api/users/**/*.js": {
      "maxDuration": 30
    }
  }
}
```

Allowed (Next.js)

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "pages/api/users/**/*.js": {
      "maxDuration": 30
    }
  }
}
```

Cannot load project settings

If the Project configuration in `.vercel` belongs to a team you are not a member of, attempting to deploy the project will result in an e

This can occur if you clone a Git repository that includes the `.vercel` directory, or you are logged in to the wrong Vercel account.
Additionally, authentication issues can occur if you don't comply with the [two-factor enforcement](/docs/two-factor-enforcement) policy

To fix, remove the `.vercel` directory and redeploy to link the project again by running these commands.

On macOS and Linux:

On Windows:

Project name validation

Project names can only consist of up to one hundred alphanumeric lowercase characters. Hyphens can be used in between words in the name,

Repository connection limitation

The amount of Vercel Projects that can be connected with the same Git repository is [limited depending on your plan](/docs/limits#general

If you have reached the limitation and would like to connect a new project to the repository, you will need to disconnect an existing pro

To increase this limit, please [contact our Sales Team](/contact/sales).

Domain verification through CLI

To verify your domain, point the domain to Vercel by configuring our nameservers or a DNS Record. You can learn more about what to do for

Alternatively, if you already added the domain to a project, read [the configuring a domain section](/docs/projects/custom-domains#step-4

Leaving the team

You cannot leave a team if you are the last remaining [Owner](/docs/rbac/access-roles#owner-role) or the last confirmed [Member](/docs/rb

If you are the only remaining [Member](/docs/rbac/access-roles#member-role), you should instead delete the Team.

Git Default ignore list

Deployments created using Vercel CLI will automatically [ignore several files](/docs/deployments/build-features#ignored-files-and-folders

However, these files are *not* ignored for deployments created using [Git](/docs/git) and a warning is printed instead. This is because `

If the file was intentionally committed to Git, you can ignore the warning.

If the file was accidentally committed to Git, you can remove it using the following commands:

```shell filename="terminal"
git rm file.txt                  # remove the file
echo 'file.txt' >> .gitignore    # append file to .gitignore
git add .gitignore               # stage the change
git commit -m "Removed file.txt" # commit the change
git push                         # deploy the change
```

GitHub app installation not found

In some cases, signing up with GitHub fails due to GitHub's database inconsistencies.

When you connected your Hobby team with your GitHub account, the [Vercel GitHub App](https://github.com/apps/vercel) was installed on you

However, Vercel was unable to retrieve the app installation from GitHub, which made it appear as if the [Vercel GitHub App](https://githu

In order to solve this issue, wait a couple of minutes and try connecting to GitHub again. If you are still unable to connect, please con

Preview branch used as production branch

If you have configured a custom Git branch for a domain or an environment variable, it is considered a preview domain and a preview envir

When configuring the production branch in the project settings, it is not possible to use a preview branch.

If you still want to use this particular Git branch as a production branch, please follow these steps:

1. Assign your affected domains to the production environment (clear out the Git branch you've defined for them)
2. Assign your affected environment variables to the production environment (clear out the Git branch you've defined for them)

Afterwards, you can use the Git branch you originally wanted to use as a production branch.

Lost Git repository access

In order for Vercel to be able to deploy commits to your Git repository, a Project on Vercel has to be connected to it.

This connection is interrupted if the Git repository is deleted, archived, or if the Vercel App was uninstalled from the corresponding Gi

Additionally, when using GitHub, the connection is also interrupted if you or a [Team Member](/docs/rbac/access-roles#member-role) modifi

To verify the access permissions of the Vercel GitHub App installed on your personal GitHub account, navigate to the [**Applications** pa

To verify the access permissions of the Vercel GitHub App installed on your GitHub organization, select **Vercel** under **Installed GitH

Production deployment cannot be redeployed

You cannot redeploy a production deployment if a more recent one exists.

The reason is that redeploying an old production deployment would result in overwriting the most recent source code you have deployed to

To force an explicit overwrite of the current production deployment, select **Promote** instead.

SSL certificate deletion denied

Certain SSL Certificates associated with your Hobby team or team (i.e. Wildcard SSL Certificates for your Vercel Project's staging domain

Because these SSL Certificates are managed by the Vercel platform, they cannot be manually deleted on the Vercel Dashboard – nor through '

Custom SSL Certificates may be uploaded to teams on the [Enteprise plan](/docs/plans/enterprise), which are allowed to be manually delete

Production branch used as preview branch

The Git branch that is configured using the [production branch](/docs/git#production-branch) field in the project settings, is considered

If you'd like to assign a domain or environment variable to that particular Git branch, there's no need to manually fill it in.

By default, if no custom Git branch is defined for them, domains are already assigned to the production branch. The same is true for envi

If you still want to enter a specific Git branch for a domain or an environment variable, it has to be a [preview branch](/docs/git#previ

Command not found in vercel dev

The *"Command not found"* error message happens when a sub-process that `vercel dev` is attempting to create is not installed on your loc

For example, you may see the error *"Command not found: go"* if you are writing a Vercel Function in Go, but do not have the `go` binary

Recursive invocation of commands

Why this error occurred

You have configured one of the following for your Project:

- The [Build Command](/docs/deployments/configure-a-build#build-command) defined in the Project Settings invokes `vercel build`
- The [Development Command](/docs/deployments/configure-a-build#development-command) defined in the Project Settings invokes `vercel dev`

Because the Build Command is invoked by `vercel build` when deploying, it cannot invoke `vercel build` itself, as that would cause an inf

The same applies to the Development Command: When developing locally, `vercel dev` invokes the Development Command, so it cannot invoke `

Possible ways to fix it

Adjust the Build and Development Commands defined for your Project to not invoke `vercel build` or `vercel dev`.

Instead, they should invoke the Build Command provided by your framework.

If you are unsure about which value to provide, disable the **Override** option in order to default to the preferred settings for the [Fr

Pnpm engine unsupported

`ERR_PNPM_UNSUPPORTED_ENGINE` occurs when `package.json#engines.pnpm` does not match the currently running version of `pnpm`.

To fix, do one of the following:

- [Set the env var `ENABLE_EXPERIMENTAL_COREPACK` to 1](https://vercel.com/docs/deployments/configure-a-build#corepack), and make sure th

```json filename="package.json"
{
  "engines": {
    "pnpm": "^7.5.1"
  },
  "packageManager": "pnpm@7.5.1"
}
```

- Remove the [`engines.pnpm`](https://pnpm.io/package_json#engines) value from your `package.json`

> **💡 Note:** You cannot use [`engine-strict`](https://pnpm.io/npmrc#engine-strict) to solve
> this error. `engine-strict` only handles dependencies.

Yarn dynamic require of "util" is not supported

This error occurs when projects use yarn, corepack, and have a `package.json` with a `type` field set to `module`. This is a known [yarn

To prevent this error, consider the following options:

- Remove `"type": "module"` from the project's `package.json`
- Install yarn into the project instead of using corepack with `yarn set version [desired-version] --yarn-path`

Invalid Edge Config connection string

This error occurs when attempting to create a deployment where at least one of its environment variables contains an outdated Edge Config

To resolve this error, delete or update the environment variable that contains the connection string. In most cases, the environment vari

Globally installed `@vercel/speed-insights` or `@vercel/analytics` packages

You must reference `@vercel/speed-insights` or `@vercel/analytics` packages in your application's `package.json` file. This error occurs

To fix this error, add the packages to your `package.json` file as a dependency.

Oversized Incremental Static Regeneration page

[Incremental Static Regeneration](https://vercel.com/docs/incremental-static-regeneration) (ISR) responses that are greater than 20 MB re

---
title: "Error Codes"
description: "Use this guide to find specific solutions and insights for common Vercel errors."
last_updated: "2026-01-16T02:19:30.504Z"
source: "https://vercel.com/docs/errors"

Error Codes

When developing your application with Vercel, you may encounter a variety of errors. They can reflect issues that happen with external pr

For general error handling guidance, that covers dashboard related errors, see [General Errors](/docs/errors/error-list).

Application errors

Platform errors

---
title: "Flags SDK"
description: "The Flags SDK is a free open-source library that gives developers the tools they need to use feature flags in Next.js and S
last_updated: "2026-01-16T02:19:30.515Z"
source: "https://vercel.com/docs/feature-flags/feature-flags-pattern"

Flags SDK

- Works with any [flag provider](https://flags-sdk.dev/docs/adapters/supported-providers), [custom setups](https://flags-sdk.dev/docs/ada
- Compatible with App Router, Pages Router, and Middleware
- Built for feature flags and experimentation

Learn more about the [Flags SDK on flags-sdk.dev](https://flags-sdk.dev).

---
title: "Getting started with Flags Explorer"
description: "Learn how to set up the Flags Explorer so you can see and override your application"
last_updated: "2026-01-16T02:19:30.635Z"
source: "https://vercel.com/docs/feature-flags/flags-explorer/getting-started"

Getting started with Flags Explorer

This guide walks you through connecting your application to the Flags Explorer, so you can use it to view and override your application's

Prerequisites

- Set up the [Vercel Toolbar](/docs/vercel-toolbar) for development by following [adding the Vercel Toolbar to local and production envir
- You should have the latest version of Vercel CLI installed. To check your version, use `vercel --version`. To [install](/docs/cli#insta
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i vercel
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i vercel
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i vercel
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i vercel
      ```
    </Code>
  </CodeBlock>
- Ensure your local directory [links](/docs/cli/link) to a Vercel project. You can use `vercel link` from root of your project to link it
  ```bash filename="Terminal"
  vercel link [path-to-directory]
  ```

Quickstart

- ### Add the Flags SDK to your project
  Install the `flags` package. This package provides convenience methods, components, and types that allow your application to communicat
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i flags
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i flags
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i flags
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i flags
      ```
    </Code>
  </CodeBlock>

- ### Adding a `FLAGS_SECRET`
  This secret is used to encrypt and sign overrides, and so Flags Explorer can make authenticated requests to the `/.well-known/vercel/fl

  Run your application locally with Vercel Toolbar enabled and open Flags Explorer from the toolbar. Click on "Start setup" to begin the

  Pull your environment variables to make them available to your project locally.
  ```bash filename="Terminal"
  vercel env pull
  ```
  If you prefer to create the secret manually, see the instructions in the [Flags Explorer Reference](/docs/feature-flags/flags-explorer/

- ### Creating the Flags Discovery Endpoint
  Your application needs to expose an API endpoint that Flags Explorer queries to get your feature flags. Flags Explorer will make an aut

Using the Flags SDK for Next.js

Ensure you completed the setup of the [Flags SDK for Next.js](https://flags-sdk.dev/docs/getting-started/next). You should have install
```ts filename="flags.ts" framework=nextjs
import { flag } from 'flags/next';

export const exampleFlag = flag({
  key: 'example-flag',
  description: 'An example feature flag',
  decide() {
    return false;
  },
});
```
```js filename="flags.js" framework=nextjs
import { flag } from 'flags/next';

export const exampleFlag = flag({
  key: 'example-flag',
  description: 'An example feature flag',
  decide() {
    return false;
  },
});
```
```ts filename="flags.ts" framework=nextjs-app
import { flag } from 'flags/next';

export const exampleFlag = flag({
  key: 'example-flag',
  description: 'An example feature flag',
  decide() {
    return false;
  },
});
```
```js filename="flags.js" framework=nextjs-app
import { flag } from 'flags/next';

export const exampleFlag = flag({
  key: 'example-flag',
  description: 'An example feature flag',
  decide() {
    return false;
  },
});
```
Create your Flags Discovery Endpoint using the snippet below.
```ts filename="pages/api/vercel/flags.ts" framework=nextjs
import type { NextApiRequest, NextApiResponse } from 'next';
import { verifyAccess, version } from 'flags';
import { getProviderData } from 'flags/next';
import * as flags from '../../../flags';

export default async function handler(
  request: NextApiRequest,
  response: NextApiResponse,
) {
  const access = await verifyAccess(request.headers['authorization']);
  if (!access) return response.status(401).json(null);

  const apiData = getProviderData(flags);

  response.setHeader('x-flags-sdk-version', version);
  return response.json(apiData);
}
```
```js filename="pages/api/vercel/flags.js" framework=nextjs
import { verifyAccess, version } from 'flags';
import { getProviderData } from 'flags/next';
import * as flags from '../../../flags';

export default async function handler(request, response) {
  const access = await verifyAccess(request.headers['authorization']);
  if (!access) return response.status(401).json(null);

  const apiData = getProviderData(flags);

  response.setHeader('x-flags-sdk-version', version);
  return response.json(apiData);
}
```
```ts filename="app/.well-known/vercel/flags/route.ts" framework=nextjs-app
import { getProviderData, createFlagsDiscoveryEndpoint } from 'flags/next';
import * as flags from '../../../../flags';

export const GET = createFlagsDiscoveryEndpoint(() => getProviderData(flags));
```
```js filename="app/.well-known/vercel/flags/route.js" framework=nextjs-app
import { getProviderData, createFlagsDiscoveryEndpoint } from 'flags/next';
import * as flags from '../../../../flags';

export const GET = createFlagsDiscoveryEndpoint(() => getProviderData(flags));
```
This endpoint uses `verifyAccess` to prevent unauthorized requests, and the `getProviderData` function to automatically generate the fe
> For \['nextjs']:
If you are using the Pages Router, you will need to add the following to your `next.config.js`. This is because the Pages Router can't
```js filename="next.config.js"
module.exports = {
  async rewrites() {
```

```
      return [
        {
          source: '/.well-known/vercel/flags',
          destination: '/api/vercel/flags',
        },
      ];
    },
  };
  ```

  If you are using the Flags SDK with adapters, use the `getProviderData` function exported by your flag provider's adapter to load flag r

  Using the Flags SDK for SvelteKit

  If you are using the Flags SDK for SvelteKit then the `createHandle` function will automatically create the API endpoint for you. Learn

  Using a custom setup

  Learn how to manually return feature flag definitions in the [Flags Explorer Reference](/docs/feature-flags/flags-explorer/reference#ve

- ### Handling overrides
  You can now use the Flags Explorer to create feature flag overrides. When you create an override Flags Explorer will set a cookie conta

  Using the Flags SDK for Next.js

  Feature flags defined in code using the Flags SDK for Next.js will automatically handle overrides set by the Flags Explorer.

  Using the Flags SDK for SvelteKit

  Feature flags defined in code using the Flags SDK for SvelteKit will automatically handle overrides set by the Flags Explorer.

  Using a custom setup

  If you have a custom feature flag setup, or if you are using the SDKs of feature flag providers directly, you need to manually handle t

  Learn how to read the overrides cookie in the [Flags Explorer Reference](/docs/feature-flags/flags-explorer/reference#override-cookie).

- ### Emitting flag values (optional)
  You can optionally make the Flags Explorer aware of the actual value each feature flag resolved to while rendering the current page by

  If you emit flag values to the client it's further possible to annotate your Web Analytics events with the feature flags you emitted. L

- ### Review
  You should now be able to see your feature flags in Flags Explorer. You should also be able to set overrides that your application can

More resources

- [Flags Explorer Reference](/docs/feature-flags/flags-explorer/reference)
- [Flags SDK](/docs/feature-flags/feature-flags-pattern)
- [Feature Flags using Next.js example](/templates/next.js/shirt-shop-feature-flags)

---
title: "Pricing for Flags Explorer"
description: "Learn about pricing for Flags Explorer."
last_updated: "2026-01-16T02:19:30.526Z"
source: "https://vercel.com/docs/feature-flags/flags-explorer/limits-and-pricing"

Pricing for Flags Explorer

Pricing

The following table outlines the price for each resource according to the plan you are on:

Resource	Hobby	Pro	Enterprise
Unlimited Overrides	N/A	$250.00 per month	$250.00 per month

Unlimited overrides can be managed in your [team's billing settings](/d?to=%2Fteams%2F%5Bteam%5D%2Fsettings%2Fbilling\&title=Go+to+Billin

Limits per plan

When not subscribed to the unlimited option, Hobby, Pro and Enterprise have a limited amount of monthly overrides:

	Hobby	Pro	Enterprise
Overrides	150 per month	150 per month	150 per month

---
title: "Flags Explorer"
description: "View and override your application"
last_updated: "2026-01-16T02:19:30.647Z"
source: "https://vercel.com/docs/feature-flags/flags-explorer"

Flags Explorer

The Flags Explorer is a feature of the [Vercel Toolbar](/docs/vercel-toolbar) that allows you to view and override your application's fea

Quickly override feature flags for your current session without signing into your feature flag provider, and without affecting team membe

Team members can access the Flags Explorer once they have activated the toolbar. The Flags Explorer is available in all environments your

View and override flags in the toolbar

Before you can use with the Flags Explorer, ensure that your team has set up both [feature flags](/docs/feature-flags/flags-explorer/gett

To see and override feature flags for your application:

1. You must log into the Vercel Toolbar to interact with your application's feature flag overrides.
2. Select the **Flags Explorer** option () from the Vercel Toolbar menu.
3. Find the desired feature flag in the modal by scrolling or using the search and filter controls.
4. Select an override value for the desired feature flag. Note that by default, overrides are not persisted and only affect the user appl
5. Apply the changes. This will trigger a soft reload. If you have applied changes, the Vercel Toolbar will turn blue.

Sharing flag overrides

Any overrides you apply from Vercel Toolbar usually apply to your browser session only. However, you can recommend overrides to team memb

- [Setting overrides as recommended for a given branch](#branch-based-recommendations)
- Explicitly [sharing a set of overrides through a URL](#url-based-recommendations) with a team member

Branch based recommendations

This workflow is great when you start working on a new feature in a branch, as the recommended overrides will travel with the branch from

1. First configure the overrides you would like to share as usual
2. Then, select the chevron next to the branch name at the top
3. Choose **Save Recommendations** to recommend these overrides to any team member visiting your branch locally or on a preview deploymen

When a team member visits that branch they will get a notification suggesting to apply the overrides you recommended. Notifications are d

URL based recommendations

This workflow is great when you want to share once-off overrides with team members to reproduce a bug under certain conditions or to shar

1. First configure the overrides you would like to share as usual
2. Choose **Share** to copy a link to the page you are on, along with a query parameter containing your overrides

You can send this link to team members. When they visit the link they will get a notification suggesting to apply the overrides you share

More resources

- [Flags Explorer reference](/docs/feature-flags/flags-explorer/reference)
- [Flags SDK](/docs/feature-flags/feature-flags-pattern)

---
title: "Reference"
description: "In-depth reference for configuring the Flags Explorer"
last_updated: "2026-01-16T02:19:30.796Z"
source: "https://vercel.com/docs/feature-flags/flags-explorer/reference"

Reference

The Flags Explorer has five main concepts: the [API Endpoint](/docs/feature-flags/flags-explorer/reference#api-endpoint), the [FLAGS_SEC

Definitions

The Flags Explorer needs to know about your feature flags before it can display them.

Flag definitions are metadata for your feature flags, which communicate:

- Name
- URL for where your team can manage the flag
- Description
- Possible values and their (optional) labels

A definition can never communicate the value of a flag as they load independently from [flag values](/docs/feature-flags/flags-explorer/r

```json
{
  "bannerFlag": {
    "origin": "https://example.com/flag/bannerFlag",
    "description": "Determines whether the banner is shown",
    "options": [
      { "value": true, "label": "on" },
      { "value": false, "label": "off" }
    ]
  }
}
```

This is how Vercel Toolbar shows flag definitions:

There are two ways to provide your feature flags to the Flags Explorer:

1. [Returning definitions through the Flags API Endpoint](/docs/feature-flags/flags-explorer/reference#returning-definitions-through-the-
2. [Embedding definitions through script tags](/docs/feature-flags/flags-explorer/reference#embedding-definitions-through-script-tags)

Returning definitions through the Flags API Endpoint

The Flags API Endpoint is the recommended way to provide your feature flags to the Flags Explorer. The Flags Explorer will request your a

See [Definitions properties](/docs/feature-flags/flags-explorer/reference#definitions-properties) for a full list of properties you can r

Embedding definitions through script tags

We strongly recommend communicating your feature flag definitions through the [Flags API Endpoint](/docs/feature-flags/flags-explorer/ref

If you are using React or Next.js, use the [`FlagsDefinitions`](https://flags-sdk.dev/docs/api-reference/core/react#flagdefinitions) compo

```ts
type FlagDefinitionsType = Record<
  string,
  {
    options?: {
      value: any;
```

```
      label?: string;
    }[];
    origin?: string;
    description?: string;
  }
>;
```

This example shows how to communicate a feature flag definition through the DOM:

```html
<script type="application/json" data-flag-definitions>
  {
    "showBanner": {
      "description": "Shows or hide the banner",
      "origin": "https://example.com/showBanner",
      "options": [
        { "value": false, "label": "Hide" },
        { "value": true, "label": "Show" }
      ]
    }
  }
</script>
```

You can also encrypt the definitions before emitting them to prevent leaking your feature flags through the DOM.

```js
import { safeJsonStringify } from 'flags';

<script type="application/json" data-flag-definitions>
  ${safeJsonStringify(definitions)}
</script>;
```

> **💡 Note:** Using `JSON.stringify` within script tags leads to [XSS
> vulnerabilities](https://owasp.org/www-community/attacks/xss/). Use
> `safeJsonStringify` exported by `flags` to stringify safely.

Values

Your Flags API Endpoint returns your application's feature flag definitions containing information like their key, description, origin, a

You can optionally provide the values of your feature flags to Flags Explorer in two ways:

1. [Emitting values using the React components](/docs/feature-flags/flags-explorer/reference#emitting-values-using-the-flagvalues-react-c
2. [Embedding values through script tags](/docs/feature-flags/flags-explorer/reference#embedding-values-through-script-tags)

Emitted values will show up in the Flags Explorer, and will be used by [Web Analytics to annotate events](/docs/feature-flags/integrate-w

This is how Vercel Toolbar shows flag values:

Any JSON-serializable values are supported. Flags Explorer combines these values with any definitions, if they are present.

```json
{ "bannerFlag": true, "buttonColor": "blue" }
```

Emitting values using the FlagValues React component

The `flags` package exposes React components which allow making the Flags Explorer aware of your feature flag's values.

```tsx filename="pages/index.tsx" framework=nextjs
import { FlagValues } from 'flags/react';

export default function Page() {
  return (
    <div>
      {/* Some other content */}
      <FlagValues values={{ exampleFlag: true }} />
    </div>
  );
}
```

```jsx filename="pages/index.jsx" framework=nextjs
import { FlagValues } from 'flags/react';

export default function Page() {
  return (
    <div>
      {/* Some other content */}
      <FlagValues values={{ exampleFlag: true }} />
    </div>
  );
}
```

```tsx filename="app/page.tsx" framework=nextjs-app
import { FlagValues } from 'flags/react';

export function Page() {
  return (
    <div>
      {/* Some other content */}
      <FlagValues values={{ exampleFlag: true }} />
    </div>
  );
}
```

```jsx filename="app/page.jsx" framework=nextjs-app
import { FlagValues } from 'flags/react';

export function Page() {
  return (
    <div>
      {/* Some other content */}
      <FlagValues values={{ exampleFlag: true }} />
    </div>
  );
}
```

The approaches above will add the names and values of your feature flags to the DOM in plain text. Use the `encrypt` function to keep you

```tsx filename="pages/index.tsx" framework=nextjs
import type { GetServerSideProps, GetServerSidePropsContext } from 'next';
import { encryptFlagValues, decryptOverrides } from 'flags';
import { FlagValues } from 'flags/react';

type Flags = {
  banner: boolean;
};

async function getFlags(
  request: GetServerSidePropsContext['req'],
): Promise<Flags> {
  const overridesCookieValue = request.cookies['vercel-flag-overrides'];
  const overrides = overridesCookieValue
    ? await decryptOverrides(overridesCookieValue)
    : null;

  return {
    banner: overrides?.banner ?? false,
  };
}

export const getServerSideProps: GetServerSideProps<{
  flags: Flags;
  encryptedFlagValues: string;
}> = async (context) => {
  const flags = await getFlags(context.req);
  const encryptedFlagValues = await encryptFlagValues(flags);

  return { props: { flags, encryptedFlagValues } };
};

export default function Page({
  flags,
  encryptedFlagValues,
}: {
  flags: Flags;
  encryptedFlagValues: string;
}) {
  return (
    <>
      <FlagValues values={encryptedFlagValues} />
      {flags.banner ? <div>Banner</div> : null}
    </>
  );
}
```

```jsx filename="pages/index.jsx" framework=nextjs
import { encryptFlagValues, decryptOverrides } from 'flags';
import { FlagValues } from 'flags/react';

async function getFlags(request) {
  const overridesCookieValue = request.cookies['vercel-flag-overrides'];
  const overrides = overridesCookieValue
    ? await decryptOverrides(overridesCookieValue)
    : null;

  return {
    banner: overrides?.banner ?? false,
  };
}

export const getServerSideProps = async (context) => {
  const flags = await getFlags(context.req);
  const encryptedFlagValues = await encryptFlagValues(flags);

  return { props: { flags, encryptedFlagValues } };
};

export default function Page({ flags, encryptedFlagValues }) {
  return (
    <>
      <FlagValues values={encryptedFlagValues} />
      {flags.banner ? <div>Banner</div> : null}
    </>
  );
}
```

```tsx filename="app/page.tsx" framework=nextjs-app
import { Suspense } from 'react';
import { encryptFlagValues, type FlagValuesType } from 'flags';
import { FlagValues } from 'flags/react';
```

```
async function ConfidentialFlagValues({ values }: { values: FlagValuesType }) {
  const encryptedFlagValues = await encryptFlagValues(values);
  return <FlagValues values={encryptedFlagValues} />;
}

export default function Page() {
  const values: FlagValuesType = { exampleFlag: true };

  return (
    <div>
      {/* Some other content */}
      <Suspense fallback={null}>
        <ConfidentialFlagValues values={values} />
      </Suspense>
    </div>
  );
}
```

```jsx filename="app/page.jsx" framework=nextjs-app
import { Suspense } from 'react';
import { encryptFlagValues } from 'flags';
import { FlagValues } from 'flags/react';

async function ConfidentialFlagValues({ values }) {
  const encryptedFlagValues = await encryptFlagValues(values);
  return <FlagValues values={encryptedFlagValues} />;
}

export default function Page() {
  const values = { exampleFlag: true };

  return (
    <div>
      {/* Some other content */}
      <Suspense fallback={null}>
        <ConfidentialFlagValues values={values} />
      </Suspense>
    </div>
  );
}
```

The `FlagValues` component will emit a script tag with a `data-flag-values` attribute, which get picked up by the Flags Explorer. Flags E

Embedding values through script tags

Flags Explorer scans the DOM for script tags with the `data-flag-values` attribute. Any changes to content get detected by a mutation obs

You can emit the values of feature flags to the Flags Explorer by rendering script tags with the `data-flag-values` attribute.

```html
<script type="application/json" data-flag-values>
  {
    "showBanner": true,
    "showAds": false,
    "pricing": 5
  }
</script>
```

> **💡 Note:** Be careful when creating these script tags. Using `JSON.stringify` within
> script tags leads to [XSS
> vulnerabilities](https://owasp.org/www-community/attacks/xss/). Use
> `safeJsonStringify` exported by `flags` to stringify safely.

The expected shape is:

```ts
type FlagValues = Record<string, any>;
```

To prevent disclosing feature flag names and values to the client, the information can be encrypted. This keeps the feature flags confide

```tsx
import { encryptFlagValues, safeJsonStringify } from 'flags';

// Encrypt your flags and their values on the server.
const encryptedFlagValues = await encryptFlagValues({
  showBanner: true,
  showAds: false,
  pricing: 5,
});

// Render the encrypted values on the client.
// Note: Use `safeJsonStringify` to ensure `encryptedFlagValues` is correctly formatted as JSON.
//       This step may vary depending on your framework or setup.
<script type="application/json" data-flag-values>
  {safeJsonStringify(encryptedFlagValues)}
</script>;
```

`FLAGS_SECRET` environment variable

This secret gates access to the Flags API endpoint, and optionally enables signing and encrypting feature flag overrides set by Vercel To

You can create this secret by following the instructions in the [Flags Explorer Quickstart](/docs/feature-flags/flags-explorer/getting-st

Manually creating the `FLAGS_SECRET`

The `FLAGS_SECRET` value must have a specific length (32 random bytes encoded in base64) to work as an encryption key. You can create one

```bash filename="Terminal"
node -e "console.log(crypto.randomBytes(32).toString('base64url'))"
```

In your local environment, pull your environment variables with `vercel env pull` to make them available to your project.

> **💡 Note:** The `FLAGS_SECRET` environment variable must be defined in your project
> settings on the Vercel dashboard. Defining the environment variable locally is
> not enough as Flags Explorer reads the environment variable from your project
> settings.

API endpoint

When you have set the [`FLAGS_SECRET`](/docs/feature-flags/flags-explorer/reference#flags_secret-environment-variable) environment variab

Verifying a request to the API endpoint

Your endpoint should call `verifyAccess` to ensure the request to load flags originates from Vercel Toolbar. This prevents your feature f

If the `verifyAccess` check fails, you should return status code `401` and no response body. When the `verifyAccess` check is successful,

Using the Flags SDK

```ts filename="pages/api/vercel/flags.ts" framework=nextjs
import type { NextApiRequest, NextApiResponse } from 'next';
import { verifyAccess, version } from 'flags';
import { getProviderData } from 'flags/next';
import * as flags from '../../../flags';

export default async function handler(
  request: NextApiRequest,
  response: NextApiResponse,
) {
  const access = await verifyAccess(request.headers['authorization']);
  if (!access) return response.status(401).json(null);

  const apiData = getProviderData(flags);

  response.setHeader('x-flags-sdk-version', version);
  return response.json(apiData);
}
```

```js filename="pages/api/vercel/flags.js" framework=nextjs
import { verifyAccess, version } from 'flags';
import { getProviderData } from 'flags/next';
import * as flags from '../../../flags';

export default async function handler(request, response) {
  const access = await verifyAccess(request.headers['authorization']);
  if (!access) return response.status(401).json(null);

  const apiData = getProviderData(flags);

  response.setHeader('x-flags-sdk-version', version);
  return response.json(apiData);
}
```

```ts filename="app/.well-known/vercel/flags/route.ts" framework=nextjs-app
import { getProviderData, createFlagsDiscoveryEndpoint } from 'flags/next';
import * as flags from '../../../../flags';

export const GET = createFlagsDiscoveryEndpoint(() => getProviderData(flags));
```

```js filename="app/.well-known/vercel/flags/route.js" framework=nextjs-app
import { getProviderData, createFlagsDiscoveryEndpoint } from 'flags/next';
import * as flags from '../../../../flags';

export const GET = createFlagsDiscoveryEndpoint(() => getProviderData(flags));
```

Using a custom setup

If you are not using the Flags SDK to define feature flags in code, or if you are not using Next.js or SvelteKit, you need to manually re

```ts filename="pages/api/vercel/flags.ts" framework=nextjs
import { verifyAccess } from 'flags';

export default async function handler(request, response) {
  const access = await verifyAccess(request.headers['authorization'] as string);
  if (!access) return response.status(401).json(null);

  return response.json({
    definitions: {
      newFeature: {
        description: 'Controls whether the new feature is visible',
        origin: 'https://example.com/#new-feature',
        options: [
          { value: false, label: 'Off' },
          { value: true, label: 'On' },
        ],
      },
    },
  });
}
```

```

```js filename="pages/api/vercel/flags.js" framework=nextjs
import { verifyAccess } from 'flags';

export default async function handler(request, response) {
  const access = await verifyAccess(request.headers['authorization']);
  if (!access) return response.status(401).json(null);

  return response.json({
    definitions: {
      newFeature: {
        description: 'Controls whether the new feature is visible',
        origin: 'https://example.com/#new-feature',
        options: [
          { value: false, label: 'Off' },
          { value: true, label: 'On' },
        ],
      },
    },
  });
}
```

```ts filename="app/.well-known/vercel/flags/route.ts" framework=nextjs-app
import { NextResponse, type NextRequest } from 'next/server';
import { verifyAccess, type ApiData } from 'flags';

export async function GET(request: NextRequest) {
  const access = await verifyAccess(request.headers.get('Authorization'));
  if (!access) return NextResponse.json(null, { status: 401 });

  return NextResponse.json<ApiData>({
    definitions: {
      newFeature: {
        description: 'Controls whether the new feature is visible',
        origin: 'https://example.com/#new-feature',
        options: [
          { value: false, label: 'Off' },
          { value: true, label: 'On' },
        ],
      },
    },
  });
}
```

```js filename="app/.well-known/vercel/flags/route.js" framework=nextjs-app
import { NextResponse } from 'next/server';
import { verifyAccess } from 'flags';

export async function GET(request) {
  const access = await verifyAccess(request.headers.get('Authorization'));
  if (!access) return NextResponse.json(null, { status: 401 });

  return NextResponse.json({
    definitions: {
      newFeature: {
        description: 'Controls whether the new feature is visible',
        origin: 'https://example.com/#new-feature',
        options: [
          { value: false, label: 'Off' },
          { value: true, label: 'On' },
        ],
      },
    },
  });
}
```

Valid JSON response

The JSON response must have the following shape

```ts
type ApiData = {
  definitions: Record<
    string,
    {
      description?: string;
      origin?: string;
      options?: { value: any; label?: string }[];
    }
  >;
  hints?: { key: string; text: string }[];
  overrideEncryptionMode?: 'plaintext' | 'encrypted';
};
```

Definitions properties

These are your application's feature flags. You can return the following data for each definition:

Property	Type	Description
`description` (optional)	string	A description of what this feature flag is for.
`origin` (optional)	string	The URL where feature flag is managed. This usually points to the flag
`options` (optional)	`{ value: any, label?: string }[]`	An array of options. These options will be available as overrides in Ve

You can optionally tell Vercel Toolbar about the actual value flags resolved to. The Flags API Endpoint cannot return this as the value m
```

Hints

In some cases you might need to fetch your feature flag definitions from your feature flag provider before you can return them from the F

In case this request fails you can use `hints`. Any hints returned will show up in the UI.

This is useful when you are fetching your feature flags from multiple sources. In case one request fails you might still want to show the

Override mode

When you create an override, Vercel Toolbar will set a cookie called `vercel-flag-overrides`. You can read this cookie in your applicatio

The `overrideEncryptionMode` setting controls the value of the cookie:

- `plaintext`: The cookie will contain the overrides as plain JSON. Be careful not to trust those overrides as users can manipulate the v
- `encrypted`: Vercel Toolbar will encrypt overrides using the `FLAGS_SECRET` before storing them in the cookie. This prevents manipulati

We highly recommend using `encrypted` mode as it protects against manipulation.

Override cookie

The Flags Explorer will set a cookie called `vercel-flag-overrides` containing the overrides.

Using the Flags SDK

If you use the Flags SDK for Next.js or SvelteKit, the SDK will automatically handle the overrides set by the Flags Explorer.

Manual setup

Read this cookie and use the `decrypt` function to decrypt the overrides and use them in your application. The decrypted value is a JSON

```ts filename="app/getFlags.ts" framework=nextjs
import { decryptOverrides, type FlagOverridesType } from 'flags';
import { type NextRequest } from 'next/server';

async function getFlags(request: NextRequest) {
  const overrideCookie = request.cookies.get('vercel-flag-overrides')?.value;
  const overrides = overrideCookie
    ? await decryptOverrides<FlagOverridesType>(overrideCookie)
    : null;

  const flags = {
    exampleFlag: overrides?.exampleFlag ?? false,
  };

  return flags;
}
```

```js filename="app/getFlags.js" framework=nextjs
import { decryptOverrides } from 'flags';

async function getFlags(request) {
  const overrideCookie = request.cookies.get('vercel-flag-overrides')?.value;
  const overrides = overrideCookie
    ? await decryptOverrides(overrideCookie)
    : null;

  const flags = {
    exampleFlag: overrides?.exampleFlag ?? false,
  };

  return flags;
}
```

```ts filename="app/getFlags.ts" framework=nextjs-app
import { type FlagOverridesType, decryptOverrides } from 'flags';
import { cookies } from 'next/headers';

async function getFlags() {
  const overrideCookie = cookies().get('vercel-flag-overrides')?.value;
  const overrides = overrideCookie
    ? await decryptOverrides<FlagOverridesType>(overrideCookie)
    : null;

  return {
    exampleFlag: overrides?.exampleFlag ?? false,
  };
}
```

```js filename="app/getFlags.js" framework=nextjs-app
import { decryptOverrides } from 'flags';
import { cookies } from 'next/headers';

async function getFlags() {
  const overrideCookie = cookies().get('vercel-flag-overrides')?.value;
  const overrides = overrideCookie
    ? await decryptOverrides(overrideCookie)
    : null;

  return {
    exampleFlag: overrides?.exampleFlag ?? false,
  };
}
```

Script tags

Vercel Toolbar uses a [MutationObserver](https://developer.mozilla.org/docs/Web/API/MutationObserver) to find all script tags with `data-

For more information, see the following sections:

- [Embedding definitions through script tags](/docs/feature-flags/flags-explorer/reference#embedding-definitions-through-script-tags)
- [Embedding values through script tags](/docs/feature-flags/flags-explorer/reference#embedding-values-through-script-tags)

---
title: "Integrating with the Vercel Platform"
description: "Integrate your feature flags with the Vercel Platform."
last_updated: "2026-01-16T02:19:30.662Z"
source: "https://vercel.com/docs/feature-flags/integrate-vercel-platform"

Integrating with the Vercel Platform

Feature flags play a crucial role in the software development lifecycle, enabling safe feature rollouts, experimentation, and A/B testing

By making the Vercel platform aware of the feature flags used in your application, you can gain insights in the following ways:

- **Runtime Logs**: See your feature flag's values in [Runtime Logs](/docs/runtime-logs)
- **Web Analytics**: Break down your pageviews and custom events by feature flags in [Web Analytics](/docs/analytics)

To get started, follow these guides:

- [Integrate Feature Flags with Runtime Logs](/docs/feature-flags/integrate-with-runtime-logs)
- [Integrate Feature Flags with Web Analytics](/docs/feature-flags/integrate-with-web-analytics)

---
title: "Integrate flags with Runtime Logs"
description: "Integrate your feature flag provider with runtime logs."
last_updated: "2026-01-16T02:19:30.668Z"
source: "https://vercel.com/docs/feature-flags/integrate-with-runtime-logs"

Integrate flags with Runtime Logs

On your dashboard, the **[Logs](/docs/runtime-logs)** tab displays your [runtime logs](/docs/runtime-logs#what-are-runtime-logs). It can

To make the runtime logs aware of your feature flag call `reportValue(name, value)` with the flag name and value to be reported. Each cal

```ts {1,8} filename="app/api/test/route.ts" framework=nextjs-app
import { reportValue } from 'flags';

export async function GET() {
  reportValue('summer-sale', false);
  return Response.json({ ok: true });
}
```

```js {1,8} filename="app/api/test/route.js" framework=nextjs-app
import { reportValue } from 'flags';

export async function GET() {
  reportValue('summer-sale', false);
  return Response.json({ ok: true });
}
```

```ts {1,4} filename="api/test/page.tsx" framework=nextjs
import { reportValue } from "flags";

export default function Test() {
  reportValue("summer-sale", false);
  return <p>test</p>;
}
```

```js {1,4} filename="api/test/page.jsx" framework=nextjs
import { reportValue } from 'flags';

export default function Test() {
  reportValue('summer-sale', false);
  return <p>test</p>;
}
```

> **💡 Note:** If you are using an implementation of the [Feature Flags
> pattern](/docs/feature-flags/feature-flags-pattern) you don't need to call
> `reportValue`. The respective implementation will automatically call
> `reportValue` for you.

Limits

The following limits apply to reported values:

- Keys are truncated to 256 characters
- Values are truncated to 256 characters
- Reported values must be JSON serializable or they will be ignored

---
title: "Integrate flags with Vercel Web Analytics"
description: "Learn how to tag your page views and custom events with feature flags"
last_updated: "2026-01-16T02:19:30.675Z"
source: "https://vercel.com/docs/feature-flags/integrate-with-web-analytics"

Integrate flags with Vercel Web Analytics

Client-side tracking

Vercel Web Analytics can look up the values of evaluated feature flags in the DOM. It can then enrich page views and client-side events w

- ### Emit feature flags and connect them to Vercel Web Analytics
  To share your feature flags with Web Analytics you have to emit your feature flag values to the DOM as described in [Supporting Feature

  This will automatically annotate all page views and client-side events with your feature flags.

- ### Tracking feature flags in client-side events
  Client-side events in Web Analytics will now automatically respect your flags and attach those to custom events.

  To manually overwrite the tracked flags for a specific `track` event, call:
  ```ts filename="component.ts"
  import { track } from '@vercel/analytics';

  track('My Event', {}, { flags: ['summer-sale'] });
  ```
  If the flag values on the client are encrypted, the entire encrypted string becomes part of the event payload. This can lead to the eve

Server-side tracking

To track feature flags in server-side events:

1. First, report the feature flag value using `reportValue` to make the flag show up in [Runtime Logs](/docs/runtime-logs):

   ```ts {1, 8} filename="app/api/test/route.ts"
   import { reportValue } from 'flags';

   export async function GET() {
     reportValue('summer-sale', false);
     return Response.json({ ok: true });
   }
   ```

2. Once reported, any calls to `track` can look up the feature flag while handling a specific request:

   ```ts {1, 10} filename="app/api/test/route.ts"
   import { track } from '@vercel/analytics/server';
   import { reportValue } from 'flags';

   export async function GET() {
     reportValue('summer-sale', false);
     track('My Event', {}, { flags: ['summer-sale'] });

     return Response.json({ ok: true });
   }
   ```

> **💡 Note:** If you are using an implementation of the [Feature Flags
> Pattern](/docs/feature-flags/feature-flags-pattern) you don't need to call
> `reportValue`. The respective implementation will automatically call
> `reportValue` for you.

---
title: "Feature Flags"
description: "Learn how to use feature flags with Vercel"
last_updated: "2026-01-16T02:19:30.682Z"
source: "https://vercel.com/docs/feature-flags"

Feature Flags

Feature flags are a powerful tool that allows you to control the visibility of features in your application, enabling you to ship, test,

Choose how you work with flags

Vercel provides a flexible approach to working with flags, allowing you to tailor the process to your team's workflow at any stage of the

- [Implement flags as code](#implementing-feature-flags-in-your-codebase), using the [Flags SDK](/docs/feature-flags/feature-flags-patter
- [Manage feature flags](#managing-feature-flags-from-the-toolbar) through the Vercel Toolbar to view, override, and share your applicati
- [Observe your flags](#observing-your-flags) using Vercel's observability features.
- [Optimize your feature flags](#optimizing-your-feature-flags) by using an [Edge Config integration](/docs/edge-config/integrations).

Implementing Feature Flags in your codebase

If you're using **Next.js** or **SvelteKit** for your application, you can implement feature flags directly in your codebase. In Next.js,

- Vercel is compatible with any feature flag provider including [LaunchDarkly](https://launchdarkly.com/), [Optimizely](https://www.optim
- [Flags SDK](/docs/feature-flags/feature-flags-pattern): A free open-source library that gives you the tools you need to use feature fla

Managing Feature Flags from the Toolbar

Using the [Vercel Toolbar](/docs/vercel-toolbar), you're able to view, override, and share feature flags for your application without lea

You can manage feature flags from the toolbar in any development environment that your team has [enabled the toolbar for](/docs/vercel-to

- [Using Feature Flags in the Vercel Toolbar](/docs/feature-flags/flags-explorer): Learn how to view and override your application's feat
- [Implementing Feature Flags in the Vercel Toolbar](/docs/feature-flags/flags-explorer/getting-started): Learn how to set up the Vercel

Observing your flags

Feature flags play a crucial role in the software development lifecycle, enabling safe feature rollouts, experimentation, and A/B testing

- [Integrate Feature Flags with Runtime Logs](/docs/feature-flags/integrate-with-runtime-logs): Learn how to send feature flag data to Ve
- [Integrate Feature Flags with Web Analytics](/docs/feature-flags/integrate-with-web-analytics): Learn how to tag your page views and cu

Optimizing your feature flags

An Edge Config is a global data store that enables experimentation with feature flags, A/B testing, critical redirects, and IP blocking.

With Vercel's optimizations, you can read Edge Config data at negligible latency. The vast majority of your reads will complete within 15

- [Vercel Edge Config](/docs/edge-config): Experiment with A/B testing by storing feature flags in your Edge Config.
- [Vercel Edge Config Quickstart](/docs/edge-config/get-started): Get started with reading data from Edge Config.

---
title: "Fluid compute"
description: "Learn about fluid compute, an execution model for Vercel Functions that provides a more flexible and efficient way to run y
last_updated: "2026-01-16T02:19:30.698Z"
source: "https://vercel.com/docs/fluid-compute"

Fluid compute

Fluid compute offers a blend of serverless flexibility and server-like capabilities. Unlike traditional [serverless architectures](/docs/

- [**Zero configuration out of the box**](/docs/fluid-compute#default-settings-by-plan): Fluid compute comes with preset defaults that au
- [**Optimized concurrency**](/docs/fluid-compute#optimized-concurrency): Optimize resource usage by handling multiple invocations within
- **Dynamic scaling**: Fluid compute automatically optimizes existing resources before scaling up to meet traffic demands. This ensures l
- **Background processing**: After fulfilling user requests, you can continue executing background tasks using [`waitUntil`](/docs/functi
- **Automatic cold start optimizations**: Reduces the effects of cold starts through [automatic bytecode optimization](/docs/fluid-comput
- **Cross-region and availability zone failover**: Ensure high availability by first failing over to [another availability zone (AZ)](/do
- **Error isolation**: Unhandled errors won't crash other concurrent requests running on the same instance, maintaining reliability witho

See [What is compute?](/docs/fundamentals/what-is-compute) to learn more about fluid compute and how it compares to traditional serverles

Enabling fluid compute

> **💡 Note:** As of April 23, 2025, fluid compute is enabled by default for new projects.

You can enable fluid compute through the Vercel dashboard or by configuring your `vercel.json` file for specific environments or deployme

Enable for entire project

To enable fluid compute through the dashboard:

1. Navigate to your project's [Functions Settings](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Fsettings%2Ffunctions\&title=Go
2. Locate the **Fluid Compute** section
3. Toggle the switch to enable fluid compute for your project
4. Click **Save** to apply the changes
5. Deploy your project for the changes to take effect

When you enable it through the dashboard, fluid compute applies to all deployments for that project by default.

Enable for specific environments and deployments

You can programmatically enable fluid compute using the [`fluid` property](/docs/project-configuration#fluid) in your `vercel.json` file.

- **Testing on specific environments**: Enable fluid compute only for custom environments environments when using branch tracking
- **Per-deployment configuration**: Test fluid compute on individual deployments before enabling it project-wide

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "fluid": true
}
```

Available runtime support

Fluid compute is available for the following runtimes:

- [Node.js](/docs/functions/runtimes/node-js)
- [Python](/docs/functions/runtimes/python)
- [Edge](/docs/functions/runtimes/edge)
- [Bun](/docs/functions/runtimes/bun)
- [Rust](/docs/functions/runtimes/rust)

Optimized concurrency

Fluid compute allows multiple invocations to share a single function instance, this is especially valuable for AI applications, where tas

Vercel Functions prioritize existing idle resources before allocating new ones, reducing unnecessary compute usage. This in-function-conc

Optimized concurrency in fluid compute is available when using Node.js or Python runtimes. See the [efficient serverless Node.js with in-

Bytecode caching

When using [Node.js version 20+](/docs/functions/runtimes/node-js/node-js-versions), Vercel Functions use bytecode caching to reduce cold

As a result, the first request isn't cached yet. However, subsequent requests benefit from the cached bytecode, enabling faster initializ

Bytecode caching is only applied to production environments, and is not available in development or preview deployments.

> **💡 Note:** For [frameworks](/docs/frameworks) that output ESM, all CommonJS dependencies
> (for example, `react`, `node-fetch`) will be opted into bytecode caching.

Isolation boundaries and global state

On traditional serverless compute, the isolation boundary refers to the separation of individual instances of a function to ensure they d

However, because each function uses a microVM for isolation, which can lead to slower start-up times, you can see an increase in resource

Fluid compute uses a different approach to isolation. Instead of using a microVM for each function invocation, multiple invocations can s

When [uncaught exceptions](https://nodejs.org/api/process.html#event-uncaughtexception) or [unhandled rejections](https://nodejs.org/api/

Default settings by plan

Fluid Compute includes default settings that vary by plan:

Settings	**Hobby**	**
[**CPU configuration**](/docs/functions/configuring-functions/memory#memory-/-cpu-type)	Standard	St
[**Default / Max duration**](/docs/functions/limitations#max-duration)	300s (5 minutes) / 300s (5 minutes)	30
[**Multi-region failover**](/docs/functions/configuring-functions/region#automatic-failover)	
[**Multi-region functions**](/docs/functions/runtimes#location)		Up to 3

Order of settings precedence

The settings you configure in your [function code](/docs/functions/configuring-functions), [dashboard](/dashboard), or [`vercel.json`](/d

The following order of precedence determines which settings take effect. Settings you define later in the sequence will always override t

Precedence	**Stage**	**Explanation**
1	**Function code**	Settings in your function code always take top priority. These include max duration defined direc
2	**`vercel.json`**	Any settings in your [`vercel.json`](/docs/project-configuration) file, like max duration, and re
3	**Dashboard**	Changes made in the dashboard, such as max duration, region, or CPU, override Fluid defaults.
4	**Fluid defaults**	These are the default settings applied automatically when fluid compute is enabled, and do not co

Pricing and usage

See the [fluid compute pricing](/docs/functions/usage-and-pricing) documentation for details on how fluid compute is priced, including ac

---
title: "Elysia on Vercel"
description: "Build fast TypeScript backends with Elysia and deploy to Vercel. Learn the project structure, plugins, middleware, and how
last_updated: "2026-01-16T02:19:30.708Z"
source: "https://vercel.com/docs/frameworks/backend/elysia"

Elysia on Vercel

Elysia is an ergonomic web framework for building backend servers with Bun. Designed with simplicity and type-safety in mind, Elysia offe

You can deploy an Elysia app to Vercel with zero configuration.

Elysia applications on Vercel benefit from:

- [Fluid compute](/docs/fluid-compute): Active CPU billing, automatic cold start prevention, optimized concurrency, background processing
- [Preview deployments](/docs/deployments/environments#preview-environment-pre-production): Test your changes on a copy of your productio
- [Instant Rollback](/docs/instant-rollback): Recover from unintended changes or bugs in milliseconds
- [Vercel Firewall](/docs/vercel-firewall): Protect your applications from a wide range of threats with a multi-layered security system
- [Secure Compute](/docs/secure-compute): Create private links between your Vercel-hosted backend and other clouds

Get started with Elysia on Vercel

Get started by initializing a new Elysia project using [Vercel CLI init command](/docs/cli/init):

```bash filename="terminal"
vc init elysia
```

> **💡 Note:** Minimum CLI version required: 49.0.0

This will clone the [Elysia example repository](https://github.com/vercel/vercel/tree/main/examples/elysia) in a directory called `elysia

To deploy, [connect your Git repository](/new) or [use Vercel CLI](/docs/cli):

```bash filename="terminal"
vc deploy
```

> **💡 Note:** Minimum CLI version required: 49.0.0

Entrypoint detection

To run an Elysia application on Vercel, create a file that imports the `elysia` package at any one of the following locations:

- `app.{js,cjs,mjs,ts,cts,mts}`
- `index.{js,cjs,mjs,ts,cts,mts}`
- `server.{js,cjs,mjs,ts,cts,mts}`
- `src/app.{js,cjs,mjs,ts,cts,mts}`
- `src/index.{js,cjs,mjs,ts,cts,mts}`
- `src/server.{js,mjs,cjs,ts,cts,mts}`

The file must also export the application as a default export of the module or use a port listener.

Using a default export

For example, use the following code that exports your Elysia app:

```js filename="src/index.js" framework=all
// For Node.js, ensure "type": "module" in package.json
// (Not required for Bun)
import { Elysia } from 'elysia';

const app = new Elysia().get('/', () => ({
  message: 'Hello from Elysia on Vercel!',
}));
```

```ts
// Export the Elysia app
export default app;
```

```ts filename="src/index.ts" framework=all
// For Node.js, ensure "type": "module" in package.json
// (Not required for Bun)
import { Elysia } from 'elysia';

const app = new Elysia().get('/', () => ({
  message: 'Hello from Elysia on Vercel!',
}));

// Export the Elysia app
export default app;
```

Using a port listener

Running your application using `app.listen` is currently not supported. For now, prefer `export default app`.

Local development

To run your Elysia application locally, you can use [Vercel CLI](https://vercel.com/docs/cli/dev):

```bash filename="terminal"
vc dev
```

> **💡 Note:** Minimum CLI version required: 49.0.0

Using Node.js

Ensure `type` is set to `module` in your `package.json` file:

```json filename="package.json"
{
  "name": "elysia-app",
  "type": "module",
}
```

> **💡 Note:** Minimum CLI version required: 49.0.0

Using the Bun runtime

To use the Bun runtime on Vercel, configure the runtime in `vercel.json`:

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "bunVersion": "1.x"
}
```

For more information, [visit the Bun runtime on Vercel documentation](/docs/functions/runtimes/bun).

Middleware

Elysia Plugins and Lifecycle Hooks

In Elysia, you can use plugins and lifecycle hooks to run code before and after request handling. This is commonly used for logging, auth

```ts filename="src/index.ts" framework="elysia"
import { Elysia } from 'elysia';

const app = new Elysia()
  .onBeforeHandle(({ request }) => {
    // Runs before route handler
    console.log('Request:', request.url);
  })
  .onAfterHandle(({ response }) => {
    // Runs after route handler
    console.log('Response:', response.status);
  })
  .get('/', () => 'Hello Elysia!');

export default app;
```

Vercel Routing Middleware

In Vercel, [Routing Middleware](/docs/routing-middleware) executes before a request is processed by your application. Use it for rewrites

Vercel Functions

When you deploy an Elysia app to Vercel, your server endpoints automatically run as [Vercel Functions](/docs/functions) and use [Fluid co

More resources

- [Elysia documentation](https://elysiajs.com)
- [Backend templates on Vercel](https://vercel.com/templates?type=backend)

---
title: "Express on Vercel"
description: "Deploy Express applications to Vercel with zero configuration. Learn about middleware and Vercel Functions."
last_updated: "2026-01-16T02:19:30.758Z"
source: "https://vercel.com/docs/frameworks/backend/express"

Express on Vercel

Express is a fast, unopinionated, minimalist web framework for Node.js. You can deploy an Express app to Vercel with zero configuration.

Express applications on Vercel benefit from:

- [Fluid compute](/docs/fluid-compute): Active CPU billing, automatic cold start prevention, optimized concurrency, background processing
- [Preview deployments](/docs/deployments/environments#preview-environment-pre-production): Test your changes on a copy of your productio
- [Instant Rollback](/docs/instant-rollback): Recover from unintended changes or bugs in milliseconds
- [Vercel Firewall](/docs/vercel-firewall): Protect your applications from a wide range of threats with a multi-layered security system
- [Secure Compute](/docs/secure-compute): Create private links between your Vercel-hosted backend and other clouds

Get started with Express on Vercel

You can quickly deploy an Express application to Vercel by creating an Express app or using an existing one:

Get started with Vercel CLI

Get started by initializing a new Express project using [Vercel CLI init command](/docs/cli/init):

```bash filename="terminal"
vc init express
```

This will clone the [Express example repository](https://github.com/vercel/vercel/tree/main/examples/express) in a directory called `expr

Exporting the Express application

To run an Express application on Vercel, create a file that imports the `express` package at any one of the following locations:

- `app.{js,cjs,mjs,ts,cts,mts}`
- `index.{js,cjs,mjs,ts,cts,mts}`
- `server.{js,cjs,mjs,ts,cts,mts}`
- `src/app.{js,cjs,mjs,ts,cts,mts}`
- `src/index.{js,cjs,mjs,ts,cts,mts}`
- `src/server.{js,mjs,cjs,ts,cts,mts}`

The file must also export the application as a default export of the module or use a port listener.

Using a default export

For example, use the following code that exports your Express app:

```js filename="src/index.js" framework=express
// Use "type: commonjs" in package.json to use CommonJS modules
const express = require('express');
const app = express();

// Define your routes
app.get('/', (req, res) => {
  res.json({ message: 'Hello from Express on Vercel!' });
});

// Export the Express app
module.exports = app;
```

```ts filename="src/index.ts" framework=express
// Use "type: module" in package.json to use ES modules
import express from 'express';
const app = express();

// Define your routes
app.get('/', (req, res) => {
  res.json({ message: 'Hello from Express on Vercel!' });
});

// Export the Express app
export default app;
```

Using a port listener

You may also run your application using the `app.listen` pattern that exposes the server on a port.

```js filename="src/index.js" framework=express
// Use "type: commonjs" in package.json to use CommonJS modules
const express = require('express');
const app = express();
const port = 3000;

// Define your routes
app.get('/', (req, res) => {
  res.json({ message: 'Hello from Express on Vercel!' });
});

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`);
});
```

```ts filename="src/index.ts" framework=express
// Use "type: module" in package.json to use ES modules
import express from 'express';
const app = express();
const port = 3000;

// Define your routes
app.get('/', (req, res) => {
```

```
  res.json({ message: 'Hello from Express on Vercel!' });
});

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`);
});
```

Local development

Use `vercel dev` to run your application locally

```bash filename="terminal"
vercel dev
```

> **💡 Note:** Minimum CLI version required: 47.0.5

Deploying the application

To deploy, [connect your Git repository](/new) or [use Vercel CLI](/docs/cli/deploy):

```bash filename="terminal"
vc deploy
```

> **💡 Note:** Minimum CLI version required: 47.0.5

Serving static assets

To serve static assets, place them in the `public/**` directory. They will be served as a part of our [CDN](/docs/cdn) using default [hea

`express.static()` will be ignored and will not serve static assets.

Vercel Functions

When you deploy an Express app to Vercel, your Express application becomes a single [Vercel Function](/docs/functions) and uses [Fluid co

Limitations

- `express.static()` will not serve static assets. You must use [the `public/**` directory](#serving-static-assets).

Additionally, all [Vercel Functions limitations](/docs/functions/limitations) apply to the Express application, including:

- **Application size**: The Express application becomes a single bundle, which must fit within the 250MB limit of Vercel Functions. Our b
- **Error handling**: Express.js will swallow errors that can put the main function into an undefined state unless properly handled. Expr

More resources

Learn more about deploying Express projects on Vercel with the following resources:

- [Express official documentation](https://expressjs.com/)
- [Vercel Functions documentation](/docs/functions)
- [Backend templates on Vercel](https://vercel.com/templates?type=backend)
- [Express middleware guide](https://expressjs.com/en/guide/using-middleware.html)

---
title: "FastAPI on Vercel"
description: "Deploy FastAPI applications to Vercel with zero configuration. Learn about the Python runtime, ASGI, static assets, and Ver
last_updated: "2026-01-16T02:19:30.766Z"
source: "https://vercel.com/docs/frameworks/backend/fastapi"

FastAPI on Vercel

FastAPI is a modern, high-performance, web framework for building APIs with Python based on standard Python type hints. You can deploy a

Get started with FastAPI on Vercel

You can quickly deploy a FastAPI application to Vercel by creating a FastAPI app or using an existing one:

Get started with Vercel CLI

Get started by initializing a new FastAPI project using [Vercel CLI init command](/docs/cli/init):

```bash filename="terminal"
vc init fastapi
```

This will clone the [FastAPI example repository](https://github.com/vercel/vercel/tree/main/examples/fastapi) in a directory called `fast

Exporting the FastAPI application

To run a FastAPI application on Vercel, define an `app` instance that initializes `FastAPI` at any of the following entrypoints:

- `app.py`
- `index.py`
- `server.py`
- `src/app.py`
- `src/index.py`
- `src/server.py`
- `app/app.py`
- `app/index.py`
- `app/server.py`

For example:

```py filename="src/index.py"
from fastapi import FastAPI
```

```
app = FastAPI()

@app.get("/")
def read_root():
    return {"Python": "on Vercel"}
```

You can also define an application script in `pyproject.toml` to point to your FastAPI app in a different module:

```toml filename="pyproject.toml"
[project.scripts]
app = "backend.server:app"
```

This script tells Vercel to look for a `FastAPI` instance named `app` in `./backend/server.py`.

Build command

The `build` property in `[tool.vercel.scripts]` defines the Build Command for FastAPI deployments. It runs after dependencies are install

```toml filename="pyproject.toml"
[tool.vercel.scripts]
build = "python build.py"
```

For example:

```py filename="build.py"
def main():
    print("Running build command...")
    with open("build.txt", "w") as f:
        f.write("BUILD_COMMAND")

if __name__ == "__main__":
    main()
```

> **💡 Note:** If you define a [Build Command](https://vercel.com/docs/project-configuration#buildcommand) in `vercel.json` or in the Pro

Local development

Use `vercel dev` to run your application locally.

```bash filename="terminal"
python -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
vercel dev
```

> **💡 Note:** Minimum CLI version required: 48.1.8

Deploying the application

To deploy, [connect your Git repository](/new) or [use Vercel CLI](/docs/cli/deploy):

```bash filename="terminal"
vc deploy
```

> **💡 Note:** Minimum CLI version required: 48.1.8

Serving static assets

To serve static assets, place them in the `public/**` directory. They will be served as a part of our [CDN](/docs/cdn) using default [hea

```py filename="app.py" highlight={6}
from fastapi import FastAPI
from fastapi.responses import RedirectResponse

app = FastAPI()

@app.get("/favicon.ico", include_in_schema=False)
async def favicon():
    # /vercel.svg is automatically served when included in the public/** directory.
    return RedirectResponse("/vercel.svg", status_code=307)
```

> **💡 Note:** `app.mount("/public", ...)` is not needed and should not be used.

Startup and shutdown

You can use [FastAPI lifespan events](https://fastapi.tiangolo.com/advanced/events/) to manage startup and shutdown logic, such as initia

```python filename="main.py"
from contextlib import asynccontextmanager
from fastapi import FastAPI

@asynccontextmanager
async def lifespan(app: FastAPI):
    # Startup logic
    print("Starting up...")
    await startup_tasks()
    yield
    # Shutdown logic
    await cleanup_tasks()

app = FastAPI(lifespan=lifespan)
```

> **💡 Note:** Cleanup logic during shutdown is limited to a maximum of **500ms** after receiving the [SIGTERM signal](https://vercel.com

Vercel Functions

When you deploy a FastAPI app to Vercel, the application becomes a single [Vercel Function](/docs/functions) and uses [Fluid compute](/do

Limitations

All [Vercel Functions limitations](/docs/functions/limitations) apply to FastAPI applications, including:

- **Application size**: The FastAPI application becomes a single bundle, which must fit within the 250MB limit of Vercel Functions. Our b

More resources

Learn more about deploying FastAPI projects on Vercel with the following resources:

- [FastAPI official documentation](https://fastapi.tiangolo.com/)
- [Vercel Functions documentation](/docs/functions)
- [Backend templates on Vercel](https://vercel.com/templates?type=backend)

---
title: "Fastify on Vercel"
description: "Deploy Fastify applications to Vercel with zero configuration."
last_updated: "2026-01-16T02:19:30.808Z"
source: "https://vercel.com/docs/frameworks/backend/fastify"

Fastify on Vercel

Fastify is a web framework highly focused on providing the best developer experience with the least overhead and a powerful plugin archit

Fastify applications on Vercel benefit from:

- [Fluid compute](/docs/fluid-compute): Pay for the CPU you use, automatic cold start reduction, optimized concurrency, background proces
- [Preview deployments](/docs/deployments/environments#preview-environment-pre-production): Test your changes in a copy of your productio
- [Instant Rollback](/docs/instant-rollback): Recover from breaking changes or bugs in milliseconds
- [Vercel Firewall](/docs/vercel-firewall): Protect your applications from a wide range of threats with a robust, multi-layered security
- [Secure Compute](/docs/secure-compute): Create private links between your Vercel-hosted backend and other clouds

Get started with Fastify on Vercel

You can quickly deploy a Fastify application to Vercel by creating a Fastify app or using an existing one:

Fastify entrypoint detection

To allow Vercel to deploy your Fastify application and process web requests, your server entrypoint file should be named one of the follo

- `src/app.{js,mjs,cjs,ts,cts,mts}`
- `src/index.{js,mjs,cjs,ts,cts,mts}`
- `src/server.{js,mjs,cjs,ts,cts,mts}`
- `app.{js,mjs,cjs,ts,cts,mts}`
- `index.{js,mjs,cjs,ts,cts,mts}`
- `server.{js,mjs,cjs,ts,cts,mts}`

For example, use the following code as an entrypoint:

```js filename="src/index.ts"
import Fastify from 'fastify'

const fastify = Fastify({ logger: true })

fastify.get('/', async (request, reply) => {
  return { hello: 'world' }
})

fastify.listen({ port: 3000 })
```

Local development

Use `vercel dev` to run your application locally

```bash filename="terminal"
vercel dev
```

> **💡 Note:** Minimum CLI version required: 48.6.0

Deploying the application

To deploy, [connect your Git repository](/new) or [use Vercel CLI](/docs/cli/deploy):

```bash filename="terminal"
vc deploy
```

> **💡 Note:** Minimum CLI version required: 48.6.0

Vercel Functions

When you deploy a Fastify app to Vercel, your Fastify application becomes a single [Vercel Function](/docs/functions) and uses [Fluid com

Limitations

All [Vercel Functions limitations](/docs/functions/limitations) apply to the Fastify application, including the size of the application b

More resources

Learn more about deploying Fastify projects on Vercel with the following resources:

- [Fastify official documentation](https://fastify.dev/docs/latest/)
- [Vercel Functions documentation](/docs/functions)
- [Backend templates on Vercel](https://vercel.com/templates?type=backend)

---
title: "Flask on Vercel"
description: "Deploy Flask applications to Vercel with zero configuration. Learn about the Python runtime, WSGI, static assets, and Verce
last_updated: "2026-01-16T02:19:30.842Z"
source: "https://vercel.com/docs/frameworks/backend/flask"

Flask on Vercel

Flask is a lightweight WSGI web application framework for Python. It's designed with simplicity and flexibility in mind, making it easy t

Get started with Flask on Vercel

You can quickly deploy a Flask application to Vercel by creating a Flask app or using an existing one:

Get started with Vercel CLI

Get started by initializing a new Flask project using [Vercel CLI init command](/docs/cli/init):

```bash filename="terminal"
vc init flask
```

This will clone the [Flask example repository](https://github.com/vercel/vercel/tree/main/examples/flask) in a directory called `flask`.

Exporting the Flask application

To run a Flask application on Vercel, define an `app` instance that initializes `Flask` at any of the following entrypoints:

- `app.py`
- `index.py`
- `server.py`
- `src/app.py`
- `src/index.py`
- `src/server.py`
- `app/app.py`
- `app/index.py`
- `app/server.py`

For example:

```py filename="src/index.py"
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return {"message": "Hello, World!"}
```

You can also define an application script in `pyproject.toml` to point to your Flask app in a different module:

```toml filename="pyproject.toml"
[project.scripts]
app = "backend.server:app"
```

This script tells Vercel to look for a `Flask` instance named `app` in `./backend/server.py`.

Build command

The `build` property in `[tool.vercel.scripts]` defines the Build Command for Flask deployments. It runs after dependencies are installed

```toml filename="pyproject.toml"
[tool.vercel.scripts]
build = "python build.py"
```

For example:

```py filename="build.py"
def main():
    print("Running build command...")
    with open("build.txt", "w") as f:
        f.write("BUILD_COMMAND")

if __name__ == "__main__":
    main()
```

> **💡 Note:** If you define a [Build Command](https://vercel.com/docs/project-configuration#buildcommand) in `vercel.json` or in the Pro

Local development

Use `vercel dev` to run your application locally.

```bash filename="terminal"
python -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
vercel dev
```

> **💡 Note:** Minimum CLI version required: 48.2.10

Deploying the application

To deploy, [connect your Git repository](/new) or [use Vercel CLI](/docs/cli/deploy):

```bash filename="terminal"
vc deploy
```

> **💡 Note:** Minimum CLI version required: 48.2.10

Serving static assets

To serve static assets, place them in the `public/**` directory. They will be served as a part of our [CDN](/docs/cdn) using default [hea

```py filename="app.py" highlight={5-7}
from flask import Flask, redirect

app = Flask(__name__)

@app.route("/favicon.ico")
def favicon():
    # /vercel.svg is automatically served when included in the public/** directory.
    return redirect("/vercel.svg", code=307)
```

> **💡 Note:** Flask's `app.static_folder` should not be used for static files on Vercel. Use
> the `public/**` directory instead.

Vercel Functions

When you deploy a Flask app to Vercel, the application becomes a single [Vercel Function](/docs/functions) and uses [Fluid compute](/docs

Limitations

All [Vercel Functions limitations](/docs/functions/limitations) apply to Flask applications, including:

- **Application size**: The Flask application becomes a single bundle, which must fit within the 250MB limit of Vercel Functions. Our bun

More resources

Learn more about deploying Flask projects on Vercel with the following resources:

- [Flask official documentation](https://flask.palletsprojects.com/)
- [Vercel Functions documentation](/docs/functions)
- [Backend templates on Vercel](https://vercel.com/templates?type=backend)

---
title: "Hono on Vercel"
description: "Deploy Hono applications to Vercel with zero configuration. Learn about observability, ISR, and custom build configurations
last_updated: "2026-01-16T02:19:30.820Z"
source: "https://vercel.com/docs/frameworks/backend/hono"

Hono on Vercel

Hono is a fast and lightweight web application framework built on Web Standards. You can deploy a Hono app to Vercel with zero configurat

Get started with Hono on Vercel

Start with Hono on Vercel by using the following Hono template to deploy to Vercel with zero configuration:

Vercel deployments can [integrate with your git provider](/docs/git) to [generate preview URLs](/docs/deployments/environments#preview-en

Get started with Vercel CLI

Get started by initializing a new Hono project using [Vercel CLI init command](/docs/cli/init):

```bash filename="terminal"
vc init hono
```

This will clone the [Hono example repository](https://github.com/vercel/vercel/tree/main/examples/hono) in a directory called `hono`.

Exporting the Hono application

To run a Hono application on Vercel, create a file that imports the `hono` package at any one of the following locations:

- `app.{js,cjs,mjs,ts,cts,mts}`
- `index.{js,cjs,mjs,ts,cts,mts}`
- `server.{js,cjs,mjs,ts,cts,mts}`
- `src/app.{js,cjs,mjs,ts,cts,mts}`
- `src/index.{js,cjs,mjs,ts,cts,mts}`
- `src/server.{js,mjs,cjs,ts,cts,mts}`

```ts filename="server.ts"
import { Hono } from 'hono';

const app = new Hono();

// ...

export default app;
```

Local development

To run your Hono application locally, use [Vercel CLI](https://vercel.com/docs/cli/dev):

```filename="terminal"
vc dev
```

This ensures that the application will use the default export to run the same as when deployed to Vercel. The application will be availab

Middleware

Hono has the concept of "Middleware" as a part of the framework. This is different from [Vercel Routing Middleware](/docs/routing-middlew

Hono Middleware

In Hono, [Middleware](https://hono.dev/docs/concepts/middleware) runs before a request handler in the framework's router. This is commonly

```ts filename="src/index.ts" framework="hono"
app.use(logger());
app.use('/posts/*', cors());
app.post('/posts/*', basicAuth());
```

More examples of Hono Middleware can be found in [the Hono documentation](https://hono.dev/docs/middleware/builtin/basic-auth).

Vercel Routing Middleware

In Vercel, [Routing Middleware](/docs/routing-middleware) executes code before a request is processed by the application. This gives you

Serving static assets

To serve static assets, place them in the `public/**` directory. They will be served as a part of our [CDN](/docs/cdn) using default [hea

[Hono's `serveStatic()`](https://hono.dev/docs/getting-started/nodejs#serve-static-files) will be ignored and will not serve static asset

Vercel Functions

When you deploy a Hono app to Vercel, your server routes automatically become [Vercel Functions](/docs/functions) and use [Fluid compute]

Streaming

Vercel Functions support streaming which can be used with [Hono's `stream()` function](https://hono.dev/docs/helpers/streaming).

```ts filename="src/index.ts" framework="hono"
app.get('/stream', (c) => {
  return stream(c, async (stream) => {
    // Write a process to be executed when aborted.
    stream.onAbort(() => {
      console.log('Aborted!');
    });
    // Write a Uint8Array.
    await stream.write(new Uint8Array([0x48, 0x65, 0x6c, 0x6c, 0x6f]));
    // Pipe a readable stream.
    await stream.pipe(anotherReadableStream);
  });
});
```

More resources

Learn more about deploying Hono projects on Vercel with the following resources:

- [Hono templates on Vercel](https://vercel.com/templates/hono)
- [Backend templates on Vercel](https://vercel.com/templates?type=backend)

---
title: "NestJS on Vercel"
description: "Deploy NestJS applications to Vercel with zero configuration."
last_updated: "2026-01-16T02:19:30.861Z"
source: "https://vercel.com/docs/frameworks/backend/nestjs"

NestJS on Vercel

NestJS is a progressive Node.js framework for building efficient, reliable and scalable server-side applications. You can deploy a NestJS

NestJS applications on Vercel benefit from:

- [Fluid compute](/docs/fluid-compute): Pay for the CPU you use, automatic cold start reduction, optimized concurrency, background proces
- [Preview deployments](/docs/deployments/environments#preview-environment-pre-production): Test your changes in a copy of your productio
- [Instant Rollback](/docs/instant-rollback): Recover from breaking changes or bugs in milliseconds
- [Vercel Firewall](/docs/vercel-firewall): Protect your applications from a wide range of threats with a robust, multi-layered security
- [Secure Compute](/docs/secure-compute): Create private links between your Vercel-hosted backend and other clouds

Get started with NestJS on Vercel

You can quickly deploy a NestJS application to Vercel by creating a NestJS app or using an existing one:

NestJS entrypoint detection

To allow Vercel to deploy your NestJS application and process web requests, your server entrypoint file should be named one of the follow

- `src/main.{js,mjs,cjs,ts,cts,mts}`
- `src/app.{js,mjs,cjs,ts,cts,mts}`
- `src/index.{js,mjs,cjs,ts,cts,mts}`
- `src/server.{js,mjs,cjs,ts,cts,mts}`
- `app.{js,mjs,cjs,ts,cts,mts}`
- `index.{js,mjs,cjs,ts,cts,mts}`
- `server.{js,mjs,cjs,ts,cts,mts}`

For example, use the following code as an entrypoint:

```js filename="src/app.ts"
import { NestFactory } from '@nestjs/core';
import { AppModule } from './app.module';

async function bootstrap() {
  const app = await NestFactory.create(AppModule);
  await app.listen(process.env.PORT ?? 3000);
}
bootstrap();
```

Local development

Use `vercel dev` to run your application locally

```bash filename="terminal"
vercel dev
```

> **💡 Note:** Minimum CLI version required: 48.4.0

Deploying the application

To deploy, [connect your Git repository](/new) or [use Vercel CLI](/docs/cli/deploy):

```bash filename="terminal"
vc deploy
```

> **💡 Note:** Minimum CLI version required: 48.4.0

Vercel Functions

When you deploy a NestJS app to Vercel, your NestJS application becomes a single [Vercel Function](/docs/functions) and uses [Fluid compu

Limitations

All [Vercel Functions limitations](/docs/functions/limitations) apply to the NestJS application, including the size of the application be

More resources

Learn more about deploying NestJS projects on Vercel with the following resources:

- [NestJS official documentation](https://docs.nestjs.com/)
- [Vercel Functions documentation](/docs/functions)
- [Backend templates on Vercel](https://vercel.com/templates?type=backend)

---
title: "Nitro on Vercel"
description: "Deploy Nitro applications to Vercel with zero configuration. Learn about observability, ISR, and custom build configuration
last_updated: "2026-01-16T02:19:30.895Z"
source: "https://vercel.com/docs/frameworks/backend/nitro"

Nitro on Vercel

Nitro is a full-stack framework with TypeScript-first support. It includes filesystem routing, code-splitting for fast startup, built-in

You can deploy a Nitro app to Vercel with zero configuration.

Get started with Nitro on Vercel

To get started with Nitro on Vercel, use the following Nitro template to deploy to Vercel with zero configuration:

Vercel deployments can [integrate with your git provider](/docs/git) to [generate preview URLs](/docs/deployments/environments#preview-en

Get started with Vercel CLI

Get started by initializing a new Nitro project using [Vercel CLI init command](/docs/cli/init):

```bash filename="terminal"
vc init nitro
```

This will clone the [Nitro example repository](https://github.com/vercel/vercel/tree/main/examples/nitro) in a directory called `nitro`.

Using Vercel's features with Nitro

When you deploy a Nitro app to Vercel, you can use Vercel specific features such as [Incremental Static Regeneration (ISR)](#incremental-

Incremental Static Regeneration (ISR)

[ISR](/docs/incremental-static-regeneration) allows you to create or update content without redeploying your site. ISR has three main ben

On-demand revalidation

With [on-demand revalidation](/docs/incremental-static-regeneration/quickstart#on-demand-revalidation), you can purge the cache for an IS

To revalidate a path to a prerendered function:

- ### Create an Environment Variable
  Create an [Environment Variable](/docs/environment-variables) to store a revalidation secret by:
  - Using the command:
  ```bash filename="terminal"
  openssl rand -base64 32
  ```
  - Or [generating a secret](https://generate-secret.vercel.app/32) to create a random value.

- ### Update your configuration
  Update your configuration to use the revalidation secret as follows:
  ```ts filename="nitro.config.ts" framework=nitro
  export default defineNitroConfig({
    vercel: {
      config: {
        bypassToken: process.env.VERCEL_BYPASS_TOKEN,
      },
    },
  });
  ```
  ```js filename="nitro.config.js" framework=nitro
  export default defineNitroConfig({
    vercel: {
      config: {
        bypassToken: process.env.VERCEL_BYPASS_TOKEN,
      },
    },
  });
  ```
  ```ts filename="nuxt.config.ts" framework=nuxt
  export default defineNitroConfig({
    vercel: {
      config: {
        bypassToken: process.env.VERCEL_BYPASS_TOKEN,
      },
    },
  });
  ```
  ```js filename="nuxt.config.ts" framework=nuxt
  export default defineNitroConfig({
    vercel: {
      config: {
        bypassToken: process.env.VERCEL_BYPASS_TOKEN,
      },
    },
  });
  ```

- ### Trigger revalidation
  You can revalidate a path to a prerendered function by making a `GET` or `HEAD` request to that path with a header of `x-prerender-reva

  When the prerendered function endpoint is accessed with this header set, the cache will be revalidated. The next request to that functi

Fine-grained ISR configuration

To have more control over ISR caching, you can pass an options object to the `isr` route rule as shown below:

```ts filename="nitro.config.ts" framework=all
export default defineNitroConfig({
  routeRules: {
    '/products/**': {
      isr: {
        allowQuery: ['q'],
        passQuery: true,
      },
    },
  },
});
```

```js filename="nitro.config.js" framework=all
export default defineNitroConfig({
  routeRules: {
    '/products/**': {
      isr: {
        allowQuery: ['q'],
        passQuery: true,
      },
    },
  },
});
```

> **💡 Note:** By default, query parameters are ignored by cache unless you specify them in
> the `allowQuery` array.

The following options are available:

Option	Type	Description
`expiration`	`number | false`	The expiration time, in seconds, before the cached asset is re-generated by invoking the serve
`group`	`number`	Group number of the asset. Use this to revalidate multiple assets at the same time.
`allowQuery`	`string[] | undefined`	List of query string parameter names that will be cached independently. If you specify an empt
`passQuery`	`boolean`	When `true`, the query string will be present on the request argument passed to the invoked fu

Observability

With [Vercel Observability](/docs/observability), you can view detailed performance insights broken down by route and monitor function ex

Nitro (>=2.12) generates routing hints for [functions observability insights](/docs/observability/insights#vercel-functions), providing a

To enable this feature, ensure you are using a compatibility date of `2025-07-15` or later.

```ts filename="nitro.config.ts" framework=nitro
export default defineNitroConfig({
  compatibilityDate: '2025-07-15', // or "latest"
});
```

```js filename="nitro.config.js" framework=nitro
export default defineNitroConfig({
  compatibilityDate: '2025-07-15', // or "latest"
});
```

```ts filename="nuxt.config.ts" framework=nuxt
export default defineNitroConfig({
  compatibilityDate: '2025-07-15', // or "latest"
});
```

```js filename="nuxt.config.js" framework=nuxt
export default defineNitroConfig({
  compatibilityDate: '2025-07-15', // or "latest"
});
```

> **💡 Note:** Framework integrations can use the `ssrRoutes` configuration to declare SSR
> routes. For more information, see
> [#3475](https://github.com/unjs/nitro/pull/3475).

Vercel Functions

When you deploy a Nitro app to Vercel, your server routes automatically become [Vercel Functions](/docs/functions) and use [Fluid compute

More resources

Learn more about deploying Nitro projects on Vercel with the following resources:

- [Getting started with Nitro guide](https://nitro.build/guide)
- [Deploy Nitro to Vercel guide](https://nitro.build/deploy/providers/vercel)
- [Backend templates on Vercel](https://vercel.com/templates?type=backend)

---
title: "Backends on Vercel"
description: "Vercel supports a wide range of the most popular backend frameworks, optimizing how your application builds and runs no mat
last_updated: "2026-01-16T02:19:30.903Z"
source: "https://vercel.com/docs/frameworks/backend"

Backends on Vercel

Backends deployed to Vercel receive the benefits of Vercel's infrastructure, including:

- [Fluid compute](/docs/fluid-compute): Zero-configuration, optimized concurrency, dynamic scaling, background processing, automatic cold
- [Active CPU pricing](/docs/functions/usage-and-pricing): Only pay for the CPU you use, not waiting for I/O (e.g. calling AI models, dat
- [Instant Rollback](/docs/instant-rollback): Quickly revert to a previous production deployment
- [Vercel Firewall](/docs/vercel-firewall): A robust, multi-layered security system designed to protect your applications
- [Preview deployments with Deployment Protection](/docs/deployments/environments#preview-environment-pre-production): Secure your previe
- [Rolling releases](/docs/rolling-releases): Gradually roll out backends to detect errors early

Zero-configuration backends

Deploy the following backends to Vercel with zero-configuration.

- **Elysia**: Ergonomic framework for humans
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/elysia)
- **Express**: Fast, unopinionated, minimalist web framework for Node.js
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/express) | [View Demo](https
- **FastAPI**: FastAPI framework, high performance, easy to learn, fast to code, ready for production
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/fastapi) | [View Demo](https
- **Fastify**: Fast and low overhead web framework, for Node.js
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/fastify)
- **Flask**: The Python micro web framework
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/flask)
- **H3**: Universal, Tiny, and Fast Servers
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/h3)
- **Hono**: Web framework built on Web Standards
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/hono) | [View Demo](https://
- **NestJS**: Framework for building efficient, scalable Node.js server-side applications
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/nestjs)
- **Nitro**: Nitro is a next generation server toolkit.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/nitro) | [View Demo](https:/
- **xmcp**: The MCP framework for building AI-powered tools
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/xmcp) | [View Demo](https://

Adapting to Serverless and Fluid compute

If you are transitioning from a fully managed server or containerized environment to Vercel’s serverless architecture, you may need to re

The following are generally applicable to serverless, and therefore Vercel Functions (running with or without Fluid compute).

Websockets

Serverless functions have maximum execution limits and should respond as quickly as possible. They should not subscribe to data events. I

Database Connections

To manage database connections efficiently, [use the `attachDatabasePool` function from `@vercel/functions`](/docs/functions/functions-ap

---
title: "xmcp on Vercel"
description: "Build MCP-compatible backends with xmcp and deploy to Vercel. Learn the project structure, tool format, middleware, and how
last_updated: "2026-01-16T02:19:30.917Z"
source: "https://vercel.com/docs/frameworks/backend/xmcp"
---

xmcp on Vercel

`xmcp` is a TypeScript-first framework for building MCP-compatible backends. It provides an opinionated project structure, automatic tool

Get started with xmcp on Vercel

Start with xmcp on Vercel by creating a new xmcp project:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i
    ```
  </Code>
</CodeBlock>

This scaffolds a project with a `src/tools/` directory for tools, optional `src/middleware.ts`, and an `xmcp.config.ts` file.

To deploy, [connect your Git repository](/new) or [use Vercel CLI](/docs/cli):

```bash filename="terminal"
vc deploy
```

Get started with Vercel CLI

Get started by initializing a new Xmcp project using [Vercel CLI init command](/docs/cli/init):

```bash filename="terminal"
vc init xmcp
```

This will clone the [Xmcp example repository](https://github.com/vercel/vercel/tree/main/examples/xmcp) in a directory called `xmcp`.

Local development

To run your xmcp application locally, you can use [Vercel CLI](https://vercel.com/docs/cli/dev):

```bash filename="terminal"
vc dev
```

Alternatively, use your project's dev script:

```bash filename="terminal"
npm run dev
yarn dev
pnpm run dev
```

Middleware

xmcp Middleware

In xmcp, an optional `middleware.ts` lets you run code before and after tool execution. This is commonly used for logging, auth, or reque

```ts filename="src/middleware.ts" framework="xmcp"
import { type Middleware } from 'xmcp';

const middleware: Middleware = async (req, res, next) => {
  // Custom processing
  next();
};

export default middleware;
```

Vercel Routing Middleware

In Vercel, [Routing Middleware](/docs/routing-middleware) executes before a request is processed by your application. Use it for rewrites

Vercel Functions

When you deploy an xmcp app to Vercel, your server endpoints automatically run as [Vercel Functions](/docs/functions) and use [Fluid comp

More resources

- [xmcp documentation](https://xmcp.dev/docs)
- [Backend templates on Vercel](https://vercel.com/templates?type=backend)

---
title: "Astro on Vercel"
description: "Learn how to use Vercel"

last_updated: "2026-01-16T02:19:31.072Z"
source: "https://vercel.com/docs/frameworks/frontend/astro"

Astro on Vercel

Astro is an all-in-one web framework that enables you to build performant static websites. People choose Astro when they want to build co

You can deploy a static Astro app to Vercel with zero configuration.

Get Started with Astro on Vercel

Using Vercel's features with Astro

To deploy a server-rendered Astro app, or a static Astro site with Vercel features like Web Analytics and Image Optimization, you must:

1. Add [Astro's Vercel adapter](https://docs.astro.build/en/guides/integrations-guide/vercel) to your project. There are two ways to do s

   - Using `astro add`, which configures the adapter for you with default settings. Using `astro add` will generate a preconfigured with

     <CodeBlock>
       <Code tab="pnpm">
         ```bash
         pnpm i @astrojs/vercel
         ```
       </Code>
       <Code tab="yarn">
         ```bash
         yarn i @astrojs/vercel
         ```
       </Code>
       <Code tab="npm">
         ```bash
         npm i @astrojs/vercel
         ```
       </Code>
       <Code tab="bun">
         ```bash
         bun i @astrojs/vercel
         ```
       </Code>
     </CodeBlock>

   - Or, manually installing the [`@astrojs/vercel`](https://www.npmjs.com/package/@astrojs/vercel) package. You should manually install

     <CodeBlock>
       <Code tab="pnpm">
         ```bash
         pnpm i @astrojs/vercel
         ```
       </Code>
       <Code tab="yarn">
         ```bash
         yarn i @astrojs/vercel
         ```
       </Code>
       <Code tab="npm">
         ```bash
         npm i @astrojs/vercel
         ```
       </Code>
       <Code tab="bun">
         ```bash
         bun i @astrojs/vercel
         ```
       </Code>
     </CodeBlock>

2. Configure your project. In your file, import either the `serverless` or `static` plugin, and set the output to `server` or `static` r

   #### \['Serverless SSR'

   ```js filename="astro.config.mjs" framework=all
   import { defineConfig } from 'astro/config';
   // Import /serverless for a Serverless SSR site
   import vercelServerless from '@astrojs/vercel/serverless';

   export default defineConfig({
     output: 'server',
     adapter: vercelServerless(),
   });
   ```

   ```ts filename="astro.config.ts" framework=all
   import { defineConfig } from 'astro/config';
   // Import /serverless for a Serverless SSR site
   import vercelServerless from '@astrojs/vercel/serverless';

   export default defineConfig({
     output: 'server',
     adapter: vercelServerless(),
   });
   ```

   #### 'Static']

   ```js filename="astro.config.mjs" framework=all
   import { defineConfig } from 'astro/config';
   // Import /static for a static site
   import vercelStatic from '@astrojs/vercel/static';
   ```

```
export default defineConfig({
  // Must be 'static' or 'hybrid'
  output: 'static',
  adapter: vercelStatic(),
});
```

```ts filename="astro.config.ts" framework=all
import { defineConfig } from 'astro/config';
// Import /static for a static site
import vercelStatic from '@astrojs/vercel/static';

export default defineConfig({
  // Must be 'static' or 'hybrid'
  output: 'static',
  adapter: vercelStatic(),
});
```

3. Enable Vercel's features using Astro's [configuration options](#configuration-options). The following example enables Web Analytics a

```js filename="astro.config.mjs" framework=all
import { defineConfig } from 'astro/config';
// Also can be @astrojs/vercel/static
import vercel from '@astrojs/vercel/serverless';

export default defineConfig({
  // Also can be 'static' or 'hybrid'
  output: 'server',
  adapter: vercel({
    webAnalytics: {
      enabled: true,
    },
    maxDuration: 8,
  }),
});
```

```ts filename="astro.config.ts" framework=all
import { defineConfig } from 'astro/config';
// Also can be @astrojs/vercel/static
import vercel from '@astrojs/vercel/serverless';

export default defineConfig({
  // Also can be 'static' or 'hybrid'
  output: 'server',
  adapter: vercel({
    webAnalytics: {
      enabled: true,
    },
    maxDuration: 8,
  }),
});
```

Configuration options

The following configuration options enable Vercel's features for Astro deployments.

Option	type
[`maxDuration`](/docs/functions/runtimes#max-duration)	`numbe
[`webAnalytics`](/docs/analytics)	`{enab
[`imageService`](https://docs.astro.build/en/guides/integrations-guide/vercel/#imageservice)	`boole
[`devImageService`](https://docs.astro.build/en/guides/integrations-guide/vercel/#devimageservice)	`strin
[`imagesConfig`](/docs/build-output-api/v3/configuration#images)	`Verce
[`functionPerRoute`](https://docs.astro.build/en/guides/integrations-guide/vercel/#function-bundling-configuration)	`boole
[`edgeMiddleware`](https://docs.astro.build/en/guides/integrations-guide/vercel/#vercel-edge-middleware-with-astro-middleware)	`boole
[`includeFiles`](https://docs.astro.build/en/guides/integrations-guide/vercel/#includefiles)	`strin
[`excludeFiles`](https://docs.astro.build/en/guides/integrations-guide/vercel/#excludefiles)	`strin

For more details on the configuration options, see [Astro's docs](https://docs.astro.build/en/guides/integrations-guide/vercel/#configura

Server-Side Rendering

Using SSR, or [on-demand rendering](https://docs.astro.build/en/guides/server-side-rendering/) as Astro calls it, enables you to deploy y

You can enable SSR by [adding the Vercel adapter to your project](#using-vercel's-features-with-astro).

If your Astro project is statically rendered, you can opt individual routes. To do so:

1. Set your `output` option to `hybrid` in your `<PreferredExtension filename="astro.config" mjs />`:

```js filename="astro.config.mjs" framework=all
import { defineConfig } from 'astro/config';
import vercel from '@astrojs/vercel/serverless';

export default defineConfig({
  output: 'hybrid',
  adapter: vercel({
    edgeMiddleware: true,
  }),
});
```

```ts filename="astro.config.ts" framework=all
import { defineConfig } from 'astro/config';
import vercel from '@astrojs/vercel/serverless';
```

```
   export default defineConfig({
     output: 'hybrid',
     adapter: vercel({
       edgeMiddleware: true,
     }),
   });
   ```

2. Add `export const prerender = false;` to your components:

   ```tsx filename="src/pages/mypage.astro"
   ---
   export const prerender = false;
   // ...
   ---
   <html>
     <!-- Server-rendered page here... -->
   </html>
   ```

SSR with Astro on Vercel:

- Scales to zero when not in use
- Scales automatically with traffic increases
- Has zero-configuration support for [`Cache-Control` headers](/docs/cdn-cache), including `stale-while-revalidate`

[Learn more about Astro SSR](https://docs.astro.build/en/guides/server-side-rendering/)

Static rendering

Statically rendered, or pre-rendered, Astro apps can be deployed to Vercel with zero configuration. To enable Vercel features like Image (

You can opt individual routes into static rendering with `export const prerender = true` as shown below:

```tsx filename="src/pages/mypage.astro"
---
export const prerender = true;
// ...
---
<html>
  <!-- Static, pre-rendered page here... -->
</html>
```

Statically rendered Astro sites on Vercel:

- Require zero configuration to deploy
- Can use Vercel features with

[Learn more about Astro Static Rendering](https://docs.astro.build/en/core-concepts/rendering-modes/#pre-rendered)

Incremental Static Regeneration

[Incremental Static Regeneration (ISR)](/docs/incremental-static-regeneration) allows you to create or update content without redeploying

To enable ISR in Astro, you need to use the [Vercel adapter](https://docs.astro.build/en/guides/integrations-guide/vercel/) and set `isr`

```js filename="astro.config.mjs" framework=all
import { defineConfig } from 'astro/config';
import vercel from '@astrojs/vercel/serverless';

export default defineConfig({
  // ...
  output: 'server',
  adapter: vercel({
    isr: true,
  }),
});
```

> **💡 Note:** ISR function requests do not include search params, similar to requests in
> static mode.

Using ISR with Astro on Vercel offers:

- Better performance with our global [CDN](/docs/cdn)
- Zero-downtime rollouts to previously statically generated pages
- Global content updates in 300ms
- Generated pages are both cached and persisted to durable storage

[Learn more about ISR with Astro.](https://docs.astro.build/en/guides/integrations-guide/vercel/#isr)

Vercel Functions

[Vercel Functions](/docs/functions) use resources that scale up and down based on traffic demands. This makes them reliable during peak h

When you [enable SSR with Astro's Vercel adapter](#using-vercel's-features-with-astro), **all** of your routes will be server-rendered as

When defining an Endpoint, you must name each function after the HTTP method it represents. The following example defines basic HTTP meth

```ts filename="src/pages/methods.json.ts" framework=all
import { APIRoute } from 'astro/dist/@types/astro';

export const GET: APIRoute = ({ params, request }) => {
  return new Response(
    JSON.stringify({
      message: 'This was a GET!',
    }),
  );
};
```

```
export const POST: APIRoute = ({ request }) => {
  return new Response(
    JSON.stringify({
      message: 'This was a POST!',
    }),
  );
};

export const DELETE: APIRoute = ({ request }) => {
  return new Response(
    JSON.stringify({
      message: 'This was a DELETE!',
    }),
  );
};

// ALL matches any method that you haven't implemented.
export const ALL: APIRoute = ({ request }) => {
  return new Response(
    JSON.stringify({
      message: `This was a ${request.method}!`,
    }),
  );
};
```

```js filename="src/pages/methods.json.js" framework=all
export const GET = ({ params, request }) => {
  return new Response(
    JSON.stringify({
      message: 'This was a GET!',
    }),
  );
};

export const POST = ({ request }) => {
  return new Response(
    JSON.stringify({
      message: 'This was a POST!',
    }),
  );
};

export const DELETE = ({ request }) => {
  return new Response(
    JSON.stringify({
      message: 'This was a DELETE!',
    }),
  );
};

export const ALL = ({ request }) => {
  return new Response(
    JSON.stringify({
      message: `This was a ${request.method}!`,
    }),
  );
};
```

> **💡 Note:** Astro removes the final file during the build process, so the name of the file
> should include the extension of the data you want serve (for example
> will become
>).

Vercel Functions with Astro on Vercel:

- Scale to zero when not in use
- Scale automatically as traffic increases

[Learn more about Vercel Functions](/docs/functions)

Image Optimization

[Image Optimization](/docs/image-optimization) helps you achieve faster page loads by reducing the size of images and using modern image

Image Optimization with Astro on Vercel is supported out of the box with Astro's `Image` component. See [the Image Optimization quickstar

Image Optimization with Astro on Vercel:

- Requires zero-configuration for Image Optimization when using Astro's `Image` component
- Helps your team ensure great performance by default
- Keeps your builds fast by optimizing images on-demand

[Learn more about Image Optimization](/docs/image-optimization)

Middleware

[Middleware](/docs/routing-middleware) is a function that execute before a request is processed on a site, enabling you to modify the res

[Astro middleware](https://docs.astro.build/en/guides/middleware/#basic-usage) allows you to set and share information across your endpoi

```ts filename="src/middleware.ts" framework=all
// This helper automatically types middleware params
import { defineMiddleware } from 'astro:middleware';

export const onRequest = defineMiddleware(({ locals }, next) => {
  // intercept data from a request
  // optionally, modify the properties in `locals`
```

````
  locals.title = 'New title';

  // return a Response or the result of calling `next()`
  return next();
});
```

```js filename="src/middleware.js" framework=all
export function onRequest({ locals }, next) {
  // intercept data from a request
  // optionally, modify the properties in `locals`
  locals.title = 'New title';

  // return a Response or the result of calling `next()`
  return next();
}
```

> **💡 Note:** , which has to be placed at the root directory of your project, outside
> .

To add custom properties to `locals` in `middleware.ts`, you must declare a global namespace in your `env.d.ts` file:

```ts filename="src/env.d.ts"
declare namespace App {
  interface Locals {
    title?: string;
  }
}
```

You can then access the data you added to `locals` in any `.astro` file, like so:

```jsx filename="src/pages/middleware-title.astro"
---
const { title } = Astro.locals;
---
<h1>{title}</h1>
<p>The name of this page is from middleware.</p>
```

### Deploying middleware at the Edge

You can deploy Astro's middleware at the Edge, giving you access to data in the `RequestContext` and `Request`, and enabling you to use ['

To use Astro's middleware at the Edge, set `edgeMiddleware: true` in your  file:

```js filename="astro.config.mjs" framework=all
import { defineConfig } from 'astro/config';
import vercel from '@astrojs/vercel/serverless';

export default defineConfig({
  output: 'server',
  adapter: vercel({
    edgeMiddleware: true,
  }),
});
```

```ts filename="astro.config.ts" framework=all
import { defineConfig } from 'astro/config';
import vercel from '@astrojs/vercel/serverless';

export default defineConfig({
  output: 'server',
  adapter: vercel({
    edgeMiddleware: true,
  }),
});
```

> **💡 Note:** If you're using [Vercel's Routing
> Middleware](#using-vercel's-edge-middleware), you do not need to set
> `edgeMiddleware: true` in your
>  file.

See Astro's docs on [the limitations and constraints](https://docs.astro.build/en/guides/integrations-guide/vercel/#limitations-and-const

#### Using `Astro.locals` in Routing Middleware

The `Astro.locals` object exposes data to your `.astro` components, allowing you to dynamically modify your content with middleware. To m

1. Add a new middleware file next to your  and name it . This file name is required to make changes to [`Astro.locals`](https://docs.astr
2. Return an object with the properties you want to add to `Astro.locals`. :

   For TypeScript, you must install [the `@vercel/functions` package](/docs/routing-middleware/api#routing-middleware-helper-methods):

   <CodeBlock>
     <Code tab="pnpm">
       ```bash
       pnpm i @vercel/functions
       ```
     </Code>
     <Code tab="yarn">
       ```bash
       yarn i @vercel/functions
       ```
     </Code>
     <Code tab="npm">
       ```bash
````

```
      npm i @vercel/functions
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i @vercel/functions
      ```
    </Code>
  </CodeBlock>

  Then, type your middleware function like so:

  ```ts filename="src/vercel-edge-middleware.ts" framework=all
  import type { RequestContext } from '@vercel/functions';

  // Note the parameters are different from standard Astro middleware
  export default function ({
    request,
    context,
  }: {
    request: Request;
    context: RequestContext;
  }) {
    // Return an Astro.locals object with a title property
    return {
      title: "Spider-man's blog",
    };
  }
  ```

  ```js filename="src/vercel-edge-middleware.js" framework=all
  // Note the parameters are different from standard Astro middleware
  export default function ({ request, context }) {
    // Return an Astro.locals object with a title property
    return {
      title: "Spider-man's blog",
    };
  }
  ```

Using Vercel's Routing Middleware

Astro's middleware, which should be in , is distinct from Vercel Routing Middleware, which should be a file at the root of your project.

Vercel recommends using framework-native solutions. You should use Astro's middleware over Vercel's Routing Middleware wherever possible.

If you still want to use Vercel's Routing Middleware, see [the Quickstart](/docs/routing-middleware/getting-started) to learn how.

Rewrites

Rewrites only work for static files with Astro. You must use [Vercel's Routing Middleware](/docs/routing-middleware/api#match-paths-b

Redirects

In general, Vercel recommends using framework-native solutions, and Astro has [built-in support for redirects](https://docs.astro.build/e

Redirects in your Astro config

You can do redirects on Astro with the `redirects` config option as shown below:

```ts filename="astro.config.ts" framework=all
import { defineConfig } from 'astro/config';

export default defineConfig({
  redirects: {
    '/old-page': '/new-page',
  },
});
```

```js filename="astro.config.mjs" framework=all
import { defineConfig } from 'astro/config';

export default defineConfig({
  redirects: {
    '/old-page': '/new-page',
  },
});
```

Redirects in Server Endpoints

You can also return a redirect from a Server Endpoint using the [`redirect`](https://docs.astro.build/en/core-concepts/endpoints/#redirec

```ts filename="src/pages/links/[id].ts" framework=all
export async function GET({ params, redirect }): APIRoute {
  return redirect('/redirect-path', 307);
}
```

```js filename="src/pages/links/[id].js" framework=all
import { getLinkUrl } from '../db';

export async function GET({ redirect }) {
  return redirect('/redirect-path', 307);
}
```

Redirects in components

You can redirect from within Astro components with [`Astro.redirect()`](https://docs.astro.build/en/reference/api-reference/#astroredirec

```tsx filename="src/pages/account.astro"
---
import { isLoggedIn } from '../utils';

const cookie = Astro.request.headers.get('cookie');

// If the user is not logged in, redirect them to the login page
if (!isLoggedIn(cookie)) {
  return Astro.redirect('/login');
}
---

<h1>You can only see this page while logged in</h1>
```

Astro Middleware on Vercel:

- Executes before a request is processed on a site, allowing you to modify responses to user requests
- Runs on *all* requests, but can be scoped to specific paths [through a `matcher` config](/docs/routing-middleware/api#match-paths-based
- Uses Vercel's lightweight Edge Runtime to keep costs low and responses fast

[Learn more about Routing Middleware](/docs/routing-middleware)

Caching

Vercel automatically caches static files at the edge after the first request, and stores them for up to 31 days on Vercel's CDN. Dynamic

The following Astro component will show a new time every 10 seconds. It does by setting a 10 second max age on the contents of the page,

[Learn more about Cache Control options](/docs/headers#cache-control-header).

```jsx filename="src/pages/ssr-with-swr-caching.astro"
---
Astro.response.headers.set('Cache-Control', 's-maxage=10, stale-while-revalidate');
const time = new Date().toLocaleTimeString();
---

<h1>{time}</h1>
```

CDN Cache-Control headers

You can also control how the cache behaves on any CDNs you may be using outside of Vercel's CDN with CDN Cache-Control Headers.

The following example tells downstream CDNs to cache the content for 60 seconds, and Vercel's CDN to cache it for 3600 seconds:

```jsx filename="src/pages/ssr-with-swr-caching.astro"
---
Astro.response.headers.set('Vercel-CDN-Cache-Control', 'max-age=3600',);
Astro.response.headers.set('CDN-Cache-Control', 'max-age=60',);
const time = new Date().toLocaleTimeString();
---

<h1>{time}</h1>
```

[Learn more about CDN Cache-Control headers](/docs/headers/cache-control-headers#cdn-cache-control-header).

Caching on Vercel:

- Automatically optimizes and caches assets for the best performance
- Requires no additional services to procure or set up
- Supports zero-downtime rollouts

Speed Insights

[Vercel Speed Insights](/docs/speed-insights) provides you with a detailed view of your website's performance metrics, facilitating infor

To enable Speed Insights with Astro, see [the Speed Insights quickstart](/docs/speed-insights/quickstart).

To summarize, using Speed Insights with Astro on Vercel:

- Enables you to track traffic performance metrics, such as [First Contentful Paint](/docs/speed-insights/metrics#first-contentful-paint-
- Enables you to view performance metrics by page name and URL for more granular analysis
- Shows you [a score for your app's performance](/docs/speed-insights/metrics#how-the-scores-are-determined) on each recorded metric, whi

[Learn more about Speed Insights](/docs/speed-insights)

More benefits

See [our Frameworks documentation page](/docs/frameworks) to learn about the benefits available to **all** frameworks when you deploy on

More resources

Learn more about deploying Astro projects on Vercel with the following resources:

- [Vercel CLI](/docs/cli)
- [Vercel Function docs](/docs/functions)
- [Astro docs](https://docs.astro.build/en/guides/integrations-guide/vercel)

---
title: "Create React App on Vercel"
description: "Learn how to use Vercel"
last_updated: "2026-01-16T02:19:31.005Z"
source: "https://vercel.com/docs/frameworks/frontend/create-react-app"

Create React App on Vercel

Create React App (CRA) is a development environment for building single-page applications with the React framework. It sets up and config

Get Started with CRA on Vercel

Static file caching

On Vercel, static files are [replicated and deployed to every region in our global CDN after the first request](/docs/cdn-cache#static-fi

Static files are cached for up to 31 days. If a file is unchanged, it can persist across deployments, as their hash caches static files.

To summarize, using Static Files with CRA on Vercel:

- Automatically optimizes and caches assets for the best performance
- Makes files easily accessible through the `public` folder
- Supports zero-downtime rollouts
- Requires no additional services needed to procure or set up

[Learn more about static files caching](/docs/cdn-cache#static-files-caching)

Preview Deployments

When you deploy your CRA app to Vercel and connect your git repo, every pull request will generate a [Preview Deployment](/docs/deploymen

Preview Deployments allow you to preview changes to your app in a live deployment. They are available by default for all projects, and ar

Comments

You can use the comments feature to receive feedback on your Preview Deployments from Vercel Team members and [people you share the Previ

Comments allow you to start discussion threads, share screenshots, send notifications, and more.

To summarize, Preview Deployments with CRA on Vercel:

- Enable you to share previews of pull request changes in a live environment
- Come with a comment feature for improved collaboration and feedback
- Experience changes to your product without merging them to your deployment branch

[Learn more about Preview Deployments](/docs/deployments/environments#preview-environment-pre-production)

Web Analytics

Vercel's Web Analytics features enable you to visualize and monitor your application's performance over time. The Analytics tab in your p

To use Web Analytics, navigate to the Analytics tab of your project dashboard on Vercel and select **Enable** in the modal that appears.

To track visitors and page views, we recommend first installing our `@vercel/analytics` package.

You can then import the `inject` function from the package, which will add the tracking script to your app. This should only be called on

Add the following code to your main app file:

```ts filename="main.ts" framework=all
import { inject } from '@vercel/analytics';

inject();
```

```js filename="main.js" framework=all
import { inject } from '@vercel/analytics';

inject();
```

Then, [ensure you've enabled Web Analytics in your dashboard on Vercel](/docs/analytics/quickstart). You should start seeing usage data i

To summarize, using Web Analytics with CRA on Vercel:

- Enables you to track traffic and see your top-performing pages
- Offers you detailed breakdowns of visitor demographics, including their OS, browser, geolocation and more

[Learn more about Web Analytics](/docs/analytics)

Speed Insights

You can see data about your CRA project's [Core Web Vitals](/docs/speed-insights/metrics#core-web-vitals-explained) performance in your d

On Vercel, you can track your app's Core Web Vitals in your project's dashboard by enabling Speed Insights.

To summarize, using Speed Insights with CRA on Vercel:

- Enables you to track traffic performance metrics, such as [First Contentful Paint](/docs/speed-insights/metrics#first-contentful-paint-
- Enables you to view performance analytics by page name and URL for more granular analysis
- Shows you [a score for your app's performance](/docs/speed-insights/metrics#how-the-scores-are-determined) on each recorded metric, whi

[Learn more about Speed Insights](/docs/speed-insights)

Observability

Vercel's observability features help you monitor, analyze, and manage your projects. From your project's dashboard on Vercel, you can tra

[Activity Logs](/docs/observability/activity-log), which you can see in the Activity tab of your project dashboard, are available on all

- **[Monitoring](/docs/observability/monitoring)**: A query editor that allows you to visualize, explore, and monitor your usage and traf
- **[Runtime Logs](/docs/runtime-logs)**: An interface that allows you to search and filter logs from static requests and Function invoca
- **[Audit Logs](/docs/observability/audit-log)**: An interface that enables your team owners to track and analyze their team members' ac

For Pro (and Enterprise) accounts:

- **[Log Drains](/docs/drains)**: Export your log data for better debugging and analyzing, either from the dashboard, or using one of [ou
- **[OpenTelemetry (OTEL) collector](/docs/observability/audit-log)**: Send OTEL traces from your Vercel functions to application perform

To summarize, using Vercel's observability features with CRA enable you to:

- Visualize website usage data, performance metrics, and logs
- Search and filter logs for static, and Function requests
- Use queries to see in-depth information about your website's usage and traffic
- Send your metrics and data to other observability services through our integrations
- Track and analyze team members' activity

[Learn more about Observability](/docs/observability)

More benefits

See [our Frameworks documentation page](/docs/frameworks) to learn about the benefits available to **all** frameworks when you deploy on '

More resources

Learn more about deploying CRA projects on Vercel with the following resources:

- [Remote caching docs](/docs/monorepos/remote-caching)
- [React with Formspree](/kb/guide/deploying-react-forms-using-formspree-with-vercel)
- [React Turborepo template](/templates/react/turborepo-design-system)

---
title: "Gatsby on Vercel"
description: "Learn how to use Vercel"
last_updated: "2026-01-16T02:19:31.082Z"
source: "https://vercel.com/docs/frameworks/frontend/gatsby"

Gatsby on Vercel

Gatsby is an open-source static-site generator. It enables developers to build fast and secure websites that integrate different content,

Gatsby also has a large ecosystem of plugins and tools that improve the development experience. Vercel supports many Gatsby features, inc

Get started with Gatsby on Vercel

Using the Gatsby Vercel Plugin

[Gatsby v4+](https://www.gatsbyjs.com/gatsby-4/) sites deployed to Vercel will **automatically detect Gatsby usage** and install the `@ve

To deploy your Gatsby site to Vercel, **do not** install the `@vercel/gatsby-plugin-vercel-builder` plugin yourself, or add it to your `g

[Gatsby v5](https://www.gatsbyjs.com/gatsby-5/) sites require Node.js 20 or higher.

Vercel persists your Gatsby project's `.cache` directory across builds.

Server-Side Rendering

Server-Side Rendering (SSR) allows you to render pages dynamically on the server. This is useful for pages where the rendered data needs

Vercel offers SSR that scales down resource consumption when traffic is low, and scales up with traffic surges. This protects your site f

Using Gatsby's SSR API with Vercel

You can server-render pages in your Gatsby application on Vercel [using Gatsby's native Server-Side Rendering API](https://www.gatsbyjs.c

To server-render a Gatsby page, you must export an `async` function called `getServerData`. The function can return an object with severa

The following example demonstrates a server-rendered Gatsby page using `getServerData`:

```js filename="pages/example.jsx" framework=all
const Page = ({ serverData }) => {
  const { name } = serverData;
  return <div>Hello, {name}</div>;
};

export async function getServerData(props) {
  try {
    const res = await fetch(`https://example-data-source.com/api/some-data`);
    return {
      props: await res.json(),
    };
  } catch (error) {
    return {
      status: 500,
      headers: {},
      props: {},
    };
  }
}

export default Page;
```

```ts filename="pages/example.tsx" framework=all
import type { GetServerDataProps, GetServerDataReturn } from 'gatsby';

type ServerDataProps = {
  hello: string;
};

const Page = (props: PageProps) => {
  const { name } = props.serverData;
  return <div>Hello, {name}</div>;
};
```

```
export async function getServerData(
  props: GetServerDataProps,
): GetServerDataReturn<ServerDataProps> {
  try {
    const res = await fetch(`https://example-data-source.com/api/some-data`);
    return {
      props: await res.json(),
    };
  } catch (error) {
    return {
      status: 500,
      headers: {},
      props: {},
    };
  }
}

export default Page;
```

To summarize, SSR with Gatsby on Vercel:

- Scales to zero when not in use
- Scales automatically with traffic increases
- Has zero-configuration support for [`Cache-Control` headers](/docs/cdn-cache), including `stale-while-revalidate`
- Framework-aware infrastructure enables switching rendering between Edge/Node.js runtimes

[Learn more about SSR](https://www.gatsbyjs.com/docs/how-to/rendering-options/using-server-side-rendering/)

Deferred Static Generation

Deferred Static Generation (DSG) allows you to defer the generation of static pages until they are requested for the first time.

To use DSG, you must set the `defer` option to `true` in the `createPages()` function in your `gatsby-node` file.

```js filename="pages/index.jsx" framework=all
/**
 * @type {import('gatsby').GatsbyNode['createPages']}
 */
exports.createPages = async ({ actions }) => {
  const { createPage } = actions;
  createPage({
    defer: true,
    path: '/using-dsg',
    component: require.resolve('./src/templates/using-dsg.js'),
    context: {},
  });
};
```

```ts filename="pages/index.tsx" framework=all
import type { GatsbyNode } from 'gatsby';

export const createPages: GatsbyNode['createPages'] = async ({ actions }) => {
  const { createPage } = actions;
  createPage({
    defer: true,
    path: '/using-dsg',
    component: require.resolve('./src/templates/using-dsg.js'),
    context: {},
  });
};
```

[See the Gatsby docs on DSG to learn more](https://www.gatsbyjs.com/docs/how-to/rendering-options/using-deferred-static-generation/#intro

To summarize, DSG with Gatsby on Vercel:

- Allows you to defer non-critical page generation to user request, speeding up build times
- Works out of the box when you deploy on Vercel
- Can yield dramatic speed increases for large sites with content that is infrequently visited

[Learn more about DSG](https://www.gatsbyjs.com/docs/how-to/rendering-options/using-deferred-static-generation/)

Incremental Static Regeneration

Gatsby supports [Deferred Static Generation](#deferred-static-generation).

The static rendered fallback pages are not generated at build time. This differentiates it from incremental static regeneration (ISR). In

See the documentation for [Deferred Static Generation](#deferred-static-generation).

API routes

You can add API Routes to your Gatsby site using the framework's native support for the `src/api` directory. Doing so will deploy your ro

The following example demonstrates a basic API Route using Vercel functions:

```js filename="src/api/handler.js" framework=all
export default function handler(request, response) {
  response.status(200).json({
    body: request.body,
    query: request.query,
    cookies: request.cookies,
  });
}
```

```ts filename="src/api/handler.ts" framework=all
```

```
import type { VercelRequest, VercelResponse } from '@vercel/node';

export default function handler(
  request: VercelRequest,
  response: VercelResponse,
) {
  response.status(200).json({
    body: request.body,
    query: request.query,
    cookies: request.cookies,
  });
}
```

To view your route locally, run the following command in your terminal:

```bash filename="terminal"
gatsby develop
```

Then navigate to `http://localhost:8000/api/handler` in your web browser.

Dynamic API routes

**Vercel does not currently have first-class support for dynamic API routes in Gatsby. For now, using them requires the workaround descri

To use Gatsby's Dynamic API routes on Vercel, you must:

1. Define your dynamic routes in a `vercel.json` file at the root directory of your project, as shown below:

   ```json filename="vercel.json"
   {
     "$schema": "https://openapi.vercel.sh/vercel.json",
     "rewrites": [
       {
         "source": "/api/blog/:id",
         "destination": "/api/blog/[id]"
       }
     ]
   }
   ```

2. Read your dynamic parameters from `req.query`, as shown below:

   ```js filename="api/blog/[id].js" framework=all
   export default function handler(request, response) {
     console.log(`/api/blog/${request.query.id}`);
     response.status(200).json({
       body: request.body,
       query: request.query,
       cookies: request.cookies,
     });
   }
   ```

   ```ts filename="api/blog/[id].ts" framework=all
   import type { VercelRequest, VercelResponse } from '@vercel/node';

   export default function handler(
     request: VercelRequest & { params: { id: string } },
     response: VercelResponse,
   ) {
     console.log(`/api/blog/${request.query.id}`);
     response.status(200).json({
       body: request.body,
       query: request.query,
       cookies: request.cookies,
     });
   }
   ```

> **💡 Note:** Although typically you'd access the dynamic parameter with `request.param`
> when using Gatsby, you must use `request.query` on Vercel.

Splat API routes

Splat API routes are dynamic wildcard routes that will match anything after the splat (`[...]`). **Vercel does not currently have first-c

To use Gatsby's splat API routes on Vercel, you must:

1. Define your splat routes in a `vercel.json` file at the root directory of your project, as shown below:

   ```json filename="vercel.json"
   {
     "$schema": "https://openapi.vercel.sh/vercel.json",
     "rewrites": [
       {
         "source": "/api/products/:path*",
         "destination": "/api/products/[...]"
       }
     ]
   }
   ```

2. Read your dynamic parameters from `req.query.path`, as shown below:

   ```js filename="api/products/[...].js" framework=all
   export default function handler(request, response) {
     console.log(`/api/products/${request.query.path}`);
     response.status(200).json({
```

````
      body: request.body,
      query: request.query,
      cookies: request.cookies,
    });
  }
  ```

  ```ts filename="api/products/[...].ts" framework=all
  import type { VercelRequest, VercelResponse } from '@vercel/node';

  export default function handler(
    request: VercelRequest & { params: { path: string } },
    response: VercelResponse,
  ) {
    console.log(`/api/products/${request.query.path}`);
    response.status(200).json({
      body: request.body,
      query: request.query,
      cookies: request.cookies,
    });
  }
  ```

**To summarize, API Routes with Gatsby on Vercel:**

- Scale to zero when not in use
- Scale automatically with traffic increases
- Can be tested as Vercel Functions in your local environment

[Learn more about Gatsby API Routes](https://www.gatsbyjs.com/docs/reference/routing/creating-routes/)

## Routing Middleware

Gatsby does not have native framework support for using [Routing Middleware](/docs/routing-middleware).

However, you can still use Routing Middleware with your Gatsby site by creating a `middeware.js` or `middeware.ts` file in your project's

The following example demonstrates middleware that adds security headers to responses sent to users who visit the `/example` route in you

```js filename="middleware.js" framework=all
import { next } from '@vercel/functions';

export const config = {
  // Only run the middleware on the example route
  matcher: '/example',
};

export default function middleware(request) {
  return next({
    headers: {
      'Referrer-Policy': 'origin-when-cross-origin',
      'X-Frame-Options': 'DENY',
      'X-Content-Type-Options': 'nosniff',
      'X-DNS-Prefetch-Control': 'on',
      'Strict-Transport-Security':
        'max-age=31536000; includeSubDomains; preload',
    },
  });
}
```

```ts filename="middleware.ts" framework=all
import { next } from '@vercel/functions';

export const config = {
  // Only run the middleware on the example route
  matcher: '/example',
};

export default function middleware(request: Request): Response {
  return next({
    headers: {
      'Referrer-Policy': 'origin-when-cross-origin',
      'X-Frame-Options': 'DENY',
      'X-Content-Type-Options': 'nosniff',
      'X-DNS-Prefetch-Control': 'on',
      'Strict-Transport-Security':
        'max-age=31536000; includeSubDomains; preload',
    },
  });
}
```

**To summarize, Routing Middleware with Gatsby on Vercel:**

- Executes before a request is processed on a site, allowing you to modify responses to user requests
- Runs on *all* requests, but can be scoped to specific paths [through a `matcher` config](/docs/routing-middleware/api#match-paths-based
- Uses our lightweight Edge Runtime to keep costs low and responses fast

[Learn more about Routing Middleware](/docs/routing-middleware)

## Speed Insights

[Core Web Vitals](/docs/speed-insights) are supported for Gatsby v4+ projects with no initial configuration necessary.

When you deploy a Gatsby v4+ site on Vercel, we automatically install the `@vercel/gatsby-plugin-vercel-analytics` package and add it to

**We do not recommend installing the Gatsby analytics plugin yourself**.

To access your Core Web Vitals data, you must enable Vercel analytics in your project's dashboard. [See our quickstart guide to do so now
````
````

To summarize, using Speed Insights with Gatsby on Vercel:

- Enables you to track traffic performance metrics, such as [First Contentful Paint](/docs/speed-insights/metrics#first-contentful-paint-
- Enables you to view performance analytics by page name and URL for more granular analysis
- Shows you [a score for your app's performance](/docs/speed-insights/metrics#how-the-scores-are-determined) on each recorded metric, whi

[Learn more about Speed Insights](/docs/speed-insights)

Image Optimization

While Gatsby [does provide an Image plugin](https://www.gatsbyjs.com/plugins/gatsby-plugin-image), it is not currently compatible with Ve

If this is something your team is interested in, [please contact our sales team](/contact/sales).

[Learn more about Image Optimization](/docs/image-optimization)

More benefits

See [our Frameworks documentation page](/docs/frameworks) to learn about the benefits available to **all** frameworks when you deploy on '

More resources

- [Build Output API](/docs/build-output-api/v3)

---
title: "Frontends on Vercel"
description: "Vercel supports a wide range of the most popular frontend frameworks, optimizing how your application builds and runs no ma
last_updated: "2026-01-16T02:19:31.093Z"
source: "https://vercel.com/docs/frameworks/frontend"

Frontends on Vercel

The following frontend frameworks are supported with zero-configuration.

- **Angular**: Angular is a TypeScript-based cross-platform framework from Google.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/angular) | [View Demo](https
- **Astro**: Astro is a new kind of static site builder for the modern web. Powerful developer experience meets lightweight output.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/astro) | [View Demo](https:/
- **Brunch**: Brunch is a fast and simple webapp build tool with seamless incremental compilation for rapid development.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/brunch) | [View Demo](https:
- **React**: Create React App allows you to get going with React in no time.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/create-react-app) | [View De
- **Docusaurus (v1)**: Docusaurus makes it easy to maintain Open Source documentation websites.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/docusaurus) | [View Demo](ht
- **Docusaurus (v2+)**: Docusaurus makes it easy to maintain Open Source documentation websites.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/docusaurus-2) | [View Demo](
- **Dojo**: Dojo is a modern progressive, TypeScript first framework.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/dojo) | [View Demo](https://
- **Eleventy**: 11ty is a simpler static site generator written in JavaScript, created to be an alternative to Jekyll.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/eleventy) | [View Demo](http
- **Ember.js**: Ember.js helps webapp developers be more productive out of the box.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/ember) | [View Demo](https:/
- **FastHTML**: The fastest way to create an HTML app
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/fasthtml) | [View Demo](http
- **Gatsby.js**: Gatsby helps developers build blazing fast websites and apps with React.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/gatsby) | [View Demo](https:
- **Gridsome**: Gridsome is a Vue.js-powered framework for building websites & apps that are fast by default.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/gridsome) | [View Demo](http
- **Hexo**: Hexo is a fast, simple & powerful blog framework powered by Node.js.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/hexo) | [View Demo](https://
- **Hugo**: Hugo is the world's fastest framework for building websites, written in Go.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/hugo) | [View Demo](https://
- **Hydrogen (v1)**: React framework for headless commerce
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/hydrogen) | [View Demo](http
- **Ionic Angular**: Ionic Angular allows you to build mobile PWAs with Angular and the Ionic Framework.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/ionic-angular) | [View Demo]
- **Ionic React**: Ionic React allows you to build mobile PWAs with React and the Ionic Framework.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/ionic-react) | [View Demo](h
- **Jekyll**: Jekyll makes it super easy to transform your plain text into static websites and blogs.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/jekyll) | [View Demo](https:
- **Middleman**: Middleman is a static site generator that uses all the shortcuts and tools in modern web development.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/middleman) | [View Demo](htt
- **Parcel**: Parcel is a zero configuration build tool for the web that scales to projects of any size and complexity.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/parcel) | [View Demo](https:
- **Polymer**: Polymer is an open-source webapps library from Google, for building using Web Components.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/polymer) | [View Demo](https
- **Preact**: Preact is a fast 3kB alternative to React with the same modern API.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/preact) | [View Demo](https:
- **React Router**: Declarative routing for React
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/react-router) | [View Demo](
- **Saber**: Saber is a framework for building static sites in Vue.js that supports data from any source.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/saber)
- **Sanity**: The structured content platform.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/sanity) | [View Demo](https:
- **Sanity (v3)**: The structured content platform.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/sanity-v3) | [View Demo](htt
- **Scully**: Scully is a static site generator for Angular.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/scully) | [View Demo](https:
- **SolidStart (v0)**: Simple and performant reactivity for building user interfaces.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/solidstart) | [View Demo](ht
- **SolidStart (v1)**: Simple and performant reactivity for building user interfaces.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/solidstart-1) | [View Demo](
- **Stencil**: Stencil is a powerful toolchain for building Progressive Web Apps and Design Systems.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/stencil) | [View Demo](https
- **Storybook**: Frontend workshop for UI development
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/storybook)
- **UmiJS**: UmiJS is an extensible enterprise-level React application framework.

  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/umijs) | [View Demo](https:/
- **Vite**: Vite is a new breed of frontend build tool that significantly improves the frontend development experience.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/vite) | [View Demo](https://
- **VitePress**: VitePress is VuePress' little brother, built on top of Vite.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/vitepress) | [View Demo](htt
- **Vue.js**: Vue.js is a versatile JavaScript framework that is as approachable as it is performant.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/vue) | [View Demo](https://v
- **VuePress**: Vue-powered Static Site Generator
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/vuepress) | [View Demo](http
- **Zola**: Everything you need to make a static site engine in one binary.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/zola) | [View Demo](https://

Frameworks infrastructure support matrix

The following table shows which features are supported by each framework on Vercel. The framework list is not exhaustive, but a represent

We're committed to having support for all Vercel features across frameworks, and continue to work with framework authors on adding suppor

Legend: ✓ Supported | ✗ Not Supported | N/A Not Applicable

Feature	Next.js	SvelteKit	Nuxt	TanStack	Astro	Remix	Vite	CRA
[Static Assets](/docs/cdn)	✓	✓	✓	✓	✓	✓	✓	✓
[Edge Routing Rules](/docs/cdn#features)	✓	✓	✓	✓	✓	✓	✓	✓
[Routing Middleware](/docs/routing-middleware)	✓	✓	✓	✓	✓	✓	✓	✓
[Server-Side Rendering](/docs/functions)	✓	✓	✓	✓	✓	✓	N/A	N/A
[Streaming SSR](/docs/functions/streaming-functions)	✓	✓	✗	✓	✓	✓	N/A	N/A
[Incremental Static Regeneration](/docs/incremental-static-regeneration)	✓	✓	✓	✗	✓	✗	N/A	N/A
[Image Optimization](/docs/image-optimization)	✓	✓	✓	N/A	✓	✗	N/A	N/A
[Data Cache](/docs/data-cache)	✓	N/A	N/A	N/A	N/A	N/A	N/A	
[Native OG Image Generation](/docs/og-image-generation)	✓	N/A	✓	N/A	N/A	N/A	N/A	N/A
[Multi-runtime support (different routes)](/docs/functions/runtimes)	✓	✓	✓	N/A	✗	✓	N/A	N/A
[Multi-runtime support (entire app)](/docs/functions/runtimes)	✓	✓	✓	N/A	✓	✓	N/A	N/A
[Output File Tracing](/kb/guide/how-can-i-use-files-in-serverless-functions)	✓	✓	✓	✓	✓	✓	✗	N/A	N/A
[Skew Protection](/docs/skew-protection)	✓	✓	✗	N/A	✓	✗	N/A	N/A
[Framework Routing Middleware](/docs/routing-middleware)	✓	N/A	✗	✓	✓	✗	N/A	N/A

---
title: "React Router on Vercel"
description: "Learn how to use Vercel"
last_updated: "2026-01-16T02:19:31.115Z"
source: "https://vercel.com/docs/frameworks/frontend/react-router"

React Router on Vercel

React Router is a multi-strategy router for React. When used [as a framework](https://reactrouter.com/home#react-router-as-a-framework),
> Preset](#vercel-react-router-preset) when deploying to Vercel.## `@vercel/react-router`The optional `@vercel/react-router` package cont
  - Contains the [Vercel Preset](#vercel-react-router-preset) to enhance React Router functionality on Vercel
* `@vercel/react-router/entry.server` import
  - For situations where you need to [define a custom `entry.server` file](#using-a-custom-app/entry.server-file).To get started, navigat
  <Code tab="pnpm">
    ```bash
    pnpm i @vercel/react-router
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @vercel/react-router
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @vercel/react-router
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i @vercel/react-router
    ```
  </Code>
</CodeBlock>## Vercel React Router PresetWhen using the [React Router](https://reactrouter.com/start/framework/installation) as a framewo
```
import { vercelPreset } from '@vercel/react-router/vite';
import type { Config } from '@react-router/dev/config';

export default {
  // Config options...
  // Server-side render by default, to enable SPA mode set this to `false`
  ssr: true,
  presets: [vercelPreset()],
} satisfies Config;
```
```When this Preset is configured, your React Router application is enhanced with Vercel-specific functionality:* Allows function-level c
* Allows Vercel to understand the routing structure of the application, which allows for bundle splitting
* Accurate "Deployment Summary" on the deployment details page## Server-Side Rendering (SSR)Server-Side Rendering (SSR) allows you to ren
```
import { type RouteConfig, index } from '@react-router/dev/routes';

export default [index('routes/home.tsx')] satisfies RouteConfig;
```
``````js filename="/app/routes.js" framework=all
import { index } from '@react-router/dev/routes';

export default [index('routes/home.jsx')];
``````tsx filename="/app/routes/home.tsx" framework=all
import type { Route } from './+types/home';
import { Welcome } from '../welcome/welcome';

export function meta({}: Route.MetaArgs) {
  return [
    { title: 'New React Router App' },
```

```
    { name: 'description', content: 'Welcome to React Router!' },
  ];
}

export default function Home() {
  return <Welcome />;
}
``````jsx filename="/app/routes/home.jsx" framework=all
import { Welcome } from '../welcome/welcome';

export function meta({}) {
  return [
    { title: 'New React Router App' },
    { name: 'description', content: 'Welcome to React Router!' },
  ];
}

export default function Home() {
  return <Welcome />;
}
```**To summarize, Server-Side Rendering (SSR) with React Router on Vercel:*** Scales to zero when not in use
* Scales automatically with traffic increases
* Has framework-aware infrastructure to generate Vercel Functions
* Supports the use of Vercel's [Fluid compute](/docs/fluid-compute) for enhanced performance## Response streaming[Streaming HTTP response
[Streaming with Suspense](https://reactrouter.com/how-to/suspense) page in the
React Router docs for general instructions.**Streaming with React Router on Vercel:*** Offers faster Function response times, improving y
* Allows you to return large amounts of data without exceeding Vercel Function response size limits
* Allows you to display Instant Loading UI from the server with React Router's `<Await>`[Learn more about Streaming](/docs/functions/stre
* For requests repeated after 1 second, but before 60 seconds have passed, return the cached content and mark it as stale. The stale cont
```
import { Route } from './+types/some-route';

export function headers(_: Route.HeadersArgs) {
  return {
    'Cache-Control': 's-maxage=1, stale-while-revalidate=59',
  };
}

export async function loader() {
  // Fetch data necessary to render content
}
``````jsx filename="/app/routes/example.jsx" framework=all
export function headers(_) {
  return {
    'Cache-Control': 's-maxage=1, stale-while-revalidate=59',
  };
}

export async function loader() {
  // Fetch data necessary to render content
}
```See [our docs on cache limits](/docs/cdn-cache#limits) to learn the max size and lifetime of caches stored on Vercel.**To summarize, u
* Allow you to serve content from the cache *while updating the cache in the background* with `stale-while-revalidate`[Learn more about c
```
  <Code tab="pnpm">
    ```bash
    pnpm i @vercel/analytics
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @vercel/analytics
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @vercel/analytics
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i @vercel/analytics
    ```
  </Code>
</CodeBlock>Then, follow the instructions below to add the `Analytics` component to your app. The `Analytics` component is a wrapper arou
```
import { Analytics } from '@vercel/analytics/react';

export default function App() {
  return (
    <html lang="en">
      <body>
        <Analytics />
      </body>
    </html>
  );
}
``````jsx filename="app/root.jsx" framework=all
import { Analytics } from '@vercel/analytics/react';

export default function App() {
  return (
    <html lang="en">
      <body>
        <Analytics />
      </body>
    </html>
  );
}
```**To summarize, Analytics with React Router on Vercel:*** Enables you to track traffic and see your top-performing pages
* Offers you detailed breakdowns of visitor demographics, including their OS, browser, geolocation and more[Learn more about Analytics](/
for supplying a "load context" for use by the application's loaders and actions.The server entrypoint file is expected to export a Web AP
matches the following signature:```ts
```
```

```
export default async function (request: Request) => Response | Promise<Response>;
```To implement a server entrypoint using the [Hono web framework](https://hono.dev), follow these steps:First define the `build.rollupOp
```
import { reactRouter } from '@react-router/dev/vite';
import tailwindcss from '@tailwindcss/vite';
import { defineConfig } from 'vite';
import tsconfigPaths from 'vite-tsconfig-paths';

export default defineConfig(({ isSsrBuild }) => ({
  build: {
    rollupOptions: isSsrBuild
      ? {
          input: './server/app.ts',
        }
      : undefined,
  },
  plugins: [tailwindcss(), reactRouter(), tsconfigPaths()],
}));
```
``````js {7-13} filename="/vite.config.js" framework=all
```
import { reactRouter } from '@react-router/dev/vite';
import tailwindcss from '@tailwindcss/vite';
import { defineConfig } from 'vite';
import tsconfigPaths from 'vite-tsconfig-paths';

export default defineConfig(({ isSsrBuild }) => ({
  build: {
    rollupOptions: isSsrBuild
      ? {
          input: './server/app.js',
        }
      : undefined,
  },
  plugins: [tailwindcss(), reactRouter(), tsconfigPaths()],
}));
```Then, create the server entrypoint file:```ts filename="/server/app.ts" framework=all
```
import { Hono } from 'hono';
import { createRequestHandler } from 'react-router';

// @ts-expect-error - virtual module provided by React Router at build time
import * as build from 'virtual:react-router/server-build';

declare module 'react-router' {
  interface AppLoadContext {
    VALUE_FROM_HONO: string;
  }
}

const app = new Hono();

// Add any additional Hono middleware here

const handler = createRequestHandler(build);
app.mount('/', (req) =>
  handler(req, {
    // Add your "load context" here based on the current request
    VALUE_FROM_HONO: 'Hello from Hono',
  }),
);

export default app.fetch;
```
``````js filename="/server/app.js" framework=all
```
import { Hono } from 'hono';
import { createRequestHandler } from 'react-router';

import * as build from 'virtual:react-router/server-build';

const app = new Hono();

// Add any additional Hono middleware here

const handler = createRequestHandler(build);
app.mount('/', (req) =>
  handler(req, {
    // Add your "load context" here based on the current request
    VALUE_FROM_HONO: 'Hello from Hono',
  }),
);

export default app.fetch;
```
```**To summarize, using a custom server entrypoint with React Router on Vercel allows you to:*** Supply a "load context" for use in your
* Use a Web API-compatible framework alongside your React Router application## Using a custom `app/entry.server` fileBy default, Vercel s
for streaming to work with Vercel Functions. This version will be used when
no `entry.server` file is found in the project.However, your application may define a customized `app/entry.server.jsx` or `app/entry.ser
file if necessary. When doing so, your custom `entry.server` file should use the `handleRequest`
function exported by `@vercel/react-router/entry.server`.For example, to supply the `nonce` option and set the corresponding `Content-Sec
```
import { handleRequest } from '@vercel/react-router/entry.server';
import type { AppLoadContext, EntryContext } from 'react-router';

export default async function (
  request: Request,
  responseStatusCode: number,
  responseHeaders: Headers,
  routerContext: EntryContext,
  loadContext?: AppLoadContext,
): Promise<Response> {
  const nonce = crypto.randomUUID();
  const response = await handleRequest(
    request,
    responseStatusCode,
    responseHeaders,
    routerContext,
```
```

```
    loadContext,
    { nonce },
  );
  response.headers.set(
    'Content-Security-Policy',
    `script-src 'nonce-${nonce}'`,
  );
  return response;
}
``````jsx filename="/app/entry.server.jsx" framework=all
import { handleRequest } from '@vercel/react-router/entry.server';

export default async function (
  request,
  responseStatusCode,
  responseHeaders,
  routerContext,
  loadContext,
) {
  const nonce = crypto.randomUUID();
  const response = await handleRequest(
    request,
    responseStatusCode,
    responseHeaders,
    routerContext,
    loadContext,
    { nonce },
  );
  response.headers.set(
    'Content-Security-Policy',
    `script-src 'nonce-${nonce}'`,
  );
  return response;
}
```## More benefitsSee [our Frameworks documentation page](/docs/frameworks) to learn about the benefits available to **all** frameworks

---
title: "Vite on Vercel"
description: "Learn how to use Vercel"
last_updated: "2026-01-16T02:19:31.141Z"
source: "https://vercel.com/docs/frameworks/frontend/vite"
---

# Vite on Vercel

Vite is an opinionated build tool that aims to provide a faster and leaner development experience for modern web projects. Vite provides

These features make Vite more desirable than out-of-the-box CLIs when building larger projects with frameworks for many developers.

Vite powers popular frameworks like [SvelteKit](/docs/frameworks/sveltekit), and is often used in large projects built with [Vue](/kb/gui

## Getting started

## Using Vite community plugins

Although Vite offers modern features like [SSR](#server-side-rendering-ssr) and [Vercel functions](#vercel-functions) out of the box, imp

Vite's plugins are based on [Rollup's plugin interface](https://rollupjs.org/javascript-api/), giving Vite users access to [many tools fr

**We recommend using Vite plugins to configure your project when possible**.

### `vite-plugin-vercel`

[`vite-plugin-vercel`](https://github.com/magne4000/vite-plugin-vercel#readme) is a popular community Vite plugin that implements [the Bu

- [Server-Side Rendering (SSR)](#server-side-rendering-ssr)
- [Vercel functions](#vercel-functions)
- [Incremental Static Regeneration](/docs/incremental-static-regeneration)
- [Static Site Generation](/docs/build-output-api/v3/primitives#static-files)

When using the Vercel CLI, set the port as an environment variable. To allow Vite to access this, include the environment variable in you

```ts filename="vite.config.ts" framework=all
import { defineConfig } from 'vite';
import vercel from 'vite-plugin-vercel';

export default defineConfig({
  server: {
    port: process.env.PORT as unknown as number,
  },
  plugins: [vercel()],
});
```

```js filename="vite.config.js" framework=all
import { defineConfig } from 'vite';
import vercel from 'vite-plugin-vercel';

export default defineConfig({
  server: {
    port: process.env.PORT,
  },
  plugins: [vercel()],
});
```

### `vite-plugin-ssr`
```

[`vite-plugin-ssr`](https://vite-plugin-ssr.com/) is another popular community Vite plugin that implements [the Build Output API spec](/d

- [Server-Side Rendering (SSR)](#server-side-rendering-ssr)
- [Vercel functions](#vercel-functions)
- [Static Site Generation](/docs/build-output-api/v3/primitives#static-files)

## Environment Variables

Vercel provides a set of [System Environment Variables](/docs/environment-variables/system-environment-variables) that our platform autom

These environment variables will be available to your project automatically, and you can enable or disable them in your project settings (

To access Vercel's System Environment Variables in Vite during the build process, prefix the variable name with `VITE`. For example, `VIT

The following example demonstrates a Vite config file that sets `VITE_VERCEL_ENV` as a global constant available throughout the app:

```js filename="vite.config.js" framework=all
export default defineConfig(() => {
  return {
    define: {
      __APP_ENV__: process.env.VITE_VERCEL_ENV,
    },
  };
});
```

```ts filename="vite.config.ts" framework=all
export default defineConfig(() => {
  return {
    define: {
      __APP_ENV__: process.env.VITE_VERCEL_ENV,
    },
  };
});
```

If you want to read environment variables from a `.env` file, additional configuration is required. See [the Vite config docs](https://vi

**To summarize, the benefits of using System Environment Variables with Vite on Vercel include:**

- Access to Vercel deployment information, dynamically or statically, with our preconfigured System Environment Variables
- Access to automatically-configured environment variables provided by [integrations for your preferred services](/docs/environment-varia
- Searching and filtering environment variables by name and environment in Vercel's dashboard

[Learn more about System Environment Variables](/docs/environment-variables/system-environment-variables)

## Vercel Functions

Vercel Functions scale up and down their resource consumption based on traffic demands. This scaling prevents them from failing during pe

If your project uses [a Vite community plugin](#using-vite-community-plugins), such as [`vite-plugin-ssr`](https://vite-plugin-ssr.com/),

If you're using a framework built on Vite, check that framework's official documentation or [our dedicated framework docs](/docs/framewor

If you're not using a framework or plugin that supports Vercel Functions, you can still use them in your project by creating routes in an

The following example demonstrates a basic Vercel Function defined in an `api` directory:

```js filename="api/handler.js" framework=all
export default function handler(request, response) {
  response.status(200).json({
    body: request.body,
    query: request.query,
    cookies: request.cookies,
  });
}
```

```ts filename="api/handler.ts" framework=all
import type { VercelRequest, VercelResponse } from '@vercel/node';

export default function handler(
  request: VercelRequest,
  response: VercelResponse,
) {
  response.status(200).json({
    body: request.body,
    query: request.query,
    cookies: request.cookies,
  });
}
```

**To summarize, Vercel Functions on Vercel:**

- Scales to zero when not in use
- Scales automatically with traffic increases
- Support standard [Web APIs](https://developer.mozilla.org/docs/Web/API), such as `URLPattern`, `Response`, and more

[Learn more about Vercel Functions](/docs/functions)

## Server-Side Rendering (SSR)

Server-Side Rendering (SSR) allows you to render pages dynamically on the server. This is useful for pages where the rendered data needs

Vite exposes [a low-level API for implementing SSR](https://vitejs.dev/guide/ssr.html#server-side-rendering), but in most cases, **we rec

See [the SSR section of Vite's plugin repo](https://github.com/vitejs/awesome-vite#ssr) for a more comprehensive list of SSR plugins.

**To summarize, SSR with Vite on Vercel:**

- Scales to zero when not in use
- Scales automatically with traffic increases
- Has zero-configuration support for [`Cache-Control`](/docs/cdn-cache) headers, including `stale-while-revalidate`

[Learn more about SSR](https://vitejs.dev/guide/ssr.html)

## Using Vite to make SPAs

If your Vite app is [configured to deploy as a Single Page Application (SPA)](https://vitejs.dev/config/shared-options.html#apptype), dee

To enable deep linking in SPA Vite apps, create a `vercel.json` file at the root of your project, and add the following code:

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "rewrites": [
    {
      "source": "/(.*)",
      "destination": "/index.html"
    }
  ]
}
```

> **💡 Note:** If [`cleanUrls`](/docs/project-configuration#cleanurls) is set to `true` in
> your project's `vercel.json`, do not include the file extension in the source
> or destination path. For example, `/index.html` would be `/`

**Deploying your app in Multi-Page App mode is recommended for production builds**.

Learn more about [Mutli-Page App mode](https://vitejs.dev/guide/build.html#multi-page-app) in the Vite docs.

## More benefits

See [our Frameworks documentation page](/docs/frameworks) to learn about the benefits available to **all** frameworks when you deploy on '

## More resources

Learn more about deploying Vite projects on Vercel with the following resources:

- [Explore Vite's template repo](https://github.com/vitejs/vite/tree/main/packages/create-vite)

---
title: "Next.js on Vercel"
description: "Vercel is the native Next.js platform, designed to enhance the Next.js experience."
last_updated: "2026-01-16T02:19:31.323Z"
source: "https://vercel.com/docs/frameworks/full-stack/nextjs"
---

# Next.js on Vercel

[Next.js](https://nextjs.org/) is a fullstack React framework for the web, maintained by Vercel.

While Next.js works when self-hosting, deploying to Vercel is zero-configuration and provides additional enhancements for **scalability,

## Getting started

## Incremental Static Regeneration

[Incremental Static Regeneration (ISR)](/docs/incremental-static-regeneration) allows you to create or update content *without* redeployi

When self-hosting, (ISR) is limited to a single region workload. Statically generated pages are not distributed closer to visitors by def

> For \["nextjs"]:

To enable ISR with Next.js in the `pages` router, add a `revalidate` property to the object returned from `getStaticProps`:

> For \["nextjs-app"]:

To enable ISR with Next.js in the `app` router, add an options object with a `revalidate` property to your `fetch` requests:

```ts filename="apps/example/page.tsx" framework=nextjs-app
export default async function Page() {
  const res = await fetch('https://api.vercel.app/blog', {
    next: { revalidate: 10 }, // Seconds
  });

  const data = await res.json();

  return (
    <main>
      <pre>{JSON.stringify(data, null, 2)}</pre>
    </main>
  );
}
```

```js filename="apps/example/page.jsx" framework=nextjs-app
export default async function Page() {
  const res = await fetch('https://api.vercel.app/blog', {
    next: { revalidate: 10 }, // Seconds
  });

  const data = await res.json();

  return (
    <main>
      <pre>{JSON.stringify(data, null, 2)}</pre>
```

```
    </main>
  );
}
```

```ts filename="pages/example/index.tsx" framework=nextjs
export async function getStaticProps() {
  /* Fetch data here */

  return {
    props: {
      /* Add something to your props */
    },
    revalidate: 10, // Seconds
  };
}
```

```js filename="pages/example/index.jsx" framework=nextjs
export async function getStaticProps() {
  /* Fetch data here */

  return {
    props: {
      /* Add something to your props */
    },
    revalidate: 10, // Seconds
  };
}
```

**To summarize, using ISR with Next.js on Vercel:**

- Better performance with our global [CDN](/docs/cdn)
- Zero-downtime rollouts to previously statically generated pages
- Framework-aware infrastructure enables global content updates in 300ms
- Generated pages are both cached and persisted to durable storage

[Learn more about Incremental Static Regeneration (ISR)](/docs/incremental-static-regeneration)

## Server-Side Rendering (SSR)

Server-Side Rendering (SSR) allows you to render pages dynamically on the server. This is useful for pages where the rendered data needs

On Vercel, you can server-render Next.js applications through [Vercel Functions](/docs/functions).

**To summarize, SSR with Next.js on Vercel:**

- Scales to zero when not in use
- Scales automatically with traffic increases
- Has zero-configuration support for [`Cache-Control` headers](/docs/cdn-cache), including `stale-while-revalidate`
- Framework-aware infrastructure enables automatic creation of Functions for SSR

[Learn more about SSR](https://nextjs.org/docs/app/building-your-application/rendering#static-and-dynamic-rendering-on-the-server)

## Streaming

Vercel supports streaming in Next.js projects with any of the following:

- [Route Handlers](https://nextjs.org/docs/app/building-your-application/routing/router-handlers)
- [Vercel Functions](/docs/functions/streaming-functions)
- React Server Components

Streaming data allows you to fetch information in chunks rather than all at once, speeding up Function responses. You can use streams to

#### Streaming with `loading` and `Suspense`

In the Next.js App Router, you can use the `loading` file convention or a `Suspense` component to show an instant loading state from the

The `loading` file provides a way to show a loading state for a whole route or route-segment, instead of just particular sections of a pa

The following example demonstrates a basic `loading` file:

```js filename="loading.jsx" framework=all
export default function Loading() {
  return <p>Loading...</p>;
}
```

```ts filename="loading.tsx" framework=all
export default function Loading() {
  return <p>Loading...</p>;
}
```

Learn more about loading in the [Next.js docs](https://nextjs.org/docs/app/building-your-application/routing/loading-ui-and-streaming).

The `Suspense` component, introduced in React 18, enables you to display a fallback until components nested within it have finished loadi

You can specify a component to show during the loading state with the `fallback` prop on the `Suspense` component as shown below:

```ts filename="app/dashboard/page.tsx" framework=all
import { Suspense } from 'react';
import { PostFeed, Weather } from './components';

export default function Posts() {
  return (
    <section>
      <Suspense fallback={<p>Loading feed...</p>}>
        <PostFeed />
```

```
      </Suspense>
      <Suspense fallback={<p>Loading weather...</p>}>
        <Weather />
      </Suspense>
    </section>
  );
}
```

```js filename="app/dashboard/page.jsx" framework=all
import { Suspense } from 'react';
import { PostFeed, Weather } from './components';

export default function Posts() {
  return (
    <section>
      <Suspense fallback={<p>Loading feed...</p>}>
        <PostFeed />
      </Suspense>
      <Suspense fallback={<p>Loading weather...</p>}>
        <Weather />
      </Suspense>
    </section>
  );
}
```

**To summarize, using Streaming with Next.js on Vercel:**

- Speeds up Function response times, improving your app's user experience
- Display initial loading UI with incremental updates from the server as new data becomes available

Learn more about [Streaming](/docs/functions/streaming-functions) with Vercel Functions.

## Partial Prerendering

> **⚠️ Warning:** Partial Prerendering as an experimental feature. It is currently
>  environments.

Partial Prerendering (PPR) is an **experimental** feature in Next.js that allows the static portions of a page to be pre-generated and se

When a user visits a route:

- A static route *shell* is served immediately, this makes the initial load fast.
- The shell leaves *holes* where dynamic content will be streamed in to minimize the perceived overall page load time.
- The async holes are loaded in parallel, reducing the overall load time of the page.

This approach is useful for pages like dashboards, where unique, per-request data coexists with static elements such as sidebars or layou

See the [Partial Prerendering docs](https://nextjs.org/docs/app/api-reference/next-config-js/partial-prerendering) to learn more.

## Image Optimization

[Image Optimization](/docs/image-optimization) helps you achieve faster page loads by reducing the size of images and using modern image

When deploying to Vercel, images are automatically optimized on demand, keeping your build times fast while improving your page load perf

When self-hosting, Image Optimization uses the default Next.js server for optimization. This server manages the rendering of pages and se

To use Image Optimization with Next.js on Vercel, import the `next/image` component into the component you'd like to add an image to, as

```js filename="components/example-component.jsx" framework=nextjs
import Image from 'next/image';

const ExampleComponent = (props) => {
  return (
    <>
      <Image src="example.png" alt="Example picture" width={500} height={500} />
      <span>{props.name}</span>
    </>
  );
};
```

```ts filename="components/example-component.tsx" framework=nextjs
import Image from 'next/image';

interface ExampleProps {
  name: string;
}

const ExampleComponent = ({ name }: ExampleProps) => {
  return (
    <>
      <Image
        src="example.png"
        alt="Example picture"
        width={500}
        height={500}
      />
      <span>{name}</span>
    </>
  );
};

export default ExampleComponent;
```

```js filename="components/example-component.jsx" framework=nextjs-app
import Image from 'next/image';
```

```
const ExampleComponent = (props) => {
  return (
    <>
      <Image src="example.png" alt="Example picture" width={500} height={500} />
      <span>{props.name}</span>
    </>
  );
};
```

```ts filename="components/ExampleComponent.tsx" framework=nextjs-app
import Image from 'next/image';

interface ExampleProps {
  name: string;
}

const ExampleComponent = ({ name }: ExampleProps) => {
  return (
    <>
      <Image
        src="example.png"
        alt="Example picture"
        width={500}
        height={500}
      />
      <span>{name}</span>
    </>
  );
};

export default ExampleComponent;
```

**To summarize, using Image Optimization with Next.js on Vercel:**

- Zero-configuration Image Optimization when using `next/image`
- Helps your team ensure great performance by default
- Keeps your builds fast by optimizing images on-demand
- Requires No additional services needed to procure or set up

[Learn more about Image Optimization](/docs/image-optimization)

## Font Optimization

[`next/font`](https://nextjs.org/docs/app/building-your-application/optimizing/fonts) enables built-in automatic self-hosting for any fon

This also allows you to use all [Google Fonts](https://fonts.google.com/) with performance and privacy in mind. CSS and font files are do

```js filename="pages/_app.jsx" framework=nextjs
import { Inter } from 'next/font/google';

// If loading a variable font, you don't need to specify the font weight
const inter = Inter({ subsets: ['latin'] });

export default function MyApp({ Component, pageProps }) {
  return (
    <main className={inter.className}>
      <Component {...pageProps} />
    </main>
  );
}
```

```ts filename="pages/_app.tsx" framework=nextjs
import { Inter } from 'next/font/google';
import type { AppProps } from 'next/app';

// If loading a variable font, you don't need to specify the font weight
const inter = Inter({ subsets: ['latin'] });

export default function MyApp({ Component, pageProps }: AppProps) {
  return (
    <main className={inter.className}>
      <Component {...pageProps} />
    </main>
  );
}
```

```js filename="app/layout.jsx" framework=nextjs-app
import { Inter } from 'next/font/google';

// If loading a variable font, you don't need to specify the font weight
const inter = Inter({
  subsets: ['latin'],
  display: 'swap',
});

export default function RootLayout({ children }) {
  return (
    <html lang="en" className={inter.className}>
      <body>{children}</body>
    </html>
  );
}
```

```ts filename="app/layout.tsx" framework=nextjs-app
```

```
import { Inter } from 'next/font/google';

// If loading a variable font, you don't need to specify the font weight
const inter = Inter({
  subsets: ['latin'],
  display: 'swap',
});

export default function RootLayout({
  children,
}: {
  children: React.ReactNode;
}) {
  return (
    <html lang="en" className={inter.className}>
      <body>{children}</body>
    </html>
  );
}
```

**To summarize, using Font Optimization with Next.js on Vercel:**

- Enables built-in, automatic self-hosting for font files
- Loads web fonts with zero layout shift
- Allows for CSS and font files to be downloaded at build time and self-hosted with the rest of your static files
- Ensures that no requests are sent to Google by the browser

[Learn more about Font Optimization](https://nextjs.org/docs/app/building-your-application/optimizing/fonts)

## Open Graph Images

Dynamic social card images (using the [Open Graph protocol](/docs/og-image-generation "The Open Graph Protocol")) allow you to create a u

The [Vercel OG](/docs/og-image-generation) image generation library allows you generate fast, dynamic social card images using Next.js AP

The following example demonstrates using OG image generation in both the Next.js Pages and App Router:

```ts filename="pages/api/og.tsx" framework=nextjs
import { ImageResponse } from '@vercel/og';

export default function () {
  return new ImageResponse(
    (
      <div
        style={{
          fontSize: 128,
          background: 'white',
          width: '100%',
          height: '100%',
          display: 'flex',
          textAlign: 'center',
          alignItems: 'center',
          justifyContent: 'center',
        }}
      >
        Hello world!
      </div>
    ),
    {
      width: 1200,
      height: 600,
    },
  );
}
```

```js filename="pages/api/og.jsx" framework=nextjs
import { ImageResponse } from '@vercel/og';

export default function () {
  return new ImageResponse(
    (
      <div
        style={{
          fontSize: 128,
          background: 'white',
          width: '100%',
          height: '100%',
          display: 'flex',
          textAlign: 'center',
          alignItems: 'center',
          justifyContent: 'center',
        }}
      >
        Hello world!
      </div>
    ),
    {
      width: 1200,
      height: 600,
    },
  );
}
```

```ts filename="app/api/og/route.tsx" framework=nextjs-app
import { ImageResponse } from 'next/og';
// App router includes @vercel/og.
// No need to install it.
```

```
export async function GET(request: Request) {
  return new ImageResponse(
    (
      <div
        style={{
          fontSize: 128,
          background: 'white',
          width: '100%',
          height: '100%',
          display: 'flex',
          textAlign: 'center',
          alignItems: 'center',
          justifyContent: 'center',
        }}
      >
        Hello world!
      </div>
    ),
    {
      width: 1200,
      height: 600,
    },
  );
}
```

```js filename="app/api/og/route.jsx" framework=nextjs-app
import { ImageResponse } from 'next/og';
// App router includes @vercel/og.
// No need to install it.

export async function GET(request) {
  return new ImageResponse(
    (
      <div
        style={{
          fontSize: 128,
          background: 'white',
          width: '100%',
          height: '100%',
          display: 'flex',
          textAlign: 'center',
          alignItems: 'center',
          justifyContent: 'center',
        }}
      >
        Hello world!
      </div>
    ),
    {
      width: 1200,
      height: 600,
    },
  );
}
```

To see your generated image, run `npm run dev` in your terminal and visit the `/api/og` route in your browser (most likely `http://localh

**To summarize, the benefits of using Vercel OG with Next.js include:**

- Instant, dynamic social card images without needing headless browsers
- Generated images are automatically cached on the Vercel CDN
- Image generation is co-located with the rest of your frontend codebase

[Learn more about OG Image Generation](/docs/og-image-generation)

## Middleware

[Middleware](/docs/routing-middleware) is code that executes before a request is processed. Because Middleware runs before the cache, it'

When deploying middleware with Next.js on Vercel, you get access to built-in helpers that expose each request's geolocation information. `

See [the Middleware API docs](/docs/routing-middleware/api) for more information.

**To summarize, Middleware with Next.js on Vercel:**

- Runs using [Middleware](/docs/routing-middleware) which are deployed globally
- Replaces needing additional services for customizable routing rules
- Helps you achieve the best performance for serving content globally

[Learn more about Middleware](/docs/routing-middleware)

## Draft Mode

[Draft Mode](/docs/draft-mode) enables you to view draft content from your [Headless CMS](/docs/solutions/cms) immediately, while still s

See [our Draft Mode docs](/docs/draft-mode#getting-started) to learn how to use it with Next.js.

### Self-hosting Draft Mode

When self-hosting, every request using Draft Mode hits the Next.js server, potentially incurring extra load or cost. Further, by spoofing

### Draft Mode security

Deployments on Vercel automatically secure Draft Mode behind the same authentication used for Preview Comments. In order to enable or dis

### Enabling Draft Mode in Preview Deployments

You and your team members can toggle Draft Mode in the Vercel Toolbar in [production](/docs/vercel-toolbar/in-production-and-localhost/ad

Users outside your Vercel team cannot toggle Draft Mode.

**To summarize, the benefits of using Draft Mode with Next.js on Vercel include:**

- Easily server-render previews of static pages
- Adds additional security measures to prevent malicious usage
- Integrates with any headless provider of your choice
- You can enable and disable Draft Mode in [the comments toolbar](/docs/comments/how-comments-work) on Preview Deployments

[Learn more about Draft Mode](/docs/draft-mode)

## Web Analytics

Vercel's Web Analytics features enable you to visualize and monitor your application's performance over time. The Analytics tab in your p

To use Web Analytics, navigate to the Analytics tab of your project dashboard on Vercel and select **Enable** in the modal that appears.

To track visitors and page views, we recommend first installing our `@vercel/analytics` package by running the terminal command below in

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i @vercel/analytics
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @vercel/analytics
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @vercel/analytics
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i @vercel/analytics
    ```
  </Code>
</CodeBlock>

Then, follow the instructions below to add the `Analytics` component to your app either using the `pages` directory or the `app` director

> For \['nextjs']:

The `Analytics` component is a wrapper around the tracking script, offering more seamless integration with Next.js, including route suppo

If you are using the `pages` directory, add the following code to your main app file:

```tsx {2, 8} filename="pages/_app.tsx" framework=nextjs
import type { AppProps } from 'next/app';
import { Analytics } from '@vercel/analytics/next';

function MyApp({ Component, pageProps }: AppProps) {
  return (
    <>
      <Component {...pageProps} />
      <Analytics />
    </>
  );
}

export default MyApp;
```

```jsx {1, 7} filename="pages/_app.js" framework=nextjs
import { Analytics } from '@vercel/analytics/next';

function MyApp({ Component, pageProps }) {
  return (
    <>
      <Component {...pageProps} />
      <Analytics />
    </>
  );
}

export default MyApp;
```

> For \['nextjs-app']:

The `Analytics` component is a wrapper around the tracking script, offering more seamless integration with Next.js, including route suppo

Add the following code to the root layout:

```tsx {1, 15} filename="app/layout.tsx" framework=nextjs-app
import { Analytics } from '@vercel/analytics/next';

export default function RootLayout({
  children,
}: {
  children: React.ReactNode;
}) {
  return (
    <html lang="en">
      <head>
```

```
        <title>Next.js</title>
      </head>
      <body>
        {children}
        <Analytics />
      </body>
    </html>
  );
}
```

```jsx {1, 11} filename="app/layout.jsx" framework=nextjs-app
import { Analytics } from '@vercel/analytics/next';

export default function RootLayout({ children }) {
  return (
    <html lang="en">
      <head>
        <title>Next.js</title>
      </head>
      <body>
        {children}
        <Analytics />
      </body>
    </html>
  );
}
```

**To summarize, Web Analytics with Next.js on Vercel:**

- Enables you to track traffic and see your top-performing pages
- Offers you detailed breakdowns of visitor demographics, including their OS, browser, geolocation, and more

[Learn more about Web Analytics](/docs/analytics)

## Speed Insights

You can see data about your project's [Core Web Vitals](/docs/speed-insights/metrics#core-web-vitals-explained) performance in your dashb

On Vercel, you can track your Next.js app's Core Web Vitals in your project's dashboard.

### reportWebVitals

> For \['nextjs-app']:

If you're self-hosting your app, you can use the [`useWebVitals`](https://nextjs.org/docs/advanced-features/measuring-performance#build-y

```jsx filename="app/_components/web-vitals.jsx" framework=all
'use client';

import { useReportWebVitals } from 'next/web-vitals';

export function WebVitals() {
  useReportWebVitals((metric) => {
    console.log(metric);
  });
}
```

```tsx filename="app/_components/web-vitals.tsx" framework=all
'use client';

import { useReportWebVitals } from 'next/web-vitals';

export function WebVitals() {
  useReportWebVitals((metric) => {
    console.log(metric);
  });
}
```

You could then reference your custom `WebVitals` component like this:

```ts filename="app/layout.ts" framework=all
import { WebVitals } from './_components/web-vitals';

export default function Layout({ children }) {
  return (
    <html>
      <body>
        <WebVitals />
        {children}
      </body>
    </html>
  );
}
```

```js filename="app/layout.js" framework=all
import { WebVitals } from './_components/web-vitals';

export default function Layout({ children }) {
  return (
    <html>
      <body>
        <WebVitals />
        {children}
      </body>
    </html>
```

```
  );
}
```

> For \['nextjs']:

If you're self-hosting your app, you can use the [`reportWebVitals`](https://nextjs.org/docs/advanced-features/measuring-performance#buil

Then you must export a `reportWebVitals` function from your custom `app` component, as demonstrated below:

```js filename="pages/_app.js" framework=all
export function reportWebVitals(metric) {
  switch (metric.name) {
    case 'FCP':
      // handle FCP results
      break;
    case 'LCP':
      // handle LCP results
      break;
    case 'CLS':
      // handle CLS results
      break;
    case 'FID':
      // handle FID results
      break;
    case 'TTFB':
      // handle TTFB results
      break;
    case 'INP':
      // handle INP results (note: INP is still an experimental metric)
      break;
    default:
      break;
  }
}

function MyApp({ Component, pageProps }) {
  return <Component {...pageProps} />;
}

export default MyApp;
```

```ts filename="pages/_app.ts" framework=all
export function reportWebVitals(metric) {
  switch (metric.name) {
    case 'FCP':
      // handle FCP results
      break;
    case 'LCP':
      // handle LCP results
      break;
    case 'CLS':
      // handle CLS results
      break;
    case 'FID':
      // handle FID results
      break;
    case 'TTFB':
      // handle TTFB results
      break;
    case 'INP':
      // handle INP results (note: INP is still an experimental metric)
      break;
    default:
      break;
  }
}

function MyApp({ Component, pageProps }) {
  return <Component {...pageProps} />;
}

export default MyApp;
```

Next.js uses [Google's `web-vitals` library](https://github.com/GoogleChrome/web-vitals#web-vitals) to measure the Web Vitals metrics ava

**To summarize, tracking Web Vitals with Next.js on Vercel:**

- Enables you to track traffic performance metrics, such as [First Contentful Paint](/docs/speed-insights/metrics#first-contentful-paint-
- Enables you to view performance analytics by page name and URL for more granular analysis
- Shows you [a score for your app's performance](/docs/speed-insights/metrics#how-the-scores-are-determined) on each recorded metric, whi

[Learn more about Speed Insights](/docs/speed-insights)

## Service integrations

Vercel has partnered with popular service providers, such as MongoDB and Sanity, to create integrations that make using those services wi

**To summarize, Integrations on Vercel:**

- Simplify the process of connecting your preferred services to a Vercel project
- Help you achieve the optimal setup for a Vercel project using your preferred service
- Configure your environment variables for you

[Learn more about Integrations](/integrations)

## More benefits

See [our Frameworks documentation page](/docs/frameworks) to learn about the benefits available to **all** frameworks when you deploy on '

## More resources

Learn more about deploying Next.js projects on Vercel with the following resources:

- [Build a fullstack Next.js app](/kb/guide/nextjs-prisma-postgres)
- [Build a multi-tenant app](/docs/multi-tenant)
- [Next.js with Contenful](/kb/guide/integrating-next-js-and-contentful-for-your-headless-cms)
- [Next.js with Stripe Checkout and Typescript](/kb/guide/getting-started-with-nextjs-typescript-stripe)
- [Next.js with Magic.link](/kb/guide/add-auth-to-nextjs-with-magic)
- [Generate a sitemap with Next.js](/kb/guide/how-do-i-generate-a-sitemap-for-my-nextjs-app-on-vercel)
- [Next.js ecommerce with Shopify](/kb/guide/deploying-locally-built-nextjs)
- [Deploy a locally built Next.js app](/kb/guide/deploying-locally-built-nextjs)
- [Deploying Next.js to Vercel](https://www.youtube.com/watch?v=AiiGjB2AxqA)
- [Learn about combining static and dynamic rendering on the same page in Next.js 14](https://www.youtube.com/watch?v=wv7w_Zx-FMU)
- [Learn about suspense boundaries and streaming when loading your UI](https://nextjs.org/docs/app/building-your-application/routing/load

---
title: "Nuxt on Vercel"
description: "Learn how to use Vercel"
last_updated: "2026-01-16T02:19:31.285Z"
source: "https://vercel.com/docs/frameworks/full-stack/nuxt"
---

# Nuxt on Vercel

Nuxt is an open-source framework that streamlines the process of creating modern Vue apps. It offers server-side rendering, SEO features,

You can deploy Nuxt static and server-side rendered sites on Vercel with no configuration required.

## Getting started

### Choosing a build command

The following table outlines the differences between `nuxt build` and `nuxt generate` on Vercel:

| Feature | `nuxt build` | `nuxt generate` |
| --- | --- | --- |
| Default build command | Yes | No |
| Supports all Vercel features out of the box | Yes | Yes |
| [Supports SSR](#server-side-rendering-ssr) | Yes | No |
| [Supports SSG](#static-rendering) | Yes, [with nuxt config](#static-rendering) | Yes |
| [Supports ISR](#incremental-static-regeneration-isr) | Yes | No |

In general, `nuxt build` is likely best for most use cases. Consider using `nuxt generate` to build [fully static sites](#static-renderin

## Editing your Nuxt config

You can configure your Nuxt deployment by creating a Nuxt config file in your project's root directory. It can be a TypeScript, JavaScrip

Your Nuxt config file should default export `defineNuxtConfig` by default, which you can add an options object to.

The following is an example of a Nuxt config file with no options defined:

```ts filename="nuxt.config.ts" framework=all
export default defineNuxtConfig({
  // Config options here
});
```

```js filename="nuxt.config.js" framework=all
export default defineNuxtConfig({
  // Config options here
});
```

[See the Nuxt Configuration Reference docs for a list of available options](https://nuxt.com/docs/api/configuration/nuxt-config/#nuxt-con

### Using `routeRules`

With the `routeRules` config option, you can:

- Create redirects
- Modify a route's response headers
- Enable ISR
- Deploy specific routes statically
- Deploy specific routes with SSR
- and more

> **💡 Note:** At the moment, there is no way to configure route deployment options within
> your page components, but development of this feature is in progress.

The following is an example of a Nuxt config that:

- Creates a redirect
- Modifies a route's response headers
- Opts a set of routes into client-side rendering

```js filename="nuxt.config.js" framework=all
export default defineNuxtConfig({
  routeRules: {
    '/examples/*': { redirect: '/redirect-route' },
    '/modify-headers-route': { headers: { 'x-magic-of': 'nuxt and vercel' } },
    // Enables client-side rendering
    '/spa': { ssr: false },
  },
});
```

```ts filename="nuxt.config.ts" framework=all
export default defineNuxtConfig({
  routeRules: {
    '/examples/*': { redirect: '/redirect-route' },
    '/modify-headers-route': { headers: { 'x-magic-of': 'nuxt and vercel' } },
    // Enables client-side rendering
    '/spa': { ssr: false },
  },
});
```

To learn more about `routeRules`:

- [Read Nuxt's reference docs to learn more about the available route options](https://nuxt.com/docs/guide/concepts/rendering#route-rules
- [Read the Nitro Engine's Cache API docs to learn about cacheing individual routes](https://nitro.unjs.io/guide/cache)

## Vercel Functions

[Vercel Functions](/docs/functions) enable developers to write functions that uses resources that scale up and down based on traffic dema

Nuxt deploys routes defined in `/server/api`, `/server/routes`, and `/server/middleware` as one server-rendered Function by default. Nuxt

The following is an example of a basic API Route in Nuxt:

```ts filename="server/api/hello.ts" framework=all
export default defineEventHandler(() => 'Hello World!');
```

```js filename="server/api/hello.js" framework=all
export default defineEventHandler(() => 'Hello World!');
```

You can test your API Routes with `nuxt dev`.

## Reading and writing files

You can read and write server files with Nuxt on Vercel. One way to do this is by using Nitro with Vercel Functions and a Redis driver su

To access server assets, you can use Nitro's [storage API](https://nitro.unjs.io/guide/storage):

```ts filename="server/api/storage.ts" framework=all
export default defineEventHandler(async () => {
  // https://nitro.unjs.io/guide/assets#server-assets
  const assets = useStorage('assets:server');
  const users = await assets.getItem('users.json');
  return {
    users,
  };
});
```

```js filename="server/api/storage.js" framework=all
export default defineEventHandler(async () => {
  // https://nitro.unjs.io/guide/assets#server-assets
  const assets = useStorage('assets:server');
  const users = await assets.getItem('users.json');
  return {
    users,
  };
});
```

To write files, mount [Redis storage](https://nitro.unjs.io/guide/storage) with a Redis driver such as the [Upstash Redis driver](https:/

First, [install Upstash Redis from the Vercel Marketplace](https://vercel.com/marketplace/upstash) to get your Redis credentials.

Then update your  file:

```ts filename="nuxt.config.ts" framework=all
export default defineNuxtConfig({
  $production: {
    nitro: {
      storage: {
        data: { driver: 'upstash' },
      },
    },
  },
});
```

```js filename="nuxt.config.js" framework=all
export default defineNuxtConfig({
  $production: {
    nitro: {
      storage: {
        data: { driver: 'upstash' },
      },
    },
  },
});
```

Use with the storage API.

```ts filename="server/api/storage.ts" framework=all
export default defineEventHandler(async (event) => {
  const dataStorage = useStorage('data');
  await dataStorage.setItem('hello', 'world');
  return {
```

```
    hello: await dataStorage.getItem('hello'),
  };
});
```

```js filename="server/api/storage.js" framework=all
export default defineEventHandler(async (event) => {
  const dataStorage = useStorage('data');
  await dataStorage.setItem('hello', 'world');
  return {
    hello: await dataStorage.getItem('hello'),
  };
});
```

[See an example code repository](https://github.com/pi0/nuxt-server-assets/tree/main).

## Middleware

Middleware is code that executes before a request gets processed. Because Middleware runs before the cache, it's an effective way of prov

Nuxt has two forms of Middleware:

- [Server middleware](#nuxt-server-middleware-on-vercel)
- [Route middleware](#nuxt-route-middleware-on-vercel)

### Nuxt server middleware on Vercel

In Nuxt, modules defined in `/server/middleware` will get deployed as [server middleware](https://nuxt.com/docs/guide/directory-structure

Server middleware is best used to read data from or add data to a request's `context`. Doing so allows you to handle authentication or ch

The following example demonstrates Middleware that:

- Checks for a cookie
- Tries to fetch user data from a database based on the request
- Adds the user's data and the cookie data to the request's context

```ts filename="server/middleware/auth.ts" framework=all
import { getUserFromDBbyCookie } from 'some-orm-package';

export default defineEventHandler(async (event) => {
  // The getCookie method is available to all
  // Nuxt routes by default. No need to import.
  const token = getCookie(event, 'session_token');

  // getUserFromDBbyCookie is a placeholder
  // made up for this example. You can fetch
  // data from wherever you want here
  const { user } = await getUserFromDBbyCookie(event.request);

  if (user) {
    event.context.user = user;
    event.context.session_token = token;
  }
});
```

```js filename="server/middleware/auth.js" framework=all
import { getUserFromDBbyCookie } from 'some-orm-package';

export default defineEventHandler(async (event) => {
  // The getCookie method is available to all
  // Nuxt routes by default. No need to import.
  const token = getCookie(event, 'session_token');

  // getUserFromDBbyCookie is a placeholder
  // made up for this example. You can fetch
  // data from wherever you want here
  const { user } = await getUserFromDBbyCookie(event.request, event.response);

  if (user) {
    event.context.user = user;
    event.context.session_token = token;
  }
});
```

You could then access that data in a page on the frontend with the [`useRequestEvent`](https://nuxt.com/docs/api/composables/use-request-

The following example demonstrates a page fetching data with `useRequestEvent`:

```tsx filename="example.vue" framework=all
<script>
  const event = useRequestEvent();
  const user = ref(event.context?.user);
</script>

<template>
    <div v-if="user">
      <h1>Hello, {{ user.name }}!</h1>
    </div>
    <div v-else>
      <p>Authentication failed!</p>
    </div>
</template>
```

```js filename="example.vue" framework=all
<script>
```

```
  const event = useRequestEvent();
  const user = ref(event.context?.user);
</script>

<template>
    <div v-if="user">
      <h1>Hello, {{ user.name }}!</h1>
    </div>
    <div v-else>
      <p>Authentication failed!</p>
    </div>
</template>
```

### Nuxt route middleware on Vercel

Nuxt's route middleware runs before navigating to a particular route. While server middleware runs in Nuxt's [Nitro engine](https://nitro

Route middleware is best used when you want to do things that server middleware can't, such as redirecting users, or preventing them from

The following example demonstrates route middleware that redirects users to a secret route:

```ts filename="middleware/redirect.ts" framework=all
export default defineNuxtRouteMiddleware((to) => {
  console.log(
    `Heading to ${to.path} - but I think we should go somewhere else...`,
  );

  return navigateTo('/secret');
});
```

```js filename="middleware/redirect.js" framework=all
export default defineNuxtRouteMiddleware((to) => {
  console.log(
    `Heading to ${to.path} - but I think we should go somewhere else...`,
  );

  return navigateTo('/secret');
});
```

By default, route middleware code will only run on pages that specify them. To do so, within the `<script>` tag for a page, you must call

The following example demonstrates a page that runs the above redirect middleware:

```tsx filename="redirect.vue" framework=all
<script>
definePageMeta({
  middleware: 'redirect'
})
</script>

<template>
  <div>
    You should never see this page
  </div>
</template>
```

```jsx filename="redirect.vue" framework=all
<script>
definePageMeta({
  middleware: 'redirect'
})
</script>

<template>
  <div>
    You should never see this page
  </div>
</template>
```

To make a middleware global, add the `.global` suffix before the file extension. The following is an example of a basic global middleware

```ts filename="example-middleware.global.ts" framework=all
export default defineNuxtRouteMiddleware(() => {
  console.log('running global middleware');
});
```

```js filename="example-middleware.global.js" framework=all
export default defineNuxtRouteMiddleware(() => {
  console.log('running global middleware');
});
```

[See a detailed example of route middleware in Nuxt's Middleware example docs](https://nuxt.com/docs/examples/routing/middleware).

**Middleware with Nuxt on Vercel enables you to:**

- Redirect users, and prevent navigation to routes
- Run authentication checks on the server, and pass results to the frontend
- Scope middleware to specific routes, or run it on all routes

[Learn more about Middleware](https://nuxt.com/docs/guide/directory-structure/middleware)

## Server-Side Rendering (SSR)

Server-Side Rendering (SSR) allows you to render pages dynamically on the server. This is useful for pages where the rendered data needs

Nuxt allows you to deploy your projects with a strategy called [Universal Rendering](https://nuxt.com/docs/guide/concepts/rendering#unive

When you deploy your app with Universal Rendering, it renders on the server once, then your client-side JavaScript code gets interpreted

On Vercel, Nuxt apps are server-rendered by default

**SSR with Nuxt on Vercel:**

- Scales to zero when not in use
- Scales automatically with traffic increases
- Allows you to opt individual routes out of SSR [with your Nuxt config](https://nuxt.com/docs/getting-started/deployment#client-side-onl

[Learn more about SSR](https://nuxt.com/docs/guide/concepts/rendering#universal-rendering)

## Client-side rendering

If you deploy with `nuxt build`, you can opt nuxt routes into client-side rendering using `routeRules` by setting `ssr: false` as demonst

```ts filename="nuxt.config.ts" framework=all
export default defineNuxtConfig({
  routeRules: {
    // Use client-side rendering for this route
    '/client-side-route-example': { ssr: false },
  },
});
```

```js filename="nuxt.config.js" framework=all
export default defineNuxtConfig({
  routeRules: {
    // Use client-side rendering for this route
    '/client-side-route-example': { ssr: false },
  },
});
```

## Static rendering

To deploy a fully static site on Vercel, build your project with `nuxt generate`.

Alternatively, you can statically generate some Nuxt routes at build time using the `prerender` route rule in your :

```ts filename="nuxt.config.ts" framework=all
export default defineNuxtConfig({
  routeRules: {
    // prerender index route by default
    '/': { prerender: true },
    // prerender this route and all child routes
    '/prerender-multiple/**': { prerender: true },
  },
});
```

```js filename="nuxt.config.js" framework=all
export default defineNuxtConfig({
  routeRules: {
    // prerender index route by default
    '/': { prerender: true },
    // prerender this route and all child routes
    '/prerender-multiple/**': { prerender: true },
  },
});
```

> **💡 Note:** To verify that a route is prerendered at build time, check
> .

## Incremental Static Regeneration (ISR)

[Incremental Static Regeneration (ISR)](/docs/incremental-static-regeneration) allows you to create or update content *without* redeployi

To enable ISR in a Nuxt route, add a `routeRules` option to your , as shown in the example below:

```ts filename="nuxt.config.ts" framework=all
export default defineNuxtConfig({
  routeRules: {
    // all routes (by default) will be revalidated every 60 seconds, in the background
    '/**': { isr: 60 },
    // this page will be generated on demand and then cached permanently
    '/static': { isr: true },
    // this page is statically generated at build time and cached permanently
    '/prerendered': { prerender: true },
    // this page will be always fresh
    '/dynamic': { isr: false },
  },
});
```

```js filename="nuxt.config.js" framework=all
export default defineNuxtConfig({
  routeRules: {
    // all routes (by default) will be revalidated every 60 seconds, in the background
    '/**': { isr: 60 },
    // this page will be generated on demand and then cached permanently
    '/static': { isr: true },
    // this page is statically generated at build time and cached permanently
    '/prerendered': { prerender: true },
    // this page will be always fresh
```

```
    '/dynamic': { isr: false },
  },
});
```

You should use the `isr` option rather than `swr` to enable ISR in a route. The `isr` option enables Nuxt to use Vercel's Cache.

**using ISR with Nuxt on Vercel offers:**

- Better performance with our global [CDN](/docs/cdn)
- Zero-downtime rollouts to previously statically generated pages
- Global content updates in 300ms
- Generated pages are both cached and persisted to durable storage

[Learn more about ISR with Nuxt](https://nuxt.com/docs/guide/concepts/rendering#hybrid-rendering).

## Redirects and Headers

You can define redirects and response headers with Nuxt on Vercel in your :

```js filename="nuxt.config.js" framework=all
export default defineNuxtConfig({
  routeRules: {
    '/examples/*': { redirect: '/redirect-route' },
    '/modify-headers-route': { headers: { 'x-magic-of': 'nuxt and vercel' } },
  },
});
```

```ts filename="nuxt.config.ts" framework=all
export default defineNuxtConfig({
  routeRules: {
    '/examples/*': { redirect: '/redirect-route' },
    '/modify-headers-route': { headers: { 'x-magic-of': 'nuxt and vercel' } },
  },
});
```

## Image Optimization

[Image Optimization](/docs/image-optimization) helps you achieve faster page loads by reducing the size of images and using modern image

When deploying to Vercel, images are automatically optimized on demand, keeping your build times fast while improving your page load perf

To use Image Optimization with Nuxt on Vercel, follow [the Image Optimization quickstart](/docs/image-optimization/quickstart) by selecti

**Using Image Optimization with Nuxt on Vercel:**

- Requires zero-configuration for Image Optimization when using `@nuxt/image`
- Helps your team ensure great performance by default
- Keeps your builds fast by optimizing images on-demand

[Learn more about Image Optimization](/docs/image-optimization)

## Open Graph Images

Dynamic social card images allow you to create a unique image for pages of your site. This is great for sharing links on the web through

To generate dynamic social card images for Nuxt projects, you can use [`nuxt-og-image`](https://nuxtseo.com/og-image/getting-started/inst

The following example demonstrates using Open Graph (OG) image generation with [`nuxt-og-image`](https://nuxtseo.com/og-image/getting-sta

1. Create a new OG template

```ts filename="components/OgImage/Template.vue" framework=all
<script setup lang="ts">
  withDefaults(defineProps<{
    title?: string
  }>(), {
    title: 'title',
  })
</script>
<template>
  <div class="h-full w-full flex items-start justify-start border border-blue-500 border-12 bg-gray-50">
    <div class="flex items-start justify-start h-full">
      <div class="flex flex-col justify-between w-full h-full">
        <h1 class="text-[80px] p-20 font-black text-left">
          {{ title }}
        </h1>
        <p class="text-2xl pb-10 px-20 font-bold mb-0">
          acme.com
        </p>
      </div>
    </div>
  </div>
</template>
```

```js filename="components/OgImage/BlogPost.vue" framework=all
<script setup lang="js">
  withDefaults(defineProps(), {
    title: 'title',
  })
</script>
<template>
  <div class="h-full w-full flex items-start justify-start border border-blue-500 border-12 bg-gray-50">
    <div class="flex items-start justify-start h-full">
      <div class="flex flex-col justify-between w-full h-full">
        <h1 class="text-[80px] p-20 font-black text-left">
          {{ title }}
```

```
        </h1>
        <p class="text-2xl pb-10 px-20 font-bold mb-0">
          acme.com
        </p>
      </div>
    </div>
  </div>
</template>
```

2. Use that OG image in your pages. Props passed get used in your open graph images.

```ts filename="pages/index.vue" framework=all
<script lang="ts" setup>
defineOgImageComponent('Template', {
  title: 'Is this thing on?'
})
</script>
```

```js filename="pages/index.vue" framework=all
<script lang="js" setup>
defineOgImageComponent('Template', {
  title: 'Is this thing on?'
})
</script>
```

To see your generated image, run your project and use Nuxt DevTools. Or you can visit the image at its URL `/__og-image__/image/og.png`.

[Learn more about OG Image Generation with Nuxt](https://nuxtseo.com/og-image/getting-started/installation).

## Deploying legacy Nuxt projects on Vercel

The Nuxt team [does not recommend deploying legacy versions of Nuxt (such as Nuxt 2) on Vercel](https://github.com/nuxt/vercel-builder#re

- Implement [Nuxt Bridge](https://github.com/nuxt/bridge#readme)
- Or [upgrade with the Nuxt team's migration guide](https://nuxt.com/docs/migration/overview)

If you still want to use legacy Nuxt versions with Vercel, you should only do so by building a static site with `nuxt generate`. **We do

## More benefits

See [our Frameworks documentation page](/docs/frameworks) to learn about the benefits available to **all** frameworks when you deploy on '

## More resources

Learn more about deploying Nuxt projects on Vercel with the following resources:

- [Deploy our Nuxt Alpine template](/templates/nuxt/alpine)
- [See an example of Nuxt Image](/docs/image-optimization/quickstart)


---
title: "Full-stack frameworks on Vercel"
description: "Vercel supports a wide range of the most popular backend frameworks, optimizing how your application builds and runs no mat
last_updated: "2026-01-16T02:19:31.255Z"
source: "https://vercel.com/docs/frameworks/full-stack"
---

# Full-stack frameworks on Vercel

The following full-stack frameworks are supported with zero-configuration.

- **Next.js**: Next.js makes you productive with React instantly — whether you want to build static or dynamic sites.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/nextjs) | [View Demo](https:
- **Nuxt**: Nuxt is the open source framework that makes full-stack development with Vue.js intuitive.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/nuxtjs) | [View Demo](https:
- **RedwoodJS**: RedwoodJS is a full-stack framework for the Jamstack.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/redwoodjs) | [View Demo](htt
- **Remix**: Build Better Websites
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/remix) | [View Demo](https:/
- **SvelteKit**: SvelteKit is a framework for building web applications of all sizes.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/sveltekit-1) | [View Demo](h
- **TanStack Start**: Full-stack Framework powered by TanStack Router for React and Solid.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/tanstack-start)


## Frameworks infrastructure support matrix

The following table shows which features are supported by each framework on Vercel. The framework list is not exhaustive, but a represent

We're committed to having support for all Vercel features across frameworks, and continue to work with framework authors on adding suppor

**Legend:** ✓ Supported | ✗ Not Supported | N/A Not Applicable

| Feature | Next.js | SvelteKit | Nuxt | TanStack | Astro | Remix | Vite | CRA |
|---|---|---|---|---|---|---|---|---|
| [Static Assets](/docs/cdn) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| [Edge Routing Rules](/docs/cdn#features) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| [Routing Middleware](/docs/routing-middleware) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| [Server-Side Rendering](/docs/functions) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | N/A | N/A |
| [Streaming SSR](/docs/functions/streaming-functions) | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | N/A | N/A |
| [Incremental Static Regeneration](/docs/incremental-static-regeneration) | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | N/A | N/A |
| [Image Optimization](/docs/image-optimization) | ✓ | ✓ | ✓ | N/A | ✓ | ✗ | N/A | N/A |
| [Data Cache](/docs/data-cache) | ✓ | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| [Native OG Image Generation](/docs/og-image-generation) | ✓ | N/A | ✓ | N/A | N/A | N/A | N/A | N/A |
| [Multi-runtime support (different routes)](/docs/functions/runtimes) | ✓ | ✓ | ✓ | N/A | ✗ | ✓ | N/A | N/A |
| [Multi-runtime support (entire app)](/docs/functions/runtimes) | ✓ | ✓ | ✓ | N/A | ✓ | ✓ | N/A | N/A |
| [Output File Tracing](/kb/guide/how-can-i-use-files-in-serverless-functions) | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | N/A | N/A |

| [Skew Protection](/docs/skew-protection) | ✓ | ✓ | ✗ | N/A | ✓ | ✗ | N/A | N/A |
| [Framework Routing Middleware](/docs/routing-middleware) | ✓ | N/A | ✗ | ✓ | ✓ | ✗ | N/A | N/A |


---
title: "Remix on Vercel"
description: "Learn how to use Vercel"
last_updated: "2026-01-16T02:19:31.201Z"
source: "https://vercel.com/docs/frameworks/full-stack/remix"
---

# Remix on Vercel

Remix is a fullstack, [server-rendered](#server-side-rendering-ssr) React framework. Its built-in features for nested pages, error bounda
> in conjunction with the [Vercel Preset](#vercel-vite-preset), when deploying
> to Vercel.## Getting started## `@vercel/remix`The [`@vercel/remix`](https://www.npmjs.com/package/@vercel/remix) package exposes useful
* [`defer`](https://remix.run/docs/en/main/utils/defer)
* [`createCookie`](https://remix.run/docs/en/main/utils/cookies#createcookie)To best experience Vercel features such as [streaming](#resp
* [`@remix-run/cloudflare`](https://www.npmjs.com/package/@remix-run/cloudflare)
* [`@remix-run/server-runtime`](https://www.npmjs.com/package/@remix-run/server-runtime)To get started, navigate to the root directory of
  <Code tab="pnpm">
    ```bash
    pnpm i @vercel/remix
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @vercel/remix
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @vercel/remix
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i @vercel/remix
    ```
  </Code>
</CodeBlock>## Vercel Vite PresetWhen using the [Remix Vite plugin](https://remix.run/docs/en/main/future/vite) (highly recommended), you
```
import { vitePlugin as remix } from '@remix-run/dev';
import { installGlobals } from '@remix-run/node';
import { defineConfig } from 'vite';
import tsconfigPaths from 'vite-tsconfig-paths';
import { vercelPreset } from '@vercel/remix/vite';

installGlobals();

export default defineConfig({
  plugins: [
    remix({
      presets: [vercelPreset()],
    }),
    tsconfigPaths(),
  ],
});
```Using this Preset enables Vercel-specific functionality such as rendering your Remix application with Vercel Functions.## Server-Side
```
export default function IndexRoute() {
  return (
    <div style={{ fontFamily: 'system-ui, sans-serif', lineHeight: '1.4' }}>
      <h1>This route is rendered on the server</h1>
    </div>
  );
}
```jsx filename="/app/routes/_index.jsx" framework=all
export default function IndexRoute() {
  return (
    <div style={{ fontFamily: 'system-ui, sans-serif', lineHeight: '1.4' }}>
      <h1>This route is rendered on the server</h1>
    </div>
  );
}
```### Vercel FunctionsVercel Functions execute using Node.js. They enable developers to write functions that use resources that scale up
```
export default function Serverless() {
  return <h1>Welcome to Remix with Vercel</h1>;
}
```jsx filename="/app/routes/serverless-example.jsx" framework=all
export default function Serverless() {
  return <h1>Welcome to Remix with Vercel</h1>;
}
```**To summarize, Server-Side Rendering (SSR) with Remix on Vercel:*** Scales to zero when not in use
* Scales automatically with traffic increases
* Has framework-aware infrastructure to generate Vercel Functions## Response streaming[Streaming HTTP responses](/docs/functions/streamin
[Streaming](https://remix.run/docs/en/main/guides/streaming) page in the Remix
docs for general instructions.The following example demonstrates a route that simulates a throttled network by delaying a promise's resul
```
import { Suspense } from 'react';
import { Await, useLoaderData } from '@remix-run/react';
import { defer } from '@vercel/remix';

function sleep(ms: number) {
  return new Promise((resolve) => setTimeout(resolve, ms));
}

export async function loader({ request }) {
  const version = process.versions.node;

  return defer({
    // Don't let the promise resolve for 1 second
    version: sleep(1000).then(() => version),
```

```
  });
}

export default function DeferredRoute() {
  const { version } = useLoaderData();

  return (
    <Suspense fallback={'Loading…'}>
      <Await resolve={version}>{(version) => <strong>{version}</strong>}</Await>
    </Suspense>
  );
}
``````jsx filename="/app/routes/defer-route.jsx" framework=all
import { Suspense } from 'react';
import { Await, useLoaderData } from '@remix-run/react';
import { defer } from '@vercel/remix';

function sleep(ms) {
  return new Promise((resolve) => setTimeout(resolve, ms));
}

export async function loader({ request }) {
  const version = process.versions.node;

  return defer({
    // Don't let the promise resolve for 1 second
    version: sleep(1000).then(() => version),
  });
}

export default function DeferredRoute() {
  const { version } = useLoaderData();

  return (
    <Suspense fallback={'Loading…'}>
      <Await resolve={version}>{(version) => <strong>{version}</strong>}</Await>
    </Suspense>
  );
}
```**To summarize, Streaming with Remix on Vercel:*** Offers faster Function response times, improving your app's user experience
* Allows you to return large amounts of data without exceeding Vercel Function response size limits
* Allows you to display Instant Loading UI from the server with Remix's `defer()` and `Await`[Learn more about Streaming](/docs/functions
* For requests repeated after 1 second, but before 60 seconds have passed, return the cached content and mark it as stale. The stale cont
```jsx filename="/app/routes/example.jsx" framework=all
import type { HeadersFunction } from '@vercel/remix';

export const headers: HeadersFunction = () => ({
  'Cache-Control': 's-maxage=1, stale-while-revalidate=59',
});

export async function loader() {
  // Fetch data necessary to render content
}
``````jsx filename="/app/routes/example.jsx" framework=all
export const headers = () => ({
  'Cache-Control': 's-maxage=1, stale-while-revalidate=59',
});

export async function loader() {
  // Fetch data necessary to render content
}
```See [our docs on cache limits](/docs/cdn-cache#limits) to learn the max size and lifetime of caches stored on Vercel.**To summarize, u
* Allow you to serve content from the cache *while updating the cache in the background* with `stale-while-revalidate`[Learn more about c
  <Code tab="pnpm">
    ```bash
    pnpm i @vercel/analytics
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @vercel/analytics
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @vercel/analytics
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i @vercel/analytics
    ```
  </Code>
</CodeBlock>Then, follow the instructions below to add the `Analytics` component to your app. The `Analytics` component is a wrapper arou
```jsx filename="app/root.jsx" framework=all
import { Analytics } from '@vercel/analytics/react';

export default function App() {
  return (
    <html lang="en">
      <body>
        <Analytics />
      </body>
    </html>
  );
}
``````jsx filename="app/root.jsx" framework=all
import { Analytics } from '@vercel/analytics/react';

export default function App() {
  return (
    <html lang="en">
```
```

```
      <body>
        <Analytics />
      </body>
    </html>
  );
}
```**To summarize, Analytics with Remix on Vercel:*** Enables you to track traffic and see your top-performing pages
* Offers you detailed breakdowns of visitor demographics, including their OS, browser, geolocation and more[Learn more about Analytics](/
for streaming to work with Vercel Functions. This version will be used when
no `entry.server` file is found in the project, or when the existing `entry.server` file has
not been modified from the base Remix template.However, if your application requires a customized `app/entry.server.jsx` or `app/entry.se
file (for example, to wrap the `<RemixServer>` component with a React context), you should
base it off of this template:```tsx filename="/app/entry.server.tsx" framework=all
import { RemixServer } from '@remix-run/react';
import { handleRequest, type EntryContext } from '@vercel/remix';

export default async function (
  request: Request,
  responseStatusCode: number,
  responseHeaders: Headers,
  remixContext: EntryContext,
) {
  let remixServer = <RemixServer context={remixContext} url={request.url} />;
  return handleRequest(
    request,
    responseStatusCode,
    responseHeaders,
    remixServer,
  );
}
``````jsx filename="/app/entry.server.jsx" framework=all
import { RemixServer } from '@remix-run/react';
import { handleRequest } from '@vercel/remix';

export default async function (
  request,
  responseStatusCode,
  responseHeaders,
  remixContext,
) {
  let remixServer = <RemixServer context={remixContext} url={request.url} />;
  return handleRequest(
    request,
    responseStatusCode,
    responseHeaders,
    remixServer,
  );
}
```## Using a custom `server` file> **💡 Note:** Defining a custom `server` file is not supported when using the Remix Vite
> plugin on Vercel.It's usually not necessary to define a custom server.js file within your Remix application when deploying to Vercel. I
import { createRequestHandler } from '@vercel/remix/server';
import * as build from '@remix-run/dev/server-build';

export default createRequestHandler({
  build,
  mode: process.env.NODE_ENV,
  getLoadContext() {
    return {
      nodeLoadContext: true,
    };
  },
});
``````ts filename="server.ts" framework=all
import { createRequestHandler } from '@vercel/remix/server';
import * as build from '@remix-run/dev/server-build';

export default createRequestHandler({
  build,
  mode: process.env.NODE_ENV,
  getLoadContext() {
    return {
      nodeLoadContext: true,
    };
  },
});
```## More benefitsSee [our Frameworks documentation page](/docs/frameworks) to learn about the benefits available to **all** frameworks 
* [Deploy our Product Roadmap template](/templates/remix/roadmap-voting-app-rowy)
* [Explore the Remix docs](https://remix.run/docs/en/main)


---
title: "SvelteKit on Vercel"
description: "Learn how to use Vercel"
last_updated: "2026-01-16T02:19:31.349Z"
source: "https://vercel.com/docs/frameworks/full-stack/sveltekit"
---

# SvelteKit on Vercel

SvelteKit is a frontend framework that enables you to build Svelte applications with modern techniques, such as Server-Side Rendering, au

You can deploy your SvelteKit projects to Vercel with zero configuration, enabling you to use [Preview Deployments](/docs/deployments/env

## Get started with SvelteKit on Vercel

## Use Vercel features with Svelte

When you create a new SvelteKit project with `npm create svelte@latest`, it installs `adapter-auto` by default. This adapter detects that

We recommend installing the `@sveltejs/adapter-vercel` package yourself. Doing so will ensure version stability, slightly speed up your C

The following instructions will guide you through adding the Vercel adapter to your SvelteKit project.

- ### Install SvelteKit's Vercel adapter plugin
  You can add [the Vercel adapter](https://kit.svelte.dev/docs/adapter-vercel) to your SvelteKit project by running the following command
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i @sveltejs/adapter-vercel
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i @sveltejs/adapter-vercel
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i @sveltejs/adapter-vercel
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i @sveltejs/adapter-vercel
      ```
    </Code>
  </CodeBlock>

- ### Add the Vercel adapter to your Svelte config
  Add the Vercel adapter to your `svelte.config.js` file, [which should be at the root of your project directory](https://kit.svelte.dev/
  > **💡 Note:** You cannot use [TypeScript for your SvelteKit config
  > file](https://github.com/sveltejs/kit/issues/2576).
  In your `svelte.config.js` file, import `adapter` from `@sveltejs/adapter-vercel`, and add your preferred options. The following exampl
  ```js filename="svelte.config.js"
  import adapter from '@sveltejs/adapter-vercel';

  export default {
    kit: {
      adapter: adapter(),
    },
  };
  ```
  [Learn more about configuring your Vercel deployment in our configuration section below](#configure-your-sveltekit-deployment).

## Configure your SvelteKit deployment

You can configure how your SvelteKit project gets deployed to Vercel at the project-level and at the route-level.

Changes to the `config` object you define in `svelte.config.js` will affect the default settings for routes across your whole project. To

The following is an example of a `svelte.config.js` file that will deploy using server-side rendering in Vercel's Node.js serverless runt

```js filename="svelte.config.js"
import adapter from '@sveltejs/adapter-vercel';

/** @type {import('@sveltejs/kit').Config} */
const config = {
  kit: {
    adapter: adapter({
      runtime: 'nodejs20.x',
    }),
  },
};

export default config;
```

You can also configure how individual routes deploy by exporting a `config` object. The following is an example of a route that will depl

```js filename="+page.server.js" framework=all
export const config = {
  runtime: 'edge',
};

/** @type {import('./$types').PageServerLoad} */
export const load = ({ cookies }) => {
  // Load function code here
};
```

```ts filename="+page.server.ts" framework=all
import { PageServerLoad } from './$types';

export const config = {
  runtime: 'edge',
};

export const load = ({ cookies }): PageServerLoad<any> => {
  // Load function code here
};
```

[Learn about all the config options available in the SvelteKit docs](https://kit.svelte.dev/docs/adapter-vercel#deployment-configuration)

### Configuration options

SvelteKit's docs have [a comprehensive list of all config options available to you](https://kit.svelte.dev/docs/adapter-vercel#deployment

#### `split`

By default, your SvelteKit routes get bundled into one Function when you deploy your project to Vercel. This configuration typically redu

**In most cases, there is no need to modify this option**.

Setting `split: true` in your Svelte config will cause your SvelteKit project's routes to get split into separate Vercel Functions.

Splitting your Functions is not typically better than bundling them. You may want to consider setting `split: true` if you're experiencin

- **You have exceeded the Function size limit for the runtime you're using**. Batching too many routes into a single Function could cause
- **Your app is experiencing abnormally long cold start times**. Batching Vercel functions into one Function will reduce how often users

#### `regions`

Choosing a region allows you to reduce latency for requests to functions. If you choose a Function region geographically near dependencie

By default, your Vercel Functions will be deployed in *Washington, D.C., USA*, or `iad1`. Adding a region ID to the `regions` array will

## Streaming

Vercel supports streaming API responses over time with SvelteKit, allowing you to render parts of the UI early, then render the rest as d

- SvelteKit enables you to use a  file to fetch data on the server, which you can access from a `+page.svelte` file located in the same f
- You fetch data in a [`load`](https://kit.svelte.dev/docs/load) function defined in . This function returns an object
  - Top-level properties that return a promise will resolve before the page renders
  - Nested properties that return a promise [will stream](https://kit.svelte.dev/docs/load#streaming-with-promises)

The following example demonstrates a `load` function that will stream its response to the client. To simulate delayed data returned from

```ts filename="src/routes/streaming-example/+page.server.ts" framework=all
function sleep(value: any, ms: number) {
  // Use this sleep function to simulate
  // a delayed API response.
  return new Promise((fulfill) => {
    setTimeout(() => {
      fulfill(value);
    }, ms);
  });
}
export function load(event): PageServerLoad<any> {
  // Get some location data about the visitor
  const ip = event.getClientAddress();
  const city = decodeURIComponent(
    event.request.headers.get('x-vercel-ip-city') ?? 'unknown',
  );
  return {
    topLevelExample: sleep({ data: "This won't be streamed" }, 2000)
    // Stream the location data to the client
    locationData: {
      details: sleep({ ip, city }, 1000),
    },
  };
}
```

```js filename="src/routes/streaming-example/+page.server.js" framework=all
/**
 * @param {any} value
 * @param {number} ms
 */
function sleep(value, ms) {
  // Use this sleep function to simulate
  // a delayed API response.
  return new Promise((fulfill) => {
    setTimeout(() => {
      fulfill(value);
    }, ms);
  });
}
/** @type {import('./$types').PageServerLoad} */
export function load(event) {
  // Get some location data about the visitor
  const ip = event.getClientAddress();
  const city = decodeURIComponent(
    event.request.headers.get('x-vercel-ip-city') ?? 'unknown',
  );
  return {
    topLevelExample: sleep({ data: "This won't be streamed" }, 2000)
    // Stream the location data to the client
    locationData: {
      details: sleep({ ip, city }, 1000),
    },
  };
}
```

You could then display this data by creating the following `+page.svelte` file in the same directory:

```jsx filename="src/routes/streaming-example/+page.svelte" framework=all
<script lang="ts">
  import type { PageData } from './$types'
  export let data: PageData;
</script>

<h1><span>Hello!</span></h1>

<div class="info">
  {#await data.locationData.details}
    <p>streaming delayed data from the server...</p>
  {:then details}
    <div>
      <p>City is {details.city}</p>
```

```
      <p>And IP is: {details.ip} </p>
    </div>
  {/await}
</div>
```

```tsx filename="src/routes/streaming-example/+page.svelte" framework=all
<script lang="ts">
  import type { PageData } from './$types'
  export let data: PageData;
</script>

<h1><span>Hello!</span></h1>

<div class="info">
  {#await data.locationData.details}
    <p>streaming delayed data from the server...</p>
  {:then details}
    <div>
      <p>City is {details.city}</p>
      <p>And IP is: {details.ip} </p>
    </div>
  {/await}
</div>
```

**To summarize, Streaming with SvelteKit on Vercel:**

- Enables you to stream UI elements as data loads
- Supports streaming through Vercel Functions
- Improves perceived speed of your app

[Learn more about Streaming on Vercel](/docs/functions/streaming-functions).

## Server-Side Rendering

Server-Side Rendering (SSR) allows you to render pages dynamically on the server. This is useful for pages where the rendered data needs

Vercel offers SSR that scales down resource consumption when traffic is low, and scales up with traffic surges. This protects your site f

SvelteKit projects are server-side rendered by default. You can configure individual routes to prerender with the `prerender` page option

**While server-side rendered SvelteKit apps do support middleware, SvelteKit does not support URL rewrites from middleware**.

[See the SvelteKit docs on prerendering to learn more](https://kit.svelte.dev/docs/page-options#prerender).

**To summarize, SSR with SvelteKit on Vercel:**

- Scales to zero when not in use
- Scales automatically with traffic increases
- Has zero-configuration support for [`Cache-Control` headers](/docs/cdn-cache), including `stale-while-revalidate`

[Learn more about SSR](https://kit.svelte.dev/docs/page-options#ssr)

## Environment variables

Vercel provides a set of System Environment Variables that our platform automatically populates. For example, the `VERCEL_GIT_PROVIDER` v

These environment variables will be available to your project automatically, and you can enable or disable them in your project settings (

### Use Vercel environment variables with SvelteKit

SvelteKit allows you to import environment variables, but separates them into different modules based on whether they're dynamic or stati

[System Environment Variables](/docs/environment-variables/system-environment-variables) are private and you should never expose them to

The example below exposes `VERCEL_COMMIT_REF`, a variable that exposes the name of the branch associated with your project's deployment,

```js filename="+layout.server.js" framework=all
import { VERCEL_COMMIT_REF } from '$env/static/private';

/** @type {import('./$types').LayoutServerLoad} */
export function load() {
  return {
    deploymentGitBranch: VERCEL_COMMIT_REF,
  };
}
```

```ts filename="+layout.server.ts" framework=all
import { LayoutServerLoad } from './types';
import { VERCEL_COMMIT_REF } from '$env/static/private';

type DeploymentInfo = {
  deploymentGitBranch: string;
};

export function load(): LayoutServerLoad<DeploymentInfo> {
  return {
    deploymentGitBranch: 'Test',
  };
}
```

You could reference that variable in [a corresponding layout](https://kit.svelte.dev/docs/load#layout-data) as shown below:

```html filename="+layout.svelte"
<script>
  /** @type {import('./$types').LayoutData} */
  export let data;
```

````
</script>

<p>This staging environment was deployed from {data.deploymentGitBranch}.</p>
```

**To summarize, the benefits of using Environment Variables with SvelteKit on Vercel include:**

- Access to vercel deployment information, dynamically or statically, with our preconfigured System Environment Variables
- Access to automatically-configured environment variables provided by [integrations for your preferred services](/docs/environment-varia
- Searching and filtering environment variables by name and environment in Vercel's dashboard

[Learn more about Environment Variables](/docs/environment-variables)

## Incremental Static Regeneration (ISR)

Incremental Static Regeneration allows you to create or update content without redeploying your site. When you deploy a route with ISR, V

[See our ISR docs to learn more](/docs/incremental-static-regeneration).

To deploy a SvelteKit route with ISR:

- Export a `config` object with an `isr` property. Its value will be the number of seconds to wait before revalidating
- To enable on-demand revalidation, add the `bypassToken` property to the `config` object. Its value gets checked when `GET` or `HEAD` re

The following example demonstrates a SvelteKit route that Vercel will deploy with ISR, revalidating the page every 60 seconds, with on-de

```js filename="example-route/+page.server.js" framework=all
export const config = {
  isr: {
    expiration: 60,
    bypassToken: 'REPLACE_ME_WITH_SECRET_VALUE',
  },
};
```

```ts filename="example-route/+page.server.ts" framework=all
export const config = {
  isr: {
    expiration: 60,
    bypassToken: 'REPLACE_ME_WITH_SECRET_VALUE',
  },
};
```

[Learn more about ISR with SvelteKit](https://kit.svelte.dev/docs/adapter-vercel#incremental-static-regeneration).

**To summarize, the benefits of using ISR with SvelteKit on Vercel include:**

- Better performance with our global [CDN](/docs/cdn)
- Zero-downtime rollouts to previously statically generated pages
- Framework-aware infrastructure enables global content updates in 300ms
- Generated pages are both cached and persisted to durable storage

[Learn more about ISR](/docs/incremental-static-regeneration)

## Skew Protection

New project deployments can lead to **version skew**. This can happen when your users are using your app and a new version gets deployed.

SvelteKit has a skew protection solution. When it detects version skew, it triggers a hard reload of a page to sync to the latest version

[Learn more about skew protection with SvelteKit](https://kit.svelte.dev/docs/adapter-vercel#skew-protection).

**To summarize, the benefits of using ISR with SvelteKit on Vercel include:**

- Mitigates the risk of your active users encountering version skew
- Avoids hard reloads for current active users on your project

[Learn more about skew protection on Vercel](/docs/skew-protection).

## Image Optimization

[Image Optimization](/docs/image-optimization) helps you achieve faster page loads by reducing the size of images and using modern image

When deploying to Vercel, you can optimize your images on demand, keeping your build times fast while improving your page load performanc

To use Image Optimization with SvelteKit on Vercel, use the [`@sveltejs/adapter-vercel`](#use-vercel-features-with-svelte) within your  f

```js filename="svelte.config.js" framework=all
import adapter from '@sveltejs/adapter-vercel';

export default {
  kit: {
    adapter({
      images: {
        sizes: [640, 828, 1200, 1920, 3840],
        formats: ['image/avif', 'image/webp'],
        minimumCacheTTL: 300,
        domains: ['example-app.vercel.app'],
      }
    })
  }
};
```

```ts filename="svelte.config.ts" framework=all
import adapter from '@sveltejs/adapter-vercel';

export default {
  kit: {
````

```
    adapter({
      images: {
        sizes: [640, 828, 1200, 1920, 3840],
        formats: ['image/avif', 'image/webp'],
        minimumCacheTTL: 300,
        domains: ['example-app.vercel.app'],
      }
    })
  }
};
```

This allows you to specify [configuration options](https://vercel.com/docs/build-output-api/v3/configuration#images) for Vercel's native

To use image optimization with SvelteKit, you have to construct your own `srcset` URLs. You can create a library function that will optim

```js filename="src/lib/image.js" framework=all
import { dev } from '$app/environment';

export function optimize(src, widths = [640, 960, 1280], quality = 90) {
  if (dev) return src;

  return widths
    .slice()
    .sort((a, b) => a - b)
    .map((width, i) => {
      const url = `/_vercel/image?url=${encodeURIComponent(src)}&w=${width}&q=${quality}`;
      const descriptor = i < widths.length - 1 ? ` ${width}w` : '';
      return url + descriptor;
    })
    .join(', ');
}
```

```ts filename="src/lib/image.ts" framework=all
import { dev } from '$app/environment';

export function optimize(src: string, widths = [640, 960, 1280], quality = 90) {
  if (dev) return src;

  return widths
    .slice()
    .sort((a, b) => a - b)
    .map((width, i) => {
      const url = `/_vercel/image?url=${encodeURIComponent(src)}&w=${width}&q=${quality}`;
      const descriptor = i < widths.length - 1 ? ` ${width}w` : '';
      return url + descriptor;
    })
    .join(', ');
}
```

Use an `img` or any other image component with an optimized `srcset` generated by the `optimize` function:

```tsx filename="src/components/image.svelte" framework=all
<script lang="ts">
  import { optimize } from '$lib/image';
  import type { Photo } from '$lib/types';

  export let photo: Photo;
</script>

<img
  class="absolute left-0 top-0 w-full h-full"
  srcset={optimize(photo.url)}
  alt={photo.description}
/>
```

```jsx filename="src/components/image.svelte" framework=all
<script lang="js">
  import { optimize } from '$lib/image';

  export let photo;
</script>

<img
  class="absolute left-0 top-0 w-full h-full"
  srcset={optimize(photo.url)}
  alt={photo.description}
/>
```

**To summarize, using Image Optimization with SvelteKit on Vercel:**

- Configure image optimization with `@sveltejs/adapter-vercel`
- Optimize for production with a function that constructs optimized `srcset` for your images
- Helps your team ensure great performance by default
- Keeps your builds fast by optimizing images on-demand

[Learn more about Image Optimization](/docs/image-optimization)

## Web Analytics

Vercel's Web Analytics features enable you to visualize and monitor your application's performance over time. The **Analytics** tab in yo

To use Web Analytics, navigate to the Analytics tab of your project dashboard on Vercel and select **Enable** in the modal that appears.

To track visitors and page views, we recommend first installing our `@vercel/analytics` package by running the terminal command below in

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i @vercel/analytics
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @vercel/analytics
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @vercel/analytics
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i @vercel/analytics
    ```
  </Code>
</CodeBlock>

In your SvelteKit project's main `+layout.svelte` file, add the following `<script>`:

```html filename="src/routes/+layout.svelte" framework=all
<script>
  import { dev } from '$app/environment';
  import { inject } from '@vercel/analytics';
  inject({ mode: dev ? 'development' : 'production' });
</script>
```

With the above script added to your project, you'll be able to view detailed user insights in your dashboard on Vercel under the Analytic

**Your project must be deployed on Vercel to take advantage of the Web Analytics feature**. Work on making this feature more broadly avai

**To summarize, using Web Analytics with SvelteKit on Vercel:**

- Enables you to track traffic and see your top-performing pages
- Offers you detailed breakdowns of visitor demographics, including their OS, browser, geolocation, and more

[Learn more about Web Analytics](/docs/analytics)

## Speed Insights

You can see data about your project's [Core Web Vitals](/docs/speed-insights/metrics#core-web-vitals-explained) performance in your dashb

[See our Speed Insights docs to learn more](/docs/speed-insights).

**To summarize, using Speed Insights with SvelteKit on Vercel:**

- Enables you to track traffic performance metrics, such as [First Contentful Paint](/docs/speed-insights/metrics#first-contentful-paint-
- Enables you to view performance metrics by page name and URL for more
  granular analysis
- Shows you [a score for your app's performance](/docs/speed-insights/metrics#how-the-scores-are-determined) on each recorded metric, whi

[Learn more about Speed Insights](/docs/speed-insights)

## Draft Mode

[Draft Mode](/docs/draft-mode) enables you to view draft content from your [Headless CMS](/docs/solutions/cms) immediately, while still s

To use a SvelteKit route in Draft Mode, you must:

1. Export a `config` object [that enables Incremental Static Regeneration](https://kit.svelte.dev/docs/adapter-vercel#incremental-static-

```ts filename="blog/[slug]/+page.server.ts" framework=all
import { BYPASS_TOKEN } from '$env/static/private';

export const config = {
  isr: {
    // Random token that can be provided to bypass the cached version of the page with a __prerender_bypass=<token> cookie. Allows render
    bypassToken: BYPASS_TOKEN,

    // Expiration time (in seconds) before the cached asset will be re-generated by invoking the Vercel Function.
    // Setting the value to `false` means it will never expire.
    expiration: 60,
  },
};
```

```js filename="blog/[slug]/+page.server.js" framework=all
import { BYPASS_TOKEN } from '$env/static/private';

export const config = {
  isr: {
    // Random token that can be provided to bypass the cached version of the page with a __prerender_bypass=<token> cookie. Allows render
    bypassToken: BYPASS_TOKEN,

    // Expiration time (in seconds) before the cached asset will be re-generated by invoking the Vercel Function.
    // Setting the value to `false` means it will never expire.
    expiration: 60,
  },
};
```

2. Send a `__prerender_bypass` cookie with the same value as `bypassToken` in your config.

To render the draft content, SvelteKit will check for `__prerender_bypass`. If its value matches the value of `bypassToken`, it will rend

> **💡 Note:** We recommend using a cryptographically secure random number generator at build
> time as your `bypassToken` value. If a malicious actor guesses your
> `bypassToken`, they can view your pages in Draft Mode.

### Draft Mode security

Deployments on Vercel automatically secure Draft Mode behind the same authentication used for Preview Comments. In order to enable or dis

### Enabling Draft Mode in Preview Deployments

You and your team members can toggle Draft Mode in the Vercel Toolbar in [production](/docs/vercel-toolbar/in-production-and-localhost/ad

Users outside your Vercel team cannot toggle Draft Mode.

**To summarize, the benefits of using Draft Mode with SvelteKit on Vercel include:**

- Easily server-render previews of static pages
- Adds security measures to prevent malicious usage
- Integrates with any headless provider of your choice
- You can enable and disable Draft Mode in [the comments toolbar](/docs/comments/how-comments-work) on Preview Deployments

[Learn more about Draft Mode](/docs/draft-mode)

## Routing Middleware

Routing Middleware is useful for modifying responses before they're sent to a user. **We recommend [using SvelteKit's server hooks](https

## Rewrites

Adding a [`vercel.json`](/docs/project-configuration) file to the root directory of your project enables you to rewrite your app's routes

**We do not recommend using `vercel.json` rewrites with SvelteKit**.

Rewrites from `vercel.json` only apply to the Vercel proxy. At runtime, SvelteKit doesn't have access to the rewritten URL, which means i

## More benefits

See [our Frameworks documentation page](/docs/frameworks) to learn about the benefits available to **all** frameworks when you deploy on '

## More resources

Learn more about deploying SvelteKit projects on Vercel with the following resources:

- [Learn about the Build Output API](/docs/build-output-api/v3)
- [SvelteKit's official docs](https://kit.svelte.dev/docs/adapter-vercel)


---
title: "TanStack Start on Vercel"
description: "Learn how to use Vercel"
last_updated: "2026-01-16T02:19:31.206Z"
source: "https://vercel.com/docs/frameworks/full-stack/tanstack-start"
---

# TanStack Start on Vercel

TanStack Start is a fullstack framework powered by TanStack Router for React and Solid. It has support for full-document SSR, streaming,

## Getting started

You can quickly deploy a TanStack Start application to Vercel by creating a new one below or configuring an existing one with Nitro:

## Nitro Configuration

The [Nitro Vite plugin](https://v3.nitro.build/) allows deploying TanStack Start apps on Vercel, and integrates with Vercel's features.

To set up Nitro in your TanStack app, navigate to the root directory of your TanStack Start project with your terminal and install `nitro

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i nitro
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i nitro
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i nitro
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i nitro
    ```
  </Code>
</CodeBlock>

To configure Nitro with TanStack Start, add the following lines to your `vite.config` file:

```ts {4-4,9-9} filename="/vite.config.ts"
import { tanstackStart } from '@tanstack/react-start/plugin/vite'
import { defineConfig } from 'vite'
import viteReact from '@vitejs/plugin-react'
import { nitro } from 'nitro/vite'
```

```
export default defineConfig({
  plugins: [
    tanstackStart(),
    nitro(),
    viteReact(),
  ],
})
```

### Vercel Functions

TanStack Start apps on Vercel benefit from the advantages of [Vercel Functions](/docs/functions) and use [Fluid Compute](/docs/fluid-comp

## More resources

Learn more about deploying TanStack Start projects on Vercel with the following resources:

- [Explore the TanStack docs](https://tanstack.com/start/latest/docs/framework/react/overview)
- [Learn to use Vercel specific features with Nitro](https://v3.nitro.build/deploy/providers/vercel)

---
title: "Supported Frameworks on Vercel"
description: "Learn about the frameworks that can be deployed to Vercel."
last_updated: "2026-01-16T02:19:31.213Z"
source: "https://vercel.com/docs/frameworks/more-frameworks"
---

# Supported Frameworks on Vercel

## Frameworks infrastructure support matrix

The following table shows which features are supported by each framework on Vercel. The framework list is not exhaustive, but a represent

We're committed to having support for all Vercel features across frameworks, and continue to work with framework authors on adding suppor

**Legend:** ✓ Supported | ✗ Not Supported | N/A Not Applicable

| Feature | Next.js | SvelteKit | Nuxt | TanStack | Astro | Remix | Vite | CRA |
|---|---|---|---|---|---|---|---|---|
| [Static Assets](/docs/cdn) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| [Edge Routing Rules](/docs/cdn#features) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| [Routing Middleware](/docs/routing-middleware) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| [Server-Side Rendering](/docs/functions) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | N/A | N/A |
| [Streaming SSR](/docs/functions/streaming-functions) | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | N/A | N/A |
| [Incremental Static Regeneration](/docs/incremental-static-regeneration) | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | N/A | N/A |
| [Image Optimization](/docs/image-optimization) | ✓ | ✓ | ✓ | N/A | ✓ | ✗ | N/A | N/A |
| [Data Cache](/docs/data-cache) | ✓ | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| [Native OG Image Generation](/docs/og-image-generation) | ✓ | N/A | ✓ | N/A | N/A | N/A | N/A | N/A |
| [Multi-runtime support (different routes)](/docs/functions/runtimes) | ✓ | ✓ | ✓ | ✓ | N/A | ✗ | ✓ | N/A | N/A |
| [Multi-runtime support (entire app)](/docs/functions/runtimes) | ✓ | ✓ | ✓ | N/A | ✓ | ✓ | N/A | N/A |
| [Output File Tracing](/kb/guide/how-can-i-use-files-in-serverless-functions) | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | N/A | N/A |
| [Skew Protection](/docs/skew-protection) | ✓ | ✓ | ✗ | N/A | ✓ | ✗ | N/A | N/A |
| [Framework Routing Middleware](/docs/routing-middleware) | ✓ | N/A | ✗ | ✓ | ✓ | ✗ | N/A | N/A |

## All frameworks

The frameworks listed below can be deployed to Vercel with minimal configuration. See [our docs on framework presets](/docs/deployments/c

- **Angular**: Angular is a TypeScript-based cross-platform framework from Google.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/angular) | [View Demo](https
- **Astro**: Astro is a new kind of static site builder for the modern web. Powerful developer experience meets lightweight output.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/astro) | [View Demo](https:/
- **Brunch**: Brunch is a fast and simple webapp build tool with seamless incremental compilation for rapid development.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/brunch) | [View Demo](https:
- **React**: Create React App allows you to get going with React in no time.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/create-react-app) | [View De
- **Docusaurus (v1)**: Docusaurus makes it easy to maintain Open Source documentation websites.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/docusaurus) | [View Demo](ht
- **Docusaurus (v2+)**: Docusaurus makes it easy to maintain Open Source documentation websites.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/docusaurus-2) | [View Demo](
- **Dojo**: Dojo is a modern progressive, TypeScript first framework.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/dojo) | [View Demo](https://
- **Eleventy**: 11ty is a simpler static site generator written in JavaScript, created to be an alternative to Jekyll.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/eleventy) | [View Demo](http
- **Elysia**: Ergonomic framework for humans
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/elysia)
- **Ember.js**: Ember.js helps webapp developers be more productive out of the box.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/ember) | [View Demo](https:/
- **Express**: Fast, unopinionated, minimalist web framework for Node.js
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/express) | [View Demo](https
- **FastAPI**: FastAPI framework, high performance, easy to learn, fast to code, ready for production
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/fastapi) | [View Demo](https
- **FastHTML**: The fastest way to create an HTML app
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/fasthtml) | [View Demo](http
- **Fastify**: Fast and low overhead web framework, for Node.js
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/fastify)
- **Flask**: The Python micro web framework
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/flask)
- **Gatsby.js**: Gatsby helps developers build blazing fast websites and apps with React.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/gatsby) | [View Demo](https:
- **Gridsome**: Gridsome is a Vue.js-powered framework for building websites & apps that are fast by default.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/gridsome) | [View Demo](http
- **H3**: Universal, Tiny, and Fast Servers
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/h3)
- **Hexo**: Hexo is a fast, simple & powerful blog framework powered by Node.js.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/hexo) | [View Demo](https://
- **Hono**: Web framework built on Web Standards
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/hono) | [View Demo](https://

- **Hugo**: Hugo is the world's fastest framework for building websites, written in Go.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/hugo) | [View Demo](https://
- **Hydrogen (v1)**: React framework for headless commerce
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/hydrogen) | [View Demo](http
- **Ionic Angular**: Ionic Angular allows you to build mobile PWAs with Angular and the Ionic Framework.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/ionic-angular) | [View Demo]
- **Ionic React**: Ionic React allows you to build mobile PWAs with React and the Ionic Framework.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/ionic-react) | [View Demo](h
- **Jekyll**: Jekyll makes it super easy to transform your plain text into static websites and blogs.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/jekyll) | [View Demo](https:
- **Middleman**: Middleman is a static site generator that uses all the shortcuts and tools in modern web development.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/middleman) | [View Demo](htt
- **NestJS**: Framework for building efficient, scalable Node.js server-side applications
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/nestjs)
- **Next.js**: Next.js makes you productive with React instantly — whether you want to build static or dynamic sites.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/nextjs) | [View Demo](https:
- **Nitro**: Nitro is a next generation server toolkit.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/nitro) | [View Demo](https:/
- **Nuxt**: Nuxt is the open source framework that makes full-stack development with Vue.js intuitive.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/nuxtjs) | [View Demo](https:
- **Parcel**: Parcel is a zero configuration build tool for the web that scales to projects of any size and complexity.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/parcel) | [View Demo](https:
- **Polymer**: Polymer is an open-source webapps library from Google, for building using Web Components.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/polymer) | [View Demo](https
- **Preact**: Preact is a fast 3kB alternative to React with the same modern API.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/preact) | [View Demo](https:
- **React Router**: Declarative routing for React
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/react-router) | [View Demo](
- **RedwoodJS**: RedwoodJS is a full-stack framework for the Jamstack.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/redwoodjs) | [View Demo](htt
- **Remix**: Build Better Websites
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/remix) | [View Demo](https:/
- **Saber**: Saber is a framework for building static sites in Vue.js that supports data from any source.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/saber)
- **Sanity**: The structured content platform.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/sanity) | [View Demo](https:
- **Sanity (v3)**: The structured content platform.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/sanity-v3) | [View Demo](htt
- **Scully**: Scully is a static site generator for Angular.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/scully) | [View Demo](https:
- **SolidStart (v0)**: Simple and performant reactivity for building user interfaces.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/solidstart) | [View Demo](ht
- **SolidStart (v1)**: Simple and performant reactivity for building user interfaces.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/solidstart-1) | [View Demo](
- **Stencil**: Stencil is a powerful toolchain for building Progressive Web Apps and Design Systems.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/stencil) | [View Demo](https
- **Storybook**: Frontend workshop for UI development
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/storybook)
- **SvelteKit**: SvelteKit is a framework for building web applications of all sizes.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/sveltekit-1) | [View Demo](h
- **TanStack Start**: Full-stack Framework powered by TanStack Router for React and Solid.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/tanstack-start)
- **UmiJS**: UmiJS is an extensible enterprise-level React application framework.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/umijs) | [View Demo](https:/
- **Vite**: Vite is a new breed of frontend build tool that significantly improves the frontend development experience.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/vite) | [View Demo](https://
- **VitePress**: VitePress is VuePress' little brother, built on top of Vite.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/vitepress) | [View Demo](htt
- **Vue.js**: Vue.js is a versatile JavaScript framework that is as approachable as it is performant.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/vue) | [View Demo](https://v
- **VuePress**: Vue-powered Static Site Generator
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/vuepress) | [View Demo](http
- **xmcp**: The MCP framework for building AI-powered tools
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/xmcp) | [View Demo](https://
- **Zola**: Everything you need to make a static site engine in one binary.
  - [Deploy](https://vercel.com/new/clone?repository-url=https://github.com/vercel/vercel/tree/main/examples/zola) | [View Demo](https://


## More resources

Learn more about deploying your preferred framework on Vercel with the following resources:

- [Next.js on Vercel](/docs/frameworks/nextjs)
- [SvelteKit on Vercel](/docs/frameworks/sveltekit)
- [Astro on Vercel](/docs/frameworks/astro)
- [Nuxt on Vercel](/docs/frameworks/nuxt)


---
title: "Frameworks on Vercel"
description: "Vercel supports a wide range of the most popular frameworks, optimizing how your application builds and runs no matter what
last_updated: "2026-01-16T02:19:31.362Z"
source: "https://vercel.com/docs/frameworks"
---

# Frameworks on Vercel

Vercel has first-class support for [a wide range of the most popular frameworks](/docs/frameworks/more-frameworks). You can build and dep

Learn how to [get started with Vercel](/docs/getting-started-with-vercel) or clone one of our example repos to your favorite git provider

Vercel deployments can [integrate with your git provider](/docs/git) to [generate preview URLs](/docs/deployments/environments#preview-en

Deploying on Vercel with one of our [supported frameworks](/docs/frameworks/more-frameworks) gives you access to many features, such as:

- [Vercel Functions](/docs/functions) enable developers to write functions that scale based on traffic demands, preventing failures durin
- [Middleware](/docs/routing-middleware) is code that executes before a request is processed on a site, enabling you to modify the respon
- [Multi-runtime Support](/docs/functions/runtimes) allows the use of various runtimes for your functions, each with unique libraries, AP
- [Incremental Static Regeneration](/docs/incremental-static-regeneration) enables content updates without redeployment. Vercel caches th
- [Speed Insights](/docs/speed-insights) provide data on your project's Core Web Vitals performance in the Vercel dashboard, helping you

- [Analytics](/docs/analytics) offer detailed insights into your website's performance over time, including metrics like top pages, top r
- [Skew Protection](/docs/skew-protection) uses version locking to ensure that the client and server use the same version of your applica

## Frameworks infrastructure support matrix

The following table shows which features are supported by each framework on Vercel. The framework list represents the most popular framew

**Legend:** ✓ Supported | ✗ Not Supported | N/A Not Applicable

| Feature | Next.js | SvelteKit | Nuxt | TanStack | Astro | Remix | Vite | CRA |
|---|---|---|---|---|---|---|---|---|
| [Static Assets](/docs/cdn) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| [Edge Routing Rules](/docs/cdn#features) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| [Routing Middleware](/docs/routing-middleware) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| [Server-Side Rendering](/docs/functions) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | N/A | N/A |
| [Streaming SSR](/docs/functions/streaming-functions) | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | N/A | N/A |
| [Incremental Static Regeneration](/docs/incremental-static-regeneration) | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | N/A | N/A |
| [Image Optimization](/docs/image-optimization) | ✓ | ✓ | ✓ | N/A | ✓ | ✗ | N/A | N/A |
| [Data Cache](/docs/data-cache) | ✓ | N/A | N/A | N/A | N/A | N/A | N/A | |
| [Native OG Image Generation](/docs/og-image-generation) | ✓ | N/A | ✓ | N/A | N/A | N/A | N/A | N/A |
| [Multi-runtime support (different routes)](/docs/functions/runtimes) | ✓ | ✓ | ✓ | N/A | ✗ | ✓ | N/A | N/A |
| [Multi-runtime support (entire app)](/docs/functions/runtimes) | ✓ | ✓ | ✓ | N/A | ✓ | ✓ | N/A | N/A |
| [Output File Tracing](/kb/guide/how-can-i-use-files-in-serverless-functions) | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | N/A | N/A |
| [Skew Protection](/docs/skew-protection) | ✓ | ✓ | ✗ | N/A | ✓ | ✗ | N/A | N/A |
| [Framework Routing Middleware](/docs/routing-middleware) | ✓ | N/A | ✗ | ✓ | ✓ | ✗ | N/A | N/A |

## Build Output API

The [Build Output API](/docs/build-output-api/v3) is a file-system-based specification for a directory structure that produces a Vercel d

If you are not using a framework, you can still use these features by manually creating and populating the `.vercel/output` directory acc

## More resources

Learn more about deploying your preferred framework on Vercel with the following resources:

- [See a full list of supported frameworks](/docs/frameworks/more-frameworks)
- [Explore our template marketplace](/templates)
- [Learn about our deployment features](/docs/deployments)

---
title: "Concurrency scaling"
description: "Learn how Vercel automatically scales your functions to handle traffic surges."
last_updated: "2026-01-16T02:19:31.366Z"
source: "https://vercel.com/docs/functions/concurrency-scaling"
---

# Concurrency scaling

Vercel automatically scales your functions to handle traffic surges, ensuring optimal performance during increased loads.

## Automatic concurrency scaling

The concurrency model on Vercel refers to how many instances of your [functions](/docs/functions) can run simultaneously. All functions o

With automatic concurrency scaling, your Vercel Functions can scale to a maximum of **30,000** on Pro or **100,000** on Enterprise, maint

Vercel's infrastructure monitors your usage and preemptively adjusts the concurrency limit to cater to growing traffic, allowing your app

Automatic concurrency scaling is available on [all plans](/docs/plans).

## Burst concurrency limits

Burst concurrency refers to Vercel's ability to temporarily handle a sudden influx of traffic by allowing a higher concurrency limit.

Upon detecting a traffic spike, Vercel temporarily increases the concurrency limit to accommodate the additional load. The initial increa

The scaling process may take several minutes during traffic surges, especially substantial ones. While this delay aligns with natural tra

You can monitor burst concurrency events using [Log Drains](/docs/drains), or [Runtime Logs](/docs/runtime-logs) to help you understand a

If you exceed the limit, a [`503 FUNCTION_THROTTLED`](/docs/errors/FUNCTION_THROTTLED) error will trigger.

---
title: "Advanced Configuration"
description: "Learn how to add utility files to the /api directory, and bundle Vercel Functions."
last_updated: "2026-01-16T02:19:31.377Z"
source: "https://vercel.com/docs/functions/configuring-functions/advanced-configuration"
---

# Advanced Configuration

For an advanced configuration, you can create a `vercel.json` file to use [Runtimes](/docs/functions/runtimes) and other customizations. `

If your use case requires that you work asynchronously with the results of a function invocation, you may need to consider a queuing, poo

## Adding utility files to the `/api` directory

Sometimes, you need to place extra code files, such as `utils.js` or `my-types.d.ts`, inside the `/api` folder. To avoid turning these fi

- Files that start with an underscore, `_`
- Files that start with `.`
- Files that end with `.d.ts`

If your file uses any of the above, it will **not** be turned into a function.

## Bundling Vercel Functions

In order to optimize resources, Vercel uses a process to bundle as many routes as possible into a single Vercel Function.

To provide more control over the bundling process, you can use the [`functions` property](/docs/project-configuration#functions) in your

This bundling process is currently only enabled for Next.js, but it will be enabled in other scenarios in the future.

> For \['other']:

In the following example,  will be bundled separately from  since each has a different configuration:

> For \['nextjs']:

In the following example,  will be bundled separately from  since each has a different configuration:

> For \['nextjs-app']:

In the following example,  will be bundled separately from  since each has a different configuration:

```js filename="vercel.json" framework=nextjs
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "pages/api/hello.js": {
      "memory": 3009,
      "maxDuration": 60
    },
    "pages/api/another.js": {
      "memory": 1024,
      "maxDuration": 30
    }
  }
}
```

```ts filename="vercel.json" framework=nextjs
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "pages/api/hello.ts": {
      "memory": 3009,
      "maxDuration": 60
    },
    "pages/api/another.ts": {
      "memory": 1024,
      "maxDuration": 30
    }
  }
}
```

```js filename="vercel.json" framework=other
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "api/hello.js": {
      "memory": 3009,
      "maxDuration": 60
    },
    "api/another.js": {
      "memory": 1024,
      "maxDuration": 30
    }
  }
}
```

```ts filename="vercel.json" framework=other
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "api/hello.ts": {
      "memory": 3009,
      "maxDuration": 60
    },
    "api/another.ts": {
      "memory": 1024,
      "maxDuration": 30
    }
  }
}
```

```js filename="vercel.json" framework=nextjs-app
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "app/api/hello/route.js": {
      "memory": 3009,
      "maxDuration": 60
    },
    "app/api/another/route.js": {
      "memory": 1024,
      "maxDuration": 30
    }
  }
}
```

```ts filename="vercel.json" framework=nextjs-app
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "app/api/hello/route.ts": {
      "memory": 3009,
      "maxDuration": 60
    },
    "app/api/another/route.ts": {
      "memory": 1024,
      "maxDuration": 30
    }
  }
}
```

---
title: "Configuring In-function Concurrency"
description: "Learn how to allow multiple requests to share a single function instance."
last_updated: "2026-01-16T02:19:31.383Z"
source: "https://vercel.com/docs/functions/configuring-functions/concurrency"
---

# Configuring In-function Concurrency

In-function concurrency allows multiple requests to share a single function instance and is available when using the Node.js or Python ru

This feature is ideal for I/O-bound tasks like database operations or API requests, as it makes better use of system resources. However,

## Enabling in-function concurrency

> **💡 Note:** You must have enabled at least 1vCPU of memory (i.e. **Standard** or
> **Performance**) in order to enable concurrency for your functions. To learn
> more, see [Setting your default function CPU
> size](/docs/functions/configuring-functions/memory#setting-your-default-function-memory-/-cpu-size).

To enable the feature:

1. Navigate to your project in the Vercel [dashboard](/dashboard).
2. Click on the **Settings** tab and select the **Functions** section.
3. Scroll to the **In-function concurrency** section.
4. Toggle the switch to **Enabled**, and click **Save**.
5. Redeploy your project to apply the changes.

Concurrency is now enabled for all functions in that project.

## Viewing in-function concurrency metrics

Once enabled, you can view the [GB-Hours saved](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Fobservability%2Fserverless-functi

1. Choose your project from the [dashboard](/dashboard).
2. Click on the **Settings** tab and select the **Functions** section and scroll to the **In-function concurrency** section.
3. Next to the toggle, click the **View in-function concurrency metrics** link.

From here, you'll be able to see total consumed and saved GB-Hours, and the ratio of the saved usage.

---
title: "Configuring Maximum Duration for Vercel Functions"
description: "Learn how to set the maximum duration of a Vercel Function."
last_updated: "2026-01-16T02:19:31.393Z"
source: "https://vercel.com/docs/functions/configuring-functions/duration"
---

# Configuring Maximum Duration for Vercel Functions

The maximum duration configuration determines the longest time that a function can run. This guide will walk you through configuring the m

## Consequences of changing the maximum duration

You are charged based on the amount of time your function has run, also known as its *duration*. It specifically refers to the *actual ti

For this reason, Vercel has set a [default maximum duration](/docs/functions/limitations#max-duration) for functions, which can be useful

If a function runs for longer than its set maximum duration, Vercel will terminate it. Therefore, when setting this duration, it's crucia

1. Allow sufficient time for your function to complete its normal operations, including any necessary waiting periods (for example, strea
2. Set a reasonable limit to prevent abnormally long executions.

## Maximum duration for different runtimes

The method of configuring the maximum duration depends on your framework and runtime:

#### Node.js, Next.js (>= 13.5 or higher), SvelteKit, Astro, Nuxt, and Remix

For these runtimes / frameworks, you can configure the number of seconds directly in your function:

```ts v0="build" {1} filename="app/api/my-function/route.ts" framework=nextjs-app
export const maxDuration = 5; // This function can run for a maximum of 5 seconds

export function GET(request: Request) {
  return new Response('Vercel', {
    status: 200,
  });
}
```

```js v0="build" {1} filename="app/api/my-function/route.js" framework=nextjs-app
export const maxDuration = 5; // This function can run for a maximum of 5 seconds
```

```
export function GET(request) {
  return new Response('Vercel', {
    status: 200,
  });
}
```

```ts v0="build" {4-6} filename="pages/api/handler.ts" framework=nextjs
import { NextApiRequest, NextApiResponse } from 'next';

// This function can run for a maximum of 5 seconds
export const config = {
  maxDuration: 5,
};

export default function handler(
  request: NextApiRequest,
  response: NextApiResponse,
) {
  response.status(200).json({
    body: request.body,
    query: request.query,
    cookies: request.cookies,
  });
}
```

```js v0="build" {2-4} filename="pages/api/handler.js" framework=nextjs
// This function can run for a maximum of 5 seconds
export const config = {
  maxDuration: 5,
};

export default function handler(request, response) {
  response.status(200).json({
    body: request.body,
    query: request.query,
    cookies: request.cookies,
  });
}
```

```ts {2-4} filename="app/routes/function/my-function.ts" framework=remix
// This function can run for a maximum of 5 seconds
export const config = {
  maxDuration: 5,
};

export default function Serverless() {
  return (
    <div style={{ fontFamily: 'system-ui, sans-serif', lineHeight: '1.4' }}>
      <h1>Configuring maxDuration</h1>
    </div>
  );
}
```

```js {2-4} filename="app/routes/function/my-function.js" framework=remix
// This function can run for a maximum of 5 seconds
export const config = {
  maxDuration: 5,
};

export default function Serverless() {
  return (
    <div style={{ fontFamily: 'system-ui, sans-serif', lineHeight: '1.4' }}>
      <h1>Configuring maxDuration</h1>
    </div>
  );
}
```

```js {7} filename="svelte.config.js" framework=sveltekit
import adapter from '@sveltejs/adapter-vercel';

// This function can run for a maximum of 5 seconds
export default {
  kit: {
    adapter: adapter({
      maxDuration: 5,
    }),
  },
};
```

```ts {7} filename="svelte.config.js" framework=sveltekit
import adapter from '@sveltejs/adapter-vercel';

// This function can run for a maximum of 5 seconds
export default {
  kit: {
    adapter: adapter({
      maxDuration: 5,
    }),
  },
};
```

```js {8} filename="astro.config.mjs" framework=astro
```

```
import { defineConfig } from 'astro/config';
import vercel from '@astrojs/vercel/serverless';

// This function can run for a maximum of 5 seconds
export default defineConfig({
  output: 'server',
  adapter: vercel({
    maxDuration: 5,
  }),
});
```

```ts {8} filename="astro.config.mjs" framework=astro
import { defineConfig } from 'astro/config';
import vercel from '@astrojs/vercel/serverless';

// This function can run for a maximum of 5 seconds
export default defineConfig({
  output: 'server',
  adapter: vercel({
    maxDuration: 5,
  }),
});
```

```js {7} filename="nitro.config.ts" framework=nuxt
import { defineNitroConfig } from 'nitropack';

// This function can run for a maximum of 5 seconds
export default defineNitroConfig({
  vercel: {
    functions: {
      maxDuration: 5,
    },
  },
});
```

```ts {7} filename="nitro.config.ts" framework=nuxt
import { defineNitroConfig } from 'nitropack';

// This function can run for a maximum of 5 seconds
export default defineNitroConfig({
  vercel: {
    functions: {
      maxDuration: 5,
    },
  },
});
```

```json {5,8} filename="vercel.json" framework=other
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "api/test.js": {
      "maxDuration": 30 // This function can run for a maximum of 30 seconds
    },
    "api/*.js": {
      "maxDuration": 15 // These functions can run for a maximum of 15 seconds
    }
  }
}
```

#### Other Frameworks and runtimes, Next.js versions older than 13.5, Go, Python, or Ruby

For these runtimes and frameworks, configure the `maxDuration` property of the [`functions` object](/docs/project-configuration#functions

```json {5,8,11} filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "api/test.js": {
      "maxDuration": 30 // This function can run for a maximum of 30 seconds
    },
    "api/*.js": {
      "maxDuration": 15 // This function can run for a maximum of 15 seconds
    },
    "src/api/*.js": {
      "maxDuration": 25 // You must prefix functions in the src directory with /src/
    }
  }
}
```

If your Next.js project is configured to use [src directory](https://nextjs.org/docs/app/building-your-application/configuring/src-direct

> **💡 Note:** The order in which you specify file patterns is important. For more
> information, see [Glob
> pattern](/docs/project-configuration#glob-pattern-order).

## Setting a default maximum duration

While Vercel specifies [defaults](/docs/functions/limitations#max-duration) for the maximum duration of a function, you can also override

### Dashboard

1. From your [dashboard](/dashboard), select your project and go to the **Settings** tab.
2. From the left side, select the **Functions** tab and scroll to the **Function Max Duration** section.

3. Update the **Default Max Duration** field value and select **Save**.

### `vercel.json` file

```json {4-5} filename="vercel.json" framework=nextjs-app
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "app/api/**/*": {
      "maxDuration": 5
    }
  }
}
```

```json {3-4} filename="pages/api/handler.js" framework=nextjs
{
  "functions": {
    "pages/api/**/*": {
      "maxDuration": 5
    }
  }
}
```

```json {4-5} filename="vercel.json" framework=remix
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "app/routes/**/*": {
      "maxDuration": 5 // All functions can run for a maximum of 5 seconds
    }
  }
}
```

```json {4-5} filename="vercel.json" framework=other
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "path/to/dir/**/*": {
      "maxDuration": 5 // All functions can run for a maximum of 5 seconds
    }
  }
}
```

This glob pattern will match *everything* in the specified path, so you may wish to be more specific by adding a file type, such as `app/

## Duration limits

Vercel Functions have the following defaults and maximum limits for the duration of a function with [fluid compute](/docs/fluid-compute)

| | Default | Maximum |
| --- | --- | --- |
| Hobby | 300s (5 minutes) | 300s (5 minutes) |
| Pro | 300s (5 minutes) | 800s (13 minutes) |
| Enterprise | 300s (5 minutes) | 800s (13 minutes) |

If you have disabled fluid compute, the following defaults and maximum limits apply:

| | Default | Maximum |
| --- | --- | --- |
| Hobby | 10s | 60s (1 minute) |
| Pro | 15s | 300s (5 minutes) |
| Enterprise | 15s | 900s (15 minutes) |

---
title: "Configuring Memory and CPU for Vercel Functions"
description: "Learn how to set the memory / CPU of a Vercel Function."
last_updated: "2026-01-16T02:19:31.460Z"
source: "https://vercel.com/docs/functions/configuring-functions/memory"
---

# Configuring Memory and CPU for Vercel Functions

The memory configuration of a function determines how much memory and CPU a function can use while executing. By default, on **Pro** and

## Memory configuration considerations

You should consider the following points when changing the memory size of your functions:

- **Performance**: Increasing memory size can improve the performance of your functions, allowing them to run faster
- **Cost**: Vercel Functions are billed based on the function duration, which is affected by the memory size. While increasing the functi

## Setting your default function memory / CPU size

Those on the Pro or Enterprise plans can configure the default memory size for all functions in a project.

To change the default function memory size:

1. Choose the appropriate project from your [dashboard](/dashboard)
2. Navigate to the **Settings** tab
3. Scroll to **Functions**
4. Select **Advanced Settings**
5. In the **Function CPU** section, select your preferred memory size option:
6. The change will be applied to all future deployments made by your team. You must create a new deployment for your changes to take effe

> **⚠️ Warning:** You cannot set your memory size using `vercel.json`. If you try to do so, you

> will receive a warning at build time. Only Pro and Enterprise users can set
> the default memory size in the dashboard. Hobby users will always use the
> default memory size of 2 GB (1 vCPU).

### Memory / CPU type

The memory size you select will also determine the CPU allocated to your Vercel Functions. The following table shows the memory and CPU a

With [fluid compute enabled](/docs/fluid-compute) on Pro and Enterprise plans, the default memory size is 2 GB (1 vCPU) and can be upgrad

| Type | Memory / CPU | Use |
| --- | --- | --- |
| Standard | 2 GB / 1 vCPU | Predictable performance for production workloads. Default for [fluid compute](/docs/fluid-compute). |
| Performance | 4 GB / 2 vCPUs | Increased performance for latency- |

Users on the Hobby plan can only use the default memory size of 2 GB (1 vCPU). **Hobby users cannot configure this size**. If you are on

> **💡 Note:** Project created before **2019-11-08** have the default function memory size
> set to **1024 MB/0.6 vCPU** for **Hobby** plan, and **3008 MB/1.67 vCPU** for
> **Pro** and **Enterprise** plan. Although the dashboard may not have any
> memory size option selected by default for those projects, you can start using
> the new memory size options by selecting your preferred memory size in the
> dashboard.

## Viewing your function memory size

To check the memory size of your functions in the [dashboard](/dashboard), follow these steps:

1. Find the project you want to review and select the **Deployments** tab
2. Go to the deployment you want to review
3. Select the **Resources** tab
4. Search for the function by name or find it in the **Functions** section
5. Click on the name of the function to open it in **Observability**
6. Hover over the information icon next to the function name to view its memory size

## Memory limits

To learn more about the maximum size of your function's memory, see [Max memory size](/docs/functions/limitations#memory-size-limits).

## Pricing

While memory / CPU size is not an explicitly billed metric, it is fundamental in how the billed metric of  is calculated.

> **⚠️ Warning:** **Legacy Billing Model**: This describes the legacy Function duration billing
> model based on wall-clock time. For new projects, we recommend [Fluid
> Compute](/docs/functions/usage-and-pricing) which bills separately for active
> CPU time and provisioned memory time for more cost-effective and transparent
> pricing.

You are charged based on the duration your Vercel functions have run. This is sometimes called "wall-clock time", which refers to the *ac

For example, if a function [has](/docs/functions/configuring-functions/memory) 1.7 GB (1769 MB) of memory and is executed **1 million tim

- Total Seconds: 1M * (1s) = 1,000,000 Seconds
- Total GB-Seconds: 1769/1024 GB * 1,000,000 Seconds = 1,727,539.06 GB-Seconds
- Total GB-Hrs: 1,727,539.06 GB-Seconds / 3600 = 479.87 GB-Hrs
- The total Vercel Function Execution is 479.87 GB-Hrs.

To see your current usage, navigate to the **Usage** tab on your team's [Dashboard](/dashboard) and go to **Serverless Functions** > **Du

You can also view [Invocations](/docs/functions/usage-and-pricing#managing-function-invocations)
to see the number of times your Functions have been invoked. To learn more about
the cost of Vercel Functions, see [Vercel Function Pricing](/docs/pricing/serverless-functions).

---
title: "Configuring Functions"
description: "Learn how to configure the runtime, region, maximum duration, and memory for Vercel Functions."
last_updated: "2026-01-16T02:19:31.451Z"
source: "https://vercel.com/docs/functions/configuring-functions"
---

# Configuring Functions

You can configure Vercel functions in many ways, including the runtime, region, maximum duration, and memory.

With different configurations, particularly the runtime configuration, there are a number of trade-offs and limits that you should be awa

## Runtime

The runtime you select for your function determines the infrastructure, APIs, and other abilities of your function.

With Vercel, you can configure the runtime of a function in any of the following ways:

- **Node.js**: When working with a TypeScript or JavaScript function, you can use the Node.js runtime by setting a config option within t
- **Ruby**, **Python**, **Go**: These have similar functionality and limitations as Node.js functions. The configuration for these runtim
- **Community runtimes**: You can specify any other [runtime](/docs/functions/runtimes#community-runtimes), by using the [`functions`](/d

See [choosing a runtime](/docs/functions/runtimes) for more information.

## Region

Your function should execute in a location close to your data source. This minimizes latency, or delay, thereby enhancing your app's perf

See [configuring a function's region](/docs/functions/configuring-functions/region) for more information.

## Maximum duration

The maximum duration for your function defines how long a function can run for, allowing for more predictable billing.

Vercel Functions have a default duration that's dependent on your plan, but you can configure this as needed, [up to your plan's limit](/

See [configuring a function's duration](/docs/functions/configuring-functions/duration) for more information.

## Memory

Vercel Functions use an infrastructure that allows you to adjust the memory size.

See [configuring a function's memory](/docs/functions/configuring-functions/memory) for more information.

---
title: "Configuring regions for Vercel Functions"
description: "Learn how to configure regions for Vercel Functions."
last_updated: "2026-01-16T02:19:31.426Z"
source: "https://vercel.com/docs/functions/configuring-functions/region"
---

# Configuring regions for Vercel Functions

The Vercel platform caches all static content in [the CDN](/docs/cdn-cache) by default. This means your users will always get static file

In a globally distributed application, the physical distance between your function and its data source can impact latency and response ti

- By default, Vercel Functions execute in [*Washington, D.C., USA* (`iad1`)](/docs/pricing/regional-pricing/iad1) **for all new projects*
- You can define the region in your `vercel.json` using the [`regions` setting](/docs/functions/configuring-functions/region#project-conf
- You can set your region in the [Vercel CLI](#vercel-cli)

## Setting your default region

The default Function region is [*Washington, D.C., USA* (`iad1`)](/docs/pricing/regional-pricing/iad1) **for all new projects**.

### Dashboard

To change the default regions in the dashboard:

1. Choose the appropriate project from your [dashboard](/dashboard) on Vercel
2. Navigate to the **Settings** tab
3. From the left side, select **Functions**
4. Use the **Function Regions** accordion to select your project's default regions:

### Project configuration

To change the default region in your `vercel.json` [configuration file](/docs/project-configuration#regions), add the region code(s) to t

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "regions": ["sfo1"]
}
```

Additionally, Pro and Enterprise users can deploy Vercel Functions to multiple regions: Pro users can deploy to up to **three** regions,

Enterprise users can also use [`functionFailoverRegions`](/docs/project-configuration#functionfailoverregions) to specify regions that a '

### Vercel CLI

Use the `vercel --regions` command in your project's root directory to set a region. Learn more about setting regions with the `vercel --

## Available regions

To learn more about the regions that you can set for your Functions, see the [region list](/docs/regions#region-list).

## Automatic failover

Vercel Functions have multiple availability zone redundancy by default. Multi-region redundancy is available depending on your runtime.

### Node.js runtime failover

Enterprise teams can enable multi-region redundancy for Vercel Functions using Node.js.

To automatically failover to closest region in the event of an outage:

1. Select your project from your team's [dashboard](/dashboard)
2. Navigate to the **Settings** tab and select **Functions**
3. Enable the **Function Failover** toggle:

To manually specify the fallback region, you can pass one or more regions to the [`functionFailoverRegions`](/docs/project-configuration#

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functionFailoverRegions": ["dub1", "fra1"]
}
```

The region(s) set in the `functionFailoverRegions` property **must be different** from the default region(s) specified in the [`regions`]

During an automatic failover, Vercel will reroute application traffic to the next closest region, meaning the order of the regions in `fu

You can view your default and failover regions through the [deployment summary](/docs/deployments#resources-tab-and-deployment-summary):

Region failover is supported with Secure Compute. See [Region Failover](/docs/secure-compute#region-failover) to learn more.

---
title: "Configuring the Runtime for Vercel Functions"
description: "Learn how to configure the runtime for Vercel Functions."
last_updated: "2026-01-16T02:19:31.465Z"

source: "https://vercel.com/docs/functions/configuring-functions/runtime"
---

# Configuring the Runtime for Vercel Functions

The runtime of your function determines the environment in which your function will execute. Vercel supports various runtimes including N

## Node.js

By default, a function with no additional configuration will be deployed as a Vercel Function on the Node.js runtime.

> For \['nextjs']:

```ts v0="build" filename="app/api/hello/route.ts" framework=nextjs
export function GET(request: Request) {
  return new Response('Hello from Vercel!');
}
```

```js v0="build" filename="app/api/hello/route.js" framework=nextjs
export function GET(request) {
  return new Response('Hello from Vercel!');
}
```

```ts filename="api/hello.ts" framework=other
export function GET(request: Request) {
  return new Response('Hello from Vercel!');
}
```

```js filename="api/hello.js" framework=other
export function GET(request) {
  return new Response('Hello from Vercel!');
}
```

```ts v0="build" filename="app/api/hello/route.ts" framework=nextjs-app
export function GET(request: Request) {
  return new Response('Hello from Vercel!');
}
```

```js v0="build" filename="app/api/hello/route.js" framework=nextjs-app
export function GET(request) {
  return new Response('Hello from Vercel!');
}
```

> **💡 Note:** If you're not using a framework, you must either add
>  to your
>  or change your JavaScript Functions'
> file extensions from  to
>

## Go

For Go, expose a single HTTP handler from a `.go` file within an `/api` directory at your project's root. For example:

```go filename="/api/index.go"
package handler

import (
  "fmt"
  "net/http"
)

func Handler(w http.ResponseWriter, r *http.Request) {
  fmt.Fprintf(w, "<h1>Hello from Go!</h1>")
}
```

## Python

For Python, create a function by adding the following code to `api/index.py`:

```py filename="api/index.py"
from http.server import BaseHTTPRequestHandler

class handler(BaseHTTPRequestHandler):

    def do_GET(self):
        self.send_response(200)
        self.send_header('Content-type','text/plain')
        self.end_headers()
        self.wfile.write('Hello, world!'.encode('utf-8'))
        return
```

## Ruby

For Ruby, define an HTTP handler from `.rb` files within an `/api` directory at your project's root. Ruby files must have one of the follo

- `Handler` proc that matches the `do |request, response|` signature
- `Handler` class that inherits from the `WEBrick::HTTPServlet::AbstractServlet` class

For example:

```ruby filename="api/index.rb"
require 'cowsay'
```

```
Handler = Proc.new do |request, response|
  name = request.query['name'] || 'World'

  response.status = 200
  response['Content-Type'] = 'text/text; charset=utf-8'
  response.body = Cowsay.say("Hello #{name}", 'cow')
end
```

Don't forget to define your dependencies inside a `Gemfile`:

```ruby filename="Gemfile"
source "https://rubygems.org"

gem "cowsay", "~> 0.3.0"
```

## Other runtimes

You can configure other runtimes by using the `functions` property in your `vercel.json` file. For example:

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "api/test.php": {
      "runtime": "vercel-php@0.5.2"
    }
  }
}
```

In this case, the function at `api/hello.ts` would use the custom runtime specified.

For more information, see [Community runtimes](/docs/functions/runtimes#community-runtimes)

---
title: "Functions API Reference"
description: "Learn about available APIs when working with Vercel Functions."
last_updated: "2026-01-16T02:19:31.483Z"
source: "https://vercel.com/docs/functions/functions-api-reference"
---

# Functions API Reference

> For \["nextjs-app"]:

Functions are defined similar to a [Route Handler](https://nextjs.org/docs/app/building-your-application/routing/route-handlers) in Next.

> For \["nextjs"]:

While you can define a function with a traditional [Next.js API Route](https://nextjs.org/docs/api-routes/introduction), they do not supp

You can create an `app` directory at the same level as your `pages` directory.
Then, define your function in .

> For \["other"]:

You can create a function in other frameworks or with no frameworks by defining your function in a file under `/api` in your project. Ver

## Function signature

Vercel Functions use a Web Handler, which consists of the `request` parameter that is an instance of the web standard [`Request`](https:/

| Parameter | Description
| --- | ---
| `request` | An instance of the `Request` object
| `context` |  Deprecated, use [`@vercel/functions`](/docs/functions/functions-api-reference/vercel-functions-package#waituntil) instead

> For \['nextjs']:

```ts v0="build" filename="app/api/hello/route.ts" framework=nextjs
export function GET(request: Request) {
  return new Response('Hello from Vercel!');
}
```

```js v0="build" filename="app/api/hello/route.js" framework=nextjs
export function GET(request) {
  return new Response('Hello from Vercel!');
}
```

```ts filename="api/hello.ts" framework=other
export function GET(request: Request) {
  return new Response('Hello from Vercel!');
}
```

```js filename="api/hello.js" framework=other
export function GET(request) {
  return new Response('Hello from Vercel!');
}
```

```ts v0="build" filename="app/api/hello/route.ts" framework=nextjs-app
export function GET(request: Request) {
  return new Response('Hello from Vercel!');
```

}
```

```js v0="build" filename="app/api/hello/route.js" framework=nextjs-app
export function GET(request) {
  return new Response('Hello from Vercel!');
}
```

> For \["nextjs"]:

The above shows how you can use a [Route Handlers in the App Router](https://nextjs.org/docs/app/building-your-application/routing/route-

> For \["other"]:

### `fetch` Web Standard

Vercel Functions also support the `fetch` Web Standard export, used by many frameworks like [Hono](https://hono.dev), [ElysiaJS](https://

```ts filename="api/hello.ts" framework=all
export default {
  fetch(request: Request) {
    return new Response('Hello from Vercel!');
  },
};
```

```js filename="api/hello.js" framework=all
export default {
  fetch(request) {
    return new Response('Hello from Vercel!');
  },
};
```

### Cancel requests

> **💡 Note:** This feature is only available in the Node.js runtime.

Cancelling requests is useful for cleaning up resources or stopping long-running tasks when the client aborts the request — for example, v

To cancel requests in Vercel Functions

1. In your `vercel.json` file, add `"supportsCancellation": true` to the [specific paths](/docs/project-configuration#key-definition) you

   ```json filename="vercel.json"  {5}
   {
     "regions": ["iad1"],
     "functions": {
       "api/*": {
         "supportsCancellation": true
       }
     }
   }
   ```

   When you have enabled cancellation, anything that must be completed in the event of request cancellation should be put in a `waitUntil

2. Use the `AbortController` API in your function to cancel the request. This will allow you to clean up resources or stop long-running t

   ```ts filename="api/abort-controller/route.ts" {2, 4-7, 13}
   export async function GET(request: Request) {
     const abortController = new AbortController();

     request.signal.addEventListener('abort', () => {
       console.log('request aborted');
       abortController.abort();
     });

     const response = await fetch('https://my-backend-service.example.com', {
       headers: {
         Authorization: `Bearer ${process.env.AUTH_TOKEN}`,
       },
       signal: abortController.signal,
     });

     return new Response(response.body, {
       status: response.status,
       headers: response.headers,
     });
   }
   ```

> For \["nextjs", "other"]:

## `config` object

### `config` properties

The table below shows a highlight of the valid config options. For detailed information on all the config options, see the [Configuring F

| Property                                                          | Type     | Description
| ----------------------------------------------------------------- | -------- | -----------------------------------------------------------
| [`runtime`](/docs/functions/configuring-functions/runtime)        | `string` | This optional property defines the runtime to use, and if
| [`regions`](/docs/functions/configuring-functions/region)         | `string` | This optional property and can be used to specify the [reg
| [`maxDuration`](/docs/functions/configuring-functions/duration) | `int`    | This optional property can be used to specify the maximum

> For \["nextjs-app"]:

## Route segment config
   ```

To configure your function when using the App Router in Next.js, you use [segment options](https://nextjs.org/docs/app/api-reference/file

```ts filename="app/api/example/route.ts" framework=all
export const runtime = 'nodejs';
export const maxDuration = 15;
```

```js filename="app/api/example/route.ts" framework=all
export const maxDuration = 15;
```

The table below shows a highlight of the valid config options. For detailed information on all the config options, see the [Configuring F

| Property | Type | Description |
| --- | --- | --- |
| [`runtime`](/docs/functions/configuring-functions/runtime) | `string` | This optional property defines the runtime to use, and i |
| [`preferredRegion`](/docs/functions/configuring-functions/region) | `string` | This optional property and can be used to specify the [r |
| [`maxDuration`](/docs/functions/configuring-functions/duration) | `int` | This optional property can be used to specify the maximu |

## `SIGTERM` signal

> **💡 Note:** This feature is supported on the Node.js and Python runtimes.

A `SIGTERM` signal is sent to a function when it is about to be terminated, such as during scale-down events. This allows you to perform

Your code can run for up to 500 milliseconds after receiving a `SIGTERM` signal. After this period, the function instance will be termina

```ts filename="api/hello.ts" framework=all
process.on('SIGTERM', () => {
  // Perform cleanup operations here
});
```

```js filename="api/hello.js" framework=all
process.on('SIGTERM', () => {
  // Perform cleanup operations here
});
```

## The `@vercel/functions` package

The `@vercel/functions` package provides a set of helper methods and utilities for working with Vercel Functions.

### Helper methods

- [**`waitUntil()`**](/docs/functions/functions-api-reference/vercel-functions-package#waituntil): This method allows you to extend the l
- [**`getEnv`**](/docs/functions/functions-api-reference/vercel-functions-package#getenv): This function retrieves System Environment Var
- [**`geolocation()`**](/docs/functions/functions-api-reference/vercel-functions-package#geolocation): Returns location information for t
- [**`ipAddress()`**](/docs/functions/functions-api-reference/vercel-functions-package#ipaddress): Extracts the IP address of the request
- [**`invalidateByTag()`**](/docs/functions/functions-api-reference/vercel-functions-package#invalidatebytag): Marks a cache tag as stale
- [**`dangerouslyDeleteByTag()`**](/docs/functions/functions-api-reference/vercel-functions-package#dangerouslydeletebytag): Marks a cach
- [**`invalidateBySrcImage()`**](/docs/functions/functions-api-reference/vercel-functions-package#invalidatebysrcimage): Marks all cached
- [**`dangerouslyDeleteBySrcImage()`**](/docs/functions/functions-api-reference/vercel-functions-package#dangerouslydeletebysrcimage): Ma
- [**`getCache()`**](/docs/functions/functions-api-reference/vercel-functions-package#getcache): Obtain a [`RuntimeCache`](/docs/function

See the [`@vercel/functions`](/docs/functions/functions-api-reference/vercel-functions-package) documentation for more information.

## The `@vercel/oidc` package

> **💡 Note:** The `@vercel/oidc` package was previously provided by
> `@vercel/functions/oidc`.

The `@vercel/oidc` package provides helper methods and utilities for working with OpenID Connect (OIDC) tokens.

### OIDC Helper methods

- [**`getVercelOidcToken()`**](/docs/functions/functions-api-reference/vercel-functions-package#getverceloidctoken): Retrieves the OIDC t

See the [`@vercel/oidc`](/docs/functions/functions-api-reference/vercel-functions-package) documentation for more information.

## The `@vercel/oidc-aws-credentials-provider` package

> **💡 Note:** The `@vercel/oidc-aws-credentials-provider` package was previously provided by
> `@vercel/functions/oidc`.

The `@vercel/oidc-aws-credentials-provider` package provides helper methods and utilities for working with OpenID Connect (OIDC) tokens a

### AWS Helper methods

- [**`awsCredentialsProvider()`**](/docs/functions/functions-api-reference/vercel-functions-package#awscredentialsprovider): This functio

See the [`@vercel/oidc-aws-credentials-provider`](/docs/functions/functions-api-reference/vercel-functions-package) documentation for mor

## More resources

- [Streaming Data: Learn about streaming on Vercel](/docs/functions/streaming)

---
title: "@vercel/functions API Reference (Node.js)"
description: "Learn about available APIs when working with Vercel Functions."
last_updated: "2026-01-16T02:19:31.508Z"
source: "https://vercel.com/docs/functions/functions-api-reference/vercel-functions-package"
---

# @vercel/functions API Reference (Node.js)

## Install and use the package

1. Install the `@vercel/functions` package:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i @vercel/functions
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @vercel/functions
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @vercel/functions
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i @vercel/functions
    ```
  </Code>
</CodeBlock>

2. Import the `@vercel/functions` package (non-Next.js frameworks or Next.js versions below 15.1):

```ts {1} filename="api/hello.ts" framework=other
import { waitUntil, attachDatabasePool } from '@vercel/functions';

export default {
  fetch(request: Request) {
    // ...
  },
};
```

```js {1} filename="api/hello.js" framework=other
import { waitUntil, attachDatabasePool } from '@vercel/functions';

export default {
  fetch(request) {
    // ...
  },
};
```

For [OIDC](/docs/functions/functions-api-reference/vercel-functions-package#oidc-methods) methods, import `@vercel/oidc`

## Usage with Next.js

If you're using **Next.js 15.1 or above**, we recommend using the built-in [`after()`](https://nextjs.org/docs/app/api-reference/function

`after()` allows you to schedule work that runs **after** the response has been sent or the prerender has completed. This is especially u

```ts v0="build" filename="app/api/hello/route.ts"
import { after } from 'next/server';

export async function GET(request: Request) {
  const country = request.headers.get('x-vercel-ip-country') || 'unknown';

  // Returns a response immediately
  const response = new Response(`You're visiting from ${country}`);

  // Schedule a side-effect after the response is sent
  after(async () => {
    // For example, log or increment analytics in the background
    await fetch(
      `https://my-analytics-service.example.com/log?country=${country}`,
    );
  });

  return response;
}
```

- `after()` does **not** block the response. The callback runs once rendering or the response is finished.
- `after()` is not a [Dynamic API](https://nextjs.org/docs/app/building-your-application/rendering/server-components#dynamic-apis); calli
- If you need to configure or extend the timeout for tasks, you can use [`maxDuration`](https://nextjs.org/docs/app/api-reference/file-co
- For more usage examples (including in **Server Components**, **Server Actions**, or **Middleware**), see [after() in the Next.js docs](

## Helper methods (non-Next.js usage or older Next.js versions)

If you're **not** using Next.js 15.1 or above (or you are using other frameworks), you can use the methods from `@vercel/functions` below

### `waitUntil`

**Description**: Extends the lifetime of the request handler for the lifetime of the given Promise. The `waitUntil()` method enqueues an

Promises passed to `waitUntil()` will have the same timeout as the function itself. If the function times out, the promises will be cance

| Name | Type | Description |
| :-------- | :---------- | :---------- |
| `promise` | [`Promise`](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise) | The promise to wait

> **💡 Note:** If you're using Next.js 15.1 or above, use [`after()`](#using-after-in-nextjs)
> from `next/server` instead. Otherwise, see below.

```ts v0="build" {1,9} filename="api/hello.ts"
import { waitUntil } from '@vercel/functions';
```

```
async function getBlog() {
  const res = await fetch('https://my-analytics-service.example.com/blog/1');
  return res.json();
}

export default {
  fetch(request: Request) {
    waitUntil(getBlog().then((json) => console.log({ json })));
    return new Response(`Hello from ${request.url}, I'm a Vercel Function!`);
  },
};
```

### `getEnv`

**Description**: Gets the [System Environment Variables](/docs/environment-variables/system-environment-variables#system-environment-vari

```ts filename="api/example.ts"
import { getEnv } from '@vercel/functions';

export default {
  fetch(request) {
    const { VERCEL_REGION } = getEnv();
    return new Response(`Hello from ${VERCEL_REGION}`);
  },
};
```

### `geolocation`

**Description**: Returns the location information for the incoming request, in the following way:

```json
{
  "city": "New York",
  "country": "US",
  "flag": "🇺🇸",
  "countryRegion": "NY",
  "region": "iad1",
  "latitude": "40.7128",
  "longitude": "-74.0060",
  "postalCode": "10001"
}
```

| Name | Type | Description |
| :--- | :--- | :--- |
| `request` | [`Request`](https://developer.mozilla.org/en-US/docs/Web/API/Request) | The incoming request object which provides the IP |

```ts filename="api/example.ts"
import { geolocation } from '@vercel/functions';

export default {
  fetch(request) {
    const details = geolocation(request);
    return Response.json(details);
  },
};
```

### `ipAddress`

**Description**: Returns the IP address of the request from the headers.

| Name | Type | Description |
| :--- | :--- | :--- |
| `request` | [`Request`](https://developer.mozilla.org/en-US/docs/Web/API/Request) | The incoming request object which provides the IP |

```ts filename="api/example.ts"
import { ipAddress } from '@vercel/functions';

export default {
  fetch(request) {
    const ip = ipAddress(request);
    return new Response(`Your ip is ${ip}`);
  },
};
```

### `invalidateByTag`

**Description**: Marks a cache tag as stale, causing cache entries associated with that tag to be revalidated in the background on the ne

| Name | Type | Description |
| :--- | :--- | :--- |
| `tag` | `string` or `string[]` | The cache tag (or multiple tags) to invalidate. |

```ts filename="api/example.ts"
import { invalidateByTag } from '@vercel/functions';

export default {
  async fetch(request) {
    await invalidateByTag('my-tag-name');
    return new Response('Success');
  },
};
```

### `dangerouslyDeleteByTag`

**Description**: Marks a cache tag as deleted, causing cache entries associated with that tag to be revalidated in the foreground on the

| Name | Type | Description |
| :-------- | :---------------------------------------- | :---------------------------------------------------------------------------- |
| `tag` | `string` or `string[]` | The cache tag (or multiple tags) to dangerously delete. |
| `options` | `{ revalidationDeadlineSeconds: number }` | The time in seconds before the delete deadline. If a request is made before the |

```ts filename="api/example.ts"
import { dangerouslyDeleteByTag } from '@vercel/functions';

export default {
  async fetch(request) {
    await dangerouslyDeleteByTag('my-tag-name', {
      revalidationDeadlineSeconds: 10,
    });
    return new Response('Success');
  },
};
```

### `invalidateBySrcImage`

**Description**: Marks all cached content associated with a source image as stale, causing those cache entries to be revalidated in the b

Learn more about [purging Vercel CDN cache](/docs/cdn-cache/purge).

| Name | Type | Description |
| :--------- | :------- | :------------------------------ |
| `srcImage` | `string` | The source image to invalidate. |

```ts filename="api/example.ts"
import { invalidateBySrcImage } from '@vercel/functions';

export default {
  async fetch(request) {
    await invalidateBySrcImage('/api/avatar/1');
    return new Response('Success');
  },
};
```

### `dangerouslyDeleteBySrcImage`

**Description**: Marks all cached content associated with a source image as deleted, causing those cache entries to be revalidated in the

Learn more about [purging Vercel CDN cache](/docs/cdn-cache/purge).

| Name | Type | Description |
| :--------- | :---------------------------------------- | :---------------------------------------------------------------------------- |
| `srcImage` | `string` | The source image to dangerously delete. |
| `options` | `{ revalidationDeadlineSeconds: number }` | The time in seconds before the delete deadline. If a request is made before th |

```ts filename="api/example.ts"
import { dangerouslyDeleteBySrcImage } from '@vercel/functions';

export default {
  async fetch(request) {
    await dangerouslyDeleteBySrcImage('/api/avatar/1', {
      revalidationDeadlineSeconds: 10,
    });
    return new Response('Success');
  },
};
```

### `addCacheTag`

**Description**: Adds one or more tags to a cached response, so that you can later invalidate the cache associated with these tag(s) usin

| Name | Type | Description |
| :---- | :---------------------- | :------------------------------------------- |
| `tag` | `string` or `string[]` | One or more tags to add to the cached response. |

```ts filename="api/example.ts"
import { addCacheTag } from '@vercel/functions';

export default {
  async fetch(request) {
    const url = new URL(request.url);
    const id = url.searchParams.get('id');
    const res = await fetch(`https://example.com/api/${id}`);
    const { name, description } = await res.json();
    await addCacheTag(`product-${id}`);
    return Response.json(
      { name, description },
      {
        headers: {
          'Cache-Control': 'public, max-age=86400',
        },
      }
    );
  },
};
```

#### Limits

- A cached response can have a maximum of 128 tags.
- The maximum tag length is 256 bytes (UTF-8 encoded).
- Tag names cannot contain commas.

### `getCache`

**Description**: Returns a `RuntimeCache` object that allows you to interact with the Vercel Runtime Cache in any Vercel region. Use this

| Name | Type | Description |
| --- | --- | --- |
| `keyHashFunction` | `(key: string) => string` | Optional custom hash function for generating keys. |
| `namespace` | `String` | Optional namespace to prefix cache keys. |
| `namespaceSeparator` | `String` | Optional separator string for the namespace. |

#### Specification

`RuntimeCache` provides the following methods:

| Method | Description |
| :--- | :--- |
| `get` | Retrieves a value from the Vercel Runtime Cache. |
| `set` | Stores a value in the Vercel Runtime Cache with optional `ttl` and/or `tags`. The `name` option allows a human-readable l |
| `delete` | Removes a value from the Vercel Runtime Cache by key |
| `expireTag` | Expires all cache entries associated with one or more tags |

```ts filename="api/example.ts"
import { getCache } from '@vercel/functions';

export default {
  async fetch(request) {
    const cache = getCache();

    // Get a value from cache
    const value = await cache.get('somekey');

    if (value) {
      return new Response(JSON.stringify(value));
    }

    const res = await fetch('https://api.vercel.app/blog');
    const originValue = await res.json();

    // Set a value in cache with TTL and tags
    await cache.set('somekey', originValue, {
      ttl: 3600, // 1 hour in seconds
      tags: ['example-tag'],
    });

    return new Response(JSON.stringify(originValue));
  },
};
```

After assigning tags to your cached data, use the `expireTag` method to invalidate all cache entries associated with that tag. This opera

```ts filename="app/actions.ts"
'use server';

import { getCache } from '@vercel/functions';

export default async function action() {
  await getCache().expireTag('blog');
}
```

#### Limits and usage

The Runtime Cache is isolated per Vercel project and deployment environment (`preview` and `production`). Cached data is persisted across

The Runtime Cache API does not have first class integration with [Incremental Static Regeneration](/docs/incremental-static-regeneration)

- Runtime Cache entry tags will not apply to ISR pages, so you cannot use expireTag to invalidate both caches.
- Runtime Cache entry TTLs will have no effect on the ISR revalidation time and
- Next.js's `revalidatePath` and `revalidateTag`API does not invalidate the Runtime Cache.

The following Runtime Cache limits apply:

- The maximum size of an item in the cache is 2 MB. Items larger than this will not be cached.
- A cached item can have a maximum of 128 tags.
- The maximum tag length is 256 bytes.

Usage of the Vercel Runtime Cache is charged, learn more about pricing in the [regional pricing docs](/docs/pricing/regional-pricing).

### Database Connection Pool Management

#### `attachDatabasePool`

Call this function right after creating a database pool to ensure proper connection
management in [Fluid Compute](/docs/fluid-compute). This function ensures that idle pool clients are
properly released before functions suspend.

Supports PostgreSQL (pg), MySQL2, MariaDB, MongoDB, Redis (ioredis), Cassandra (cassandra-driver), and other compatible pool types.

| Name | Type | Description |
| :--- | :--- | :--- |
| `dbPool` | `DbPool` | The database pool object. |

```ts {8} filename="api/database.ts" framework=all
import { Pool } from 'pg';
import { attachDatabasePool } from '@vercel/functions';

const pool = new Pool({
  connectionString: process.env.DATABASE_URL,
```

```
});

attachDatabasePool(pool);

export default {
  async fetch() {
    const client = await pool.connect();
    try {
      const result = await client.query('SELECT NOW()');
      return Response.json(result.rows[0]);
    } finally {
      client.release();
    }
  },
};
```

### OIDC methods

#### `awsCredentialsProvider`

> **💡 Note:** This function has moved from @vercel/functions/oidc to
> @vercel/oidc-aws-credentials-provider. It is now deprecated from
> @vercel/functions and will be removed in a future release.

**Description**: Obtains the Vercel OIDC token and creates an AWS credential provider function that gets AWS credentials by calling the S

| Name | Type | Description |
| --- | --- | --- |
| `roleArn` | `string` | ARN of the role that the caller is assuming. |
| `clientConfig` | `Object` | Custom STS client configurations overriding the default ones. |
| `clientPlugins` | `Array` | Custom STS client middleware plugin to modify the client default behavior. |
| `roleAssumerWithWebIdentity` | `Function` | A function that assumes a role with web identity and returns a promise fulfilled with crede |
| `roleSessionName` | `string` | An identifier for the assumed role session. |
| `providerId` | `string` | The fully qualified host component of the domain name of the identity provider. |
| `policyArns` | `Array` | ARNs of the IAM managed policies that you want to use as managed session policies. |
| `policy` | `string` | An IAM policy in JSON format that you want to use as an inline session policy. |
| `durationSeconds` | `number` | The duration, in seconds, of the role session. Defaults to 3600 seconds. |

```ts filename="api/example.ts"
import * as s3 from '@aws-sdk/client-s3';
import { awsCredentialsProvider } from '@vercel/oidc-aws-credentials-provider';

const s3Client = new s3.S3Client({
  credentials: awsCredentialsProvider({
    roleArn: process.env.AWS_ROLE_ARN,
  }),
});
```

#### `getVercelOidcToken`

> **💡 Note:** This function has moved from @vercel/functions/oidc to @vercel/oidc. It is now
> deprecated from @vercel/functions and will be removed in a future release.

**Description**: Returns the OIDC token from the request context or the environment variable. This function first checks if the OIDC toke
`VERCEL_OIDC_TOKEN`. If it is not found there, it retrieves the token from the request context headers.

```ts filename="api/example.ts"
import { ClientAssertionCredential } from '@azure/identity';
import { CosmosClient } from '@azure/cosmos';
import { getVercelOidcToken } from '@vercel/oidc';

const credentialsProvider = new ClientAssertionCredential(
  process.env.AZURE_TENANT_ID,
  process.env.AZURE_CLIENT_ID,
  getVercelOidcToken,
);

const cosmosClient = new CosmosClient({
  endpoint: process.env.COSMOS_DB_ENDPOINT,
  aadCredentials: credentialsProvider,
});

export const GET = () => {
  const container = cosmosClient
    .database(process.env.COSMOS_DB_NAME)
    .container(process.env.COSMOS_DB_CONTAINER);
  const items = await container.items.query('SELECT * FROM f').fetchAll();
  return Response.json({ items: items.resources });
};
```

---
title: "vercel.functions API Reference (Python)"
description: "Learn about available APIs when working with Vercel Functions in Python."
last_updated: "2026-01-16T02:19:31.518Z"
source: "https://vercel.com/docs/functions/functions-api-reference/vercel-sdk-python"
---

# vercel.functions API Reference (Python)

## Install and use the package

1. Install the `vercel` package:

   ```python
   pip install vercel
   ```

2. Import the `vercel.functions` package:

   ```python
   from vercel.functions import get_env
   ```

## Helper methods

### `get_env`

**Description**: Gets the [System Environment Variables](/docs/environment-variables/system-environment-variables#system-environment-vari

```python filename="src/example.py"
from vercel.functions import get_env

print(get_env().VERCEL_REGION)
```

### `geolocation`

**Description**: Returns the location information for the incoming request, in the following way:

```json
{
  "city": "New York",
  "country": "US",
  "flag": "🇺🇸",
  "countryRegion": "NY",
  "region": "iad1",
  "latitude": "40.7128",
  "longitude": "-74.0060",
  "postalCode": "10001"
}
```

| Name | Type | Description |
| :--- | :--- | :--- |
| `request_or_headers` | `RequestLike \| HeadersLike` | The incoming request object which provides the IP |

```python filename="src/main.py"
from fastapi import FastAPI, Request
from vercel.functions import geolocation

app = FastAPI()

@app.get("/api/geo")
async def geo_info(request: Request):
    info = geolocation(request)
    return info
```

### `ip_address`

**Description**: Returns the IP address of the request from the headers.

| Name | Type | Description |
| :--- | :--- | :--- |
| `request_or_headers` | `RequestLike \| HeadersLike` | The incoming request object which provides the IP |

```python filename="src/main.py"
from fastapi import FastAPI, Request
from vercel.functions import ip_address

app = FastAPI()

@app.get("/api/ip")
async def get_ip_address(request: Request):
    ip = ip_address(request)  # you can also pass request.headers
    return {"ip": ip}
```

### `RuntimeCache`

**Description**: Allows you to interact with the Vercel Runtime Cache in any Vercel region. Use this for storing and retrieving data acro

| Name | Type | Description |
| --- | --- | --- |
| `key_hash_function` | `Callable[[str], str]` | Optional custom hash function for generating keys. |
| `namespace` | `str` | Optional namespace to prefix cache keys. |
| `namespace_separator` | `str` | Optional separator string for the namespace. |

#### Specification

`RuntimeCache | AsyncRuntimeCache` provide the following methods:

| Method | Description |
| :--- | :--- |
| `get` | Retrieves a value from the Vercel Runtime Cache. |
| `set` | Stores a value in the Vercel Runtime Cache with optional `ttl` and/or `tags`. The `name` option allows a human-readable |
| `delete` | Removes a value from the Vercel Runtime Cache by key |
| `expire_tag` | Expires all cache entries associated with one or more tags |

Use `AsyncRuntimeCache` in async code. It has the same API and uses the same underlying cache as `RuntimeCache`, and exposes awaitable me

```python filename="src/main.py"
import requests
import httpx
from fastapi import FastAPI, Request
from vercel.functions import RuntimeCache, AsyncRuntimeCache
```

```
app = FastAPI()
cache = RuntimeCache()
acache = AsyncRuntimeCache()

@app.get("/blog")
def get_blog(request: Request):
    key = "blog"
    value = cache.get(key)
    if value is not None:
        return value

    res = requests.get("https://api.vercel.app/blog")
    origin_value = res.json()
    cache.set(key, origin_value, {"ttl": 3600, "tags": ["blog"]})

    return origin_value

@app.get("/blog-async")
async def get_blog_async(request: Request):
    key = "blog"
    value = await acache.get(key)
    if value is not None:
        return value

    async with httpx.AsyncClient() as client:
        res = await client.get("https://api.vercel.app/blog")
        origin_value = res.json()
    await acache.set(key, origin_value, {"ttl": 3600, "tags": ["blog"]})
    return origin_value
```

After assigning tags to your cached data, use the `expire_tag` method to invalidate all cache entries associated with that tag. This oper

```python filename="src/main.py"
from fastapi import FastAPI, Request
from vercel.functions import RuntimeCache

app = FastAPI()
cache = RuntimeCache()

@app.get("/expire-blog")
def expire_blog(request: Request):
    cache.expire_tag("blog")
    return {"ok": True}

```

#### Limits and usage

The Runtime Cache is isolated per Vercel project and deployment environment (`preview` and `production`). Cached data is persisted across

The Runtime Cache API does not have first class integration with [Incremental Static Regeneration](/docs/incremental-static-regeneration)

- Runtime Cache entry tags will not apply to ISR pages, so you cannot use `expire_tag` to invalidate both caches.
- Runtime Cache entry TTLs will have no effect on the ISR revalidation time and

The following Runtime Cache limits apply:

- The maximum size of an item in the cache is 2 MB. Items larger than this will not be cached.
- A cached item can have a maximum of 128 tags.
- The maximum tag length is 256 bytes.

Usage of the Vercel Runtime Cache is charged, learn more about pricing in the [regional pricing docs](/docs/pricing/regional-pricing).

---
title: "Vercel Functions Limits"
description: "Learn about the limits and restrictions of using Vercel Functions with the Node.js runtime."
last_updated: "2026-01-16T02:19:31.593Z"
source: "https://vercel.com/docs/functions/limitations"
---

# Vercel Functions Limits

The table below outlines the limits and restrictions of using Vercel Functions with the Node.js runtime:

| Feature | Node.js |
| --- | --- |
| [Maximum memory](/docs/functions/limitations#memory-size-limits) | Hobby: 2 GB, Pro and Ent: 4 GB |
| [Maximum duration](/docs/functions/limitations#max-duration) | Hobby: 300s (default) - [configurable up to 300s](/d |
| [Size](/docs/functions/runtimes#bundle-size-limits) (after gzip compression) | 250 MB |
| [Concurrency](/docs/functions/concurrency-scaling#automatic-concurrency-scaling) | Auto-scales up to 30,000 (Hobby and Pro) or 100,000+ |
| [Cost](/docs/functions/runtimes) | Pay for active CPU time and provisioned memory time |
| [Regions](/docs/functions/runtimes#location) | Executes region-first, [can customize location](/doc |
| [API Coverage](/docs/functions/limitations#api-support) | Full Node.js coverage |
| [File descriptors](/docs/functions/limitations#file-descriptors) | 1,024 shared across concurrent executions (including |

## Functions name

The following limits apply to the function's name when using [Node.js runtime](/docs/functions/runtimes/node-js):

- Maximum length of 128 characters. This includes the extension of the file (e.g. `apps/admin/api/my-function.js` is 29 characters)
- No spaces are allowed. Replace them with a `-` or `_` (e.g. `api/my function.js` isn't allowed)

## Bundle size limits

Vercel places restrictions on the maximum size of the deployment bundle for functions to ensure that they execute in a timely manner.

For Vercel Functions, the maximum uncompressed size is **250 MB** including layers which are automatically used depending on [runtimes](/

You can use [`includeFiles` and `excludeFiles`](/docs/project-configuration#functions) to specify items which may affect the function siz

## Max duration

This refers to the longest time a function can process an HTTP request before responding.

While Vercel Functions have a default duration, this duration can be extended using the [maxDuration config](/docs/functions/configuring-

With [fluid compute](/docs/fluid-compute) enabled, Vercel Functions have the following defaults and maximum limits (applies to the Node.j

### Node.js and python runtimes

| | Default | Maximum |
| --- | --- | --- |
| Hobby | 300s (5 minutes) | 300s (5 minutes) |
| Pro | 300s (5 minutes) | 800s (13 minutes) |
| Enterprise | 300s (5 minutes) | 800s (13 minutes) |

### Edge runtime

Vercel Functions using the [Edge runtime](/docs/functions/runtimes/edge) must begin sending a response within 25 seconds to maintain stre

## Memory size limits

Vercel Functions have the following defaults and maximum limits:

| | Default | Maximum |
| --- | --- | --- |
| Hobby | 2 GB / 1 vCPU | 2 GB / 1 vCPU |
| Pro /Enterprise | 2 GB / 1 vCPU | 4 GB / 2 vCPU |

Users on Pro and Enterprise plans can [configure the default memory size](/docs/functions/configuring-functions/memory#setting-your-defau

The maximum size for a Function includes your JavaScript code, imported libraries and files (such as fonts), and all files bundled in the

If you reach the limit, make sure the code you are importing in your function is used
and is not too heavy. You can use a package size checker tool like [bundle](https://bundle.js.org/) to
check the size of a package and search for a smaller alternative.

## Request body size

In Vercel, the request body size is the maximum amount of data that can be included in the body of a request to a function.

The maximum payload size for the request body or the response body of a Vercel Function is **4.5 MB**. If a Vercel Function receives a pa

## File descriptors

File descriptors are unique identifiers that the operating system uses to track and manage open resources like files, network connections

Vercel Functions have a limit of **1,024 file descriptors** shared across all concurrent executions. This limit includes file descriptors

File descriptors are used for:

- Open files
- Network connections (TCP sockets, HTTP requests)
- Database connections
- File system operations

If your function exceeds this limit, you might encounter errors related to "too many open files" or similar resource exhaustion issues.

To manage file descriptors effectively, consider the following:

- Close files, database connections, and HTTP connections when they're no longer needed
- Use connection pooling for database connections
- Implement proper resource cleanup in your function code

## API support

| | Node.js runtime (and more) |
| --- | --- |
| Geolocation data | [Yes](/docs/headers/request-headers#x-vercel-ip-country) |
| Access request headers | Yes |
| Cache responses | [Yes](/docs/cdn-cache#using-vercel-functions) |

## Cost and usage

The Hobby plan offers functions for free, within [limits](/docs/limits). The Pro plan extends these limits, and charges usage based on ac

Active CPU time is based on the amount of CPU time your code actively consumes, measured in milliseconds. Waiting for I/O (e.g. calling A

It is important to make sure you've set a reasonable [maximum duration](/docs/functions/configuring-functions/duration) for your function

## Environment variables

If you have [fluid compute](/docs/fluid-compute) enabled, the following environment variables are not accessible and you cannot log them:

- `AWS_EXECUTION_ENV`
- `AWS_LAMBDA_EXEC_WRAPPER`
- `AWS_LAMBDA_FUNCTION_MEMORY_SIZE`
- `AWS_LAMBDA_FUNCTION_NAME`
- `AWS_LAMBDA_FUNCTION_VERSION`
- `AWS_LAMBDA_INITIALIZATION_TYPE`
- `AWS_LAMBDA_LOG_GROUP_NAME`
- `AWS_LAMBDA_LOG_STREAM_NAME`
- `AWS_LAMBDA_RUNTIME_API`
- `AWS_XRAY_CONTEXT_MISSING`
- `AWS_XRAY_DAEMON_ADDRESS`
- `LAMBDA_RUNTIME_DIR`
- `LAMBDA_TASK_ROOT`

- `_AWS_XRAY_DAEMON_ADDRESS`
- `_AWS_XRAY_DAEMON_PORT`
- `_HANDLER`
- `_LAMBDA_TELEMETRY_LOG_FD`

---
title: "Vercel Function Logs"
description: "Use runtime logs to debug and monitor your Vercel Functions."
last_updated: "2026-01-16T02:19:31.601Z"
source: "https://vercel.com/docs/functions/logs"
---

# Vercel Function Logs

Vercel Functions allow you to debug and monitor your functions using runtime logs. Users on the Pro and Enterprise plans can use Vercel's

## Runtime Logs

You can view [runtime logs](/docs/runtime-logs#what-are-runtime-logs) for all Vercel Functions in real-time from [the **Logs** tab](/docs

When your function is [streaming](/docs/functions/streaming-functions), you'll get the following:

- You can [view the logs](/docs/runtime-logs#view-runtime-logs) in real-time from the **Logs** tab of your project's dashboard.
- Each action of writing to standard output, such as using `console.log`, results in a separate log entry.
- Each of the logs are 256 KB **per line**.
- The path in streaming logs will be prefixed with a forward slash (`/`).

For more information, see [Runtime Logs](/docs/runtime-logs).

> **💡 Note:** These changes in the frequency and format of logs will affect Log Drains. If
> you are using Log Drains we recommend ensuring that your ingestion can handle
> both the new format and frequency.

### Number of logs per request

When a Function on a specific path receives a user request, you *may* see more than one log when the application renders or regenerates t

This can occur in the following situations:

1. When a new page is rendered
2. When you are using [Incremental Static Regeneration (ISR)](/docs/incremental-static-regeneration)

In the case of ISR, multiple logs are the result of:

- A [stale](/docs/cdn-cache#cache-invalidation) page having to be regenerated. For stale pages, both HTML (for direct browser navigation)
- On-demand ISR happening with `fallback` set as [`blocking`](/docs/incremental-static-regeneration/quickstart). During on-demand ISR, th

### Next.js logs

In Next.js projects, logged functions include API Routes (those defined in  or ).

Pages that use SSR, such as those that call `getServerSideProps` or export [`revalidate`](https://nextjs.org/docs/app/building-your-appli

---
title: "Vercel Functions"
description: "Vercel Functions allow you to run server-side code without managing a server."
last_updated: "2026-01-16T02:19:31.615Z"
source: "https://vercel.com/docs/functions"
---

# Vercel Functions

Vercel Functions lets you run server-side code without managing servers. They adapt automatically to user demand, handle connections to A

When you deploy your application, Vercel automatically sets up the tools and optimizations for your chosen [framework](/docs/frameworks).

## Getting started

To get started with creating your first function, copy the code below:

```ts filename="api/hello.ts" framework=all
export default {
  fetch(request: Request) {
    return new Response('Hello from Vercel!');
  },
};
```

```js filename="api/hello.js" framework=all
export default {
  fetch(request) {
    return new Response('Hello from Vercel!');
  },
};
```

While using `fetch` is the recommended way to create a Vercel Function, you can still use HTTP methods like `GET` and `POST`.

```ts v0="build" filename="app/api/hello/route.ts" framework=nextjs-app
export function GET(request: Request) {
  return new Response('Hello from Vercel!');
}
```

```js v0="build" filename="app/api/hello/route.js" framework=nextjs-app
export function GET(request) {
  return new Response('Hello from Vercel!');
```

}
```

```ts v0="build" filename="pages/api/hello.ts" framework=nextjs
export function GET(request: Request) {
  return new Response('Hello from Vercel!');
}
```

```js v0="build" filename="pages/api/hello.js" framework=nextjs
export function GET(request) {
  return new Response('Hello from Vercel!');
}
```

```ts filename="api/hello.ts" framework=other
export function GET(request: Request) {
  return new Response('Hello from Vercel!');
}
```

```js filename="api/hello.js" framework=other
export function GET(request) {
  return new Response('Hello from Vercel!');
}
```

> For \['nextjs']:

When using Next.js Pages, we recommend using [Route Handlers in the App Router](https://nextjs.org/docs/app/building-your-application/rou

To learn more, see the [quickstart](/docs/functions/quickstart) or [deploy a template](/templates).

## Functions lifecycle

Vercel Functions run in a single [region](/docs/functions/configuring-functions/region) by default, although you can configure them to ru

When a user sends a request to your site, Vercel can automatically run a function based on your application code. You do not need to mana

Vercel creates a new function invocation for each incoming request. If another request arrives soon after the previous one, Vercel [reuse

By allowing concurrent execution within the same instance (and so using idle time for compute), fluid compute reduces cold starts, lowers

### Functions and your data source

**Functions** should always execute close to where your data source is to reduce latency. By default, functions using the Node.js runtime

## Viewing Vercel Function metrics

You can view various performance and cost efficiency metrics using Vercel Observability:

1. Choose your project from the [dashboard](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D\&title=Go+to+dashboard).
2. Click on the **Observability** tab and select the **Vercel Functions** section.
3. Click on the chevron icon to expand and see all charts.

From here, you'll be able to see total consumed and saved GB-Hours, and the ratio of the saved usage. When you have [fluid](/docs/fluid-c

## Pricing

Vercel Functions are priced based on active CPU, provisioned memory, and invocations. See the [fluid compute pricing](/docs/functions/usag

If your project is not using fluid compute, see the [legacy pricing documentation](/docs/functions/usage-and-pricing/legacy-pricing) for '

## Related

- [What is compute?](/docs/fundamentals/what-is-compute)
- [What is streaming?](/docs/fundamentals/what-is-streaming)
- [Fluid compute](/docs/fluid-compute)
- [Runtimes](/docs/functions/runtimes)
- [Configuring functions](/docs/functions/configuring-functions)
- [Streaming](/docs/functions/streaming-functions)
- [Limits](/docs/functions/limitations)
- [Functions logs](/docs/functions/logs)

---
title: "Getting started with Vercel Functions"
description: "Build your first Vercel Function in a few steps."
last_updated: "2026-01-16T02:19:31.524Z"
source: "https://vercel.com/docs/functions/quickstart"
---

# Getting started with Vercel Functions

In this guide, you'll learn how to get started with Vercel Functions using your favorite [frontend framework](/docs/frameworks) (or no fr

## Prerequisites

- You can use an existing project or create a new one. If you don't have one, you can run the following terminal command to create a Next

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i
    ```
    ```

  </Code>
  <Code tab="npm">
    ```bash
    npm i
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i
    ```
  </Code>
</CodeBlock>

## Create a Vercel Function

Open the code block in  for a walk through on creating a Vercel Function with the below code, or copy the code into your project. The fun

```ts v0="build" filename="app/api/hello/route.ts" framework=nextjs-app
export async function GET(request: Request) {
  const response = await fetch('https://api.vercel.app/products');
  const products = await response.json();
  return Response.json(products);
}
```

```js v0="build" filename="app/api/hello/route.js" framework=nextjs-app
export async function GET(request) {
  const response = await fetch('https://api.vercel.app/products');
  const products = await response.json();
  return Response.json(products);
}
```

```ts v0="build" filename="pages/api/hello.ts" framework=nextjs
export async function GET(request: Request) {
  const response = await fetch('https://api.vercel.app/products');
  const products = await response.json();
  return Response.json(products);
}
```

```js v0="build" filename="pages/api/hello.js" framework=nextjs
export async function GET(request) {
  const response = await fetch('https://api.vercel.app/products');
  const products = await response.json();
  return Response.json(products);
}
```

```ts filename="api/hello" framework=other
export default {
  async fetch(request: Request) {
    const response = await fetch('https://api.vercel.app/products');
    const products = await response.json();
    return Response.json(products);
  },
};
```

```js filename="api/hello" framework=other
export default {
  async fetch(request) {
    const response = await fetch('https://api.vercel.app/products');
    const products = await response.json();
    return Response.json(products);
  },
};
```

While using `fetch` is the recommended way to create a Vercel Function, you can still use HTTP methods like `GET` and `POST`.

## Next steps

Now that you have set up a Vercel Function, you can explore the following topics to learn more:

- [Explore the functions API reference](/docs/functions/functions-api-reference): Learn more about creating a Vercel Function.
- [Learn about streaming functions](/docs/functions/streaming-functions): Learn how to fetch streamable data with Vercel Functions.
- [Choosing a Runtime](/docs/functions/runtimes): Learn more about the differences between the Node.js and Edge runtimes.
- [Configuring Functions](/docs/functions/configuring-functions): Learn about the different options for configuring a Vercel Function.

---
title: "Using the Bun Runtime with Vercel Functions"
description: "Learn how to use the Bun runtime with Vercel Functions to create fast, efficient functions."
last_updated: "2026-01-16T02:19:31.542Z"
source: "https://vercel.com/docs/functions/runtimes/bun"
---

# Using the Bun Runtime with Vercel Functions

Bun is a fast, all-in-one JavaScript runtime that serves as an alternative to Node.js.

Bun provides Node.js API compatibility and is generally faster than Node.js for CPU-bound tasks. It includes a bundler, test runner, and

## Configuring the runtime

For all frameworks, including Next.js, you can configure the runtime in your `vercel.json` file using the [`bunVersion`](/docs/project-co

Once you configure the runtime version, Vercel manages the Bun minor and patch versions automatically, meaning you only need to set the m

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "bunVersion": "1.x"
}
```

> **💡 Note:** Vercel manages the Bun minor and patch versions automatically. `1.x` is the
> only valid value currently.

## Framework-specific considerations

### Next.js

When using Next.js, and [ISR](/docs/incremental-static-regeneration), you must change your `build` and `dev` commands in your package.jso

**Before:**

```json filename="package.json"
{
  "scripts": {
    "dev": "next dev",
    "build": "next build"
  }
}
```

**After:**

```json filename="package.json"
{
  "scripts": {
    "dev": "bun run --bun next dev",
    "build": "bun run --bun next build"
  }
}
```

### Routing Middleware

The Bun runtime works with [Routing Middleware](/docs/routing-middleware) the same way as the Node.js runtime once you set the `bunVersio

## Feature support

The Bun runtime on Vercel supports most Node.js features. The main differences relate to automatic source maps, bytecode caching, and req

See the table below for a detailed comparison:

## Supported APIs

Vercel Functions using the Bun runtime support [most Node.js APIs](https://bun.sh/docs/runtime/nodejs-apis), including standard Web APIs

## Using TypeScript with Bun

Bun has built-in TypeScript support with zero configuration required. The runtime supports files ending with `.ts` inside of the `/api` d

```typescript filename="api/hello.ts"
export default {
  async fetch(request: Request) {
    const url = new URL(request.url);
    const name = url.searchParams.get('name') || 'World';

    return Response.json({ message: `Hello ${name}!` });
  },
};
```

## Performance considerations

Bun is generally faster than Node.js, especially for CPU-bound tasks. Performance varies by workload, and in some cases Node.js may be fa

## When to use Bun

Bun is best suited for new workloads where you want a fast, all-in-one toolkit with built-in support for TypeScript, JSX, and modern Java

- You want faster execution for CPU-bound tasks
- You prefer zero-config TypeScript and JSX support
- You're starting a new project and want to use modern tooling

Consider using Node.js instead if:

- Node.js is already installed on your project and is working for you
- You need automatic source maps for debugging
- You need request metrics on the `node:http` or `node:https` modules

Both runtimes run on [Fluid compute](/docs/fluid-compute) and support the same core Vercel Functions features.

---
title: "Edge Functions"
description: "Run minimal code at the network edge."
last_updated: "2026-01-16T02:19:31.634Z"
source: "https://vercel.com/docs/functions/runtimes/edge/edge-functions"
---

# Edge Functions

Edge Functions are Vercel Functions that run on the Edge Runtime, a minimal JavaScript runtime that exposes a set of Web Standard APIs.

- **Lightweight runtime**: With a smaller API surface area and using V8 isolates, Edge runtime-powered functions have a slim runtime with

- **Globally distributed by default**: Vercel deploys all Edge Functions globally across its CDN, which means your site's visitors will g

> **⚠️ Warning:** We recommend migrating from edge to Node.js for improved performance and
> reliability. Both runtimes run on [Fluid compute](/docs/fluid-compute) with
> [Active CPU pricing](/docs/functions/usage-and-pricing).

## Edge Functions and your data source

**Edge Functions** execute in the region closest to the user, which could result in longer response times when the function relies on a d

To avoid these long roundtrips, you can limit your Edge Functions to [regions near your database](/docs/functions/configuring-functions/r

## Feature support

| Feature | Support Status |
| --- | --- |
| Secure Compute | Not Supported |
| [Streaming](#streaming) | Supported |
| [Cron jobs](#cron-jobs) | Supported |
| [Vercel Storage](/docs/storage) | Supported |
| [Edge Config](#edge-config) | Supported |
| OTEL | Not supported |

### Streaming

Streaming refers to the ability to send or receive data in a continuous flow.

The [Edge runtime](/docs/functions/runtimes/edge) supports streaming by default.

Edge Functions **do not** have a maximum duration, but you **must** send an *initial* response within 25 seconds. You can continue [strea

Node.js and Edge runtime streaming functions support the [`waitUntil` method](/docs/functions/functions-api-reference/vercel-functions-pa

### Cron jobs

[Cron jobs](/docs/cron-jobs) are time-based scheduling tools used to automate repetitive tasks. When a cron job is triggered through the

### Edge Config

An [Edge Config](/docs/edge-config) is a global data store that enables experimentation with feature flags, A/B testing, critical redirec

## Location

Edge Functions are executed close to the end-users across Vercel's global network.

When you deploy Edge Functions, there are considerations you need to make about where it's deployed and executes. Edge Functions are exec

## Failover mode

Vercel's [failover mode](/docs/security#failover-strategy) refers to the system's behavior when a function fails to execute because of da

Vercel provides [redundancy](/docs/regions#outage-resiliency) and automatic failover for Edge Functions to ensure high availability.

## File system support

Edge Functions do not have filesystem access due to their ephemeral nature.

## Isolation boundary

In Vercel, the isolation boundary refers to the separation of individual instances of a function to ensure they don't interfere with each

As the Edge runtime is built on the [V8 engine](https://developers.google.com/apps-script/guides/v8-runtime), it uses V8 isolates to sepa

## Bundle size limits

Vercel places restrictions on the maximum size of the deployment bundle for functions to ensure that they execute in a timely manner. Edg

## Memory size limits

Edge Functions have a fixed memory limit. When you exceeds this limit, the execution will be aborted and we will return a `502` error.

The maximum size for a Function includes your JavaScript code, imported libraries and files (such as fonts), and all files bundled in the

If you reach the limit, make sure the code you are importing in your function is used and is not too heavy. You can use a package size ch

### Request body size

In Vercel, the request body size is the maximum amount of data that can be included in the body of a request to a function.

Edge Functions have the following limits applied to the request size:

| Name | Limit |
| --- | --- |
| Maximum URL length | 14 KB |
| Maximum request body length | 4 MB |
| Maximum number of request headers | 64 |
| Maximum request headers length | 16 KB |

## Edge Function API support

Edge Functions are neither Node.js nor browser applications, which means they don't have access to all browser and Node.js APIs. Currentl

There are some restrictions when writing Edge Functions:

- Use ES modules
- Most libraries which use Node.js APIs as dependencies can't be used in Edge Functions yet. See [available APIs](/docs/functions/runtime
- Dynamic code execution (such as `eval`) is not allowed for security reasons. You must ensure **libraries used in your Edge Functions do
  | API | Description |
  | --- | --- |
  | [`eval`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/eval) | Evaluates JavaScript code represented as

  | [`new Function(evalString)`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Function) | Creates a new fun
  | [`WebAssembly.instantiate`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/WebAssembly/instantiate) | Com

See the [Edge Runtime supported APIs](/docs/functions/runtimes/edge#edge-runtime-supported-apis) for more information.

### Limits on fetch API

- You cannot set non-standard port numbers in the fetch URL (e.g., `https://example.com:8080`). Only `80` and `443` are allowed. If you s
- The maximum number of requests from `fetch` API is **950** per Edge Function invocation.
- The maximum number of open connections is **6**.
  - Each function invocation can have up to 6 open connections. For example, if you try to send 10 simultaneous fetch requests, only 6 of
- If in-flight requests have been waiting for a response for more than 15 seconds with no active reads/writes, the runtime may cancel the
  - If you attempt to use a canceled connection, the `Network connection lost.` exception will be thrown.
  - You can `catch` on the `fetch` promise to handle this exception gracefully (e.g. with retries). Additionally, you can use the [`Abort

### Limited Date API

To avoid CPU timing attacks, like Spectre, date and time functionality is not generally available. In particular, the time returned from

```ts filename="app/api/date/route.ts" framework=all
export const runtime = 'edge';

export async function GET(request: Request) {
  const currentDate = () => new Date().toISOString();
  for (let i = 0; i < 500; i++) {
    console.log(`Current Date before fetch: ${currentDate()}`); // Prints the same value 1000 times.
  }

  await fetch('https://worldtimeapi.org/api/timezone/Etc/UTC');
  console.log(`Current Date after fetch: ${currentDate()}`); // Prints the new time

  return Response.json({ date: currentDate() });
}
```

```js filename="app/api/date/route.js" framework=all
export const runtime = 'edge';

export async function GET(request) {
  const currentDate = () => new Date().toISOString();
  for (let i = 0; i < 500; i++) {
    console.log(`Current Date before fetch: ${currentDate()}`); // Prints the same value 1000 times.
  }

  await fetch('https://worldtimeapi.org/api/timezone/Etc/UTC');
  console.log(`Current Date after fetch: ${currentDate()}`); // Prints the new time

  return Response.json({ date: currentDate() });
}
```

## Limits

The table below outlines the limits and restrictions of using Edge Functions on Vercel:

| Feature | Edge Runtime |
| ---------------------------------------------------------------------------------Autoscaled concurrency | ------------------------------
| [Maximum memory](/docs/functions/limitations#memory-size-limits) | 128 MB |
| [Maximum duration](/docs/functions/limitations#max-duration) | 25s (to begin returning a response, but can continue [streaming](/docs/f
| [Size](/docs/functions/runtimes#bundle-size-limits) (after gzip compression) | Hobby: 1 MB, Pro: 2 MB, Ent: 4 MB |
| [Concurrency](/docs/functions/concurrency-scaling#automatic-concurrency-scaling) | Autoscaled concurrency based on your plan |
| [Cost](/docs/functions/usage-and-pricing) | Pay for CPU time |
| [Regions](/docs/functions/runtimes#location) | Executes global-first, [can specify a region](/docs/functions/configuring-functions/regi
| [API Coverage](/docs/functions/limitations#api-support) | Limited API support |

### Routing Middleware CPU Limit

Routing Middleware can use no more than **50 ms** of CPU time on average.

This limitation refers to actual net CPU time, which is the time spent performing calculations, not the total elapsed execution or "wall

## Logs

See the Vercel Functions [Logs](/docs/functions/logs) documentation for more information on how to debug and monitor your Edge Functions.

## Pricing

The Hobby plan offers functions for free, within [limits](/docs/limits). The Pro plan extends these limits, and charges CPU Time for Edge

Edge runtime-powered functions usage is based on [CPU Time](/docs/pricing/edge-functions#managing-cpu-time). CPU time is the time spent a

Functions using the Edge Runtime are measured in the number of [**execution units**](/docs/limits/usage#execution-units), which are the a

A function can use up to 50 ms of CPU time per execution unit. If a function uses more than 50 ms, it will be divided into multiple 50 ms

See [viewing function usage](#viewing-function-usage) for more information on how to track your usage.

### Resource pricing

The following table outlines the price for each resource according to the plan you are on.

Edge Functions are available for free with the included usage limits. If you exceed the included usage and are on the Pro plan, you will

| Resource | Hobby Included | Pro Included | Pro Additional |
| ---------------------------- | -------------- | --------------- | ---------------------------------- |
| Edge Function Execution Units | First 500,000 | First 1,000,000 | $2.00 per 1,000,000 Execution Units |
| Function Invocations | First 100,000 | First 1,000,000 | $0.60 per 1,000,000 Invocations |

### Hobby

Vercel will send you emails as you are nearing your usage limits. On the Hobby plan you **will not pay for any additional usage**. Howeve

When your [Hobby team](/docs/plans/hobby) is set to **paused**, it remains in this state indefinitely unless you take action. This means

> **💡 Note:** If you have reached this state, your application is likely a good candidate
> for a [Pro account](/docs/plans/pro-plan).

To unpause your account, you have two main options:

- **Contact Support**: You can reach out to our [support team](/help) to discuss the reason for the pause and potential resolutions
- **Transfer to a Pro team**:
  If your Hobby team is paused, you won't have the option to initiate a [Pro trial](/docs/plans/pro-plan/trials). Instead, you can set up
  1. [Create a Pro team account](/docs/accounts/create-a-team)
  2. Add a valid credit card to this account. Select the **Settings** tab, then select **Billing** and **Payment Method**

Once set up, a transfer modal will appear, prompting you to [transfer your previous Hobby projects](/docs/projects/overview#transferring-

### Pro

For teams on a Pro trial, the [trial will end](/docs/plans/pro-plan/trials#post-trial-decision) when your team reaches the [trial limits]

Once your team exceeds the included usage, you will continue to be charged the on-demand costs going forward.

Pro teams can [set up Spend Management](/docs/spend-management#managing-your-spend-amount) to get notified or to automatically take actio

### Enterprise

Enterprise agreements provide custom usage and pricing for Edge Functions, including:

- Custom [execution units](/docs/functions/runtimes/edge/edge-functions#managing-execution-units)
- Multi-region deployments

See [Vercel Enterprise plans](/docs/plans/enterprise) for more information.

### Viewing function usage

Usage metrics can be found in the [Usage tab](/dashboard/usage) on your [dashboard](/dashboard). Functions are invoked for every request

You can see the usage for **functions using the Edge Runtime** on the **Edge Functions** section of the [Usage tab](/docs/limits/usage#ed

| Metric | Description | Priced
| --- | --- | ---
| Invocations | The number of times your Functions have been invoked | | [Learn
| Execution Units | The number of execution units that your Edge Functions have used. An execution unit is 50 ms of CPU time. | Yes
| CPU Time | The time your Edge Functions have spent computing responses to requests | No

## Managing Functions invocations

You are charged based on the number of times your [functions](/docs/functions) are invoked, including both successful and errored respons

When viewing your Invocations graph, you can group by **Count** to see the total of all invocations across your team's projects.

### Optimizing Function invocations

- Use the **Projects** option to see the total number of invocations for each project within your team. This can help you identify which
- Cache your responses using [the CDN](/docs/cdn-cache#using-vercel-functions) and [Cache-Control headers](/docs/headers#cache-control-he

## Managing execution units

You are charged based on number of **execution units** that your Edge Functions have used. Each invocation of an Edge Function has a **To

Each execution unit is 50ms. Vercel will work out the number of execution units (**total CPU time of the invocation / 50ms**) used for ea

For example:

- If your function gets invoked *250,000* times and uses *350* ms of CPU time at each invocation, then the function will incur **(350 ms
- Your usage is: 250,000 * 7 = **1,750,000** execution units
  Pro users have 1,000,000 execution units included in their plan, so you will be charged for the additional 750,000 execution units. The

### Optimizing execution units

- Execution units are comprised of a calculation of invocation count and CPU time. You can optimize your Edge Functions by [reducing the

## Managing CPU time

There is no time limit on the amount of CPU time your Edge Function can use during a single invocation. However, you are charged for each

You can view CPU time by **Average** to show the average time for computation across all projects using Edge Functions within your team. 

### Optimizing CPU time

- View the CPU time by **Project** to understand which Projects are using the most CPU time
- CPU time is calculated based on the actual time your function is running, not the time it takes to respond to a request. Therefore you

---
title: "Edge Runtime"
description: "Learn about the Edge runtime, an environment in which Vercel Functions can run."
last_updated: "2026-01-16T02:19:31.682Z"
source: "https://vercel.com/docs/functions/runtimes/edge"
---

# Edge Runtime

> **⚠️ Warning:** We recommend migrating from edge to Node.js for improved performance and
> reliability. Both runtimes run on [Fluid compute](/docs/fluid-compute) with
> [Active CPU pricing](/docs/functions/usage-and-pricing).

To convert your Vercel Function to use the Edge runtime, add the following code to your function:

```ts {1} filename="app/api/my-function/route.ts" framework=nextjs-app
export const runtime = 'edge'; // 'nodejs' is the default

export function GET(request: Request) {
  return new Response(`I am an Vercel Function!`, {
    status: 200,
  });
}
```

```js {1} filename="app/api/my-function/route.js" framework=nextjs-app
export const runtime = 'edge'; // 'nodejs' is the default

export function GET(request) {
  return new Response(`I am an Vercel Function!`, {
    status: 200,
  });
}
```

```ts {4-6} filename="pages/api/handler.ts" framework=nextjs
import { NextResponse } from 'next/server';
import type { NextRequest } from 'next/server';

export const config = {
  runtime: 'edge', // 'nodejs' is the default
};

export default function handler(request: NextRequest) {
  return NextResponse.json({
    name: `I am an Vercel Function!`,
  });
}
```

```js {1-3} filename="pages/api/handler.js" framework=nextjs
export const config = {
  runtime: 'edge', // 'nodejs' is the default
};

export default function handler(request) {
  return NextResponse.json({
    name: `I am an Vercel Function!`,
  });
}
```

```ts {3-5} filename="api/runtime-example.ts" framework=other
import type { VercelRequest, VercelResponse } from '@vercel/node';

export const config = {
  runtime: 'edge', // this is a pre-requisite
};

export default function handler(
  request: VercelRequest,
  response: VercelResponse,
) {
  return response.status(200).json({ text: 'I am an Vercel Function!' });
}
```

```js {1-3} filename="api/runtime-example.js" framework=other
export const config = {
  runtime: 'edge', // this is a pre-requisite
};

export default function handler(request, response) {
  return response.status(200).json({ text: 'I am an Vercel Function!' });
}
```

> **💡 Note:** If you're not using a framework, you must either add
>  to your
>  or change your JavaScript Functions'
> file extensions from  to
>

## Region

By default, Vercel Functions using the Edge runtime execute in the region closest to the incoming request. You can set one or more prefer

### Setting regions in your function

If your function depends on a data source, you may want it to be close to that source for fast responses.

To configure which region (or multiple regions) you want your function to execute in, pass the [ID of your preferred region(s)](/docs/reg

```ts {5-7} filename="pages/api/regional-example.ts" framework=nextjs
import { NextResponse } from 'next/server';
import type { NextRequest } from 'next/server';

export const config = {
  runtime: 'edge', // this must be set to `edge`
  // execute this function on iad1 or hnd1, based on the connecting client location
  regions: ['iad1', 'hnd1'],
};

export default function handler(request: NextRequest) {
  return NextResponse.json({
```

```
    name: `I am an Vercel Function! (executed on ${process.env.VERCEL_REGION})`,
  });
}
```

```js {2-4} filename="pages/api/regional-example.js" framework=nextjs
export const config = {
  runtime: 'edge', // this must be set to `edge`
  // execute this function on iad1 or hnd1, based on the connecting client location
  regions: ['iad1', 'hnd1'],
};

export default function handler(request) {
  return NextResponse.json({
    name: `I am an Vercel Function! (executed on ${process.env.VERCEL_REGION})`,
  });
}
```

> For \['nextjs-app']:

The `preferredRegion` option can be used to specify a single region using a string value, or multiple regions using a string array. See t

```ts {1-3} filename="app/api/regional-example/route.ts" framework=nextjs-app
export const runtime = 'edge'; // 'nodejs' is the default
// execute this function on iad1 or hnd1, based on the connecting client location
export const preferredRegion = ['iad1', 'hnd1'];
export const dynamic = 'force-dynamic'; // no caching

export function GET(request: Request) {
  return new Response(
    `I am an Vercel Function! (executed on ${process.env.VERCEL_REGION})`,
    {
      status: 200,
    },
  );
}
```

```js {1-3} filename="app/api/regional-example/route.js" framework=nextjs-app
export const runtime = 'edge'; // 'nodejs' is the default
// execute this function on iad1 or hnd1, based on the connecting client location
export const preferredRegion = ['iad1', 'hnd1'];
export const dynamic = 'force-dynamic'; // no caching

export function GET(request) {
  return new Response(
    `I am an Vercel Function! (executed on ${process.env.VERCEL_REGION})`,
    {
      status: 200,
    },
  );
}
```

```ts {2-4} filename="api/regional-example.ts" framework=other
export const config = {
  runtime: 'edge', // this must be set to `edge`
  // execute this function on iad1 or hnd1, based on the connecting client location
  regions: ['iad1', 'hnd1'],
};

export default function handler(request: Request, response: Response) {
  return response.status(200).json({
    text: `I am an Vercel Function! (executed on ${process.env.VERCEL_REGION})`,
  });
}
```

```js {2-4} filename="api/regional-example.js" framework=other
export const config = {
  runtime: 'edge', // this must be set to `edge`
  // execute this function on iad1 or hnd1, based on the connecting client location
  regions: ['iad1', 'hnd1'],
};

export default function handler(request, response) {
  return response.status(200).json({
    text: `I am an Vercel Function! (executed on ${process.env.VERCEL_REGION})`,
  });
}
```

> **💡 Note:** If you're not using a framework, you must either add
>  to your
>  or change your JavaScript Functions'
> file extensions from  to
>

## Failover mode

In the event of regional downtime, Vercel will automatically reroute traffic to the next closest CDN region on all plans. For more inform

## Maximum duration

Vercel Functions using the Edge runtime must begin sending a response within 25 seconds to maintain streaming capabilities beyond this pe

## Concurrency

Vercel automatically scales your functions to handle traffic surges, ensuring optimal performance during increased loads. For more inform

## Edge Runtime supported APIs

The Edge runtime is built on top of the [V8 engine](https://v8.dev/), allowing it to run in isolated execution environments that don't re

### Supported APIs

The Edge runtime provides a subset of Web APIs such as [`fetch`](https://developer.mozilla.org/docs/Web/API/Fetch_API), [`Request`](https

The following tables list the APIs that are available in the Edge runtime.

### Network APIs

| API | Description |
| --- | --- |
| [`fetch`](https://developer.mozilla.org/docs/Web/API/Fetch_API) | Fetches a resource |
| [`Request`](https://developer.mozilla.org/docs/Web/API/Request) | Represents an HTTP request |
| [`Response`](https://developer.mozilla.org/docs/Web/API/Response) | Represents an HTTP response |
| [`Headers`](https://developer.mozilla.org/docs/Web/API/Headers) | Represents HTTP headers |
| [`FormData`](https://developer.mozilla.org/docs/Web/API/FormData) | Represents form data |
| [`File`](https://developer.mozilla.org/docs/Web/API/File) | Represents a file |
| [`Blob`](https://developer.mozilla.org/docs/Web/API/Blob) | Represents a blob |
| [`URLSearchParams`](https://developer.mozilla.org/docs/Web/API/URLSearchParams) | Represents URL search parameters |
| [`Blob`](https://developer.mozilla.org/docs/Web/API/Blob) | Represents a blob |
| [`Event`](https://developer.mozilla.org/docs/Web/API/Event) | Represents an event |
| [`EventTarget`](https://developer.mozilla.org/docs/Web/API/EventTarget) | Represents an object that can handle events |
| [`PromiseRejectEvent`](https://developer.mozilla.org/docs/Web/API/PromiseRejectionEvent) | Represents an event that is sent to the glob |

### Encoding APIs

| API | Description |
| --- | --- |
| [`TextEncoder`](https://developer.mozilla.org/docs/Web/API/TextEncoder) | Encodes a string into a Uint8Array |
| [`TextDecoder`](https://developer.mozilla.org/docs/Web/API/TextDecoder) | Decodes a Uint8Array into a string |
| [`atob`](https://developer.mozilla.org/docs/Web/API/WindowOrWorkerGlobalScope/atob) | Decodes a base-64 encoded string |
| [`btoa`](https://developer.mozilla.org/docs/Web/API/WindowOrWorkerGlobalScope/btoa) | Encodes a string in base-64 |

### Stream APIs

| API | Description |
| --- | --- |
| [`ReadableStream`](https://developer.mozilla.org/docs/Web/API/ReadableStream) | Represents a readable stream |
| [`WritableStream`](https://developer.mozilla.org/docs/Web/API/WritableStream) | Represents a writable stream |
| [`WritableStreamDefaultWriter`](https://developer.mozilla.org/docs/Web/API/WritableStreamDefaultWriter) | Represents a writer of a Writ |
| [`TransformStream`](https://developer.mozilla.org/docs/Web/API/TransformStream) | Represents a transform stream |
| [`ReadableStreamDefaultReader`](https://developer.mozilla.org/docs/Web/API/ReadableStreamDefaultReader) | Represents a reader of a Read |
| [`ReadableStreamBYOBReader`](https://developer.mozilla.org/docs/Web/API/ReadableStreamBYOBReader) | Represents a reader of a Read |

### Crypto APIs

| API | Description |
| --- | --- |
| [`crypto`](https://developer.mozilla.org/docs/Web/API/Window/crypto) | Provides access to the cryptographic functionality of the p |
| [`SubtleCrypto`](https://developer.mozilla.org/docs/Web/API/SubtleCrypto) | Provides access to common cryptographic primitives, like ha |
| [`CryptoKey`](https://developer.mozilla.org/docs/Web/API/CryptoKey) | Represents a cryptographic key |

### Other Web Standard APIs

| API | Description |
| --- | --- |
| [`AbortController`](https://developer.mozilla.org/docs/Web/API/AbortController) | Allows you to a |
| [`AbortSignal`](https://developer.mozilla.org/docs/Web/API/AbortSignal) | Represents a si |
| [`DOMException`](https://developer.mozilla.org/docs/Web/API/DOMException) | Represents an e |
| [`structuredClone`](https://developer.mozilla.org/docs/Web/API/Web_Workers_API/Structured_clone_algorithm) | Creates a deep |
| [`URLPattern`](https://developer.mozilla.org/docs/Web/API/URLPattern) | Represents a UR |
| [`Array`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array) | Represents an a |
| [`ArrayBuffer`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/ArrayBuffer) | Represents a ge |
| [`Atomics`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Atomics) | Provides atomic |
| [`BigInt`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/BigInt) | Represents a wh |
| [`BigInt64Array`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/BigInt64Array) | Represents a ty |
| [`BigUint64Array`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/BigUint64Array) | Represents a ty |
| [`Boolean`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Boolean) | Represents a lo |
| [`clearInterval`](https://developer.mozilla.org/docs/Web/API/WindowOrWorkerGlobalScope/clearInterval) | Cancels a timed |
| [`clearTimeout`](https://developer.mozilla.org/docs/Web/API/WindowOrWorkerGlobalScope/clearTimeout) | Cancels a timed |
| [`console`](https://developer.mozilla.org/docs/Web/API/Console) | Provides access |
| [`DataView`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/DataView) | Represents a ge |
| [`Date`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Date) | Represents a si |
| [`decodeURI`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/decodeURI) | Decodes a Unifo |
| [`decodeURIComponent`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/decodeURIComponent) | Decodes a Unifo |
| [`encodeURI`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/encodeURI) | Encodes a Unifo |
| [`encodeURIComponent`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/encodeURIComponent) | Encodes a Unifo |
| [`Error`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Error) | Represents an e |
| [`EvalError`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/EvalError) | Represents an e |
| [`Float32Array`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Float32Array) | Represents a ty |
| [`Float64Array`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Float64Array) | Represents a ty |
| [`Function`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Function) | Represents a fu |
| [`Infinity`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Infinity) | Represents the |
| [`Int8Array`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Int8Array) | Represents a ty |
| [`Int16Array`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Int16Array) | Represents a ty |
| [`Int32Array`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Int32Array) | Represents a ty |
| [`Intl`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Intl) | Provides access |
| [`isFinite`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/isFinite) | Determines whet |
| [`isNaN`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/isNaN) | Determines whet |
| [`JSON`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/JSON) | Provides functi |
| [`Map`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Map) | Represents a co |
| [`Math`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Math) | Provides access |
| [`Number`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | Represents a nu |
| [`Object`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | Represents the |
| [`parseFloat`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/parseFloat) | Parses a string |
| [`parseInt`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/parseInt) | Parses a string |

| [`Promise`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Promise) | Represents the
| [`Proxy`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Proxy) | Represents an o
| [`RangeError`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/RangeError) | Represents an e
| [`ReferenceError`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/ReferenceError) | Represents an e
| [`Reflect`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Reflect) | Provides method
| [`RegExp`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/RegExp) | Represents a re
| [`Set`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Set) | Represents a co
| [`setInterval`](https://developer.mozilla.org/docs/Web/API/setInterval) | Repeatedly call
| [`setTimeout`](https://developer.mozilla.org/docs/Web/API/setTimeout) | Calls a functio
| [`SharedArrayBuffer`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/SharedArrayBuffer) | Represents a ge
| [`String`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Represents a se
| [`Symbol`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Symbol) | Represents a un
| [`SyntaxError`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/SyntaxError) | Represents an e
| [`TypeError`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/TypeError) | Represents an e
| [`Uint8Array`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Uint8Array) | Represents a ty
| [`Uint8ClampedArray`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Uint8ClampedArray) | Represents a ty
| [`Uint32Array`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Uint32Array) | Represents a ty
| [`URIError`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/URIError) | Represents an e
| [`URL`](https://developer.mozilla.org/docs/Web/API/URL) | Represents an o
| [`URLSearchParams`](https://developer.mozilla.org/docs/Web/API/URLSearchParams) | Represents a co
| [`WeakMap`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/WeakMap) | Represents a co
| [`WeakSet`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/WeakSet) | Represents a co
| [`WebAssembly`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/WebAssembly) | Provides access

## Check if you're running on the Edge runtime

You can check if your function is running on the Edge runtime by checking the global `globalThis.EdgeRuntime` property. This can be helpf

```ts
if (typeof EdgeRuntime !== 'string') {
  // dead-code elimination is enabled for the code inside this block
}
```

## Compatible Node.js modules

The following modules can be imported with and without the `node:` prefix when using the `import` statement:

| Module | Description
| ---------- | -----
| [`async_hooks`](https://nodejs.org/api/async_hooks.html) | Manage asynchronous resources lifecycles with `AsyncLocalStorage`. Supports
| [`events`](https://nodejs.org/api/events.html) | Facilitate event-driven programming with custom event emitters and listeners
| [`buffer`](https://nodejs.org/api/buffer.html) | Efficiently manipulate binary data using fixed-size, raw memory allocations
| [`assert`](https://nodejs.org/api/assert.html) | Provide a set of assertion functions for verifying invariants in your code
| [`util`](https://nodejs.org/api/util.html) | Offer various utility functions where we include `promisify`/`callbackify` a

Also, `Buffer` is globally exposed to maximize compatibility with existing Node.js modules.

## Unsupported APIs

The Edge runtime has some restrictions including:

- Some Node.js APIs other than the ones listed above **are not supported**. For example, you can't read or write to the filesystem
- `node_modules` *can* be used, as long as they implement ES Modules and do not use native Node.js APIs
- Calling `require` directly is **not allowed**. Use `import` instead

The following JavaScript language features are disabled, and **will not work:**

| API | Descr
| ---------- | -----
| [`eval`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/eval) | Evalu
| [`new Function(evalString)`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Function) | Creat
| [`WebAssembly.compile`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/WebAssembly/compile) | Compi
| [`WebAssembly.instantiate`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/WebAssembly/instantiate) | Compi

> **💡 Note:** While `WebAssembly.instantiate` is supported in Edge Runtime, it requires the
> Wasm source code to be provided using the import statement. This means you
> cannot use a buffer or byte array to dynamically compile the module at
> runtime.

## Environment Variables

You can use `process.env` to access [Environment Variables](/docs/environment-variables).

## Many Node.js APIs are not available

Middleware with the `edge` runtime configured is neither a Node.js nor browser application, which means it doesn't have access to all bro

In summary:

- Use ES modules
- Most libraries that use Node.js APIs as dependencies can't be used in Middleware with the `edge` runtime configured.
- Dynamic code execution (such as `eval`) is not allowed (see the next section for more details)

## Dynamic code execution leads to a runtime error

Dynamic code execution is not available in Middleware with the `edge` runtime configured for security reasons. For example, the following

| API | Descr
| ---------- | -----
| [`eval`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/eval) | Evalu
| [`new Function(evalString)`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Function) | Creat
| [`WebAssembly.instantiate`](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/WebAssembly/instantiate) | Compi

You need to make sure libraries used in your Middleware with the `edge` runtime configured don't rely on dynamic code execution because i

## Maximum Execution Duration

Middleware with the `edge` runtime configured must begin sending a response within **25 seconds**.

You may continue streaming a response beyond that time and you can continue with asynchronous workloads in the background, after returning

## Code size limit

| Plan | Limit (after gzip compression) |
| --- | --- |
| Hobby | 1 MB |
| Pro | 2 MB |
| Enterprise | 4 MB |

The maximum size for an Vercel Function using the Edge runtime includes your JavaScript code, imported libraries and files (such as fonts

If you reach the limit, make sure the code you are importing in your function is used and is not too heavy. You can use a package size ch

## Ignored Environment Variable Names

Environment Variables can be accessed through the `process.env` object.
Since JavaScript objects have methods to allow some operations on them, there are limitations
on the names of Environment Variables to avoid having ambiguous code.

The following names will be ignored as Environment Variables to avoid overriding the `process.env` object prototype:

- `constructor`
- `__defineGetter__`
- `__defineSetter__`
- `hasOwnProperty`
- `__lookupGetter__`
- `__lookupSetter__`
- `isPrototypeOf`
- `propertyIsEnumerable`
- `toString`
- `valueOf`
- `__proto__`
- `toLocaleString`

Therefore, your code will always be able to use them with their expected behavior:

```js
// returns `true`, if `process.env.MY_VALUE` is used anywhere & defined in the Vercel dashboard
process.env.hasOwnProperty('MY_VALUE');
```

---
title: "Using the Go Runtime with Vercel functions"
description: "Learn how to use the Go runtime to compile Go Vercel functions on Vercel."
last_updated: "2026-01-16T02:19:31.575Z"
source: "https://vercel.com/docs/functions/runtimes/go"
---

# Using the Go Runtime with Vercel functions

The Go runtime is used by Vercel to compile Go Vercel functions that expose a single HTTP handler, from a `.go` file within an `/api` dir

For example, define an `index.go` file inside an `/api` directory as follows:

```go filename="/api/index.go"
package handler

import (
  "fmt"
  "net/http"
)

func Handler(w http.ResponseWriter, r *http.Request) {
  fmt.Fprintf(w, "<h1>Hello from Go!</h1>")
}
```

For advanced usage, such as using private packages with your Go projects, see the [Advanced Go Usage section](#advanced-go-usage).

> **💡 Note:** The exported function needs to include the [
>
> ](https://golang.org/pkg/net/http/#HandlerFunc) signature type, but can use
> any valid Go exported function declaration as the function name.

## Go Version

The Go runtime will automatically detect the `go.mod` file at the root of your Project to determine the version of Go to use.

If `go.mod` is missing or the version is not defined, the default version 1.20 will be used.

The first time the Go version is detected, it will be automatically downloaded and cached. Subsequent deployments using the same Go versi

## Go Dependencies

The Go runtime will automatically detect the `go.mod` file at the root of your Project to install dependencies.

## Go Build Configuration

You can provide custom build flags by using the `GO_BUILD_FLAGS` [Environment Variable](/docs/environment-variables).

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "build": {
    "env": {
      "GO_BUILD_FLAGS": "-ldflags '-s -w'"
    }
  }
```

}
```

## Advanced Go Usage

In order to use this runtime, no configuration is needed. You only need to create a file inside the `api` directory.

**The entry point of this runtime is a global matching `.go` files** that export a function that implements the `http.HandlerFunc` signat

### Private Packages for Go

To install private packages with `go get`, add an [Environment Variable](/docs/environment-variables) named `GIT_CREDENTIALS`.

The value should be the URL to the Git repo including credentials, such as `https://username:token@github.com`.

All major Git providers are supported including GitHub, GitLab, Bitbucket, as well as a self-hosted Git server.

With GitHub, you will need to [create a personal token](https://github.com/settings/tokens) with permission to access your private reposi

---
title: "Advanced Node.js Usage"
description: "Learn about advanced configurations for Vercel functions on Vercel."
last_updated: "2026-01-16T02:19:31.691Z"
source: "https://vercel.com/docs/functions/runtimes/node-js/advanced-node-configuration"
---

# Advanced Node.js Usage

To use Node.js, create a file inside your project's `api` directory. No additional configuration is needed.

**The entry point for `src` must be a glob matching `.js`, `.mjs`, or `.ts` files** that export a default function.

### Disabling helpers for Node.js

To disable [helpers](/docs/functions/runtimes/node-js#node.js-helpers):

1. From the dashboard, select your project and go to the **Settings** tab.
2. Select Environment Variables from the left side in settings.
3. Add a new environment variable with the **Key**: `NODEJS_HELPERS` and the **Value**: `0`. You should ensure this is set for all enviro
4. Pull your env vars into your local project with the [following command](/docs/cli/env):
   ```bash filename="terminal"
   vercel env pull
   ```

For more information, see [Environment Variables](/docs/environment-variables).

### Private npm modules for Node.js

To install private npm modules:

1. From the dashboard, select your project and go to the **Settings** tab.
2. Select Environment Variables from the left side in settings.
3. Add a new environment variable with the **Key**: `NPM_TOKEN` and enter your [npm token](https://docs.npmjs.com/about-access-tokens) as
4. Pull your env vars into your local project with the [following command](/docs/cli/env):
   ```bash filename="terminal"
   vercel env pull
   ```

For more information, see [Environment Variables](/docs/environment-variables).

### Custom build step for Node.js

In some cases, you may wish to include build outputs inside your Vercel Function. To do this:

1. Add a `vercel-build` script within your `package.json` file, in the same directory as your Vercel Function or any parent directory. Th

```json filename="package.json"
{
  "scripts": {
    "vercel-build": "node ./build.js"
  }
}
```

2. Create the build script named `build.js`:

```javascript filename="build.js"
const fs = require('fs');
fs.writeFile('built-time.js', `module.exports = '${new Date()}'`, (err) => {
  if (err) throw err;
  console.log('Build time file created successfully!');
});
```

3. Finally, create a `.js` file for the built Vercel functions, `index.js` inside the `/api` directory:

```javascript filename="api/index.js"
const BuiltTime = require('./built-time');
module.exports = (request, response) => {
  response.setHeader('content-type', 'text/plain');
  response.send(`
    This Vercel Function was built at ${new Date(BuiltTime)}.
    The current time is ${new Date()}
  `);
};
```

### Experimental Node.js require() of ES Module
```

By default, we disable experimental support for [requiring ES Modules](https://nodejs.org/docs/latest-v24.x/api/modules.html#loading-ecma

- `NODE_OPTIONS=--experimental-require-module`

---
title: "Supported Node.js versions"
description: "Learn about the supported Node.js versions on Vercel."
last_updated: "2026-01-16T02:19:31.698Z"
source: "https://vercel.com/docs/functions/runtimes/node-js/node-js-versions"
---

# Supported Node.js versions

## Default and available versions

By default, a new project uses the latest Node.js LTS version available on Vercel.

Current available versions are:

- **24.x** (default)
- **22.x**
- **20.x**

Only major versions are available. Vercel automatically rolls out minor and patch updates when needed, such as to fix a security issue.

## Setting the Node.js version in project settings

To override the [default](#default-and-available-versions) version and set a different Node.js version for new deployments:

1. From your dashboard, select your project.
2. Select the **Settings** tab.
3. On the **Build and Deployment** page, navigate to the **Node.js Version** section.
4. Select the version you want to use from the dropdown. This Node.js version will be used for new deployments.

## Version overrides in `package.json`

You can define the major Node.js version in the `engines#node` section of the `package.json` to override the one you have selected in the

```json filename="package.json"
{
  "engines": {
    "node": "24.x"
  }
}
```

For instance, when you set the Node.js version to **20.x** in the **Project Settings** and you specify a valid [semver range](https://sem

The following table lists some example version ranges and the available Node.js version they map to:

| Version in `package.json` | Version deployed |
| --- | --- |
| `24.x` `^24.0.0` `>=20.0.0` | latest **24.x** version |
| `22.x` `^22.0.0` | latest **22.x** version |
| `20.x` `^20.0.0` | latest **20.x** version |

## Checking your deployment's Node.js version

To verify the Node.js version your Deployment is using, either run `node -v` in the Build Command or log `process.version`.

---
title: "Using the Node.js Runtime with Vercel Functions"
description: "Learn how to use the Node.js runtime with Vercel Functions to create functions."
last_updated: "2026-01-16T02:19:31.714Z"
source: "https://vercel.com/docs/functions/runtimes/node-js"
---

# Using the Node.js Runtime with Vercel Functions

You can create Vercel Function in JavaScript or TypeScript by using the Node.js runtime. By default, the runtime builds and serves any fu

[Node.js](/docs/functions/runtimes/node-js)-powered functions are suited to computationally intense or large functions and provide benefi

- **More RAM and CPU power**: For computationally intense workloads, or functions that have bundles up to 250 MB in size, this runtime is
- **Complete Node.js compatibility**: The Node.js runtime offers access to all Node.js APIs, making it a powerful tool for many applicati

## Creating a Node.js function

In order to use the Node.js runtime, create a file inside the `api` directory with a function using the [`fetch` Web Standard export](/do

```ts filename="api/hello.ts"
export default {
  fetch(request: Request) {
    return new Response('Hello from Vercel!');
  },
};
```

Alternatively, you can export each HTTP method as a separate export instead of using the `fetch` Web Standard export:

```ts filename="api/hello.ts"
export function GET(request: Request) {
  return new Response('Hello from Vercel!');
}
```

To learn more about creating Vercel Functions, see the [Functions API Reference](/docs/functions/functions-api-reference). If you need mo

> **💡 Note:** The entry point for `src` must be a glob matching `.js`, `.mjs`, or `.ts`
> files** that export a default function.

## Supported APIs

Vercel Functions using the Node.js runtime support [all Node.js APIs](https://nodejs.org/docs/latest/api/), including standard Web APIs s

## Node.js version

To learn more about the supported Node.js versions on Vercel, see [Supported Node.js Versions](/docs/functions/runtimes/node-js/node-js-v

## Node.js dependencies

For dependencies listed in a `package.json` file at the root of a project, the following behavior is used:

- If `bun.lock` or `bun.lockb` is present, `bun install` is executed
- If `yarn.lock` is present `yarn install` is executed
- If `pnpm-lock.yaml` is present, `pnpm install` is executed
  - See [supported package managers](/docs/package-managers#supported-package-managers) for pnpm detection details
- If `package-lock.json` is present, `npm install` is executed
- If `vlt-lock.json` is present, `vlt install` is executed
- Otherwise, `npm install` is executed

If you need to select a specific version of a package manager, see [corepack](/docs/deployments/configure-a-build#corepack).

## Using TypeScript with the Node.js runtime

The Node.js runtime supports files ending with `.ts` inside of the `/api` directory as TypeScript files to compile and serve when deployi

An example TypeScript file that exports a Web signature handler is as follows:

```typescript filename="api/hello.ts"
export default {
  async fetch(request: Request) {
    const url = new URL(request.url);
    const name = url.searchParams.get('name') || 'World';

    return Response.json({ message: `Hello ${name}!` });
  },
};
```

You can use a `tsconfig.json` file at the root of your project to configure the TypeScript compiler. Most options are supported aside fro

## Node.js request and response objects

Each request to a Node.js Vercel Function gives access to Request and Response objects. These objects are the [standard](https://nodejs.o

### Node.js helpers

Vercel additionally provides helper methods inside of the Request and Response objects passed to Node.js Vercel Functions. These methods

| method | description |
| --- | --- |
| `request.query` | An object containing the request's [query string](https://en.wikipedia.org/wi |
| `request.cookies` | An object containing the cookies sent by the request, or `{}` if the request |
| [`request.body`](#node.js-request-and-response-objects) | An object containing the body sent by the request, or `null` if no body is se |
| `response.status(code)` | A function to set the status code sent with the response where `code` must be |
| `response.send(body)` | A function to set the content of the response where `body` can be a `string`, |
| `response.json(obj)` | A function to send a JSON response where `obj` is the JSON object to send. |
| `response.redirect(url)` | A function to redirect to the URL derived from the specified path with status |
| `response.redirect(statusCode, url)` | A function to redirect to the URL derived from the specified path, with speci |

The following Node.js Vercel Function example showcases the use of `request.query`, `request.cookies` and `request.body` helpers:

```javascript filename="api/hello.ts"
import { VercelRequest, VercelResponse } from "@vercel/node";

module.exports = (request: VercelRequest, response: VercelResponse) => {
  let who = 'anonymous';

  if (request.body && request.body.who) {
    who = request.body.who;
  } else if (request.query.who) {
    who = request.query.who;
  } else if (request.cookies.who) {
    who = request.cookies.who;
  }

  response.status(200).send(`Hello ${who}!`);
};
```

> **💡 Note:** If needed, you can opt-out of Vercel providing `helpers` using [advanced
> configuration](#disabling-helpers-for-node.js).

### Request body

We populate the `request.body` property with a parsed version of the content sent with the request when possible.

We follow a set of rules on the `Content-type` header sent by the request to do so:

| `Content-Type` header | Value of `request.body` |
| --- | --- |
| No header | `undefined` |
| `application/json` | An object representing the parsed JSON sent by the request. |
| `application/x-www-form-urlencoded` | An object representing the parsed data sent by with the request. |
| `text/plain` | A string containing the text sent by the request. |
| `application/octet-stream` | A [Buffer](https://nodejs.org/api/buffer.html) containing the data sent by the request. |

With the `request.body` helper, you can build applications without extra dependencies or having to parse the content of the request manua

> **💡 Note:** The `request.body` helper is set using a [JavaScript
> getter](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Functions/get).
> In turn, it is only computed when it is accessed.

When the request body contains malformed JSON, accessing `request.body` will throw an error. You can catch that error by wrapping `reques

```javascript filename="api/hello.ts"
try {
  request.body;
} catch (error) {
  return response.status(400).json({ error: 'My custom 400 error' });
}
```

### Cancelled Requests

Request cancellation must be enabled on a per-route basis. See [Functions API Reference](/docs/functions/functions-api-reference#cancel-r

You can listen for the `error` event on the request object to detect request cancellation:

```typescript filename="api/cancel.ts" {5-8}
import { VercelRequest, VercelResponse } from '@vercel/node';

export default async (request: VercelRequest, response: VercelResponse) => {
  let cancelled = false;
  request.on('error', (error) => {
    if (error.message === 'aborted') {
      console.log('request aborted');
    }
    cancelled = true;
  });

  response.writeHead(200);

  for (let i = 1; i < 5; i++) {
    if (cancelled) {
      // the response must be explicitly ended
      response.end();
      return;
    }

    response.write(`Count: ${i}\n`);

    await new Promise((resolve) => setTimeout(resolve, 1000));
  }

  response.end('All done!');
};
```

## Using Express with Vercel

Express.js is a popular framework used with Node.js. For information on how to use Express with Vercel, see the guide: [Using Express.js w

## Using Node.js with middleware

The Node.js runtime can be used as an experimental feature to run middleware. To enable, add the flag to your `next.config.ts` file:

```ts filename="next.config.ts" framework=all
import type { NextConfig } from 'next';

const nextConfig: NextConfig = {
  experimental: {
    nodeMiddleware: true,
  },
};

export default nextConfig;
```

```js filename="next.config.ts" framework=all
const nextConfig = {
  experimental: {
    nodeMiddleware: true,
  },
};

export default nextConfig;
```

Then in your middleware file, set the runtime to `nodejs` in the `config` object:

```js {3} filename="middleware.ts" framework=all
export const config = {
  matcher: '/about/:path*',
  runtime: 'nodejs',
};
```

```ts {3} filename="middleware.ts" framework=all
export const config = {
  matcher: '/about/:path*',
  runtime: 'nodejs',
};
```

> **💡 Note:** Running middleware on the Node.js runtime incurs charges under [Vercel
> Functions pricing](/docs/functions/usage-and-pricing#pricing). These functions

> only run using [Fluid compute](/docs/fluid-compute#fluid-compute).

---
title: "Runtimes"
description: "Runtimes transform your source code into Functions, which are served by our CDN. Learn about the official runtimes supporte
last_updated: "2026-01-16T02:19:31.728Z"
source: "https://vercel.com/docs/functions/runtimes"
---

# Runtimes

Vercel supports multiple runtimes for your functions. Each runtime has its own set of libraries, APIs, and functionality that provides di

Runtimes transform your source code into [Functions](/docs/functions), which are served by our [CDN](/docs/cdn).

## Official runtimes

Vercel Functions support the following official runtimes:

| Runtime | Description |
| --- | --- |
| [Node.js](/docs/functions/runtimes/node-js) | The Node.js runtime takes an entrypoint of a Node.js function, builds its dependencies (i |
| [Bun](/docs/functions/runtimes/bun) | The Bun runtime takes an entrypoint of a Bun function, builds its dependencies (if any) a |
| [Python](/docs/functions/runtimes/python) | The Python runtime takes in a Python program that defines a singular HTTP handler and out |
| [Rust](/docs/functions/runtimes/rust) | The Rust runtime takes an entrypoint of a Rust function using the `vercel_runtime` crate |
| | The Go runtime takes in a Go program that defines a singular HTTP handler and outputs it as a Vercel Funct |
| [Ruby](/docs/functions/runtimes/ruby) | The Ruby runtime takes in a Ruby program that defines a singular HTTP handler and outputs |
| [Wasm](/docs/functions/runtimes/wasm) | The Wasm takes in a pre-compiled WebAssembly program and outputs it as a Vercel Function. |
| [Edge](/docs/functions/runtimes/edge) | The Edge runtime is built on top of the V8 engine, allowing it to run in isolated executi |

## Community runtimes

If you would like to use a language that Vercel does not support by default, you can use a community runtime by setting the [`functions`

The following community runtimes are recommended by Vercel:

| Runtime | Runtime Module | Docs |
| --- | --- | --- |
| Bash | `vercel-bash` | https://github.com/importpw/vercel-bash |
| Deno | `vercel-deno` | https://github.com/vercel-community/deno |
| PHP | `vercel-php` | https://github.com/vercel-community/php |

You can create a community runtime by using the [Runtime API](https://github.com/vercel/vercel/blob/main/DEVELOPING_A_RUNTIME.md). Altern

## Features

- **Location**: Deployed as region-first, [can customize location](/docs/functions/configuring-functions/region#setting-your-default-regi
- [**Failover**](/docs/functions/runtimes#failover-mode): Automatic failover to [defined regions](/docs/functions/configuring-functions/r
- [**Automatic concurrency scaling**](/docs/functions/concurrency-scaling#automatic-concurrency-scaling): Auto-scales up to 30,000 (Hobby
- [**Isolation boundary**](/docs/functions/runtimes#isolation-boundary): microVM
- [**File system support**](/docs/functions/runtimes#file-system-support): Read-only filesystem with writable `/tmp` scratch space up to
- [**Archiving**](/docs/functions/runtimes#archiving): Functions are archived when not invoked
- [**Functions created per deployment**](/docs/functions/runtimes#functions-created-per-deployment): Hobby: Framework-dependent, Pro and

### Location

Location refers to where your functions are **executed**. Vercel Functions are region-first, and can be [deployed](/docs/functions/config

### Failover mode

Vercel's failover mode refers to the system's behavior when a function fails to execute because of data center downtime.

Vercel provides [redundancy](/docs/regions#outage-resiliency) and automatic failover for Vercel Functions using the Edge runtime. For Ver

### Isolation boundary

In Vercel, the isolation boundary refers to the separation of individual instances of a function to ensure they don't interfere with each

With traditional serverless infrastructure, each function uses a microVM for isolation, which provides strong security but also makes ther

### File system support

Filesystem support refers to a function's ability to read and write to the filesystem. Vercel functions have a read-only filesystem with

### Archiving

Vercel Functions are archived when they are not invoked:

- **Within 2 weeks** for [Production Deployments](/docs/deployments)
- **Within 48 hours** for [Preview Deployments](/docs/deployments/environments#preview-environment-pre-production)

Archived functions will be unarchived when they're invoked, which can make the initial [cold start](/docs/infrastructure/compute#cold-and

### Functions created per deployment

When using [Next.js](/docs/frameworks/nextjs) or [SvelteKit](/docs/frameworks/sveltekit) on Vercel, dynamic code (APIs, server-rendered p

When using other [frameworks](/docs/frameworks), or Vercel Functions [directly without a framework](/docs/functions), every API maps dire

## Caching data

A runtime can retain an archive of up to **100 MB** of the filesystem at build time. The cache key is generated as a combination of:

- Project name
- [Team ID](/docs/accounts#find-your-team-id) or User ID
- Entrypoint path (e.g., `api/users/index.go`)
- Runtime identifier including version (e.g.: `@vercel/go@0.0.1`)

The cache will be invalidated if any of those items changes. You can bypass the cache by running `vercel -f`.

## Environment variables

You can use [environment variables](/docs/environment-variables#environment-variable-size) to manage dynamic values and sensitive informa

You can use a total of **64 KB** in environments variables per-deployment on Vercel. This limit is for all variables combined, and so no

## Vercel features support

The following features are supported by Vercel Functions:

### Secure Compute

Vercel's [Secure Compute](/docs/secure-compute) feature offers enhanced security for your Vercel Functions, including dedicated IP addres

### Streaming

Streaming refers to the ability to send or receive data in a continuous flow.

The Node.js runtime supports streaming by default. Streaming is also supported when using the [Python runtime](/docs/functions/streaming-

Vercel Functions have a [maximum duration](/docs/functions/configuring-functions/duration), meaning that it isn't possible to stream inde

Node.js and Edge runtime streaming functions support the [`waitUntil` method](/docs/functions/functions-api-reference/vercel-functions-pa

### Cron jobs

[Cron jobs](/docs/cron-jobs) are time-based scheduling tools used to automate repetitive tasks. When a cron job is triggered through the

### Vercel Storage

From your function, you can communicate with a choice of [data stores](/docs/storage). To ensure low-latency responses, it's crucial to h

### Edge Config

An [Edge Config](/docs/edge-config) is a global data store that enables experimentation with feature flags, A/B testing, critical redirec

### Tracing

Vercel supports [Tracing](/docs/tracing) that allows you to send OpenTelemetry traces from your Vercel Functions to any application perfo

---
title: "Using the Python Runtime with Vercel Functions"
description: "Learn how to use the Python runtime to compile Python Vercel Functions on Vercel."
last_updated: "2026-01-16T02:19:31.740Z"
source: "https://vercel.com/docs/functions/runtimes/python"
---

# Using the Python Runtime with Vercel Functions

The Python runtime enables you to write Python code, including using [FastAPI](https://vercel.com/new/git/external?repository-url=https:/

You can create your first function, available at the `/api` route, as follows:

```py filename="api/index.py"
from http.server import BaseHTTPRequestHandler

class handler(BaseHTTPRequestHandler):

    def do_GET(self):
        self.send_response(200)
        self.send_header('Content-type','text/plain')
        self.end_headers()
        self.wfile.write('Hello, world!'.encode('utf-8'))
        return
```

## Python version

The current available version is **Python 3.12**. This cannot be changed.

## Dependencies

You can install dependencies for your Python projects by defining them in a `pyproject.toml` with or without a corresponding `uv.lock`, `

```python filename="requirements.txt"
fastapi==0.117.1
```

```toml filename="pyproject.toml"
[project]
name = "my-python-api"
version = "0.1.0"
description = "Add your description here"
readme = "README.md"
requires-python = ">=3.11"
dependencies = [
    "fastapi>=0.117.1",
]
```

## Streaming Python functions

Vercel Functions support streaming responses when using the Python runtime. This allows you to render parts of the UI as they become read

## Controlling what gets bundled

By default, Python Vercel Functions include all files from your project that are reachable at build time. Unlike the Node.js runtime, the

You should make sure your `pyproject.toml` or `requirements.txt` only lists packages necessary for runtime and you should also explicitly

> **💡 Note:** Python functions have a maximum uncompressed bundle size of . See the
> .

To exclude unnecessary files (for example: tests, static assets, and test data), configure `excludeFiles` in `vercel.json` under the `fun

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "api/**/*.py": {
      "excludeFiles": "{tests/**,__tests__/**,**/*.test.py,**/test_*.py,fixtures/**,__fixtures__/**,testdata/**,sample-data/**,static/**,
    }
  }
}
```

## Using FastAPI with Vercel

FastAPI is a modern, high-performance, web framework for building APIs with Python. For information on how to use FastAPI with Vercel, re

## Using Flask with Vercel

Flask is a lightweight WSGI web application framework. For information on how to use Flask with Vercel, review this [guide](/docs/framewo

## Other Python Frameworks

For FastAPI, Flask, or basic usage of the Python runtime, no configuration is required. Usage of the Python runtime with other frameworks

**The entry point of this runtime is a glob matching `.py` source files** with one of the following variables defined:

- `handler` that inherits from the `BaseHTTPRequestHandler` class
- `app` that exposes a WSGI or ASGI Application

### Reading Relative Files in Python

Python uses the current working directory when a relative file is passed to [open()](https://docs.python.org/3/library/functions.html#ope

The current working directory is the base of your project, not the `api/` directory.

For example, the following directory structure:

```py filename="directory"
├── README.md
├── api
│   ├── user.py
├── data
│   └── file.txt
└── requirements.txt
```

With the above directory structure, your function in `api/user.py` can read the contents of `data/file.txt` in a couple different ways.

You can use the path relative to the project's base directory.

```py filename="api/user.py"

from http.server import BaseHTTPRequestHandler
from os.path import join

class handler(BaseHTTPRequestHandler):

    def do_GET(self):
        self.send_response(200)
        self.send_header('Content-type','text/plain')
        self.end_headers()
        with open(join('data', 'file.txt'), 'r') as file:
          for line in file:
            self.wfile.write(line.encode())
        return
```

Or you can use the path relative to the current file's directory.

```py filename="api/user.py"

from http.server import BaseHTTPRequestHandler
from os.path import dirname, abspath, join
dir = dirname(abspath(__file__))

class handler(BaseHTTPRequestHandler):

    def do_GET(self):
        self.send_response(200)
        self.send_header('Content-type','text/plain')
        self.end_headers()
        with open(join(dir, '..', 'data', 'file.txt'), 'r') as file:
          for line in file:
            self.wfile.write(line.encode())
        return
```

### Web Server Gateway Interface

The Web Server Gateway Interface (WSGI) is a calling convention for web servers to forward requests to web applications written in Python

- [Deploy an example with Flask](https://vercel.com/new/git/external?repository-url=https://github.com/vercel/examples/tree/main/python/f

- [Deploy an example with Django](https://vercel.com/new/git/external?repository-url=https://github.com/vercel/examples/tree/main/python/

### Asynchronous Server Gateway Interface

The Asynchronous Server Gateway Interface (ASGI) is a calling convention for web servers to forward requests to asynchronous web applicat

Instead of defining a `handler`, define an `app` variable in your Python file.

For example, define a `api/index.py` file as follows:

```python filename="api/index.py"
from sanic import Sanic
from sanic.response import json
app = Sanic()


@app.route('/')
@app.route('/<path:path>')
async def index(request, path=""):
    return json({'hello': path})
```

Inside `requirements.txt` define:

```py filename="requirements.txt"
sanic==19.6.0
```

---
title: "Using the Ruby Runtime with Vercel Functions"
description: "Learn how to use the Ruby runtime to compile Ruby Vercel Functions on Vercel."
last_updated: "2026-01-16T02:19:31.746Z"
source: "https://vercel.com/docs/functions/runtimes/ruby"
---

# Using the Ruby Runtime with Vercel Functions

The Ruby runtime is used by Vercel to compile Ruby Vercel functions that define a singular HTTP handler from `.rb` files within an `/api`

Ruby files must have one of the following variables defined:

- `Handler` proc that matches the `do |request, response|` signature.
- `Handler` class that inherits from the `WEBrick::HTTPServlet::AbstractServlet` class.

For example, define a `index.rb` file inside a `/api` directory as follows:

```ruby filename="api/index.rb"
require 'cowsay'

Handler = Proc.new do |request, response|
  name = request.query['name'] || 'World'

  response.status = 200
  response['Content-Type'] = 'text/text; charset=utf-8'
  response.body = Cowsay.say("Hello #{name}", 'cow')
end
```

Inside a `Gemfile` define:

```ruby filename="Gemfile"
source "https://rubygems.org"

gem "cowsay", "~> 0.3.0"
```

## Ruby Version

New deployments use Ruby 3.3.x as the default version.

You can specify the version of Ruby by defining `ruby` in a `Gemfile`, like so:

```ruby filename="Gemfile"
source "https://rubygems.org"
ruby "~> 3.3.x"
```

> **💡 Note:** If the patch part of the version is defined, like
>  it will be ignored and assume the latest
> .

## Ruby Dependencies

This runtime supports installing dependencies defined in the `Gemfile`. Alternatively, dependencies can be vendored with the `bundler ins

---
title: "Using the Rust Runtime with Vercel functions"
description: "Build fast, memory-safe serverless functions with Rust on Vercel."
last_updated: "2026-01-16T02:19:31.752Z"
source: "https://vercel.com/docs/functions/runtimes/rust"
---

# Using the Rust Runtime with Vercel functions

Use Rust to build high-performance, memory-safe serverless functions. The Rust runtime runs on [Fluid compute](/docs/fluid-compute) for o

## Getting Started

1. [**Configure your project**](#cargo.toml-configuration) - Add a `Cargo.toml` file with required dependencies
2. [**Create your function**](#creating-api-handlers) - Write handlers in the `api/` directory
3. [**Deploy**](#deployment) - Push to GitHub or use the Vercel CLI

## Project setup

### Cargo.toml configuration

Create a `Cargo.toml` file in your project root:

```toml filename="Cargo.toml"
[package]
name = "rust-hello-world"
version = "0.1.0"
edition = "2024"

[dependencies]
tokio = { version = "1", features = ["full"] } # async runtime
vercel_runtime = { version = "2" } # handles communicating with Vercel's function bridge
serde = { version = "1.0", features = ["derive"] }
serde_json = "1.0"

# Each handler has to be specified as [[bin]]
# Note that you need to provide unique names for each binary
[[bin]]
name = "hello"
path = "api/hello.rs"

# This section configures settings for the release profile, which optimizes the build for performance.
[profile.release]
codegen-units = 1
lto = "fat"
opt-level = 3
```

### Creating API handlers

Create Rust files in your `api/` directory. Each file becomes a serverless function:

```rust filename="api/hello.rs"
use serde_json::{Value, json};
use vercel_runtime::{Error, Request, run, service_fn};

#[tokio::main]
async fn main() -> Result<(), Error> {
    let service = service_fn(handler);
    run(service).await
}

async fn handler(_req: Request) -> Result<Value, Error> {
    Ok(json!({
        "message": "Hello, world!",
    }))
}
```

For more code examples, please refer to our templates:

- [Rust Hello World](https://vercel.com/templates/template/rust-hello-world)
- [Rust Axum](https://vercel.com/templates/template/rust-axum)

[vercel/examples](https://github.com/vercel/examples/tree/main/rust).

## Deployment

### Git deployment

Push your code to a connected GitHub repository for automatic deployments.

### CLI deployment

Deploy directly using the Vercel CLI:

```bash
vercel deploy
```

### Build optimization

For prebuilt deployments, optimize your `.vercelignore`:

```bash filename=".vercelignore"
# Ignore everything in the target directory except for release binaries
target/**
!target/release
!target/x86_64-unknown-linux-gnu/release/**
!target/aarch64-unknown-linux-gnu/release/**
```

## Feature support

---
title: "Using WebAssembly (Wasm)"
description: "Learn how to use WebAssembly (Wasm) to enable low-level languages to run on Vercel Functions and Routing Middleware."
last_updated: "2026-01-16T02:19:31.758Z"
source: "https://vercel.com/docs/functions/runtimes/wasm"
---

# Using WebAssembly (Wasm)

[WebAssembly](https://webassembly.org), or Wasm, is a portable, low-level, assembly-like language that can be used as a compilation targe

With Vercel, you can use Wasm in [Vercel Functions](/docs/functions) or [Routing Middleware](/docs/routing-middleware) when the runtime i

Pre-compiled WebAssembly can be imported with the `?module` suffix. This will provide an array of the Wasm data that can be instantiated

> **💡 Note:** While `WebAssembly.instantiate` is supported in Edge Runtime, it requires the
> Wasm source code to be provided using the import statement. This means you
> cannot use a buffer or byte array to dynamically compile the module at
> runtime.

## Using a Wasm file

You can use Wasm in your production deployment or locally, using [`vercel dev`](/docs/cli/dev).

- ### Get your Wasm file ready
  - Compile your existing C, Go, and Rust project to create a binary `.wasm` file. For this example, we use a [rust](https://github.com/v
  - Copy the compiled file (in our example, [`add.wasm`](https://github.com/vercel/next.js/blob/canary/examples/with-webassembly/add.wasm

- ### Create an API route for calling the Wasm file
  With `nodejs` runtime that uses [Fluid compute](/docs/fluid-compute) by default:
  ```ts filename="api/wasm/route.ts"
  import path from 'node:path';
  import fs from 'node:fs';
  import type * as addWasmModule from '../../../add.wasm'; // import type definitions at the root of your project

  const wasmBuffer = fs.readFileSync(path.resolve(process.cwd(), './add.wasm')); // path from root
  const wasmPromise = WebAssembly.instantiate(wasmBuffer);

  export async function GET(request: Request) {
    const url = new URL(request.url);
    const num = Number(url.searchParams.get('number') || 10);
    const { add_one: addOne } = (await wasmPromise).instance
      .exports as typeof addWasmModule;

    return new Response(`got: ${addOne(num)}`);
  }
  ```

- ### Call the Wasm endpoint
  - Run the project locally with `vercel dev`
  - Browse to `http://localhost:3000/api/wasm?number=12` which should return `got: 13`


---
title: "Streaming"
description: "Learn how to stream responses from Vercel Functions."
last_updated: "2026-01-16T02:19:31.775Z"
source: "https://vercel.com/docs/functions/streaming-functions"
---

# Streaming

AI providers can be slow when producing responses, but many make their responses available in chunks as they're processed. Streaming enab

**Vercel recommends using [Vercel's AI SDK](https://sdk.vercel.ai/docs) to stream responses from LLMs and AI APIs**. It reduces the boile

## Getting started

The following example shows how to send a message to one of OpenAI's models and streams:

### Prerequisites

1. You should understand how to setup a Vercel Function. See the [Functions quickstart](/docs/functions/quickstart) for more information.
2. You should also have a fundamental understanding of how streaming works on Vercel. To learn more see [What is streaming?](/docs/fundam
3. You should be using Node.js 20 or later and the [latest version](/docs/cli#updating-vercel-cli) of the Vercel CLI.
4. You should copy your OpenAI API key in the `.env.local` file with name `OPENAI_API_KEY`. See the [AI SDK docs](https://sdk.vercel.ai/d
5. Install the `ai` and `@ai-sdk/openai` packages:
   <CodeBlock>
     <Code tab="pnpm">
       ```bash
       pnpm i ai openai
       ```
     </Code>
     <Code tab="yarn">
       ```bash
       yarn i ai openai
       ```
     </Code>
     <Code tab="npm">
       ```bash
       npm i ai openai
       ```
     </Code>
     <Code tab="bun">
       ```bash
       bun i ai openai
       ```
     </Code>
   </CodeBlock>

```ts v0="build" filename="app/api/streaming-example/route.ts" framework=nextjs-app
import { streamText } from 'ai';
import { openai } from '@ai-sdk/openai';

// This method must be named GET
export async function GET() {
  // Make a request to OpenAI's API based on
  // a placeholder prompt
  const response = streamText({
```

```
    model: openai('gpt-4o-mini'),
    messages: [{ role: 'user', content: 'What is the capital of Australia?' }],
  });
  // Respond with the stream
  return response.toTextStreamResponse({
    headers: {
      'Content-Type': 'text/event-stream',
    },
  });
}
```

```js v0="build" filename="app/api/streaming-example/route.js" framework=nextjs-app
import { streamText } from 'ai';
import { openai } from '@ai-sdk/openai';

// This method must be named GET
export async function GET() {
  // Make a request to OpenAI's API based on
  // a placeholder prompt
  const response = streamText({
    model: openai('gpt-4o-mini'),
    messages: [{ role: 'user', content: 'What is the capital of Australia?' }],
  });
  // Respond with the stream
  return response.toTextStreamResponse({
    headers: {
      'Content-Type': 'text/event-stream',
    },
  });
}
```

```ts v0="build" filename="app/api/streaming-example/route.ts" framework=nextjs
// Streaming Functions must be defined in an
// app directory, even if the rest of your app
// is in the pages directory.
import { streamText } from 'ai';
import { openai } from '@ai-sdk/openai';

// This method must be named GET
export async function GET() {
  // Make a request to OpenAI's API based on
  // a placeholder prompt
  const response = streamText({
    model: openai('gpt-4o-mini'),
    messages: [{ role: 'user', content: 'What is the capital of Australia?' }],
  });
  // Respond with the stream
  return response.toTextStreamResponse({
    headers: {
      'Content-Type': 'text/event-stream',
    },
  });
}
```

```js v0="build" filename="app/api/streaming-example/route.js" framework=nextjs
// Streaming Functions must be defined in an
// app directory, even if the rest of your app
// is in the pages directory.
import { streamText } from 'ai';
import { openai } from '@ai-sdk/openai';

// This method must be named GET
export async function GET() {
  // Make a request to OpenAI's API based on
  // a placeholder prompt
  const response = streamText({
    model: openai('gpt-4o-mini'),
    messages: [{ role: 'user', content: 'What is the capital of Australia?' }],
  });
  // Respond with the stream
  return response.toTextStreamResponse({
    headers: {
      'Content-Type': 'text/event-stream',
    },
  });
}
```

```ts filename="api/chat-example.ts" framework=other
import { streamText } from 'ai';
import { openai } from '@ai-sdk/openai';

// This method must be named GET
export async function GET() {
  // Make a request to OpenAI's API based on
  // a placeholder prompt
  const response = streamText({
    model: openai('gpt-4o-mini'),
    messages: [{ role: 'user', content: 'What is the capital of Australia?' }],
  });
  // Respond with the stream
  return response.toTextStreamResponse({
    headers: {
      'Content-Type': 'text/event-stream',
    },
  });
}
```

```

```js filename="api/chat-example.js" framework=other
import { streamText } from 'ai';
import { openai } from '@ai-sdk/openai';

// This method must be named GET
export async function GET() {
  // Make a request to OpenAI's API based on
  // a placeholder prompt
  const response = streamText({
    model: openai('gpt-4o-mini'),
    messages: [{ role: 'user', content: 'What is the capital of Australia?' }],
  });
  // Respond with the stream
  return response.toTextStreamResponse({
    headers: {
      'Content-Type': 'text/event-stream',
    },
  });
}
```

## Function duration

If your workload requires longer durations, you should consider enabling [fluid compute](/docs/fluid-compute), which has [higher default l

Maximum durations can be configured for Node.js functions to enable streaming responses for longer periods. See [max durations](/docs/fun

## Streaming Python functions

You can stream responses from Vercel Functions that use the Python runtime.

When your function is streaming, it will be able to take advantage of the extended [runtime logs](/docs/functions/logs#runtime-logs), whi

## More resources

- [What is streaming?](/docs/functions/streaming)
- [AI SDK](https://sdk.vercel.ai/docs/getting-started)
- [Vercel Functions](/docs/functions)
- [Fluid compute](/docs/fluid-compute)
- [Streaming and SEO: Does streaming affect SEO?](/kb/guide/does-streaming-affect-seo)
- [Processing data chunks: Learn how to process data chunks](/kb/guide/processing-data-chunks)
- [Handling backpressure: Learn how to handle backpressure](/kb/guide/handling-backpressure)

---
title: "Legacy Usage & Pricing for Functions"
description: "Learn about legacy usage and pricing for Vercel Functions."
last_updated: "2026-01-16T02:19:31.794Z"
source: "https://vercel.com/docs/functions/usage-and-pricing/legacy-pricing"
---

# Legacy Usage & Pricing for Functions

> **⚠️ Warning:** **Legacy Billing Model**: This page describes the legacy billing model and
> relates to functions which  use Fluid Compute. All new projects
> use [Fluid Compute](/docs/fluid-compute) by default, which bills separately
> for active CPU time and provisioned memory time for more cost-effective and
> transparent pricing.

Functions using the Node.js runtime are measured in [GB-hours](/docs/limits/usage#execution), which is the [memory allocated](/docs/funct

A function can use up to 50 ms of CPU time per execution unit. If a function uses more than 50 ms, it will be divided into multiple 50 ms

See [viewing function usage](#viewing-function-usage) for more information on how to track your usage.

## Pricing

> **💡 Note:** This information relates to functions which  use Fluid Compute.
> Fluid Compute is the default for all new functions. To learn about pricing for
> functions that use Fluid Compute, see
> [Pricing](/docs/functions/usage-and-pricing).

The following table outlines the price for functions which do not use [Fluid Compute](/docs/fluid-compute).

Vercel Functions are available for free with the included usage limits:

| Resource | Hobby Included | Pro Included | On-demand with Pro |
| --- | --- | --- | --- |
| Function Duration | First 100 GB-Hours | N/A | $0.18 per 1 GB-Hour |
| Function Invocations | First 100,000 | N/A | $0.60 per 1,000,000 Invocations |

### Hobby

Vercel will send you emails as you are nearing your usage limits. On the Hobby plan you **will not pay for any additional usage**. Howeve

When your [Hobby team](/docs/plans/hobby) is set to **paused**, it remains in this state indefinitely unless you take action. This means

> **💡 Note:** If you have reached this state, your application is likely a good candidate
> for a [Pro account](/docs/plans/pro-plan).

To unpause your account, you have two main options:

- **Contact Support**: You can reach out to our [support team](/help) to discuss the reason for the pause and potential resolutions
- **Transfer to a Pro team**:
  If your Hobby team is paused, you won't have the option to initiate a [Pro trial](/docs/plans/pro-plan/trials). Instead, you can set up
  1. [Create a Pro team account](/docs/accounts/create-a-team)
  2. Add a valid credit card to this account. Select the **Settings** tab, then select **Billing** and **Payment Method**
```

Once set up, a transfer modal will appear, prompting you to [transfer your previous Hobby projects](/docs/projects/overview#transferring-

### Pro

For teams on a Pro trial, the [trial will end](/docs/plans/pro-plan/trials#post-trial-decision) when your team reaches the [trial limits]

Once your team exceeds the included usage, you will continue to be charged the on-demand costs going forward.

Pro teams can [set up Spend Management](/docs/spend-management#managing-your-spend-amount) to get notified or to automatically take actio

### Enterprise

Enterprise agreements provide custom usage and pricing for Vercel Functions, including:

- Custom [execution units](/docs/functions/runtimes/edge/edge-functions#managing-execution-units)
- Increased [maximum duration](/docs/functions/configuring-functions/duration) up to 900 seconds
- Multi-region deployments
- [Vercel Function failover](/docs/functions/configuring-functions/region#automatic-failover)

See [Vercel Enterprise plans](/docs/plans/enterprise) for more information.

## Viewing Function Usage

Usage metrics can be found in the [**Usage** tab](/dashboard/usage) on your [dashboard](/dashboard). Functions are invoked for every requ

You can see the usage for **functions using the Node.js runtime** on the **Serverless Functions** section of the **Usage** tab.

| Metric | Description | Priced |
| --- | --- | --- |
| Function Invocations | The number of times your Functions have been invoked | [Learn More |
| Function Duration | The time your Vercel Functions have spent responding to requests | [Learn M |
| Throttling | The number of instances where Functions did not execute due to concurrency limits being reached | No |

## Managing function invocations

You are charged based on the number of times your [functions](/docs/functions) are invoked, including both successful and errored invocat

When using [Incremental Static Regeneration](/docs/incremental-static-regeneration) with Next.js, both the `revalidate` option for `getSt

When viewing your Functions Invocations graph, you can group by **Ratio** to see a total of all invocations across your team's projects t

Executing a Vercel Function will increase Edge Request usage as well. Caching your Vercel Function reduces the GB-hours of your functions

### Optimizing function invocations

- Use the **Projects** option to identify which projects have the most invocations and where you can optimize.
- Cache your responses using [caching in the CDN](/docs/cdn-cache#using-vercel-functions) and [Cache-Control headers](/docs/headers#cache
- See [How can I reduce my Serverless Execution usage on Vercel?](/kb/guide/how-can-i-reduce-my-serverless-execution-usage-on-vercel) for

## Managing function duration

> **⚠️ Warning:** **Legacy Billing Model**: This describes the legacy Function duration billing
> model based on wall-clock time. For new projects, we recommend [Fluid
> Compute](/docs/functions/usage-and-pricing) which bills separately for active
> CPU time and provisioned memory time for more cost-effective and transparent
> pricing.

You are charged based on the duration your Vercel functions have run. This is sometimes called "wall-clock time", which refers to the *ac

For example, if a function [has](/docs/functions/configuring-functions/memory) 1.7 GB (1769 MB) of memory and is executed **1 million tim

- Total Seconds: 1M * (1s) = 1,000,000 Seconds
- Total GB-Seconds: 1769/1024 GB * 1,000,000 Seconds = 1,727,539.06 GB-Seconds
- Total GB-Hrs: 1,727,539.06 GB-Seconds / 3600 = 479.87 GB-Hrs
- The total Vercel Function Execution is 479.87 GB-Hrs.

To see your current usage, navigate to the **Usage** tab on your team's [Dashboard](/dashboard) and go to **Serverless Functions** > **Du

### Optimizing function duration

**Recommended: Upgrade to Fluid compute**

- **Enable [Fluid compute](/docs/fluid-compute)** for more cost-effective billing that separates active CPU time from provisioned memory

**Legacy optimization strategies:**

- Use the **Projects** option to identify which projects have the most execution time and where you can optimize.
- You can adjust the [maximum duration](/docs/functions/configuring-functions/duration) for your functions to prevent excessive run times
- To reduce the GB-hours (Execution) of your functions, ensure you are [caching in the CDN](/docs/cdn-cache#using-vercel-functions) with
- For troubleshooting issues causing functions to run longer than expected or timeout, see [What can I do about Vercel Serverless Functio

## Throttles

This counts the number of times that a request to your Functions could not be served because the [concurrency limit](/docs/functions/conc

While this is not a chargeable metric, it will cause a `503: FUNCTION_THROTTLED` error. To learn more, see [What should I do if I receive

---
title: "Fluid compute pricing"
description: "Learn about usage and pricing for fluid compute on Vercel."
last_updated: "2026-01-16T02:19:31.803Z"
source: "https://vercel.com/docs/functions/usage-and-pricing"
---

# Fluid compute pricing

Vercel Functions on fluid compute are priced based on your plan and resource usage. Each plan includes a set amount of resources per mont

| Resource | Hobby | Pro |

| ---------------------------------------------- | ------------------ | ---------------------------------------- |
| [**Active CPU**](#active-cpu-1)                | 4 hours included   | N/A                                      |
| *On-demand Active CPU*                         | -                  | Costs vary by [region](#regional-pricing) |
| [**Provisioned Memory**](#provisioned-memory-1) | 360 GB-hrs included | N/A                                      |
| *On-demand Provisioned Memory*                 | -                  | Costs vary by [region](#regional-pricing) |
| [**Invocations**](#invocations-1)              | 1 million included | N/A                                      |
| *On-demand Invocations*                        | -                  | $0.60 per million                        |

Enterprise plans have custom terms. Speak to your Customer Success Manager (CSM) or Account Executive (AE) for details.

### Resource Details

#### Active CPU

- This is the CPU time your code actively consumes in milliseconds
- You are only billed during actual code execution and not during I/O operations (database queries, like AI model calls, etc.)
- Billed per CPU-hour
- Pauses billing when your code is waiting for external services

For example: If your function takes 100ms to process data but spends 400ms waiting for a database query, you're only billed for the 100ms

#### Provisioned Memory

- Memory allocated to your function instances (in GB)
- Billed for the entire instance lifetime in GB-hours
- Continues billing while handling requests, even during I/O operations
- Each instance can handle multiple requests with [optimized concurrency](/docs/fluid-compute#optimized-concurrency)
- Memory is reserved for your function even when it's waiting for I/O
- Billing continues until the last in-flight request completes

For example: If you have a 1GB function instance running for 1 hour handling multiple requests, you're billed for 1 GB-hour of provisione

#### Invocations

- Counts each request to your function
- Billed per incoming request
- First million requests included in both Hobby and Pro plans
- Counts regardless of request success or failure

For example: If your function receives 1.5 million requests on a Pro plan, you'll be billed for the 500,000 requests beyond your included

## Regional pricing

The following table shows the regional pricing for fluid compute resources on Vercel. The prices are per hour for CPU and per GB-hr for m

| Region                          | Active CPU time (per hour) | Provisioned Memory (GB-hr) |
| ------------------------------- | -------------------------- | -------------------------- |
| Washington, D.C., USA (iad1)    | $0.128                     | $0.0106                    |
| Cleveland, USA (cle1)           | $0.128                     | $0.0106                    |
| San Francisco, USA (sfo1)       | $0.177                     | $0.0147                    |
| Portland, USA (pdx1)            | $0.128                     | $0.0106                    |
| Cape Town, South Africa (cpt1)  | $0.200                     | $0.0166                    |
| Hong Kong (hkg1)                | $0.176                     | $0.0146                    |
| Mumbai, India (bom1)            | $0.140                     | $0.0116                    |
| Osaka, Japan (kix1)             | $0.202                     | $0.0167                    |
| Seoul, South Korea (icn1)       | $0.169                     | $0.0140                    |
| Singapore (sin1)                | $0.160                     | $0.0133                    |
| Sydney, Australia (syd1)        | $0.180                     | $0.0149                    |
| Tokyo, Japan (hnd1)             | $0.202                     | $0.0167                    |
| Frankfurt, Germany (fra1)       | $0.184                     | $0.0152                    |
| Dublin, Ireland (dub1)          | $0.168                     | $0.0139                    |
| London, UK (lhr1)               | $0.177                     | $0.0146                    |
| Paris, France (cdg1)            | $0.177                     | $0.0146                    |
| Stockholm, Sweden (arn1)        | $0.160                     | $0.0133                    |
| Dubai, UAE (dxb1)               | $0.185                     | $0.0153                    |
| São Paulo, Brazil (gru1)        | $0.221                     | $0.0183                    |
| Montréal, Canada (yul1)         | $0.147                     | $0.0121                    |

## How pricing works

A function instance runs in a region, and its pricing is based on the resources it uses in that region. The cost for each invocation is c

When the first request arrives, Vercel starts an instance with your configured memory. Provisioned memory is billed continuously until th

After all requests complete, the instance is paused, and no CPU or memory charges apply until the next invocation. This means, you pay fo

### Example

Suppose you deploy a function with 4 GB of memory in the São Paulo, Brazil region, where the rates are $0.221/hour for CPU and $0.0183/GB

- CPU: (4 seconds / 3600) × $0.221 = $0.0002456
- Memory: (4 GB × 10 seconds / 3600) × $0.0183 = $0.0002033
- Total: $0.0002456 + $0.0002033 = $0.0004489 for each invocation.

---
title: "Vercel fundamentals"
description: "Learn about the core concepts of Vercel"
last_updated: "2026-01-16T02:19:31.806Z"
source: "https://vercel.com/docs/fundamentals"
---

# Vercel fundamentals

The articles below outline key ideas that shape how Vercel handles computation and streaming:

- [What is Compute?](/docs/fundamentals/what-is-compute): Explains how Vercel manages building, rendering, and on-demand tasks. Including
  - Dedicated servers vs. Vercel functions
  - Cold starts, region models, and maximum durations
  - Fluid compute

- [What is Streaming?](/docs/fundamentals/what-is-streaming): Shows how to send data progressively from Vercel Functions using the Web St
  - Chunks, backpressure, and flow control
  - Real-time use cases like AI responses or ecommerce updates
  - Strategies to enhance perceived performance and responsiveness


---
title: "What is Compute?"
description: "Learn about the different models for compute and how they can be used with Vercel."
last_updated: "2026-01-16T02:19:31.822Z"
source: "https://vercel.com/docs/fundamentals/what-is-compute"
---

# What is Compute?

## Where does compute happen?

Traditionally with web applications, we talk about two main locations:

- **Client**: This is the browser on your *user's* device that sends a request to a server for your application code. It then turns the r
- **Server**: This is the computer in a data center that stores your application code. It receives requests from a client, does some comp
  - **Origin Server**: The server that stores and runs the original version of your app code. When the origin server receives a request,
  - **CDN (Content Delivery Network)**: This stores static content, such as HTML, in multiple locations around the globe, placed between
  - **Global Network**: Vercel's global network consists of Points of Presence (PoPs) and compute regions distributed around the world. T

## Compute in practice

To demonstrate an example of what this looks like in practice, we'll use the example of a Next.js app deployed to Vercel.

When you start a deployment of your Next.js app to Vercel, Vercel's [build process](/docs/deployments/builds#build-process) creates a bui

Now that the deployment is ready to serve traffic, a user can visit your site. When they do, the request is sent to the closest region, w

1. **User Action**: The user interacts with a website by clicking a link, submitting a form, or entering a URL.
2. **HTTP Request**: The user's browser sends a request to the server, asking for the resources needed to display the webpage.
3. **Server Processing**: The server receives the request, processes it, and prepares the necessary resources. For Vercel Functions, Verc
4. **HTTP Response**: The server sends back a response to the browser, which includes the requested resources and a status code indicatin

In this lifecycle, the "Server Processing" step can look very different depending on your needs, the artifacts being requested, and the m

## Servers

Servers provide a specific environment and resources for your applications. This means that you have control over the environment, but yo

Managing your own servers can work well if you have a highly predictable workload. You don't have a need to scale up or down and you have

### Server advantages

Servers give you complete control to configure the environment to suit your needs. You can set the CPU power and RAM for consistent perfo

### Server disadvantages

If you have peaks of traffic, you'll need to anticipate and provision additional resources in advance, which can lead to 2 possible scena

- Under provisioning: leads to degraded performance due to lack of compute availability.
- Over provisioning: leads to increased costs due to wasted compute capacity.

Furthermore, because scaling resources can be slow, you will need to apply it in advance of the time where traffic peaks are expected.

## Serverless

Serverless is a cloud computing model that allows you to build and run applications and services without having to manage your own server

The term "Serverless" has been used by several cloud providers to describe the compute used for functions, such as AWS Lambda functions, (

The difference between serverless and servers, is that there is no single server assigned to your application. Instead, when a request is

### Serverless advantages

With serverless, applications are automatically scaled up or down based on demand, ensuring that resources are used efficiently and costs

### Serverless disadvantages

#### Cold starts

When adding additional capacity to a serverless application there is a short period of initialization time that happens as the first requ

Reusing a function means the underlying instance that hosts it does not get discarded. State, such as temporary files, memory caches, and

By their very nature of being on-demand, serverless applications will always have the notion of cold starts.

With Vercel, pre-warmed instances are enabled for paid plans on production environments. This prevents cold starts by keeping a minimum o

#### Region model

Serverless compute typically happens in a single specified location (or [region](/docs/functions/regions)). Having a single region (or sm

However, a single region can be a disadvantage if you have user request coming from different region, as the response latency will be hig

All of this means that it's left up to teams to determine which region (or regions) they want Vercel to deploy their functions to. This r

#### High maximum duration

AI-driven workloads have stretched the limits of serverless compute, through the requirement of long-running processes, data-intensive ta

The maximum duration of a function describes the maximum amount of time that a function can run before it is terminated. As a user, you h

This can be a challenge, as you may not know how long a task will take to complete, and if you set the duration too low, the function wil

## Fluid compute

Fluid compute is a hybrid approach between [serverless](#serverless) and [servers](#servers), and it builds upon the benefits of serverle

### How does Fluid compute work

In the serverless compute model, one serverless instance can process only one request at a time so that the number of instances needed ca

In the Fluid compute model, when a request requires a function to be executed, a new compute instance is started if there are no existing

### Benefits of Fluid compute

#### Optimized concurrency

Resource usage is optimized by handling multiple request with invocations in one function instance and dynamically routing traffic to ins

#### Reduction in cold starts

Optimized concurrency reduces the likelihood of [cold starts](#cold-starts), a disadvantage of serverless, as there is less chance that a

- **Bytecode caching**: It automatically pre-compiles function code to minimize startup time during cold invocations.
- **Pre-warmed instances**: It keeps functions ready to handle requests without cold start delays.

#### Dynamic scaling

Fluid compute includes one of the advantages of serverless with the ability of automatically adjusting the number of concurrent instances

#### Background processing

Serverless computing is designed for quick tasks that are short-lived. With Fluid compute, you can execute background tasks with [`waitUn

#### Cross-region failover

Fluid compute includes backup regions where it can launch function instances and route traffic to in case of outages in the regions where

#### Compute instance sharing

As opposed to traditional serverless where instances are completely isolated, Fluid compute allows multiple invocations to share the same

### Enabling Fluid compute

You can enable Fluid compute from the [Functions Settings](https://vercel.com/d?to=/%5Bteam%5D/%5Bproject%5D/settings/functions%fluid-com

---
title: "Buy a domain"
description: "Purchase your domain with Vercel. Expand your online reach and establish a memorable online identity."
last_updated: "2026-01-16T02:19:31.841Z"
source: "https://vercel.com/docs/getting-started-with-vercel/buy-domain"
---

# Buy a domain

Use Vercel to find and buy a domain that resonates with your brand, establishes credibility, and captures your visitors' attention.

> **💡 Note:** All domains purchased on Vercel have WHOIS privacy enabled by default.

- ### Find a domain
  Go to [https://vercel.com/domains](/domains) and search for a domain that matches you or your brand. You could try "SuperDev"!

  Depending on the TLD (top-level domain), you'll see the purchase price. Domains with **Premium** badges are more expensive. You can sor

- ### Select your domain(s)
  - Select an address by clicking the  button next to the available domain, or continue searching until you find the perfect one.
  - When you click the  button, Vercel adds the domain to your domains cart. You can continue to add more domains from the same results o

- ### Purchase your domain(s)
  - Click on the **Cart**  button on the top right and review the list of domains and prices that you added.
  - Then, click **Proceed to Checkout**. You can also change the team under which you are making this purchase at this stage.

- ### Enter payment details and registrant information
  - You'll need to enter your billing and credit card details to purchase the domain on the checkout page. These details are saved for [a
  - You'll also need to enter your registrant information and confirm it for [ICANN](https://www.icann.org/) purposes.
  - Click **Buy** to complete the purchase.
  > **💡 Note:** For the ICANN registrant information:

- ### Configure your domain
  - Once the purchase is complete, you can click **Configure** next to each purchased domain on the checkout page.
  - You'll have the following options:
    - Connect the domain to an existing project
    - Create a new project to connect the domain to
    - Manage the domain's DNS records
  You can also configure your domain from the [project's domains dashboard page](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2F

## Next steps

Next, learn how to take advantage of Vercel's collaboration features as part of your developer workflow:

---
title: "Collaborate on Vercel"
description: "Amplify collaboration and productivity with Vercel"
last_updated: "2026-01-16T02:19:31.867Z"
source: "https://vercel.com/docs/getting-started-with-vercel/collaborate"
---

# Collaborate on Vercel

Collaboration is key in successful development projects, and Vercel offers robust features to enhance collaboration among developers. Fro

## Make Changes

Now that your project is publicly available on your domain of choice, it’s time to begin making changes to it. With Vercel's automatic de

> **💡 Note:** A Production environment is one built from the `main` or development branch of
> your Git repository. A preview environment is created when you deploy from any
> other branch.

Vercel provides a [URL](/docs/deployments/generated-urls#generated-from-git) that reflects the latest pushes to that branch. You can find

This connection was established for you automatically, so all you have to do is push commits, and you will start receiving links to deplo

## Create a preview deployment

- ### Make your changes
  Create a new branch in your project and make some changes

- ### Commit your changes
  Commit those changes and create a pull request. After a few seconds, Vercel picks up the changes and starts to build and deploy your pr

- ### Inspect your deployment information
  Select **Inspect** to explore the build within your dashboard. You can see the build is within the preview environment and additional i

- ### View your deployment URL
  Return to your pull request. At this point your build should be deployed and you can select **Visit Preview**. You can now see your cha

## Commenting on previews

[Comments](/docs/comments) provide a way for your team [or friends](/docs/comments/how-comments-work#sharing) to give direct feedback on

- ### Open your deployment
  Open the preview deployment that you’d like to share by selecting the **Domain** from the deployment information as shown in step 3 abo

- ### Authenticate with your Vercel account
  From the Comments toolbar at the bottom of the screen, select **Log in to comment** and sign in with your Vercel account.

- ### Adjust the share settings
  Select **Share** in the [Toolbar](/docs/vercel-toolbar) menu. Add the emails of people you would like to share the preview with. If you

  To learn more, including other ways to share, see [Sharing Deployments](/docs/deployments/sharing-deployments).

- ### Collaborator needs to sign-in
  The person you are sharing the preview with needs to have a Vercel account. To do so, they'll need to select **Log in to comment** and

- ### Collaborator can comment
  Once the person you are sharing the preview with goes through the security options, they'll be ready to comment. You'll be notified of

For more information on using Comments, see [Using comments](/docs/comments/using-comments).

---
title: "Add a domain"
description: "Easily add a custom domain to your Vercel project. Enhance your brand presence and optimize SEO with just a few clicks."
last_updated: "2026-01-16T02:19:31.871Z"
source: "https://vercel.com/docs/getting-started-with-vercel/domains"
---

# Add a domain

Assigning a custom domain to your project guarantees that visitors to your application will have a tailored experience that aligns with y

On Vercel, this domain can have any format of your choosing:

- `acme.com` ([apex domain](/docs/domains/working-with-domains#apex-domain))
- `blog.acme.com` ([subdomain](/docs/domains/working-with-domains#subdomain))
- `*.acme.com` ([wildcard domain](/docs/domains/working-with-domains#wildcard-domain))

If you already own a domain, you can point it to Vercel, or transfer it over. If you don't own one yet, you can purchase a new one. For t

For more information on domains at Vercel, see [Domains overview](/docs/domains).

### Next steps

Now that your site is deployed, you can to personalize it by setting up a custom domain. With Vercel you can either **buy a new domain** 

- [Buy a new domain](/docs/getting-started-with-vercel/buy-domain)
- [Use an existing domain](/docs/getting-started-with-vercel/use-existing)

---
title: "Import an existing project"
description: "Create a new project on Vercel by importing your existing frontend project, built on any of our supported frameworks."
last_updated: "2026-01-16T02:19:31.879Z"
source: "https://vercel.com/docs/getting-started-with-vercel/import"
---

# Import an existing project

Your existing project can be any web project that outputs static HTML content (such as a website that contains HTML, CSS, and JavaScript)

- ### Connect to your Git provider
  On the [New Project](/new) page, under the **Import Git Repository** section, select the Git provider that you would like to import you

  Follow the prompts to sign in to either your [GitHub](/docs/git/vercel-for-github), [GitLab](/docs/git/vercel-for-gitlab), or [BitBucke

- ### Import your repository
  Find the repository in the list that you would like to import and select **Import**.

- ### Optionally, configure any settings
  Vercel will automatically detect the framework and any necessary build settings. However, you can also configure the Project settings a
  - To update the [framework](/docs/deployments/configure-a-build#framework-preset), [build command](/docs/deployments/configure-a-build#

  - To set environment variables, expand the **Environment Variables** section and either paste or copy them in.
  - You can also configure additional properties by adding a **[vercel.json](/docs/project-configuration)** to your project. You can eith

- ### Deploy your project
  Press the **Deploy** button. Vercel will create the Project and deploy it based on the chosen configurations.

- ### Enjoy the confetti!
  To view your deployment, select the Project in the dashboard and then select the **Domain**. This page is now visible to anyone who has

## Next Steps

Next, learn how to assign a domain to your new deployment.

---
title: "Next Steps"
description: "Discover the next steps to take on your Vercel journey. Unlock new possibilities and harness the full potential of your pro
last_updated: "2026-01-16T02:19:31.847Z"
source: "https://vercel.com/docs/getting-started-with-vercel/next-steps"
---

# Next Steps

Congratulations on getting started with Vercel!

Now, let's explore what's next on your journey. At this point, you can either continue learning more about Vercel's many features, or you

Alternatively, you can start learning about many of the products and features that Vercel provides:

## Infrastructure

Learn about Vercel's CDN and implement scalable infrastructure in your app using Functions. Get started today by implementing a Vercel Fu

- [Vercel functions quickstart](/docs/functions/quickstart)

## Storage

Vercel offers a suite of managed, serverless storage products that integrate with your frontend framework.

Learn more about [which storage option is right for you](/docs/storage#choosing-a-storage-product) and get started with implementing them

- [Vercel Blob](/docs/vercel-blob/server-upload)
- [Vercel Edge Config](/docs/edge-config/get-started)

## Observability

Vercel provides a suite of observability tools to allow you to monitor, analyze, and manage your site.

- [Monitoring](/docs/observability/monitoring)
- [Web Analytics](/docs/analytics/quickstart)
- [Speed Insights](/docs/speed-insights/quickstart)

## Security

Vercel takes security seriously. It uses HTTPS by default for secure data transmission, regularly updates its platform to mitigate potent

- [Security overview](/docs/security)
- [DDoS Mitigation](/docs/security/ddos-mitigation)

---
title: "Getting started with Vercel"
description: "This step-by-step tutorial will help you get started with Vercel, an end-to-end platform for developers that allows you to
last_updated: "2026-01-16T02:19:31.852Z"
source: "https://vercel.com/docs/getting-started-with-vercel"
---

# Getting started with Vercel

Vercel is a platform for developers that provides the tools, workflows, and infrastructure you need to build and deploy your web apps fas

Vercel supports [popular frontend frameworks](/docs/frameworks) out-of-the-box, and its scalable, secure infrastructure is globally distr

During development, Vercel provides tools for real-time collaboration on your projects such as automatic preview and production environme

## Before you begin

To get started, create an account with Vercel. You can [select the plan](/docs/plans) that's right for you.

- [Sign up for a new Vercel account](/signup)
- [Log in to your existing Vercel account](/login)

Once you create an account, you can choose to authenticate either with a Git provider or by using an email. When using email authenticati

## Customizing your journey

This tutorial is framework agnostic but Vercel supports many frontend [frameworks](/docs/frameworks/more-frameworks). As you go through t

While many of our instructions use the dashboard, you can also use [Vercel CLI](/docs/cli) to carry out most tasks on Vercel. In this tut

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i vercel
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i vercel
    ```

  </Code>
  <Code tab="npm">
    ```bash
    npm i vercel
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i vercel
    ```
  </Code>
</CodeBlock>


---
title: "Projects and deployments"
description: "Streamline your workflow with Vercel"
last_updated: "2026-01-16T02:19:31.857Z"
source: "https://vercel.com/docs/getting-started-with-vercel/projects-deployments"
---

# Projects and deployments

To get started with Vercel, it's helpful to understand **projects** and **deployments**:

- **Projects**: A [project](/docs/projects/overview) is the application that you have deployed to Vercel. You can have multiple projects
- **Deployments**: A [deployment](/docs/deployments) is the result of a successful [build](/docs/deployments/builds# "Build Step") of you

### More resources

To get started you'll create a new project by either **deploying a template** or **importing and deploying** an existing project:

- [Deploy a template](/docs/getting-started-with-vercel/template)
- [Import an existing project](/docs/getting-started-with-vercel/import)


---
title: "Use a template"
description: "Create a new project on Vercel by using a template"
last_updated: "2026-01-16T02:19:31.899Z"
source: "https://vercel.com/docs/getting-started-with-vercel/template"
---

# Use a template

Accelerate your development on Vercel with [Templates](/templates). This guide will show you how to use templates to fast-track project s

- ### Find a template
  From [https://vercel.com/templates](/templates), select the template you'd like to deploy. You can use the filters to select a template

  Not sure which one to use? How about [exploring Next.js](https://vercel.com/templates/next.js/nextjs-boilerplate).

- ### Deploy the template to Vercel
  Once you've selected a template, Click **Deploy** on the template page to start the process.

- ### Connect your Git provider
  To ensure you can easily update your project after deploying it, Vercel will create a new repository with your chosen [Git provider](/d

  First, select the Git provider that you'd like to connect to. Once you've signed in, you'll need to set the scope and repository name. A

- ### Project deployment
  Once the project has been cloned to your git provider, Vercel will automatically start deploying the project. This starts with [buildin

- ### View your dashboard
  At this point, you've created a **production** deployment, with its very own domain assigned. If you continue to your [dashboard](/dash

- ### Clone the project to your machine
  Finally, you'll want to clone the source files to your local machine so that you can make some changes later. To do this from your dash

> **💡 Note:** Because you used a template, we've automatically included any additional
> environment set up as part of the template. You can customize your project by
> configuring environment variables and build options.Environment Variables are key-value pairs that can be defined in your project
> settings for each [Environment](/docs/environment-variables#environments).
> Teams can also use [shared environment
> variables](/docs/environment-variables/shared-environment-variables) that are
> linked between multiple projects.Vercel automatically configures builds settings based on your framework, but
> you can [customize the build](/docs/deployments/configure-a-build) in your
> project settings or within a [vercel.json](/docs/project-configuration) file.

## Next Steps

Next, learn how to assign a domain to your new deployment.


---
title: "Use an existing domain"
description: "Seamlessly integrate your existing domain with Vercel. Maximize flexibility and maintain your established online presence."
last_updated: "2026-01-16T02:19:31.906Z"
source: "https://vercel.com/docs/getting-started-with-vercel/use-existing"
---

# Use an existing domain

Already have a domain you love? Seamlessly integrate it with Vercel to leverage the platform's powerful features and infrastructure. Whet

- ### Go to your project's domains settings
  Select your project and select the **Settings** tab. Then, select the **Domains** menu item or click on this [link](https://vercel.com/

- ### Add your existing domain to your project

  From the **Domains** page, enter the domain you wish to add to the project.

  If you add an apex domain (e.g. `example.com`) to the project, Vercel will prompt you to add the `www`subdomain prefix, the apex domain

  For more information on which redirect option to choose, see [Redirecting `www` domains](/docs/domains/deploying-and-redirecting#redire

- ### Configure your DNS records
  Configure the DNS records of your domain with your registrar so it can be used with your Project. The dashboard will automatically disp
  - If the domain is in use by another Vercel account, you will need to [verify access to the domain](/docs/domains/add-a-domain#verify-d
  - If you're using an **[Apex domain](/docs/domains/add-a-domain#apex-domains)** (e.g. example.com), you will need to configure it with
  - If you're using a **[Subdomain](/docs/domains/add-a-domain#subdomains)** (e.g. docs.example.com), you will need to configure it with
  Both apex domains and subdomains can also be configured using the **[Nameservers](/docs/domains/add-a-domain#vercel-nameservers)** meth

## Next steps

Next, learn how to take advantage of Vercel's collaboration features as part of your developer workflow:

---
title: "Deploying Git Repositories with Vercel"
description: "Vercel allows for automatic deployments on every branch push and merges onto the production branch of your GitHub, GitLab,
last_updated: "2026-01-16T02:19:31.930Z"
source: "https://vercel.com/docs/git"
---

# Deploying Git Repositories with Vercel

Vercel allows for **automatic deployments on every branch push** and merges onto the [production branch](#production-branch) of your [Git

Using Git with Vercel provides the following benefits:

- [Preview deployments](/docs/deployments/environments#preview-environment-pre-production#) for every push.
- [Production deployments](/docs/deployments/environments#production-environment) for the most recent changes from the [production branch
- Instant rollbacks when reverting changes assigned to a custom domain.

When working with Git, have a branch that works as your production branch, often called `main`. After you create a pull request (PR) to t
Vercel will create a production deployment.

You can choose to use a different branch as the [production branch](#production-branch).

## Supported Git Providers

- [GitHub Free](https://github.com/pricing)
- [GitHub Team](https://github.com/pricing)
- [GitHub Enterprise Cloud](https://docs.github.com/en/get-started/learning-about-github/githubs-products#github-enterprise)
- [GitLab Free](https://about.gitlab.com/pricing/)
- [GitLab Premium](https://about.gitlab.com/pricing/)
- [GitLab Ultimate](https://about.gitlab.com/pricing/)
- [GitLab Enterprise](https://about.gitlab.com/enterprise/)
- [Bitbucket Free](https://www.atlassian.com/software/bitbucket/pricing)
- [Bitbucket Standard](https://www.atlassian.com/software/bitbucket/pricing)
- [Bitbucket Premium](https://www.atlassian.com/software/bitbucket/pricing)
- [Azure DevOps Pipelines](https://learn.microsoft.com/en-us/azure/devops/pipelines/get-started/what-is-azure-pipelines)

### Self-Hosted examples

- [GitHub Enterprise Server](/kb/guide/how-can-i-use-github-actions-with-vercel)
- [Self-Managed GitLab](https://vercel.com/kb/guide/how-can-i-use-gitlab-pipelines-with-vercel)
- [Bitbucket Data Center (Self-Hosted)](/kb/guide/how-can-i-use-bitbucket-pipelines-with-vercel)

If your provider is not listed here, you can also use the [Vercel CLI to deploy](/kb/guide/using-vercel-cli-for-custom-workflows) with an

## Deploying a Git repository

Setting up your GitHub, GitLab, or Bitbucket repository on Vercel is only a matter of clicking the ["New Project"](/new) button on the to

> **💡 Note:** For Azure DevOps repositories, use the [Vercel Deployment
> Extension](/docs/git/vercel-for-azure-pipelines)

After clicking it, you'll be presented with a list of Git repositories that the Git account you've signed up with has write access to.

To select a different Git namespace or provider, you can use the dropdown list on the top left of the section.

You can also:

- Select a third-party Git repository by clicking on [Import Third-Party Git Repository](/new/git/third-party) on the bottom of the secti
- Select a pre-built solution from the section on the right.

After you've selected the Git repository or template you want to use for your new project, you'll be taken to a page where you can config

You can:

- Customize the project's name
- Select [a **Framework Preset**](/docs/deployments/configure-a-build#framework-preset)
- Select the root directory of your project
- Configure [Build Output Settings](/docs/deployments/configure-a-build#build-command)
- Set [Environment Variables](/docs/environment-variables)

When your settings are correct, you can select the **Deploy** button to initiate a deployment.

### Creating a deployment from a Git reference

You can initiate new deployments directly from the Vercel Dashboard using a Git reference. This approach is ideal when automatic deployme

To create a deployment from a Git reference:

1. From your [dashboard](/dashboard), select the project you'd like to create a deployment for

2. Select the **Deployments** tab. Once on the Deployments page, select the **Create Deployment** button

3. Depending on how you would like to deploy, enter the following:
   - **Targeted Deployments:** Provide the unique ID (SHA) of a commit to build a deployment based on that specific commit
   - **Branch-Based Deployments:** Provide the full name of a branch when you want to build the most recent changes from that specific br

4. Select **Create Deployment**. Vercel will build and deploy your commit or branch as usual

When the same commit appears in multiple branches, Vercel will prompt you to choose the appropriate branch configuration. This choice is

## Deploying private Git repositories

As an additional security measure, commits on private Git repositories (and commits of forks that are targeting those Git repositories) w

Depending on whether the owner of the connected Vercel project is a Hobby or a Pro team, the behavior changes as mentioned in the section

This only applies to commit authors on GitHub organizations, GitLab groups and non-personal Bitbucket workspaces. It does not apply to co

For public Git repositories, [a different behavior](/docs/git#deploying-forks-of-public-git-repositories) applies.

### Using Pro teams

To deploy commits under a Vercel Pro team, the commit author must be a member of the team containing the Vercel project connected to the

Membership is verified by finding the Vercel user associated with the commit author through [**Login Connections**](/docs/accounts#login-

If the commit author is not a member, the deployment will be prevented, and the commit author can request to join the team. The team owne

If the request is declined, the commit will remain undeployed. If the commit author is accepted as a member of the Pro team, their most r

Commit authors are automatically considered part of the Pro team on Vercel if one of the existing members has connected their account on

### Using Hobby teams

You cannot deploy to a Hobby team from a private repository in a GitHub organization, GitLab group, or Bitbucket workspace. Consider maki

To deploy commits under a Hobby team, the commit author must be the owner of the Hobby team containing the Vercel project connected to th

If the commit author is not the owner of the destination Hobby team, the deployment will be prevented, and a recommendation to transfer t

After transferring the project to a Pro team, commit authors can be added as members of that team. The behavior mentioned in the [section

## Deploying forks of public Git repositories

When a public repository is forked, commits from it will usually deploy automatically. However, when you receive a pull request from a fo

The authorization step will be skipped if the commit author is already a [team member](/docs/accounts/team-members-and-roles) on Vercel.

## Production branch

A [Production deployment](/docs/deployments/environments#production-environment "Production deployment") will be created each time you me

### Default configuration

When you create a new Project from a Git repository on Vercel, the Production Branch will be selected in the following order:

- The `main` branch.
- If not present, the `master` branch ([more details](https://vercel.com/blog/custom-production-branch#a-note-on-the-master-branch)).
- \[Only for Bitbucket]: If not present, the "production branch" setting of your Git repository is used.
- If not present, the Git repository's default branch.

### Customizing the production branch

On the **Environments** page in the **Project Settings**, you can change your production branch:

- Click on the **Production** environment and go to **Branch Tracking**
- Change the name of the branch and click **Save**

Whenever a new commit is then pushed to the branch you configured here, a [production deployment](/docs/deployments/environments#producti

## Preview branches

While the [production branch](/docs/git#production-branch) is a single Git branch that contains the code that is served to your visitors,

For example, if your production branch is `main`, then [by default](/docs/git#using-custom-environments) all the Git branches that are no

To learn more about previews, see the [Preview Deployments](/docs/deployments/environments#preview-environment-pre-production) page.

By default, every preview branch automatically receives its own domain similar to the one shown below, whenever a commit is pushed to it.

### Multiple preview phases

For most use cases, the default preview behavior mentioned above is enough. If you'd like your changes to pass through multiple phases of

For example, you could create a phase called "Staging" where you can accumulate Preview changes before merging them onto production by fo

1. Create a Git branch called "staging" in your Git repository.
2. Add a domain of your choice (like `staging.example.com`) on your Vercel project and assign it to the "staging" Git branch [like this](
3. Add Environment Variables that you'd like to use for your new Staging phase on your Vercel project [like this](/docs/environment-varia
4. Push to the "staging" Git branch to update your Staging phase and automatically receive the domain and environment variables you've de
5. Once you're happy with your changes, you would then merge the respective Preview Branch into your production branch. However, unlike w

Alternatively, teams on the Pro plan can use [custom environments](/docs/deployments/environments#custom-environments).

### Using custom environments

[Custom environments](/docs/deployments/environments#custom-environments) allow you to create and define a pre-production environment. As

---

title: "Deploying Azure DevOps Pipelines with Vercel"
description: "Vercel for Azure DevOps allows you to deploy Azure Pipelines to Vercel automatically."
last_updated: "2026-01-16T02:19:31.963Z"
source: "https://vercel.com/docs/git/vercel-for-azure-pipelines"
---

# Deploying Azure DevOps Pipelines with Vercel

The [Vercel Deployment Extension](https://marketplace.visualstudio.com/items?itemName=Vercel.vercel-deployment-extension) allows you to a

Once the Vercel extension is set up, your Azure DevOps project is connected to your [Vercel Project](/docs/projects/overview). You can th

This page will help you use the extension in your own use case. You can:

- Follow the [quickstart](#quickstart) to set up the extension and trigger a production deployment based on commits to the `main` branch
- Use the [full-featured pipeline](#full-featured-azure-pipeline-creation) for a similar setup as [Vercel's other git integrations](/docs
- Review the [extension task reference](#extension-task-reference) to customize the pipeline for your specific use case

## Quickstart

At the end of this quickstart, your Azure DevOps Pipeline will trigger a Vercel production deployment whenever you commit a change to the

1. Create a Vercel Personal Access Token
2. Create secret variables
3. Set up the Vercel Deployment Extension from the Visual Studio marketplace
4. Set up a basic Azure Pipeline to trigger production deployments on Vercel
5. Test your workflow

Once you have the Vercel Deployment extension set up, you only need to modify your Azure DevOps Pipeline (Steps 4 and 5) to change the de

### Prerequisites

To create an empty Vercel project:

1. Use the [Vercel CLI](/docs/cli/project) with the `add` command

```bash filename="terminal"
vercel project add
```

2. Or through the [dashboard](/docs/projects/overview#creating-a-project) and then disconnect the [Git integration](/docs/projects/overvi

### Extension and Pipeline set up

- ### Create a Vercel Personal Access Token
  - Follow [Creating an Access Token](/docs/rest-api#creating-an-access-token) to create an access token with the scope of access set to
  - Copy this token to a secure location

- ### Create secret variables
  For security purposes, you should use the above created token in your Azure Pipeline through [secret variables](https://learn.microsoft
  - For this quickstart, we will create the secret variables when we create the pipeline. Once created, these variables will always be ac
  - Otherwise, you can create them before you create the pipeline in a [variable group](https://learn.microsoft.com/en-us/azure/devops/pi

- ### Set up the Vercel Deployment Extension
  - Go to the [Vercel Deployment Extension Visual Studio marketplace page](https://marketplace.visualstudio.com/items?itemName=Vercel.ver
  - Click **Get it free** and select the Azure DevOps organization where your Azure Project is located

- ### Set up a basic Azure Pipeline
  This step assumes that your code exists as a repository in your Azure Project's **Repos** and that your Vercel Project is named `azure-
  - From your Azure DevOps Project, select **Pipelines** from the left side bar
  - Select the **New Pipeline** button
  - Select where your code is located (In this example, we uploaded the code as an **Azure Repos Git**. Select **Azure Repos Git** and th
  - Select **Node.js** for the pipeline configuration
  - In the **Review your pipeline YAML** step, select **Variables** on the top right
    - Select **New Variable**, use `VERCEL_TOKEN` as the name and the value of the Vercel Personal Access Token you created earlier. Chec
  - Close the **Variables** window and paste the following code to replace the code in `azure-pipelines.yml` that you can rename to `verc
  ```json filename="vercel-pipeline.yml"
  trigger:
  - main

  pool:
    vmImage: ubuntu-latest

  steps:
  - task: vercel-deployment-task@1
    inputs:
      vercelProjectId: 'prj_mtYjOMP83muZkYDs2DIDfasdas' //Example Vercel Project ID
      vercelOrgId: '3Gcd2ASTsPxwxTsYBwJTB11p' //Example Vercel Personal Account ID
      vercelToken: $(VERCEL_TOKEN)
      production: true
  ```
  #### Value of `vercelProjectId`
  Look for **Project ID** located on the Vercel Project's Settings page at **Project Settings > General**.
  #### Value of `vercelOrgId`
  - If your Project is located under your Hobby team, look for **Your ID** under your Vercel Personal Account [Settings](https://vercel.c
  - If your Project is located under a Team, look for **Team ID** under **Team Settings > General**
  - Select **Save and Run**
  - This should trigger a production deployment in your Vercel Project as no code was committed before

- ### Test your workflow
  - Make a change in your code inside **Azure Repos** from the `main` branch and commit the change
  - This should trigger another deployment in your Vercel Project

Your Azure DevOps project is now connected to your Vercel project with automatic production deployments on the `main` branch. You can upd

## Full-featured Azure Pipeline creation

In a production environment, you will often want the following to happen:

- Trigger preview deployments for pull requests to the `main` branch

- Trigger production deployments only for commits to the `main` branch

Before you update your pipeline file to enable preview deployments, you need to configure Azure DevOps with pull requests.

### Triggers and comments on pull requests

In order to allow pull requests in your Azure repository to create a deployment and report back with a comment, you need the following:

- An Azure DevOps Personal Access Token
- A build validation policy for your branch

### Create an Azure DevOps Personal Access Token

1. Go to your [Azure DevOps account](https://dev.azure.com) and select the **user settings** icon on the top right
2. Select **Personal access tokens** from the menu option
3. Select the **New Token** button
4. After completing the basic token information such as Name, Organization, and Expiration, select the **Custom defined** option under **
5. At the bottom of the form, select **Show all scopes**
6. Browse down the scopes list until **Pull Request Threads**. Select the **Read & Write** checkbox
7. Select **Create** at the bottom of the form
8. Make sure you copy the token to a secure location before you close the prompt

### Create a build validation policy

1. Go to your Azure DevOps Project's page
2. Select **Project settings** in the lower left corner
3. From the Project settings left side bar, select **Repositories** under **Repos**
4. Select the repository where your vercel pipeline is set up
5. Select the **Policies** tab on the right side
6. Scroll down to **Branch Policies**, and select the `main` branch
7. Scroll down to **Build Validation** and select on the **+** button to create a new validation policy
8. Select the pipeline you created earlier and keep the policy marked as **Required** so that commits directly to main are prevented
9. Select **Save**

Create a pull request to the `main` branch. This will trigger the pipeline, run the deployment and comment back on the pull request with

### Update your pipeline

- From your Azure DevOps Project, select **Pipelines** from the left side bar
- Select the pipeline that you want to edit by selecting the  icon
- Select the **Variables** button and add a new secret variable called `AZURE_TOKEN` with the value of the Azure DevOps Personal Access T
- Close the **Variables** window and paste the following code to replace the code in `vercel-pipelines.yml`

```json filename="vercel-pipeline.yml"
trigger:
- main

pool:
  vmImage: ubuntu-latest

variables:
  isMain: $[eq(variables['Build.SourceBranch'], 'refs/heads/main')]

steps:
- task: vercel-deployment-task@1
  name: 'Deploy'
  inputs:
    condition: or(eq(variables.isMain, true), eq(variables['Build.Reason'], 'PullRequest'))
    vercelProjectId: 'azure-devops-extension'
    vercelOrgId: '3Gcd2ASTsPxwxTsYBwJTB11p' //Example Vercel Personal Account ID
    vercelToken: $(VERCEL_TOKEN)
    production: $(isMain)
- task: vercel-azdo-pr-comment-task@1
  inputs:
    azureToken: $(AZURE_TOKEN)
    deploymentTaskMessage: $(Deploy.deploymentTaskMessage)
```

- Select **Save**

> **💡 Note:** The `vercel-deployment-task` creates an [output
> variable](https://learn.microsoft.com/en-us/azure/devops/pipelines/process/variables)
> called `deploymentTaskMessage`. By setting the `name:` of the step to
> `'Deploy'`, you can access it using `$(Deploy.deploymentTaskMessage)` which
> you can then assign to the input option `deploymentTaskMessage` of the
> `vercel-azdo-pr-comment-task` task step.

### Create a pull request and test

- Create a new branch in your Azure DevOps repository and push a commit
- Open a pull request against the `main` branch
- This will trigger a pipeline execution and create a preview deployment on Vercel
- Once the deployment has completed, you will see a comment on the pull request in Azure DevOps with the preview URL

## Extension task reference

Here, you can find a list of available properties for each of the available tasks in the Vercel Deployment Extension.

### `vercel-deployment-task`

#### Input properties

| Property | Required | Type | Description |
| --- | --- | --- | --- |
| `vercelProjectId` | No | string | The [ID of your Vercel Project](#value-of-vercelprojectid). It can alternatively be set as the |
| `vercelOrgId` | No | string | The [ID of your Vercel Org](#value-of-vercelorgid). It can alternatively be set as the environ |
| `vercelToken` | No | string | A [Vercel personal access](/docs/rest-api#creating-an-access-token) token with deploy permissi |
| `vercelCWD` | No | string | The working directory where the Vercel deployment task will run. When omitted, the task will r |
| `production` | No | boolean | Boolean value specifying if the task should create a production deployment. When omitted or se |
| `debug` | No | boolean | Boolean value that enables the `--debug` option for the internal Vercel CLI operations |

#### Output variables

| Variable | Type | Description |
| --- | --- | --- |
| `deploymentTaskMessage` | string | The message output taken from the deployment. It can be used in tasks that follow |

### `vercel-azdo-pr-comment-task`

#### Input properties

| Property | Required | Type | Description |
| --- | --- | --- | --- |
| `azureToken` | Yes | string | An [Azure personal access token](#create-an-azure-devops-personal-access-token) with the |
| `deploymentTaskMessage` | Yes | string | The message that will added as a comment on the pull request. It is normally created by t |

---
title: "Deploying Bitbucket Projects with Vercel"
description: "Vercel for Bitbucket automatically deploys your Bitbucket projects with Vercel, providing Preview Deployment URLs, and auto
last_updated: "2026-01-16T02:19:31.974Z"
source: "https://vercel.com/docs/git/vercel-for-bitbucket"
---

# Deploying Bitbucket Projects with Vercel

Vercel for Bitbucket automatically deploys your Bitbucket projects with [Vercel](/), providing [Preview Deployment URLs](/docs/deployment

## Supported Bitbucket Products

- [Bitbucket Free](https://www.atlassian.com/software/bitbucket/pricing)
- [Bitbucket Standard](https://www.atlassian.com/software/bitbucket/pricing)
- [Bitbucket Premium](https://www.atlassian.com/software/bitbucket/pricing)
- [Bitbucket Data Center (Self-Hosted)](#using-bitbucket-pipelines)

## Deploying a Bitbucket Repository

The [Deploying a Git repository](/docs/git#deploying-a-git-repository) guide outlines how to create a new Vercel Project from a Bitbucket

## Changing the Bitbucket Repository of a Project

If you'd like to connect your Vercel Project to a different Bitbucket repository or disconnect it, you can do so from the [Git section](/

### A Deployment for Each Push

Vercel for Bitbucket will **deploy each push by default**. This
includes pushes and pull requests made to branches. This allows those working
within the project to preview the changes made before they are pushed to
production.

With each new push, if Vercel is already building a previous commit on the same branch, the current build will complete and any commit pu

### Updating the Production Domain

If [Custom Domains](/docs/projects/custom-domains) are set from a project domains dashboard, pushes and merges to the [Production Branch]

If you decide to revert a commit that has already been deployed to production, the previous [Production Deployment](/docs/deployments/env

### Preview URLs for Each Pull Request

The latest push to any [pull request](https://www.atlassian.com/git/tutorials/making-a-pull-request) will automatically be made available

### System environment variables

You may want to use different workflows and APIs based on Git information. To support this, the following [System Environment Variables](

We require some permissions through our Vercel for Bitbucket integration. Below are listed the permissions required and a description for

### Repository Permissions

Repository permissions allow us to interact with repositories belonging to or associated with (if permitted) the connected account.

| Permission | Read | Write | Description |
| --- | --- | --- | --- |
| `Web Hooks` | Y | N | Allows us to react to various Bitbucket events. |
| `Issues` | Y | Y | Allows us to interact with Pull Requests as with the `Pull Requests` permissions due to Bitbucket requ |
| `Repository` | N | N | Allows us to access admin features of a Bitbucket repository. |
| `Pull requests` | Y | Y | Allows us create deployments for each Pull Request (PR) and comment on those PR's with status updates. |

#### Organization Permissions

Organization permissions allow us to offer an enhanced experience through information about the connected organization.

| Permission | Read | Write | Description |
| --- | --- | --- | --- |
| `Team` | Y | N | Allows us to offer a better team onboarding experience. |

#### User Permissions

User permissions allow us to offer an enhanced experience through information about the connected user.

| Permission | Read | Write | Description |
| --- | --- | --- | --- |
| `Account` | Y | N | Allows us to associate an email with a Bitbucket account. |

> **💡 Note:** We use the permissions above in order to provide you with the best possible
> deployment experience. If you have any questions or concerns about any of the
> permission scopes, please [contact Vercel Support](/help#issues).

To sign up on Vercel with a different Bitbucket account, sign out of your current Bitbucket account. Then, restart the Vercel [signup pro

## Missing Git repository

When importing or connecting a Bitbucket repository, we require that you have  access to the corresponding repository, so that we can con

If a repository is missing when you try to import or connect it, make sure that you have [Admin access configured for the repository](htt

## Silence comments

By default, comments from the Vercel bot will appear on your pull requests and commits. You can silence these comments in your project's

1. From the Vercel [dashboard](/dashboard), select your project
2. From the **Settings** tab, select **Git**
3. Under **Connected Git Repository**, toggle the switches to your preference

> **💡 Note:** It is currently not possible to prevent comments for specific branches.

## Using Bitbucket Pipelines

You can use Bitbucket Pipelines to build and deploy your Vercel Application.

`vercel build` allows you to build your project inside Bitbucket Pipelines, without exposing your source code to Vercel. Then, `vercel de

[Learn more about how to configure Bitbucket Pipelines and Vercel](/kb/guide/how-can-i-use-bitbucket-pipelines-with-vercel) for custom CI

---
title: "Deploying GitHub Projects with Vercel"
description: "Vercel for GitHub automatically deploys your GitHub projects with Vercel, providing Preview Deployment URLs, and automatic
last_updated: "2026-01-16T02:19:31.993Z"
source: "https://vercel.com/docs/git/vercel-for-github"
---

# Deploying GitHub Projects with Vercel

Vercel for GitHub automatically deploys your GitHub projects with [Vercel](/), providing [Preview Deployment URLs](/docs/deployments/envi

## Supported GitHub Products

- [GitHub Free](https://github.com/pricing)
- [GitHub Team](https://github.com/pricing)
- [GitHub Enterprise Cloud](https://docs.github.com/en/get-started/learning-about-github/githubs-products#github-enterprise)
- [GitHub Enterprise Server](#using-github-actions) (When used with GitHub Actions)

> **💡 Note:** When using [Data Residency with a unique subdomain](https://docs.github.com/en/get-started/learning-about-github/githubs-p

## Deploying a GitHub Repository

The [Deploying a Git repository](/docs/git#deploying-a-git-repository) guide outlines how to create a new Vercel Project from a GitHub re

## Changing the GitHub Repository of a Project

If you'd like to connect your Vercel Project to a different GitHub repository or disconnect it, you can do so from the [Git section](/doc

### A Deployment for Each Push

Vercel for GitHub will **deploy every push by default**. This includes
pushes and pull requests made to branches. This allows those working within the
repository to preview changes made before they are pushed to production.

With each new push, if Vercel is already building a previous commit on the same branch, the current build will complete and any commit pu

You can disable this feature for GitHub by configuring the [github.autoJobCancellation](/docs/project-configuration/git-configuration#git

### Updating the Production Domain

If [Custom Domains](/docs/projects/custom-domains) are set from a project domains dashboard, pushes and merges to the [Production Branch]

If you decide to revert a commit that has already been deployed to production, the previous [Production Deployment](/docs/deployments/env

### Preview URLs for the Latest Changes for Each Pull Request

The latest push to any pull request will automatically be made available at a unique [preview URL](/docs/deployments/environments#preview

### Deployment Authorizations for Forks

If you receive a pull request from a fork of your repository, Vercel will require authorization from you or a [team member](/docs/rbac/ma

This behavior protects you from leaking sensitive project information such as environment variables and the [OIDC Token](/docs/oidc).

You can disable [Git Fork Protection](/docs/projects/overview#git-fork-protection) in the Security section of your Project Settings.

Vercel for GitHub uses the deployment API to bring you an extended user interface both in GitHub, when showing deployments, and Slack, if

You will see all of your deployments, production or preview, from within GitHub on its own page.

Due to using GitHub's Deployments API, you will also be able to integrate with other services through [GitHub's checks](https://help.gith

### Configuring for GitHub

To configure the Vercel for GitHub integration, see [the configuration reference for Git](/docs/project-configuration/git-configuration).

### System environment variables

You may want to use different workflows and APIs based on Git information. To support this, the following [System Environment Variables](

We require some permissions through our Vercel for GitHub integration. Below are listed the permissions required and a description for wh

### Repository Permissions

Repository permissions allow us to interact with repositories belonging to or associated with (if permitted) the connected account.

| Permission | Read | Write | Description |
| --- | --- | --- | --- |
| `Administration` | Y | Y | Allows us to create repositories on the user's behalf. |
| `Checks` | Y | Y | Allows us to add checks against source code on push. |
| `Contents` | Y | Y | Allows us to fetch and write source code for new project templates for the connected user or organiz |
| `Deployments` | Y | Y | Allows us to synchronize deployment status between GitHub and the Vercel infrastructure. |
| `Pull Requests` | Y | Y | Allows us create deployments for each Pull Request (PR) and comment on those PR's with status update |
| `Issues` | Y | Y | Allows us to interact with Pull Requests as with the `Pull Requests` permissions due to GitHub requi |
| `Metadata` | Y | N | Allows us to read basic repository metadata to provide a detailed dashboard. |
| `Web Hooks` | Y | Y | Allows us to react to various GitHub events. |
| `Commit Statuses` | Y | Y | Allows us to synchronize commit status between GitHub and Vercel. |

### Organization Permissions

Organization permissions allow us to offer an enhanced experience through information about the connected organization.

| Permission | Read | Write | Description |
| --- | --- | --- | --- |
| `Members` | Y | N | Allows us to offer a better team onboarding experience. |

### User Permissions

User permissions allow us to offer an enhanced experience through information about the connected user.

| Permission | Read | Write | Description |
| --- | --- | --- | --- |
| `Email addresses` | Y | N | Allows us to associate an email with a GitHub account. |

> **💡 Note:** We use the permissions above in order to provide you with the best possible
> deployment experience. If you have any questions or concerns about any of the
> permission scopes, please [contact Vercel Support](/help#issues).

To sign up on Vercel with a different GitHub account, sign out of your current GitHub account.

Then, restart the Vercel [signup process](/signup).

## Missing Git repository

When importing or connecting a GitHub repository, we require that you have  access to the corresponding repository, so that we can config

If a repository is missing when you try to import or connect it, make sure that you have  access configured for the repository. For an or

## Silence GitHub comments

By default, comments from the Vercel GitHub bot will appear on your pull requests and commits. You can silence these comments in your pro

1. From the Vercel [dashboard](/dashboard), select your project
2. From the **Settings** tab, select **Git**
3. Under **Connected Git Repository**, toggle the switches to your preference

If you had previously used the, now deprecated, [`github.silent`](/docs/project-configuration/git-configuration#github.silent) property i

> **💡 Note:** It is currently not possible to prevent comments for specific branches.

## Silence deployment notifications on pull requests

By default, Vercel notifies GitHub of deployments using [the `deployment_status` webhook event](https://docs.github.com/en/actions/writing

Because Vercel also adds a comment to the pull request with a link to the deployment, unwanted noise can accumulate from the list of depl

You can disable `deployment_status` events by:

- [Going to the Git settings for your project](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Fsettings%2Fgit\&title=Project+Git+
- Disabling the `deployment_status` Events toggle

> **⚠️ Warning:** Before doing this, ensure that you aren't depending on `deployment_status`
> events in your GitHub Actions workflows. If you are, we encourage [migrating
> to `repository_dispatch` events](#migrating-from-deployment_status).

## Using GitHub Actions

You can use GitHub Actions to build and deploy your Vercel Application. This approach is necessary to enable Vercel with GitHub Enterpris

1. Create a GitHub Action to build your project and deploy it to Vercel. Make sure to install the Vercel CLI (`npm install --global verce
2. Use `vercel build` to build your project inside GitHub Actions, without exposing your source code to Vercel
3. Then use `vercel deploy --prebuilt` to skip the build step on Vercel and upload the previously generated `.vercel/output` folder from

You'll need to make GitHub Actions for preview (non-`main` pushes) and production (`main` pushes) deployments. [Learn more about how to c

### Repository dispatch events

> **💡 Note:** This event will only trigger a workflow run if the workflow file exists on the
> default branch (e.g. `main`). If you'd like to test the workflow prior to
> merging to `main`, we recommend adding a [`workflow_dispatch`
> trigger](https://docs.github.com/en/actions/writing-workflows/choosing-when-your-workflow-runs/events-that-trigger-workflows#workflow_d

Vercel sends [`repository_dispatch` events](https://docs.github.com/en/actions/writing-workflows/choosing-when-your-workflow-runs/events-

GitHub Actions can trigger on the following events:

```yaml
on:
  repository_dispatch:
    - 'vercel.deployment.ready'
    - 'vercel.deployment.success'
    - 'vercel.deployment.error'
```

```
    - 'vercel.deployment.canceled'
    # canceled as a result of the ignored build script
    - 'vercel.deployment.ignored'
    # canceled as a result of automatic deployment skipping https://vercel.com/docs/monorepos#skipping-unaffected-projects
    - 'vercel.deployment.skipped'
    - 'vercel.deployment.pending'
    - 'vercel.deployment.failed'
    - 'vercel.deployment.promoted'
```

`repository_dispatch` events contain a JSON payload with information about the deployment, such as deployment `url` and deployment `envir

Read more and see the [full schema](https://github.com/vercel/repository-dispatch/blob/main/packages/repository-dispatch/src/types.ts) in

#### Migrating from `deployment_status`

With `repository_dispatch`, the dispatch event `client_payload` contains details about your deployment allowing you to reduce GitHub Acti

For example, to migrate the GitHub Actions trigger for preview deployments for end-to-end tests:

Previously, we needed to check if the status of a deployment was successful. Now, with `repository_dispatch` we can trigger our workflow

Since we're no longer using the `deployment_status` event, we need to get the `url` from the `vercel.deployment.success` event's `client_

```diff
name: End to End Tests

on:
- deployment_status:
+ repository_dispatch:
+   types:
+     - 'vercel.deployment.success'
jobs:
  run-e2es:
-   if: github.event_name == 'deployment_status' && github.event.deployment_status.state == 'success'
+   if: github.event_name == 'repository_dispatch'
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - name: Install dependencies
        run: npm ci && npx playwright install --with-deps
      - name: Run tests
        run: npx playwright test
        env:
-         BASE_URL: ${{ github.event.deployment_status.environment_url }}
+         BASE_URL: ${{ github.event.client_payload.url }}
```

---
title: "Deploying GitLab Projects with Vercel"
description: "Vercel for GitLab automatically deploys your GitLab projects with Vercel, providing Preview Deployment URLs, and automatic
last_updated: "2026-01-16T02:19:32.004Z"
source: "https://vercel.com/docs/git/vercel-for-gitlab"
---

# Deploying GitLab Projects with Vercel

Vercel for GitLab automatically deploys your GitLab projects with [Vercel](/), providing [Preview Deployment URLs](/docs/deployments/envi

## Supported GitLab Products

- [GitLab Free](https://about.gitlab.com/pricing/)
- [GitLab Premium](https://about.gitlab.com/pricing/)
- [GitLab Ultimate](https://about.gitlab.com/pricing/)
- [GitLab Enterprise](https://about.gitlab.com/enterprise/)
- [Self-Managed GitLab](#using-gitlab-pipelines)

## Deploying a GitLab Repository

The [Deploying a Git repository](/docs/git#deploying-a-git-repository) guide outlines how to create a new Vercel Project from a GitLab re

## Changing the GitLab Repository of a Project

If you'd like to connect your Vercel Project to a different GitLab repository or disconnect it, you can do so from the [Git section](/doc

### A Deployment for Each Push

Vercel for GitLab will **deploy each push by default**. This includes
pushes and pull requests made to branches. This allows those working within the
project to preview the changes made before they are pushed to production.

With each new push, if Vercel is already building a previous commit on the same branch, the current build will complete and any commit pu

### Updating the Production Domain

If [Custom Domains](/docs/projects/custom-domains) are set from a project domains dashboard, pushes and merges to the [Production Branch]

If you decide to revert a commit that has already been deployed to production, the previous [Production Deployment](/docs/deployments/env

### Preview URLs for Each Merge Request

The latest push to any [merge request](https://docs.gitlab.com/ee/user/project/merge_requests/) will automatically be made available at a

### System environment variables

You may want to use different workflows and APIs based on Git information. To support this, the following [System Environment Variables](

We require some permissions through our Vercel for GitLab integration. Below are listed the permissions required and a description for wh

| Permission | Read | Write | Description
| ---------- | ---- | ----- | ----------------------------------------------------------------------------------------------------------------
| `API`      | Y    | Y     | Allows us access to the API—including all groups and projects, the container registry, and the package regi

> **💡 Note:** We use the permissions above in order to provide you with the best possible
> deployment experience. If you have any questions or concerns about any of the
> permission scopes, please [contact Vercel Support](/help#issues).

To sign up on Vercel with a different GitLab account, sign out of your current GitLab account.

Then, restart the Vercel [signup process](/signup).

## Missing Git repository

When importing or connecting a GitLab repository, we require that you have **Maintainer** access to the corresponding repository, so that

If a repository is missing when you try to import or connect it, make sure that you have [Maintainer access configured for the repository

## Silence comments

By default, comments from the Vercel bot will appear on your pull requests and commits. You can silence these comments in your project's

1. From the Vercel [dashboard](/dashboard), select your project
2. From the **Settings** tab, select **Git**
3. Under **Connected Git Repository**, toggle the switches to your preference

> **💡 Note:** It is currently not possible to prevent comments for specific branches.

## Using GitLab Pipelines

You can use GitLab Pipelines to build and deploy your Vercel Application.

`vercel build` allows you to build your project inside GitLab Pipelines, without exposing your source code to Vercel. Then, `vercel deplo

[Learn more about how to configure GitLab Pipelines and Vercel](/kb/guide/how-can-i-use-gitlab-pipelines-with-vercel) for custom CI/CD wo

> **💡 Note:** In some cases, your GitLab merge pipeline can fail while your branch pipeline
> succeeds, allowing your merge requests to [merge with failing
> tests](https://gitlab.com/gitlab-org/gitlab/-/issues/384927#top). This is a
> GitLab issue. To avoid it, we recommend using [Vercel
> CLI](/docs/cli/deploying-from-cli) to deploy your projects.


---
title: "Glossary"
description: "Learn about the terms and concepts used in Vercel"
last_updated: "2026-01-16T02:19:32.057Z"
source: "https://vercel.com/docs/glossary"
---

# Glossary

A full glossary of terms used in Vercel's products and documentation.

## A

### Active CPU

A pricing model for [Fluid Compute](/docs/fluid-compute) where you only pay for the actual CPU time your functions use while executing, r

### AI Gateway

A proxy service from Vercel that routes model requests to various AI providers, offering a unified API, budget management, usage monitori

### AI SDK

A TypeScript toolkit designed to help developers build AI-powered applications with React, Next.js, Vue, Svelte, and Node.js by providing

### Analytics

See [Web Analytics](#web-analytics).

### Anycast Network

A network topology that shares an IP address among multiple nodes, routing requests to the nearest available node based on network condit

## B

### Build

The process that Vercel performs every time you deploy your code, compiling, bundling, and optimizing your application so it's ready to s

### Build Cache

A cache that stores build artifacts and dependencies to speed up subsequent deployments. Each build cache can be up to 1 GB and is retain

### Build Command

The command used to build your project during deployment. Vercel automatically configures this based on your framework, but it can be ove

### Build Output API

A file-system-based specification for a directory structure that can produce a Vercel deployment, primarily targeted at framework authors

### Bot Protection

Security features that help identify and block malicious bots and crawlers from accessing your applications.

## C

### CDN (Content Delivery Network)

A distributed network of servers that stores static content in multiple locations around the globe to serve content from the closest serv

### CI/CD (Continuous Integration/Continuous Deployment)

Development practices where code changes are automatically built, tested, and deployed. Vercel provides built-in CI/CD through Git integr

### CLI (Command Line Interface)

The Vercel CLI is a command-line tool that allows you to deploy projects, manage deployments, and configure Vercel from your terminal.

### Compute

The processing power and execution environment where your application code runs. Vercel offers serverless compute through Functions and E

### Concurrency

The ability to handle multiple requests simultaneously. Vercel Functions support concurrency scaling and [Fluid Compute](/docs/fluid-comp

### Core Web Vitals

Key metrics defined by Google that assess your web application's loading speed, responsiveness, and visual stability, including LCP, FID,

### Cron Jobs

Scheduled tasks that run at specified intervals. Vercel supports cron jobs for automating recurring processes.

### Custom Domain

A domain that you own and configure to point to your Vercel deployment, replacing the default `.vercel.app` domain.

## D

### Data Cache

A specialized cache that stores responses from data fetches in frameworks like Next.js, allowing for granular caching per fetch rather th

### DDoS (Distributed Denial of Service)

A type of cyber attack where multiple systems flood a target with traffic. Vercel provides built-in DDoS protection and mitigation.

### Deploy Hooks

URLs that accept HTTP POST requests to trigger deployments without requiring a new Git commit.

### Deployment

The result of a successful build of your project on Vercel. Each deployment generates a unique URL and represents a specific version of y

### Deployment Protection

Security features that restrict access to your deployments using methods like Vercel Authentication, Password Protection, or Trusted IPs.

### Directory

A file system structure used to organize and store files, also known as a folder. Often abbreviated as "dir" in programming contexts.

## E

### Edge

The edge refers to servers closest to users in a distributed network. Vercel's CDN runs code and serves content from edge locations globa

### Edge Config

A global data store that enables ultra-fast data reads in the region closest to the user (typically under 1ms) for configuration data lik

### CDN (Content Delivery Network)

Vercel's global infrastructure consisting of Points of Presence (PoPs) and compute-capable regions that serve content and run code close

### Edge Runtime

A minimal JavaScript runtime that exposes Web Standard APIs, used for Vercel Functions and Routing Middleware.

### Environment

A context for running your application, such as Local Development, Preview, or Production. Each environment can have its own configuratio

### Environment Variables

Configuration values that can be accessed by your application at build time or runtime, used for API keys, database connections, and othe

## F

### Fast Data Transfer

Data transfer between the Vercel CDN and user devices, optimized for performance and charged based on usage.

### Feature Flags

Configuration switches that allow you to enable or disable features in your application without deploying new code, often stored in Edge

### Firewall

See [Vercel Firewall](#vercel-firewall).

### Fluid Compute

An enhanced execution model for Vercel Functions that provides in-function concurrency, and a new pricing model where you only pay for th

### Framework

A software library that provides a foundation for building applications. Vercel supports over 30 frameworks including Next.js, React, Vue

### Framework Preset

A configuration setting that tells Vercel which framework your project uses, enabling automatic optimization and build configuration.

### Functions

See [Vercel Functions](#vercel-functions).

## G

### Git Integration

Automatic connection between your Git repository (GitHub, GitLab, Bitbucket, Azure DevOps) and Vercel for continuous deployment.

## H

### Headers

HTTP headers that can be configured to modify request and response behavior, improving security, performance, and functionality.

### HTTPS/SSL

Secure HTTP protocol that encrypts communication between clients and servers. All Vercel deployments automatically use HTTPS with SSL cer

## I

### I/O-bound

Processes limited by input/output operations rather than CPU speed, such as database queries or API requests. Optimized through concurren

### Image Optimization

Automatic optimization of images including format conversion, resizing, and compression to improve performance and reduce bandwidth.

### Incremental Static Regeneration (ISR)

A feature that allows you to update static content without redeployment by rebuilding pages in the background on a specified interval.

### Install Command

The command used to install dependencies before building your project, such as `npm install` or `pnpm install`.

### Integration

Third-party services and tools that connect with Vercel to extend functionality, available through the Vercel Marketplace.

## J

### JA3/JA4 Fingerprints

TLS fingerprinting techniques used by Vercel's security systems to identify and restrict malicious traffic patterns.

## L

### Drains

A feature that allows you to send observability data (logs, traces, speed insights, and analytics) to external services for long-term ret

## M

### Managed Infrastructure

Vercel's fully managed platform that handles server provisioning, scaling, security, and maintenance automatically.

### MCP (Model Context Protocol)

A protocol for AI applications that enables secure and standardized communication between AI models and external data sources.

### Middleware

Code that executes before a request is processed, running on the global network to modify responses, implement authentication, or perform

### Microfrontends

A development approach that allows you to split a single application into smaller, independently deployable units that render as one cohe

### Monorepo

A version control strategy where multiple packages or modules are stored in a single repository, facilitating code sharing and collaborat

### Multi-repo

A version control strategy where each package or module has its own separate repository, also known as "polyrepo."

### Multi-tenant

Applications that serve multiple customers (tenants) from a single codebase, with each tenant getting their own domain or subdomain.

## N

### Node.js

A JavaScript runtime environment that Vercel supports for Vercel Functions and applications.

## O

### Observability

Tools and features that help you monitor, analyze, and understand your application's performance, traffic, and behavior in production.

### OIDC (OpenID Connect)

A federation protocol that issues short-lived, non-persistent tokens for secure backend access without storing long-lived credentials.

### Origin Server

The server that stores and runs the original version of your application code, where requests are processed when not served from cache.

### Output Directory

The folder containing your final build output after the build process completes, such as `dist`, `build`, or `.next`.

## P

### Package

A collection of files and directories grouped together for a common purpose, such as libraries, applications, or development tools.

### Password Protection

A deployment protection method that restricts access to deployments using a password, available on Enterprise plans or as a Pro add-on.

### Points of Presence (PoPs)

Distributed servers in Vercel's CDN that provide the first point of contact for requests, handling routing, DDoS protection, and SSL term

### Preview Deployment

A deployment created from non-production branches that allows you to test changes in a live environment before merging to production.

### Production Deployment

The live version of your application that serves end users, typically deployed from your main branch.

### Project

An application that you have deployed to Vercel, which can have multiple deployments and is connected to a Git repository.

## R

### Real Experience Score (RES)

A performance metric in Speed Insights that uses real user data to measure your application's actual performance in production.

### Redirects

HTTP responses that tell clients to make a new request to a different URL, useful for enforcing HTTPS or directing traffic.

### Region

Geographic locations where Vercel can run your functions and store data. Vercel has 19 compute-capable regions globally.

### Repository

A location where files and source code are stored and managed in version control systems like Git, maintaining history of all changes.

### Rewrites

URL transformations that change what the server fetches internally without changing the URL visible to the client.

### Runtime

The execution environment for your functions, such as Node.js, Edge Runtime, Python, or other supported runtimes.

### Runtime Logs

Logs generated by your functions during execution, useful for debugging and monitoring application behavior.

## S

### SAML SSO (Single Sign-On)

An authentication protocol that allows teams to log into Vercel using their organization's identity provider.

### Sandbox

See [Vercel Sandbox](#vercel-sandbox).

### Secure Compute

An Enterprise feature that creates private connections between Vercel Functions and backend infrastructure using dedicated IP addresses.

### Serverless

A cloud computing model where code runs without managing servers, automatically scaling based on demand and charging only for actual usag

### Speed Insights

Performance monitoring that provides detailed insights into your website's Core Web Vitals and loading performance metrics.

### Storage

Vercel's suite of storage products including Blob storage for files and Edge Config for configuration data.

### Streaming

A technique for sending data progressively from functions to improve perceived performance and responsiveness.

## T

### Trusted IPs

A deployment protection method that restricts access to deployments based on IP address allowlists, available on Enterprise plans.

### Turborepo

A high-performance build system for monorepos that provides fast incremental builds and remote caching capabilities.

## V

### v0

An AI-powered tool that converts natural language descriptions into React code and UI components, integrated with Vercel for deployment.

### Vercel Authentication

A deployment protection method that restricts access to team members and authorized users with Vercel accounts.

### Vercel Blob

Scalable object storage service for static assets like images, videos, and files, optimized for global content delivery.

### Vercel Firewall

A multi-layered security system that protects applications from threats, including platform-wide DDoS protection and customizable WAF rul

### Vercel Functions

Serverless compute that allows you to run server-side code without managing servers, automatically scaling based on demand.

### Vercel Sandbox

An ephemeral compute primitive for safely running untrusted or user-generated code in isolated Linux VMs.

### Virtual Experience Score (VES)

A predictive performance metric that anticipates the impact of changes on application performance before deployment.

## W

### WAF (Web Application Firewall)

A customizable security layer that allows you to define rules to protect against attacks, scrapers, and unwanted traffic.

### Web Analytics

Privacy-friendly analytics that provide insights into website visitors, page views, and user behavior without using cookies.

### Workspace

In JavaScript, an entity in a repository that can be either a single package or a collection of packages, often at the repository root.

---
title: "Cache-Control headers"
description: "Learn about the cache-control headers sent to each Vercel deployment and how to use them to control the caching behavior of
last_updated: "2026-01-16T02:19:32.025Z"
source: "https://vercel.com/docs/headers/cache-control-headers"
---

# Cache-Control headers

You can control how Vercel's CDN caches your Function responses by setting a [Cache-Control headers](https://developer.mozilla.org/docs/W

## Default `cache-control` value

The default value is `cache-control: public, max-age=0, must-revalidate` which instructs both the CDN and the browser not to cache.

## Recommended settings

We recommend that you set your cache to`max-age=0, s-maxage=86400`, adjusting 86400 to the number of seconds you want the response cached

## `s-maxage`

This directive sets the number of seconds a response is considered "fresh" by the CDN. After this period ends, Vercel's CDN will serve th

`s-maxage` is consumed by Vercel's proxy and not included as part the final HTTP response to the client.

### `s-maxage` example

The following example instructs the CDN to cache the response for 60 seconds. A response can be cached a minimum of `1` second and maximu

```js filename="cache-response"
Cache-Control: s-maxage=60
```

## `stale-while-revalidate`

This `cache-control` directive allows you to serve content from the Vercel CDN cache while simultaneously updating the cache in the backg

- Your content changes frequently, but regeneration is slow, such as content that relies on an expensive database query or upstream API r
- Your content changes infrequently but you want to have the flexibility to update it without waiting for the cache to expire

`stale-while-revalidate` is consumed by Vercel's proxy and not included as part the final HTTP response to the client. This allows you to

### SWR example

The following example instructs the CDN to:

- Serve content from the cache for 1 second
- Return a stale request (if requested after 1 second)
- Update the cache **in the background** asynchronously (if requested after 1 second)

```js filename="swr-on-edge-network"
Cache-Control: s-maxage=1, stale-while-revalidate=59
```

The first request is served synchronously. Subsequent requests are served from the cache and revalidated asynchronously if the cache is "

If you need to do a *synchronous* revalidation you can set the `pragma: no-cache` header along with the `cache-control` header. This can

> **💡 Note:** Many browser developer tools set `pragma: no-cache` by default, which reveals
> the true load time of the page with the synchronous update to the cache.

## `stale-if-error`

This directive is currently not supported. `stale-if-error` is consumed by Vercel's proxy, and will not be included in the HTTP response

## `proxy-revalidate`

This directive is currently not supported.

## Using `private`

Using the `private` directive specifies that the response can only be cached by the client and **not by Vercel's CDN**. Use this directiv

## `Pragma: no-cache`

When Vercel's CDN receives a request with `Pragma: no-cache` (such as when the browser devtools are open), it will revalidate any stale r

## CDN-Cache-Control Header

Sometimes the directives you set in a `Cache-Control` header can be interpreted differently by the different CDNs and proxies your conten

The `CDN-Cache-Control` and `Vercel-CDN-Cache-Control` headers are response headers that can be used to specify caching behavior on the C

You can use the same directives as [`Cache-Control`](#default-cache-control-value), but `CDN-Cache-Control` is only used by the CDN.

## Behavior

Origins can set the following headers:

- `Vercel-CDN-Cache-Control`
- `CDN-Cache-Control`
- `Cache-Control`

When multiple of the above headers are set, Vercel's CDN will use the following priority to determine the caching behavior:

### `Vercel-CDN-Cache-Control`

`Vercel-CDN-Cache-Control` is exclusive to Vercel and has top priority, whether it's defined in a Vercel Function response or a `vercel.j

### `CDN-Cache-Control`

`CDN-Cache-Control` is second in priority after `Vercel-CDN-Cache-Control`, and **always** overrides `Cache-Control` headers, whether def

By default, `CDN-Cache-Control` configures Vercel's Cache and is used by other CDNs, allowing you to configure intermediary caches. If `V

### `Cache-Control`

`Cache-Control` is a web standard header and last in priority. If neither `CDN-Cache-Control` nor `Vercel-CDN-Cache-Control` are set, thi

You can still set `Cache-Control` while using the other two, and it will be forwarded to the client as is.

> **💡 Note:** If only `Cache-Control` is used, Vercel strips the `s-maxage` directive from
> the header before it's sent to the client.

## Cache-Control comparison tables

The following tables demonstrate how Vercel's Cache behaves in different scenarios:

### Functions have priority over config files

`Cache-Control` headers returned from Vercel Functions take priority over `Cache-Control` headers from `next.config.js` or `vercel.json`

| Parameter | Value |
| --- | --- |
| Vercel Function response headers | `Cache-Control: s-maxage=60` |
| `vercel.json` or `next.config.js` headers | `Cache-Control: s-maxage: 120` |
| Cache behavior | 60s TTL |
| Headers sent to the client | `Cache-Control: public, max-age: 0` |

### `CDN-Cache-Control` priority

`CDN-Cache-Control` has priority over `Cache-Control`, even if defined in `vercel.json` or `next.config.js`.

| Parameter | Value |
| --- | --- |
| Vercel Function response headers | `Cache-Control: s-maxage=60` |
| `vercel.json` or `next.config.js` headers | `CDN-Cache-Control: max-age=120` |
| Cache behavior | 120s TTL |
| Headers sent to the client | `Cache-Control: s-maxage=60 CDN-Cache-Control: max-age=120` |

### `Vercel-CDN-Cache-Control` priority

`Vercel-CDN-Cache-Control` has priority over both `CDN-Cache-Control` and `Cache-Control`. It only applies to Vercel, so it is not return

| Parameter | Value |
| --- | --- |
| Vercel Function response headers | `CDN-Cache-Control: max-age=120` |
| `vercel.json` or `next.config.js` headers | `Cache-Control: s-maxage=60 Vercel-CDN-Cache-Control: max-age=300` |
| Cache behavior | 300s TTL |
| Headers sent to the client | `Cache-Control: s-maxage=60 CDN-Cache-Control: max-age=120` |

## Which Cache-Control headers to use with CDNs

- If you want to control caching similarly on Vercel, CDNs, and the client, use `Cache-Control`
- If you want to control caching on Vercel and also on other CDNs, use `CDN-Cache-Control`
- If you want to control caching only on Vercel, use `Vercel-CDN-Cache-Control`
- If you want to specify different caching behaviors for Vercel, other CDNs, and the client, you can set all three headers

## Example usage

The following example demonstrates `Cache-Control` headers that instruct:

- Vercel's Cache to have a [TTL](https://en.wikipedia.org/wiki/Time_to_live "TTL – Time To Live") of `3600` seconds
- Downstream CDNs to have a TTL of `60` seconds
- Clients to have a TTL of `10` seconds

```js filename="app/api/cache-control-headers/route.js" framework=nextjs
export async function GET() {
  return new Response('Cache Control example', {
    status: 200,
    headers: {
      'Cache-Control': 'max-age=10',
      'CDN-Cache-Control': 'max-age=60',
      'Vercel-CDN-Cache-Control': 'max-age=3600',
    },
  });
}
```

```ts filename="app/api/cache-control-headers/route.ts" framework=nextjs
export async function GET() {
  return new Response('Cache Control example', {
    status: 200,
    headers: {
      'Cache-Control': 'max-age=10',
      'CDN-Cache-Control': 'max-age=60',
      'Vercel-CDN-Cache-Control': 'max-age=3600',
    },
  });
}
```

```js filename="app/api/cache-control-headers/route.js" framework=nextjs-app
export async function GET() {
  return new Response('Cache Control example', {
    status: 200,
    headers: {
      'Cache-Control': 'max-age=10',
      'CDN-Cache-Control': 'max-age=60',
      'Vercel-CDN-Cache-Control': 'max-age=3600',
    },
  });
}
```

```ts filename="app/api/cache-control-headers/route.ts" framework=nextjs-app
export async function GET() {
  return new Response('Cache Control example', {
    status: 200,
    headers: {
      'Cache-Control': 'max-age=10',
      'CDN-Cache-Control': 'max-age=60',
      'Vercel-CDN-Cache-Control': 'max-age=3600',
    },
  });
}
```

```js filename="api/cache-control-headers.js" framework=other
export default function handler(request, response) {
  response.setHeader('Vercel-CDN-Cache-Control', 'max-age=3600');
  response.setHeader('CDN-Cache-Control', 'max-age=60');
  response.setHeader('Cache-Control', 'max-age=10');

  return response.status(200).json({ name: 'John Doe' });
}
```

```ts filename="api/cache-control-headers.ts" framework=other
import type { VercelResponse } from '@vercel/node';

export default function handler(response: VercelResponse) {
  response.setHeader('Vercel-CDN-Cache-Control', 'max-age=3600');
  response.setHeader('CDN-Cache-Control', 'max-age=60');
  response.setHeader('Cache-Control', 'max-age=10');

  return response.status(200).json({ name: 'John Doe' });
}
```

## Custom Response Headers

Using configuration, you can assign custom headers to each response.

Custom headers can be configured with the `headers` property in [`next.config.js`](https://nextjs.org/docs/api-reference/next.config.js/h

Alternatively, a [Vercel Function](/docs/functions) can assign headers to the [Response](https://nodejs.org/api/http.html#http_response_s

> **💡 Note:** Response headers `x-matched-path`, `server`, and `content-length` are reserved
> and cannot be modified.

---
title: "Headers"
description: "This reference covers the list of request, response, cache-control, and custom response headers included with deployments w
last_updated: "2026-01-16T02:19:32.008Z"
source: "https://vercel.com/docs/headers"
---

# Headers

Headers are small pieces of information that are sent between the client (usually a web browser) and the server. They contain metadata ab

## Using headers

By using headers effectively, you can optimize the performance and security of your application on Vercel's global network. Here are some

1. [Use caching headers](#cache-control-header): Caching headers instruct the client and server to cache resources like images, CSS files
2. [Use compression headers](/docs/compression#compression-with-vercel-cdn): Use the `Accept-Encoding` header to tell the client and serv
3. Use custom headers: You can also use custom headers in your `vercel.json` file to add metadata specific to your application. For examp

## Request headers

To learn about the request headers sent to each Vercel deployment and how to use them to process requests before sending a response, see

## Response headers

To learn about the response headers included in Vercel deployment responses and how to use them to process responses before sending a res

## Cache-Control header

To learn about the cache-control headers sent to each Vercel deployment and how to use them to control the caching behavior of your appli

## More resources

- [Set Caching Header](/kb/guide/set-cache-control-headers)

---
title: "Request headers"
description: "Learn about the request headers sent to each Vercel deployment and how to use them to process requests before sending a res
last_updated: "2026-01-16T02:19:32.244Z"
source: "https://vercel.com/docs/headers/request-headers"
---

# Request headers

The following headers are sent to each Vercel deployment and can be used to process the request before sending back a response. These hea

## `host`

This header represents the domain name as it was accessed by the client. If the deployment has been assigned to a preview URL or productio

```ts filename="app/api/header/route.ts" framework=nextjs
export function GET(request: Request) {
  const host = request.headers.get('host');
  return new Response(`Host: ${host}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs
export function GET(request) {
  const host = request.headers.get('host');
  return new Response(`Host: ${host}`);
}
```

```ts filename="api/header.ts" framework=other
export function GET(request: Request) {
  const host = request.headers.get('host');
  return new Response(`Host: ${host}`);
}
```

```js filename="api/header.js" framework=other
export function GET(request) {
  const host = request.headers.get('host');
  return new Response(`Host: ${host}`);
}
```

```ts filename="app/api/header/route.ts" framework=nextjs-app
export function GET(request: Request) {
  const host = request.headers.get('host');
  return new Response(`Host: ${host}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs-app
```

```
export function GET(request) {
  const host = request.headers.get('host');
  return new Response(`Host: ${host}`);
}
```

## `x-vercel-id`

This header contains a list of [Vercel regions](/docs/regions) your request hit, as well as the region the function was executed in (for

It also allows Vercel to automatically prevent infinite loops.

```ts filename="app/api/header/route.ts" framework=nextjs
export function GET(request: Request) {
  const vercelId = request.headers.get('x-vercel-id');
  return new Response(`Vercel ID: ${vercelId}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs
export function GET(request) {
  const vercelId = request.headers.get('x-vercel-id');
  return new Response(`Vercel ID: ${vercelId}`);
}
```

```ts filename="api/header.ts" framework=other
export function GET(request: Request) {
  const vercelId = request.headers.get('x-vercel-id');
  return new Response(`Vercel ID: ${vercelId}`);
}
```

```js filename="api/header.js" framework=other
export function GET(request) {
  const vercelId = request.headers.get('x-vercel-id');
  return new Response(`Vercel ID: ${vercelId}`);
}
```

```ts filename="app/api/header/route.ts" framework=nextjs-app
export function GET(request: Request) {
  const vercelId = request.headers.get('x-vercel-id');
  return new Response(`Vercel ID: ${vercelId}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs-app
export function GET(request) {
  const vercelId = request.headers.get('x-vercel-id');
  return new Response(`Vercel ID: ${vercelId}`);
}
```

## `x-forwarded-host`

This header is identical to the `host` header.

```ts filename="app/api/header/route.ts" framework=nextjs
export function GET(request: Request) {
  const host = request.headers.get('x-forwarded-host');
  return new Response(`Host: ${host}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs
export function GET(request) {
  const host = request.headers.get('x-forwarded-host');
  return new Response(`Host: ${host}`);
}
```

```ts filename="api/header.ts" framework=other
export function GET(request: Request) {
  const host = request.headers.get('x-forwarded-host');
  return new Response(`Host: ${host}`);
}
```

```js filename="api/header.js" framework=other
export function GET(request) {
  const host = request.headers.get('x-forwarded-host');
  return new Response(`Host: ${host}`);
}
```

```ts filename="app/api/header/route.ts" framework=nextjs-app
export function GET(request: Request) {
  const host = request.headers.get('x-forwarded-host');
  return new Response(`Host: ${host}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs-app
export function GET(request) {
  const host = request.headers.get('x-forwarded-host');
  return new Response(`Host: ${host}`);
}
```

## `x-forwarded-proto`

This header represents the protocol of the forwarded server, typically `https` in production and `http`in development.

```ts filename="app/api/header/route.ts" framework=nextjs
export function GET(request: Request) {
  const protocol = request.headers.get('x-forwarded-proto');
  return new Response(`Protocol: ${protocol}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs
export function GET(request) {
  const protocol = request.headers.get('x-forwarded-proto');
  return new Response(`Protocol: ${protocol}`);
}
```

```ts filename="api/header.ts" framework=other
export function GET(request: Request) {
  const protocol = request.headers.get('x-forwarded-proto');
  return new Response(`Protocol: ${protocol}`);
}
```

```js filename="api/header.js" framework=other
export function GET(request) {
  const protocol = request.headers.get('x-forwarded-proto');
  return new Response(`Protocol: ${protocol}`);
}
```

```ts filename="app/api/header/route.ts" framework=nextjs-app
export function GET(request: Request) {
  const protocol = request.headers.get('x-forwarded-proto');
  return new Response(`Protocol: ${protocol}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs-app
export function GET(request) {
  const protocol = request.headers.get('x-forwarded-proto');
  return new Response(`Protocol: ${protocol}`);
}
```

## `x-forwarded-for`

The public IP address of the client that made the request.

If you are trying to use Vercel behind a proxy, we currently overwrite the [`X-Forwarded-For`](https://developer.mozilla.org/docs/Web/HTT

```ts filename="app/api/header/route.ts" framework=nextjs
export function GET(request: Request) {
  const ip = request.headers.get('x-forwarded-for');
  return new Response(`IP: ${ip}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs
export function GET(request) {
  const ip = request.headers.get('x-forwarded-for');
  return new Response(`IP: ${ip}`);
}
```

```ts filename="api/header.ts" framework=other
export function GET(request: Request) {
  const ip = request.headers.get('x-forwarded-for');
  return new Response(`IP: ${ip}`);
}
```

```js filename="api/header.js" framework=other
export function GET(request) {
  const ip = request.headers.get('x-forwarded-for');
  return new Response(`IP: ${ip}`);
}
```

```ts filename="app/api/header/route.ts" framework=nextjs-app
export function GET(request: Request) {
  const ip = request.headers.get('x-forwarded-for');
  return new Response(`IP: ${ip}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs-app
export function GET(request) {
  const ip = request.headers.get('x-forwarded-for');
  return new Response(`IP: ${ip}`);
}
```

### Custom `X-Forwarded-For` IP

**Enterprise customers** can purchase and enable a trusted proxy to allow your custom `X-Forwarded-For` IP. [Contact us](/contact/sales)

## `x-vercel-forwarded-for`

This header is identical to the `x-forwarded-for` header. However, `x-forwarded-for` could be overwritten if you're using a proxy on top

## `x-real-ip`

This header is identical to the `x-forwarded-for` header.

## `x-vercel-deployment-url`

This header represents the unique deployment, not the preview URL or production domain. For example, `*.vercel.app`.

```ts filename="app/api/header/route.ts" framework=nextjs
export function GET(request: Request) {
  const deploymentUrl = request.headers.get('x-vercel-deployment-url');
  return new Response(`Deployment URL: ${deploymentUrl}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs
export function GET(request) {
  const deploymentUrl = request.headers.get('x-vercel-deployment-url');
  return new Response(`Deployment URL: ${deploymentUrl}`);
}
```

```ts filename="api/header.ts" framework=other
export function GET(request: Request) {
  const deploymentUrl = request.headers.get('x-vercel-deployment-url');
  return new Response(`Deployment URL: ${deploymentUrl}`);
}
```

```js filename="api/header.js" framework=other
export function GET(request) {
  const deploymentUrl = request.headers.get('x-vercel-deployment-url');
  return new Response(`Deployment URL: ${deploymentUrl}`);
}
```

```ts filename="app/api/header/route.ts" framework=nextjs-app
export function GET(request: Request) {
  const deploymentUrl = request.headers.get('x-vercel-deployment-url');
  return new Response(`Deployment URL: ${deploymentUrl}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs-app
export function GET(request) {
  const deploymentUrl = request.headers.get('x-vercel-deployment-url');
  return new Response(`Deployment URL: ${deploymentUrl}`);
}
```

## `x-vercel-ip-continent`

A two-character [ISO 3166-1](https://en.wikipedia.org/wiki/ISO_3166-1) code representing the continent associated with the location of th

- `AF` for Africa
- `AN` for Antarctica
- `AS` for Asia
- `EU` for Europe
- `NA` for North America
- `OC` for Oceania
- `SA` for South America

```ts filename="app/api/header/route.ts" framework=nextjs
export function GET(request: Request) {
  const continent = request.headers.get('x-vercel-ip-continent');
  return new Response(`Continent: ${continent}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs
export function GET(request) {
  const continent = request.headers.get('x-vercel-ip-continent');
  return new Response(`Continent: ${continent}`);
}
```

```ts filename="api/header.ts" framework=other
export function GET(request: Request) {
  const continent = request.headers.get('x-vercel-ip-continent');
  return new Response(`Continent: ${continent}`);
}
```

```js filename="api/header.js" framework=other
export function GET(request) {
  const continent = request.headers.get('x-vercel-ip-continent');
  return new Response(`Continent: ${continent}`);
}
```

```ts filename="app/api/header/route.ts" framework=nextjs-app
export function GET(request: Request) {
  const continent = request.headers.get('x-vercel-ip-continent');
  return new Response(`Continent: ${continent}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs-app
```

```ts
export function GET(request) {
  const continent = request.headers.get('x-vercel-ip-continent');
  return new Response(`Continent: ${continent}`);
}
```

## `x-vercel-ip-country`

A two-character [ISO 3166-1](https://en.wikipedia.org/wiki/ISO_3166-1) country code for the country associated with the location of the r

```ts filename="app/api/header/route.ts" framework=nextjs
export function GET(request: Request) {
  const country = request.headers.get('x-vercel-ip-country');
  return new Response(`Country: ${country}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs
export function GET(request) {
  const country = request.headers.get('x-vercel-ip-country');
  return new Response(`Country: ${country}`);
}
```

```ts filename="api/header.ts" framework=other
export function GET(request: Request) {
  const country = request.headers.get('x-vercel-ip-country');
  return new Response(`Country: ${country}`);
}
```

```js filename="api/header.js" framework=other
export function GET(request) {
  const country = request.headers.get('x-vercel-ip-country');
  return new Response(`Country: ${country}`);
}
```

```ts filename="app/api/header/route.ts" framework=nextjs-app
export function GET(request: Request) {
  const country = request.headers.get('x-vercel-ip-country');
  return new Response(`Country: ${country}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs-app
export function GET(request) {
  const country = request.headers.get('x-vercel-ip-country');
  return new Response(`Country: ${country}`);
}
```

## `x-vercel-ip-country-region`

A string of up to three characters containing the region-portion of the [ISO 3166-2](https://en.wikipedia.org/wiki/ISO_3166-2) code for t

```ts filename="app/api/header/route.ts" framework=nextjs
export function GET(request: Request) {
  const region = request.headers.get('x-vercel-ip-country-region');
  return new Response(`Region: ${region}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs
export function GET(request) {
  const region = request.headers.get('x-vercel-ip-country-region');
  return new Response(`Region: ${region}`);
}
```

```ts filename="api/header.ts" framework=other
export function GET(request: Request) {
  const region = request.headers.get('x-vercel-ip-country-region');
  return new Response(`Region: ${region}`);
}
```

```js filename="api/header.js" framework=other
export function GET(request) {
  const region = request.headers.get('x-vercel-ip-country-region');
  return new Response(`Region: ${region}`);
}
```

```ts filename="app/api/header/route.ts" framework=nextjs-app
export function GET(request: Request) {
  const region = request.headers.get('x-vercel-ip-country-region');
  return new Response(`Region: ${region}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs-app
export function GET(request) {
  const region = request.headers.get('x-vercel-ip-country-region');
  return new Response(`Region: ${region}`);
}
```

## `x-vercel-ip-city`

The city name for the location of the requester's public IP address. Non-ASCII characters are encoded according to [RFC3986](https://tool

```ts filename="app/api/header/route.ts" framework=nextjs
export function GET(request: Request) {
  const city = request.headers.get('x-vercel-ip-city');
  return new Response(`City: ${city}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs
export function GET(request) {
  const city = request.headers.get('x-vercel-ip-city');
  return new Response(`City: ${city}`);
}
```

```ts filename="api/header.ts" framework=other
export function GET(request: Request) {
  const city = request.headers.get('x-vercel-ip-city');
  return new Response(`City: ${city}`);
}
```

```js filename="api/header.js" framework=other
export function GET(request) {
  const city = request.headers.get('x-vercel-ip-city');
  return new Response(`City: ${city}`);
}
```

```ts filename="app/api/header/route.ts" framework=nextjs-app
export function GET(request: Request) {
  const city = request.headers.get('x-vercel-ip-city');
  return new Response(`City: ${city}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs-app
export function GET(request) {
  const city = request.headers.get('x-vercel-ip-city');
  return new Response(`City: ${city}`);
}
```

## `x-vercel-ip-latitude`

The latitude for the location of the requester's public IP address. For example, `37.7749`.

```ts filename="app/api/header/route.ts" framework=nextjs
export function GET(request: Request) {
  const latitude = request.headers.get('x-vercel-ip-latitude');
  return new Response(`Latitude: ${latitude}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs
export function GET(request) {
  const latitude = request.headers.get('x-vercel-ip-latitude');
  return new Response(`Latitude: ${latitude}`);
}
```

```ts filename="api/header.ts" framework=other
export function GET(request: Request) {
  const latitude = request.headers.get('x-vercel-ip-latitude');
  return new Response(`Latitude: ${latitude}`);
}
```

```js filename="api/header.js" framework=other
export function GET(request) {
  const latitude = request.headers.get('x-vercel-ip-latitude');
  return new Response(`Latitude: ${latitude}`);
}
```

```ts filename="app/api/header/route.ts" framework=nextjs-app
export function GET(request: Request) {
  const latitude = request.headers.get('x-vercel-ip-latitude');
  return new Response(`Latitude: ${latitude}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs-app
export function GET(request) {
  const latitude = request.headers.get('x-vercel-ip-latitude');
  return new Response(`Latitude: ${latitude}`);
}
```

## `x-vercel-ip-longitude`

The longitude for the location of the requester's public IP address. For example, `-122.4194`.

```ts filename="app/api/header/route.ts" framework=nextjs
export function GET(request: Request) {
  const longitude = request.headers.get('x-vercel-ip-longitude');
  return new Response(`Longitude: ${longitude}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs
export function GET(request) {
  const longitude = request.headers.get('x-vercel-ip-longitude');
  return new Response(`Longitude: ${longitude}`);
}
```

```ts filename="api/header.ts" framework=other
export function GET(request: Request) {
  const longitude = request.headers.get('x-vercel-ip-longitude');
  return new Response(`Longitude: ${longitude}`);
}
```

```js filename="api/header.js" framework=other
export function GET(request) {
  const longitude = request.headers.get('x-vercel-ip-longitude');
  return new Response(`Longitude: ${longitude}`);
}
```

```ts filename="app/api/header/route.ts" framework=nextjs-app
export function GET(request: Request) {
  const longitude = request.headers.get('x-vercel-ip-longitude');
  return new Response(`Longitude: ${longitude}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs-app
export function GET(request) {
  const longitude = request.headers.get('x-vercel-ip-longitude');
  return new Response(`Longitude: ${longitude}`);
}
```

## `x-vercel-ip-timezone`

The name of the time zone for the location of the requester's public IP address in ICANN Time Zone Database name format such as `America/

```ts filename="app/api/header/route.ts" framework=nextjs
export function GET(request: Request) {
  const timezone = request.headers.get('x-vercel-ip-timezone');
  return new Response(`Timezone: ${timezone}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs
export function GET(request) {
  const timezone = request.headers.get('x-vercel-ip-timezone');
  return new Response(`Timezone: ${timezone}`);
}
```

```ts filename="api/header.ts" framework=other
export function GET(request: Request) {
  const timezone = request.headers.get('x-vercel-ip-timezone');
  return new Response(`Timezone: ${timezone}`);
}
```

```js filename="api/header.js" framework=other
export function GET(request) {
  const timezone = request.headers.get('x-vercel-ip-timezone');
  return new Response(`Timezone: ${timezone}`);
}
```

```ts filename="app/api/header/route.ts" framework=nextjs-app
export function GET(request: Request) {
  const timezone = request.headers.get('x-vercel-ip-timezone');
  return new Response(`Timezone: ${timezone}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs-app
export function GET(request) {
  const timezone = request.headers.get('x-vercel-ip-timezone');
  return new Response(`Timezone: ${timezone}`);
}
```

## `x-vercel-ip-postal-code`

The postal code close to the user's location.

```ts filename="app/api/header/route.ts" framework=nextjs
export function GET(request: Request) {
  const postalCode = request.headers.get('x-vercel-ip-postal-code');
  return new Response(`Postal Code: ${postalCode}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs
export function GET(request) {
  const postalCode = request.headers.get('x-vercel-ip-postal-code');
  return new Response(`Postal Code: ${postalCode}`);
}
```

```ts filename="api/header.ts" framework=other
export function GET(request: Request) {
  const postalCode = request.headers.get('x-vercel-ip-postal-code');
  return new Response(`Postal Code: ${postalCode}`);
}
```

```js filename="api/header.js" framework=other
export function GET(request) {
  const postalCode = request.headers.get('x-vercel-ip-postal-code');
  return new Response(`Postal Code: ${postalCode}`);
}
```

```ts filename="app/api/header/route.ts" framework=nextjs-app
export function GET(request: Request) {
  const postalCode = request.headers.get('x-vercel-ip-postal-code');
  return new Response(`Postal Code: ${postalCode}`);
}
```

```js filename="app/api/header/route.js" framework=nextjs-app
export function GET(request) {
  const postalCode = request.headers.get('x-vercel-ip-postal-code');
  return new Response(`Postal Code: ${postalCode}`);
}
```

## `x-vercel-signature`

Vercel sends an `x-vercel-signature` header with requests from [Webhooks](/docs/webhooks), [Log Drains](/docs/drains), and other services

### 1. Reading the header value

First, let's see how to read the header value from incoming requests:

```ts filename="app/api/webhook/route.ts" framework=nextjs
export function POST(request: Request) {
  const signature = request.headers.get('x-vercel-signature');
  return new Response(`Signature: ${signature}`);
}
```

```js filename="app/api/webhook/route.js" framework=nextjs
export function POST(request) {
  const signature = request.headers.get('x-vercel-signature');
  return new Response(`Signature: ${signature}`);
}
```

```ts filename="api/webhook.ts" framework=other
export function POST(request: Request) {
  const signature = request.headers.get('x-vercel-signature');
  return new Response(`Signature: ${signature}`);
}
```

```js filename="api/webhook.js" framework=other
export function POST(request) {
  const signature = request.headers.get('x-vercel-signature');
  return new Response(`Signature: ${signature}`);
}
```

```ts filename="app/api/webhook/route.ts" framework=nextjs-app
export function POST(request: Request) {
  const signature = request.headers.get('x-vercel-signature');
  return new Response(`Signature: ${signature}`);
}
```

```js filename="app/api/webhook/route.js" framework=nextjs-app
export function POST(request) {
  const signature = request.headers.get('x-vercel-signature');
  return new Response(`Signature: ${signature}`);
}
```

### 2. Verifying the signature

When your server has a public endpoint, anyone who knows the URL can send requests to it. Verify the signature to confirm the request cam

Vercel creates the signature as an HMAC-SHA1 hash of the raw request body using a secret key. To verify it, generate the same hash with y

```ts filename="app/api/webhook/route.ts" framework=nextjs-app
import crypto from 'crypto';

export async function POST(request: Request) {
  const signatureSecret = process.env.WEBHOOK_SECRET;
  const headerSignature = request.headers.get('x-vercel-signature');

  const rawBody = await request.text();
  const bodySignature = crypto
    .createHmac('sha1', signatureSecret)
    .update(rawBody)
    .digest('hex');

  // Use constant-time comparison to prevent timing attacks
  if (
    !headerSignature ||
```

```
    headerSignature.length !== bodySignature.length ||
    !crypto.timingSafeEqual(
      Buffer.from(headerSignature),
      Buffer.from(bodySignature)
    )
  ) {
    return Response.json({ error: 'Invalid signature' }, { status: 403 });
  }

  // Process the verified request
  const payload = JSON.parse(rawBody);
  return Response.json({ success: true });
}
```

```js filename="app/api/webhook/route.js" framework=nextjs-app
import crypto from 'crypto';

export async function POST(request) {
  const signatureSecret = process.env.WEBHOOK_SECRET;
  const headerSignature = request.headers.get('x-vercel-signature');

  const rawBody = await request.text();
  const bodySignature = crypto
    .createHmac('sha1', signatureSecret)
    .update(rawBody)
    .digest('hex');

  // Use constant-time comparison to prevent timing attacks
  if (
    !headerSignature ||
    headerSignature.length !== bodySignature.length ||
    !crypto.timingSafeEqual(
      Buffer.from(headerSignature),
      Buffer.from(bodySignature)
    )
  ) {
    return Response.json({ error: 'Invalid signature' }, { status: 403 });
  }

  // Process the verified request
  const payload = JSON.parse(rawBody);
  return Response.json({ success: true });
}
```

```ts filename="pages/api/webhook.ts" framework=nextjs
import type { NextApiRequest, NextApiResponse } from 'next';
import crypto from 'crypto';
import getRawBody from 'raw-body';

export default async function handler(
  request: NextApiRequest,
  response: NextApiResponse
) {
  const signatureSecret = process.env.WEBHOOK_SECRET;
  const headerSignature = request.headers['x-vercel-signature'];

  const rawBody = await getRawBody(request);
  const bodySignature = crypto
    .createHmac('sha1', signatureSecret)
    .update(rawBody)
    .digest('hex');

  // Use constant-time comparison to prevent timing attacks
  if (
    !headerSignature ||
    typeof headerSignature !== 'string' ||
    headerSignature.length !== bodySignature.length ||
    !crypto.timingSafeEqual(
      Buffer.from(headerSignature),
      Buffer.from(bodySignature)
    )
  ) {
    return response.status(403).json({ error: 'Invalid signature' });
  }

  // Process the verified request
  const payload = JSON.parse(rawBody.toString('utf-8'));
  return response.status(200).json({ success: true });
}

export const config = {
  api: {
    bodyParser: false,
  },
};
```

```js filename="pages/api/webhook.js" framework=nextjs
import crypto from 'crypto';
import getRawBody from 'raw-body';

export default async function handler(request, response) {
  const signatureSecret = process.env.WEBHOOK_SECRET;
  const headerSignature = request.headers['x-vercel-signature'];

  const rawBody = await getRawBody(request);
  const bodySignature = crypto
    .createHmac('sha1', signatureSecret)
```

```
    .update(rawBody)
    .digest('hex');

  // Use constant-time comparison to prevent timing attacks
  if (
    !headerSignature ||
    typeof headerSignature !== 'string' ||
    headerSignature.length !== bodySignature.length ||
    !crypto.timingSafeEqual(
      Buffer.from(headerSignature),
      Buffer.from(bodySignature)
    )
  ) {
    return response.status(403).json({ error: 'Invalid signature' });
  }

  // Process the verified request
  const payload = JSON.parse(rawBody.toString('utf-8'));
  return response.status(200).json({ success: true });
}

export const config = {
  api: {
    bodyParser: false,
  },
};
```

```ts filename="api/webhook.ts" framework=other
import type { VercelRequest, VercelResponse } from '@vercel/node';
import crypto from 'crypto';
import getRawBody from 'raw-body';

export default async function handler(
  request: VercelRequest,
  response: VercelResponse
) {
  const signatureSecret = process.env.WEBHOOK_SECRET;
  const headerSignature = request.headers['x-vercel-signature'];

  const rawBody = await getRawBody(request);
  const bodySignature = crypto
    .createHmac('sha1', signatureSecret)
    .update(rawBody)
    .digest('hex');

  // Use constant-time comparison to prevent timing attacks
  if (
    !headerSignature ||
    typeof headerSignature !== 'string' ||
    headerSignature.length !== bodySignature.length ||
    !crypto.timingSafeEqual(
      Buffer.from(headerSignature),
      Buffer.from(bodySignature)
    )
  ) {
    return response.status(403).json({ error: 'Invalid signature' });
  }

  // Process the verified request
  const payload = JSON.parse(rawBody.toString('utf-8'));
  return response.status(200).json({ success: true });
}

export const config = {
  api: {
    bodyParser: false,
  },
};
```

```js filename="api/webhook.js" framework=other
import crypto from 'crypto';
import getRawBody from 'raw-body';

export default async function handler(request, response) {
  const signatureSecret = process.env.WEBHOOK_SECRET;
  const headerSignature = request.headers['x-vercel-signature'];

  const rawBody = await getRawBody(request);
  const bodySignature = crypto
    .createHmac('sha1', signatureSecret)
    .update(rawBody)
    .digest('hex');

  // Use constant-time comparison to prevent timing attacks
  if (
    !headerSignature ||
    typeof headerSignature !== 'string' ||
    headerSignature.length !== bodySignature.length ||
    !crypto.timingSafeEqual(
      Buffer.from(headerSignature),
      Buffer.from(bodySignature)
    )
  ) {
    return response.status(403).json({ error: 'Invalid signature' });
  }

  // Process the verified request
  const payload = JSON.parse(rawBody.toString('utf-8'));
```

```
  return response.status(200).json({ success: true });
}

export const config = {
  api: {
    bodyParser: false,
  },
};
```

### 3. Getting your signature secret

The secret key you need depends on what type of request you're receiving:

- **For account webhooks**: The secret displayed when [creating the webhook](/docs/webhooks#enter-your-endpoint-url)
- **For integration webhooks**: Your Integration Secret (also called Client Secret) from the [Integration Console](https://vercel.com/das
- **For log drains**: Click **Edit** in the Drains list to find or update your [Drain signature secret](/docs/drains/security)

For complete examples with additional error handling, see [Securing webhooks](/docs/webhooks/webhooks-api#securing-webhooks) and [Drain s

---
title: "Response headers"
description: "Learn about the response headers sent to each Vercel deployment and how to use them to process responses before sending a r
last_updated: "2026-01-16T02:19:32.250Z"
source: "https://vercel.com/docs/headers/response-headers"
---

# Response headers

The following headers are included in Vercel deployment responses and indicate certain factors of the environment. These headers can be v

## `cache-control`

Used to specify directives for caching mechanisms in both the [Network layer cache](/docs/cdn-cache) and the browser cache. See the [Cach

If you use this header to instruct the CDN to cache data, such as with the [`s-maxage`](/docs/headers/cache-control-headers#s-maxage) dir

-`cache-control: public, max-age=0, must-revalidate`

## `content-length`

An integer that indicates the number of bytes in the response.

## `content-type`

The [media type](https://developer.mozilla.org/docs/Web/HTTP/Basics_of_HTTP/MIME_types) that describes the nature and format of the respo

## `date`

A timestamp indicating when the response was generated.

## `server: Vercel`

Shows where the request came from. This header can be overridden by other proxies (e.g., Cloudflare).

## `strict-transport-security`

A header often abbreviated as [HSTS](https://developer.mozilla.org/docs/Glossary/HSTS) that tells browsers that the resource should only

## `x-robots-tag`

Present only on:

- [Preview deployments](/docs/deployments/environments#preview-environment-pre-production)
- Outdated [production deployments](/docs/deployments). When you [promote a new deployment to production](/docs/deployments/promoting-a-d

We add this header automatically with a value of `noindex` to **prevent** search engines from crawling your Preview Deployments and outda

You can prevent this header from being added to your Preview Deployment by:

- [Assigning a production domain](/docs/domains/working-with-domains/assign-domain-to-a-git-branch) to it
- Disabling it manually [using vercel.json](/docs/project-configuration#headers)

## `x-vercel-cache`

The `x-vercel-cache` header is primarily used to indicate the cache status of static assets and responses from Vercel's CDN. For dynamic

The following values are possible when the content being served [is static](/docs/cdn-cache#static-files-caching) or uses [a Cache-Contro

### `MISS`

The response was not found in the cache and was fetched from the origin server.

### `HIT`

The response was served from the cache.

### `STALE`

The response was served from the cache but the content is no longer fresh, so a background request to the origin server was made to updat

Cached content can go stale for several different reasons such as:

- Response included `stale-while-revalidate` Cache-Control response header.
- Response was served from [ISR](/docs/incremental-static-regeneration) with a revalidation time in frameworks like Next.js.
- On-demand using `@vercel/functions` like [`invalidateByTag()`](/docs/functions/functions-api-reference/vercel-functions-package#invalid
- On-demand using framework-specific functions like [`revalidatePath()`](https://nextjs.org/docs/app/api-reference/functions/revalidatePa
- On-demand using the Vercel dashboard [project purge settings](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Fsettings%2Fcaches

See [purging the cache](/docs/cdn-cache/purge) for more information.

### `PRERENDER`

The response was served from static storage. An example of prerender is in `Next.js`, when setting `fallback:true` in `getStaticPaths`. H

### `REVALIDATED`

The response was served from the origin server after the cache was deleted so it must be revalidated in the foreground.

The cached content can be deleted in several ways such as:

- On-demand using `@vercel/functions` like [`dangerouslyDeleteByTag()`](/docs/functions/functions-api-reference/vercel-functions-package#
- On-demand using framework-specific functions like [`revalidatePath()`](https://nextjs.org/docs/app/api-reference/functions/revalidatePa
- On-demand using the Vercel dashboard [project purge settings](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Fsettings%2Fcaches

See [purging the cache](/docs/cdn-cache/purge) for more information.

## `x-vercel-id`

This header contains a list of [Vercel regions](/docs/regions) your request hit, as well as the region the function was executed in (for

It also allows Vercel to automatically prevent infinite loops.

---
title: "Content Security Policy"
description: "Learn how the Content Security Policy (CSP) offers defense against web vulnerabilities, its key features, and best practice
last_updated: "2026-01-16T02:19:32.096Z"
source: "https://vercel.com/docs/headers/security-headers"
---

# Content Security Policy

Content Security Policy is a browser feature designed to prevent cross-site scripting (XSS) and related code-injection attacks. CSP provi

When a browser receives the `Content-Security-Policy` HTTP header from a web server it adheres to the defined policy, blocking or allowin

[XSS](/kb/guide/understanding-xss-attacks) remains one of the most prevalent web application vulnerabilities. In an XSS attack, malicious

CSP can reduce the likelihood of XSS by:

- **Allowlisting content sources** – CSP works by specifying which sources of content are legitimate for a web application. You can defin
- **Inline script blocking** – A common vector for XSS is through inline scripts, which are scripts written directly within the HTML cont
- **Disallowing `eval()`** – The `eval()` function in JavaScript can be misused to execute arbitrary code, which can be a potential XSS v
- **Nonce and hashes** – If there's a need to allow certain inline scripts (while still blocking others), CSP supports a nonce (number us
- **Reporting violations** – CSP can be set in `report-only` mode where policy violations don't result in content being blocked but inste
- **Plugin restrictions** – Some XSS attacks might exploit browser plugins. With CSP, you can limit the types of plugins that can be invo

While input sanitization and secure coding practices are essential, **CSP acts as a second line of defense**, reducing the risk of [XSS e

Beyond XSS, CSP can prevent the unauthorized loading of content, protecting users from other threats like clickjacking and data injection

## Content Security Policy headers

```bash
Content-Security-Policy: default-src 'self'; script-src 'self' cdn.example.com; img-src 'self' img.example.com; style-src 'self';
```

This policy permits:

- All content to be loaded only from the site's own origin.
- Scripts to be loaded from the site's own origin and cdn.example.com.
- Images from the site's own origin and img.example.com
- Styles only from the site's origin.

## Best Practices

- Before enforcing a CSP, start with the `Content-Security-Policy-Report-Only` header. You can do this to keep an eye on possible violati
- Avoid using `unsafe-inline` and `unsafe-eval` . The use of `eval()` and inline scripts/styles can pose security risks. Avoid enabling t
- Use nonces for inline scripts and styles. To allow that particular inline content, a nonce (number used once) can be added to a script
- Be as detailed as you can, and avoid using too general sources like `.` . List the specific subdomains you want to allow rather than al
- Keep directives updated. As your project evolves, the sources from which you load content might change. Ensure you update your CSP dire

Keep in mind that while CSP is a robust security measure, it's part of a multi-layered security strategy. Input validation, output encodi

Additionally, while CSP is supported by modern browsers, nuances exist in their implementations. Ensure you **test your policy across div

---
title: "Legacy Pricing for Image Optimization"
description: "This page outlines information on the pricing and limits for the source images-based legacy option."
last_updated: "2026-01-16T02:19:32.108Z"
source: "https://vercel.com/docs/image-optimization/legacy-pricing"
---

# Legacy Pricing for Image Optimization

## Pricing

> **💡 Note:** This legacy pricing option is only available to Enterprise teams
> created before February 18th, 2025, who are given the choice to
> [opt-in](https://vercel.com/d?to=%2F%5Bteam%5D%2F~%2Fsettings%2Fbilling%23image-optimization-new-price\&title=Go+to+Billing+Settings)
> to the [transformation images-based pricing
> plan](/docs/image-optimization/limits-and-pricing) or stay on this legacy
> source images-based pricing plan until the contract expires.

Image Optimization pricing is dependent on your plan and how many unique [source images](#source-images) you have across your projects du

| Resource | Price |

| ---------------------------------------------- | ---------------------- |
| [Image Optimization Source Images](#source-images) | $5.00 per 1,000 Images |

## Usage

The table below shows the metrics for the Image Optimization section of the **Usage** dashboard.

To view information on managing each resource, select the resource link in the **Metric** column. To jump straight to guidance on optimiz

Usage is not incurred until an image is requested.

### Source Images

A source image is the value that is passed to the `src` prop. A single source image can produce multiple optimized images. For example:

- Usage: `<Image src="/hero.png" width="700" height="745" />`
- Source image: `/hero.png`
- Optimized image: `/_next/image?url=%2Fhero.png&w=750&q=75`
- Optimized image: `/_next/image?url=%2Fhero.png&w=828&q=75`
- Optimized image: `/_next/image?url=%2Fhero.png&w=1080&q=75`

For example, if you have passed 6000 source images to the `src` prop within the last billing cycle, your bill will be $5 for image optimi

## Billing

You are billed for the **number of unique [source images](#source-images) requested during the billing period**.

Additionally, charges apply for [Fast Data Transfer](/docs/manage-cdn-usage#fast-data-transfer) when optimized images are delivered from '

### Hobby

Image Optimization is free for Hobby users within the [usage limits](/docs/limits/fair-use-guidelines#typical-monthly-usage-guidelines). A

Vercel will send you emails as you are nearing your [usage](#pricing) limits, but you will also be advised of any alerts within the [dash

Once you exceed the limits:

- New [source images](#source-images) will fail to optimize and instead return a runtime error response with [402 status code](/docs/erro
- Previously optimized images have already been cached and will continue to work as expected, without error

You will **not** be charged for exceeding the usage limits, but this usually means your application is ready to upgrade to a [Pro plan](/

If you want to continue using Hobby, read more about [Managing Usage & Costs](/docs/image-optimization/managing-image-optimization-costs)

### Pro and Enterprise

For Teams on Pro trials, the [trial will end](/docs/plans/pro-plan/trials#post-trial-decision) if your Team uses over 2500 source images.

Vercel will send you emails as you are nearing your [usage](#pricing) limits, but you will also be advised of any alerts within the [dash

Pro teams can [set up Spend Management](/docs/spend-management#managing-your-spend-amount) to get notified or to automatically take actio

## Limits

For all the images that are optimized by Vercel, the following limits apply:

- The maximum size for an optimized image is **10 MB**, as set out in the [Cacheable Responses limits](/docs/cdn-cache#how-to-cache-respo
- Each [source image](#source-images) has a maximum width and height of 8192 pixels
- A [source image](#source-images) must be one of the following formats to be optimized: `image/jpeg`, `image/png`, `image/webp`, `image/

See the [Fair Usage Policy](/docs/limits/fair-use-guidelines#typical-monthly-usage-guidelines) for typical monthly usage guidelines.


---
title: "Limits and Pricing for Image Optimization"
description: "This page outlines information on the limits that are applicable when using Image Optimization, and the costs they can incu
last_updated: "2026-01-16T02:19:32.122Z"
source: "https://vercel.com/docs/image-optimization/limits-and-pricing"
---

# Limits and Pricing for Image Optimization

## Pricing

> **💡 Note:** This is the default pricing option. For Enterprise teams created
> before February 18th, 2025, you will be given the choice to
> [opt-in](https://vercel.com/d?to=%2F%5Bteam%5D%2F~%2Fsettings%2Fbilling%23image-optimization-new-price\&title=Go+to+Billing+Settings)
> to this pricing plan or stay on the [legacy source
> images-based](/docs/image-optimization/legacy-pricing) pricing plan until the contract expires.

Image optimization pricing is dependent on your plan and on specific parameters outlined in the table below. For detailed pricing informa

| Image Usage                                  | Hobby Included | On-demand Rates                                                                |
| -------------------------------------------- | -------------- | ------------------------------------------------------------------------------ |
| [Image transformations](#image-transformations) | 5K/month       | [$0.05 - $0.0812 per 1K](/docs/pricing/regional-pricing#specific-reg |
| [Image cache reads](#image-cache-reads)         | 300K/month     | [$0.40 - $0.64 per 1M](/docs/pricing/regional-pricing#specific-regio |
| [Image cache writes](#image-cache-writes)       | 100K/month     | [$4.00 - $6.40 per 1M](/docs/pricing/regional-pricing#specific-regio |

This ensures that you only pay for the optimizations when the images are used instead of the number of images in your project.

## Image transformations

Image transformations are billed for every cache MISS and STALE. The cache key is based on several inputs and differs for [local images c

## Image cache reads

The total amount of Read Units used to access the cached image from the global cache, measured in 8KB units.

It is *not* billed for every cache HIT, only when the image needs to be retrieved from the shared global cache.

An image that has been accessed recently (several hours ago) in the same region will be cached in region and does *not* incur this cost.

## Image cache writes

The total amount of Write Units used to store the cached image in the global cache, measured in 8KB units. It is billed for every cache M

## Billing

You are billed for the number of [Image Transformations](#image-transformations), [Image Cache Reads](#image-cache-reads), and [Image Cac

Additionally, charges apply for [Fast Data Transfer](/docs/manage-cdn-usage#fast-data-transfer) and [Edge Requests](/docs/manage-cdn-usag

### Hobby

Image Optimization is free for Hobby users within the [usage limits](/docs/limits/fair-use-guidelines#typical-monthly-usage-guidelines). A

Vercel will send you emails as you are nearing your [usage](#pricing) limits, but you will also be advised of any alerts within the [dash

Once you exceed the limits:

- New images will fail to optimize and instead return a runtime error response with [402 status code](/docs/errors/platform-error-codes#4
- Previously optimized images have already been cached and will continue to work as expected, without error

You will **not** be charged for exceeding the usage limits, but this usually means your application is ready to upgrade to a [Pro plan](/

If you want to continue using Hobby, read more about [Managing Usage & Costs](/docs/image-optimization/managing-image-optimization-costs)

### Pro and Enterprise

Vercel will send you emails as you are nearing your [usage](#pricing) limits, but you will also be advised of any alerts within the [dash

Pro teams can [set up Spend Management](/docs/spend-management#managing-your-spend-amount) to get notified or to automatically take actio

## Limits

For all the images that are [optimized by Vercel](/docs/image-optimization/managing-image-optimization-costs#measuring-usage), the follow

- The maximum size for an transformed image is **10 MB**, as set out in the [Cacheable Responses limits](/docs/cdn-cache#how-to-cache-res
- Each source image has a maximum width and height of 8192 pixels
- A source image must be one of the following formats to be optimized: `image/jpeg`, `image/png`, `image/webp`, `image/avif`. Other forma

See the [Fair Usage Policy](/docs/limits/fair-use-guidelines#typical-monthly-usage-guidelines) for typical monthly usage guidelines.

---
title: "Managing Usage & Costs"
description: "Learn how to measure and manage Image Optimization usage with this guide to avoid any unexpected costs."
last_updated: "2026-01-16T02:19:32.128Z"
source: "https://vercel.com/docs/image-optimization/managing-image-optimization-costs"
---

# Managing Usage & Costs

## Measuring usage

> **💡 Note:** This document describes usage for the default pricing option.
> Enterprise teams created before February 18th, 2025 have the choice to
> [opt-in](https://vercel.com/d?to=%2F%5Bteam%5D%2F~%2Fsettings%2Fbilling%23image-optimization-new-price\&title=Go+to+Billing+Settings)
> to this pricing plan or stay on the [legacy source images-based pricing plan](/docs/image-optimization/legacy-pricing)
> until the contract expires.

Your Image Optimization usage over time is displayed under the **Image Optimization** section of the [Usage](https://vercel.com/d?to=%2F%

You can also view detailed information in the **Image Optimization** section of the [Observability](https://vercel.com/d?to=%2F%5Bteam%5D%

## Reducing usage

To help you minimize Image Optimization usage costs, consider implementing the following suggestions:

- **Cache Max Age**: If your images do not change in less than a month, set `max-age=2678400` (31 days) in the `Cache-Control` header or

- **Formats**: Check if your Next.js configuration is using [`images.formats`](https://nextjs.org/docs/app/api-reference/components/image

- **Remote and local patterns**: Configure [`images.remotePatterns`](https://nextjs.org/docs/app/api-reference/components/image#remotepat

- **Qualities**: Configure the [`images.qualities`](https://nextjs.org/docs/app/api-reference/components/image#qualities) allowlist to re

- **Image sizes**: Configure the [`images.imageSizes`](https://nextjs.org/docs/app/api-reference/components/image#imagesizes) and [`image

- **Unoptimized**: For source images that do not benefit from optimization such as small images (under 10 KB), vector images (SVG) and an

---
title: "Image Optimization with Vercel"
description: "Transform and optimize images to improve page load performance."
last_updated: "2026-01-16T02:19:32.187Z"
source: "https://vercel.com/docs/image-optimization"
---

# Image Optimization with Vercel

Vercel supports dynamically transforming unoptimized images to reduce the file size while maintaining high quality. These optimized image

## Get started

Image Optimization works with many frameworks, including Next.js, Astro, and Nuxt, enabling you to optimize images using built-in compone

- Get started by following the [Image Optimization Quickstart](/docs/image-optimization/quickstart) and selecting your framework (Next.js
- For a live example which demonstrates usage with the [`next/image`](https://nextjs.org/docs/pages/api-reference/components/image) compo

## Why should I optimize my images on Vercel?

Optimizing images on Vercel provides several advantages for your application:

- Reduces the size of images and data transferred, enhancing website performance, user experience, and [Fast Data Transfer](/docs/manage-
- Improving [Core Web Vitals](https://web.dev/vitals/), reduced bounce rates, and speeding up page loads.
- Sizing images to support different devices and use modern formats like [WebP](https://developer.mozilla.org/docs/Web/Media/Formats/Imag
- Optimized images are cached after transformation, which allows them to be reused in subsequent requests.

## How Image Optimization works

The flow of image optimization on Vercel involves several steps, starting from the image request to serving the optimized image.

1. The optimization process starts with your component choice in your codebase:
   - If you use a standard HTML `img` element, the browser will be instructed to bypass optimization and serve the image directly from it
   - If you use a framework's `Image` component (like [`next/image`](https://nextjs.org/docs/app/api-reference/components/image)) it will

2. When Next.js receives an image request, it checks the [`unoptimized`](https://nextjs.org/docs/app/api-reference/components/image#unopt
   - If you set the `unoptimized` prop on the `Image` component to `true`, Next.js bypasses optimization and serves the image directly fr
   - If you don't set the `unoptimized` prop or set it to `false`, Next.js checks the `next.config.ts` file to see if optimization is dis
   - If neither the `unoptimized` prop is set nor optimization is disabled in the `next.config.ts` file, Next.js continues with the optim

3. If optimization is enabled, Vercel validates the [loader configuration](https://nextjs.org/docs/app/api-reference/components/image#loa

4. Vercel then checks the status of the cache to see if an image has been previously cached:

- `HIT`: The image is fetched and served from the cache, either in region or from the shared global cache.
  - If fetched from the global cache, it's billed as an [image cache read](/docs/image-optimization/limits-and-pricing#image-cache-reads)
- `MISS`: The image is fetched, transformed, cached, and then served to the user.
  - Billed as an [image transformation](/docs/image-optimization/limits-and-pricing#image-transformations) and [image cache write](/docs/
- `STALE`: The image is fetched and served from the cache while revalidating in the background.
  - Billed as an [image transformation](/docs/image-optimization/limits-and-pricing#image-transformations) and [image cache write](/docs/

## When to use Image Optimization

Image Optimization is ideal for:

- Responsive layouts where images need to be optimized for different device sizes (e.g. mobile vs desktop)
- Large, high-quality images (e.g. product photos, hero images)
- User uploaded images
- Content where images play a central role (e.g. photography portfolios)

In some cases, Image Optimization may not be necessary or beneficial, such as:

- Small icons or thumbnails (under 10 KB)
- Animated image formats such as GIFs
- Vector image formats such as SVG
- Frequently changing images where caching could lead to outdated content

If your images meet any of the above criteria where Image Optimization is not beneficial, we recommend using the [`unoptimized`](https://

It's important that you are only optimizing images that need to be optimized otherwise you could end up using your [image usage](/docs/im

## Setting up remote or local patterns

An important aspect of using the `Image` component is properly setting up remote/local patterns in your `next.config.ts` file. This confi

You can set up patterns for both [local images](#local-images) (stored as static assets in your `public` folder) and [remote images](#rem

### Local images

A local image is imported from your file system and analyzed at build time. The import is added to the `src` prop: `src={myImage}`

#### Setting up local patterns

To set up local patterns, you need to specify the pathname of the images you want to optimize. This is done in the `next.config.ts` file:

```ts filename="next.config.ts"
module.exports = {
  images: {
    localPatterns: [
      {
        pathname: '/assets/images/**',
        search: '',
      },
    ],
  },
};
```

See the [Next.js documentation for local patterns](https://nextjs.org/docs/app/api-reference/components/image#localpatterns) for more inf

#### Local images cache key

The cache key for local images is based on the query string parameters, the `Accept` HTTP header, and the content hash of the image URL.

- **Cache Key**:
  - Project ID
  - Query string parameters:
    - `q`: The desired quality of the transformed image, between 1 (lowest quality) and 100 (highest quality).
    - `w`: The desired width (in pixels) of the transformed image.
    - `url`: The URL of the source image. For local images (`/assets/me.png`) the content hash is used instead (`3399d02f49253deb9f5b5d11
  - `Accept` HTTP header (normalized).
- **Local image cache invalidation**:
  - Redeploying your app doesn't invalidate the image cache.
  - To invalidate, replace the image of the same name with different content, then [redeploy](/docs/deployments/managing-deployments#rede
  - You can also [manually purge](/docs/cdn-cache/purge#manually-purging-vercel-cdn-cache) or [programatically purge](/docs/cdn-cache/pur
- **Local image cache expiration**:
  - [Cached](/docs/cdn-cache#static-files-caching) **for up to 31 days** on the Vercel CDN.

### Remote images

A remote image requires the `src` property to be a URL string, which can be relative or absolute.

#### Setting up remote patterns

To set up remote patterns, you need to specify the `hostname` of the images you want to optimize. This is done in the `next.config.ts` fi

```ts filename="next.config.ts"
module.exports = {
  images: {
    remotePatterns: [
      {
        protocol: 'https',
        hostname: 'example.com',
        port: '',
        pathname: '/account123/**',
        search: '',
      },
    ],
  },
};
```

In the case of external images, you should consider adding your account id to the `pathname` if you don't own the `hostname`. For example

See the [Next.js documentation for remote patterns](https://nextjs.org/docs/app/api-reference/components/image#remotepatterns) for more i

#### Remote images cache key

The cache key for remote images is based on the query string parameters, the `Accept` HTTP header, and the content hash of the image URL.

- **Cache Key**:
  - Project ID
  - Query string parameters:
    - `q`: The desired quality of the transformed image, between 1 (lowest quality) and 100 (highest quality).
    - `w`: The desired width (in pixels) of the transformed image.
    - `url`: The URL of the source image. Remote images use an absolute url (`https://example.com/assets/me.png`).
  - `Accept` HTTP header (normalized).
- **Remote image cache invalidation**:
  - Redeploying your app doesn't invalidate the image cache
  - You can [manually purge](/docs/cdn-cache/purge#manually-purging-vercel-cdn-cache) or [programatically purge](/docs/cdn-cache/purge#pr
  - Alternatively, you can configure the cache to expire more frequently by adjusting the TTL.
- **Remote image cache expiration**:
  - TTL is determined by the [`Cache-Control`](/docs/headers#cache-control-header) `max-age` header from the upstream image or [`minimumC

Once an image is cached, it remains so even if you update the source image. For remote images, users accessing a URL with a previously ca

See [Pricing](/docs/image-optimization/limits-and-pricing) for more information, and read more about [caching behavior](https://nextjs.or

## Image Transformation URL format

When you use the `Image` component in common frameworks and deploy your project on Vercel, Image Optimization automatically adjusts your

- Next.js: `/_next/image?url={link/to/src/image}&w=3840&q=75`
- Nuxt, Astro, etc: `/_vercel/image?url={link/to/src/image}&w=3840&q=75`

The Image Optimization API has the following query parameters:

- `url`: The URL of the source image to be transformed. This can be a local image (relative url) or remote image (absolute url).
- `w`: The width of the transformed image in pixels. No height is needed since the source image aspect ratio is preserved.
- `q`: The quality of the transformed image, between 1 (lowest quality) and 100 (highest quality).

The allowed values of those query parameters are determined by the framework you are using, such as `next.config.js` for Next.js.

If you are not using a framework that comes with an `Image` component or you are building your own framework, refer to the [Build Output A

## Opt-in

To switch to the transformation images-based pricing plan:

1. Choose your team scope on the dashboard, and go to **Settings**, then **Billing**
2. Scroll down to the **Image Optimization** section
3. Select **Review Cost Estimate**. Proceed to enable this option in the dialog that shows the cost estimate.

[View your estimate](https://vercel.com/d?to=%2F%5Bteam%5D%2F~%2Fsettings%2Fbilling%23image-optimization-new-price\&title=Go+to+Billing+S

## Related

For more information on what to do next, we recommend the following articles:

- [Image Optimization quickstart](/docs/image-optimization/quickstart)
- [Managing costs](/docs/image-optimization/managing-image-optimization-costs)
- [Pricing](/docs/image-optimization/limits-and-pricing)
- If you are building a custom web framework, you can also use the [Build Output API](/docs/build-output-api/v3/configuration#images) to

---
title: "Getting started with Image Optimization"
description: "Learn how you can leverage Vercel Image Optimization in your projects."
last_updated: "2026-01-16T02:19:32.156Z"
source: "https://vercel.com/docs/image-optimization/quickstart"
---

# Getting started with Image Optimization

This guide will help you get started with using Vercel Image Optimization in your project, showing you how to import images, add the requ

## Prerequisites

- A Vercel account. If you don't have one, you can [sign up for free](https://vercel.com/signup).
- A Vercel project. If you don't have one, you can [create a new project](https://vercel.com/new).
- The Vercel CLI installed. If you don't have it, you can install it using the following command:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i vercel
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i vercel
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i vercel
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i vercel
      ```
    </Code>
  </CodeBlock>

- ### Import images
  > For \['astro']:
  To use Astro, you must:
  1. Enable [Vercel's image service](https://docs.astro.build/en/guides/integrations-guide/vercel/#imageservice) in 

     ```js filename="astro.config.mjs" framework=all
     import { defineConfig } from 'astro/config';
     import vercel from '@astrojs/vercel/static';

     export default defineConfig({
       output: 'server',
       adapter: vercel({
         imageService: true,
       }),
     });
     ```

     ```ts filename="astro.config.mjs" framework=all
     import { defineConfig } from 'astro/config';
     import vercel from '@astrojs/vercel/static';

     export default defineConfig({
       output: 'server',
       adapter: vercel({
         imageService: true,
       }),
     });
     ```

  2. Use Astro's built-in `Image` component:

     ```jsx filename="src/components/MyComponent.astro" framework=all
     ---
     // import the Image component and the image
     import { Image } from 'astro:assets';
     import myImage from "../assets/my_image.png"; // Image is 1600x900
     ---

     {/* `alt` is mandatory on the Image component */}
     <Image src={myImage} alt="A description of my image." />
     ```

     ```tsx filename="src/components/MyComponent.astro" framework=all
     ---
     // import the Image component and the image
     import { Image } from 'astro:assets';
     import myImage from "../assets/my_image.png"; // Image is 1600x900
     ---

     {/* `alt` is mandatory on the Image component */}
     <Image src={myImage} alt="A description of my image." />
     ```
  > For \['nextjs']:
  Next.js provides a built-in [`next/image`](https://nextjs.org/docs/pages/api-reference/components/image) component.
  ```js filename="pages/index.js" framework=all
  import Image from 'next/image';
  ```
  ```ts filename="pages/index.ts" framework=all
  import Image from 'next/image';
  ```
  > For \['sveltekit']:
  To use SvelteKit, use [`@sveltejs/adapter-vercel`](https://kit.svelte.dev/docs/adapter-vercel) within your  file.
  ```js filename="svelte.config.js" framework=all
  import adapter from '@sveltejs/adapter-vercel';

  export default {
    kit: {
      adapter({
        images: {
          sizes: [640, 828, 1200, 1920, 3840],
          formats: ['image/avif', 'image/webp'],
          minimumCacheTTL: 300,
          domains: ['example-app.vercel.app'],
        }
  ```

```
    })
  }
};
```
```ts filename="svelte.config.ts" framework=all
import adapter from '@sveltejs/adapter-vercel';

export default {
  kit: {
    adapter({
      images: {
        sizes: [640, 828, 1200, 1920, 3840],
        formats: ['image/avif', 'image/webp'],
        minimumCacheTTL: 300,
        domains: ['example-app.vercel.app'],
      }
    })
  }
};
```
This allows you to specify [configuration options](https://vercel.com/docs/build-output-api/v3/configuration#images) for Vercel's nativ

You have to construct your own `srcset` URLs to use image optimization with SvelteKit. You can create a library function that will optii
```js filename="src/lib/image.js" framework=all
import { dev } from '$app/environment';

export function optimize(src, widths = [640, 960, 1280], quality = 90) {
  if (dev) return src;

  return widths
    .slice()
    .sort((a, b) => a - b)
    .map((width, i) => {
      const url = `/_vercel/image?url=${encodeURIComponent(src)}&w=${width}&q=${quality}`;
      const descriptor = i < widths.length - 1 ? ` ${width}w` : '';
      return url + descriptor;
    })
    .join(', ');
}
```
```ts filename="src/lib/image.ts" framework=all
import { dev } from '$app/environment';

export function optimize(src: string, widths = [640, 960, 1280], quality = 90) {
  if (dev) return src;

  return widths
    .slice()
    .sort((a, b) => a - b)
    .map((width, i) => {
      const url = `/_vercel/image?url=${encodeURIComponent(src)}&w=${width}&q=${quality}`;
      const descriptor = i < widths.length - 1 ? ` ${width}w` : '';
      return url + descriptor;
    })
    .join(', ');
}
```
> For \['nextjs-app']:
Next.js provides a built-in [`next/image`](https://nextjs.org/docs/app/api-reference/components/image) component.
```js filename="app/example/page.jsx" framework=all
import Image from 'next/image';
```
```ts filename="app/example/page.tsx" framework=all
import Image from 'next/image';
```
> For \['nuxt']:
Install the `@nuxt/image` package:

Then, add the module to the `modules` array in your Nuxt config:
```js filename="nuxt.config.js" framework=all
export default defineNuxtConfig({
  modules: ['@nuxt/image'],
});
```
```ts filename="nuxt.config.ts" framework=all
export default defineNuxtConfig({
  modules: ['@nuxt/image'],
});
```
When you deploy to Vercel, the Vercel provider will be automatically enabled by default. **Vercel requires you to explicitly list all t
```js filename="nuxt.config.js" framework=all
export default defineNuxtConfig({
  modules: ['@nuxt/image'],
  image: {
    // You must specify every custom width used in <NuxtImg>, <NuxtPicture> or $img
    screens: {
      xs: 320,
      sm: 640,
      md: 768,
      lg: 1024,
      xl: 1280,
      xxl: 1536,
      // Add any custom widths used in your components
      avatar: 40,
      avatar2x: 80,
      hero: 1920,
    },
    // Whitelist external domains for images not in public/ directory
    domains: ['example.com', 'images.unsplash.com'],
  },
```

```
});
```
```ts filename="nuxt.config.ts" framework=all
export default defineNuxtConfig({
  modules: ['@nuxt/image'],
  image: {
    // You must specify every custom width used in <NuxtImg>, <NuxtPicture> or $img
    screens: {
      xs: 320,
      sm: 640,
      md: 768,
      lg: 1024,
      xl: 1280,
      xxl: 1536,
      // Add any custom widths used in your components
      avatar: 40,
      avatar2x: 80,
      hero: 1920,
    },
    // Whitelist external domains for images not in public/ directory
    domains: ['example.com', 'images.unsplash.com'],
  },
});
```
**Important:** If a width is not defined in your configuration, the image will fallback to the next bigger width, which may affect perf

See the [Nuxt Image documentation](https://image.nuxt.com/providers/vercel) for more details on Vercel provider requirements and [confi

- ### Add the required props
  > For \['astro']:
  The only required props for Astro's `Image` component are `alt` and `src`. All other attributes are enforced automatically if not speci
  ```jsx filename="src/components/MyComponent.astro" framework=all
  ---
  // import the Image component and the image
  import { Image } from 'astro:assets';
  import myImage from "../assets/my_image.png"; // Image is 1600x900
  ---

  {/* `alt` is mandatory on the Image component */}
  <Image src={myImage} alt="A description of my image." />
  ```
  ```tsx filename="src/components/MyComponent.astro" framework=all
  ---
  // import the Image component and the image
  import { Image } from 'astro:assets';
  import myImage from "../assets/my_image.png"; // Image is 1600x900
  ---

  {/* `alt` is mandatory on the Image component */}
  <Image src={myImage} alt="A description of my image." />
  ```
  The output would look like this:
  ```tsx filename="src/components/MyComponent.astro" framework=all
  {
    /* Output */
  }
  {
    /* Image is optimized, proper attributes are enforced */
  }
  <img
    src="/_astro/my_image.hash.webp"
    width="1600"
    height="900"
    decoding="async"
    loading="lazy"
    alt="A description of my image."
  />;
  ```
  ```jsx filename="src/components/MyComponent.astro" framework=all
  {
    /* Output */
  }
  {
    /* Image is optimized, proper attributes are enforced */
  }
  <img
    src="/_astro/my_image.hash.webp"
    width="1600"
    height="900"
    decoding="async"
    loading="lazy"
    alt="A description of my image."
  />;
  ```
  > For \['nextjs']:
  This component takes the following [required props](https://nextjs.org/docs/pages/api-reference/components/image#required-props):
  - `src`: The URL of the image to be loaded
  - `alt`: A short description of the image
  - `width`: The width of the image
  - `height`: The height of the image
  When using [local images](https://nextjs.org/docs/pages/building-your-application/optimizing/images#local-images "Local images") you **

  The below example uses a [remote image](https://nextjs.org/docs/pages/building-your-application/optimizing/images#remote-images "Remote
  ```js filename="pages/index.jsx" framework=all
  <Image
    src="https://unsplash.com/photos/MpL4w1vb798"
    alt="Picture of a triangle"
    width={500}
    height={500}
  />
  ```

```
```ts filename="pages/index.tsx" framework=all
<Image
  src="https://unsplash.com/photos/MpL4w1vb798"
  alt="Picture of a triangle"
  width={500}
  height={500}
/>
```
If you have images with URLs that may change frequently, even if the image content remains the same, you might want to avoid optimizati

For more information on all props, caching behavior, and responsive images, visit the [`next/image`](https://nextjs.org/docs/pages/api-
> For \['sveltekit']:
To use image optimization with SvelteKit, you can use the `img` tag or any image component. Use an optimized `srcset` string generated
```tsx filename="src/components/image.svelte" framework=all
<script lang="ts">
  import { optimize } from '$lib/image';
  import type { Photo } from '$lib/types';

  export let photo: Photo;
</script>

<img
  class="absolute left-0 top-0 w-full h-full"
  srcset={optimize(photo.url)}
  alt={photo.description}
/>
```
```jsx filename="src/components/image.svelte" framework=all
<script lang="js">
  import { optimize } from '$lib/image';

  export let photo;
</script>

<img
  class="absolute left-0 top-0 w-full h-full"
  srcset={optimize(photo.url)}
  alt={photo.description}
/>
```
> For \['nextjs-app']:
This component takes the following [required props](https://nextjs.org/docs/app/api-reference/components/image#required-props):
- `src`: The URL of the image
- `alt`: A short description of the image
- `width`: The width of the image
- `height`: The height of the image
When using [local images](https://nextjs.org/docs/app/building-your-application/optimizing/images#local-images "Local images") you **do

The example below uses a [remote image](https://nextjs.org/docs/app/building-your-application/optimizing/images#remote-images "Remote I
```js filename="app/example/page.jsx" framework=all
<Image
  src="https://images.unsplash.com/photo-1627843240167-b1f9d28f732e"
  alt="Triangular frames arranged concentrically, creating a tunnel against a dark background."
  width={1920}
  height={1080}
/>
```
```ts filename="app/example/page.tsx" framework=all
<Image
  src="https://images.unsplash.com/photo-1627843240167-b1f9d28f732e"
  alt="Triangular frames arranged concentrically, creating a tunnel against a dark background."
  width={1920}
  height={1080}
/>
```
If there are some images that you wish to not optimize (for example, if the URL contains a token), you can use the [unoptimized](https:

For more information on all props, caching behavior, and responsive images, visit the [`next/image`](https://nextjs.org/docs/app/api-re
> For \['nuxt']:
The `<NuxtImg>` component will automatically optimize your images on demand. It is a wrapper around the `<img>` element, and takes all

The following example demonstrates a `<NuxtImg>` component with optimization props:
```jsx filename="pages/index.vue" framework=all
<template>
  <NuxtImg
    preset="cover"
    src="/nuxt-icon.png"
    width="100"
    height="100"
    sizes="sm:100vw md:50vw lg:400px"
    provider="static"
    format="webp"
    quality="70"
    modifiers="rounded"
  />
</template>
```
```tsx filename="pages/index.vue" framework=all
<template>
  <NuxtImg
    preset="cover"
    src="/nuxt-icon.png"
    width="100"
    height="100"
    sizes="sm:100vw md:50vw lg:400px"
    provider="static"
    format="webp"
    quality="70"
    modifiers="rounded"
```
```

```
    />
  </template>
  ```

- ### Deploy your app to Vercel
  > For \['nextjs', 'nextjs-app']:
  Push your changes and deploy your Next.js application to Vercel.

  When deployed to Vercel, this component automatically optimizes your images on-demand and serves them from the [Vercel CDN](/docs/cdn).
  > For \['sveltekit']:
  Push your changes and deploy your SvelteKit application to Vercel.

  Your images that use optimized `src` URLs will leverage Vercel's on-demand image optimization. Images get served from the [Vercel CDN](
  > For \['astro']:
  Push your changes and deploy your Astro application to Vercel.

  When deployed to Vercel, this component automatically optimizes your images on-demand and serves them from the [Vercel CDN](/docs/cdn).
  > For \['nuxt']:
  When you deploy your Nuxt application to Vercel, the Vercel provider will be automatically enabled by default and use Vercel's CDN for

  The `<NuxtImg>` components will automatically optimize your images and serve them from the [Vercel CDN](/docs/cdn). Make sure you have

  For more information on usage with external URLs and customizing your images on demand, visit the [`@nuxt/image`](https://image.nuxt.co

## Next steps

Now that you've set up Vercel Image Optimization, you can explore the following:

- [Explore limits and pricing](/docs/image-optimization/limits-and-pricing)
- [Managing costs](/docs/image-optimization/managing-image-optimization-costs)


---
title: "Incremental Migration to Vercel"
description: "Learn how to migrate your app or website to Vercel with minimal risk and high impact."
last_updated: "2026-01-16T02:19:32.205Z"
source: "https://vercel.com/docs/incremental-migration"
---

# Incremental Migration to Vercel

When migrating to Vercel you should use an incremental migration strategy. This allows your current site and your new site to operate sim

In this guide, we'll explore incremental migration benefits, strategies, and implementation approaches for a zero-downtime migration to V

## Why opt for incremental migration?

Incremental migrations offer several advantages:

- Reduced risk due to smaller migration steps
- A smoother rollback path in case of unexpected issues
- Earlier technical implementation and business value validation
- Downtime-free migration without maintenance windows

### Disadvantages of one-time migrations

One-time migration involves developing the new site separately before switching traffic over. This approach has certain drawbacks:

- Late discovery of expensive product issues
- Difficulty in assessing migration success upfront
- Potential for reaching a point of no-return, even with major problem detection
- Possible business loss due to legacy system downtime during migration

### When to use incremental migration?

Despite requiring more effort to make the new and legacy sites work concurrently, incremental migration is beneficial if:

- Risk reduction and time-saving benefits outweigh the effort
- The extra effort needed for specific increments to interact with legacy data
  doesn't exceed the time saved

## Incremental migration strategies



With incremental migration, legacy and new systems operate simultaneously. Depending on your strategy, you'll select a system aspect, lik

### Vertical migration

This strategy targets system features with the following process:

1. Identify all legacy system features
2. Choose key features for the initial migration
3. Repeat until all features have been migrated

Throughout, both systems operate in parallel with migrated features routed to the new system.

### Horizontal migration

This strategy focuses on system users with the following process:

1. Identify all user groups
2. Select a user group for initial migration to the new system
3. Repeat until all users have been migrated

During migration, a subset of users accesses the new system while others continue using the legacy system.

### Hybrid migration

A blend of vertical and horizontal strategies. For each feature subset, migrate by user group before moving to the next feature subset.

## Implementation approaches

Follow these steps to incrementally migrate your website to Vercel. Two possible strategies can be applied:

1. [Point your domain to Vercel from the beginning](#point-your-domain-to-vercel)
2. [Keep your domain on the legacy server](#keep-your-domain-on-the-legacy-server)

## Point your domain to Vercel

In this approach, you make Vercel [the entry point for all your production traffic](/docs/domains/add-a-domain). When you begin, all traf

### 1. Deploy your application

Use the [framework](/docs/frameworks) of your choice to deploy your application to Vercel

### 2. Re-route the traffic

Send all traffic to the legacy server using one of the following 3 methods:

#### Framework-specific rewrites

Use rewrites [built-in to the framework](/docs/rewrites#framework-considerations) such as configuring `next.config.ts` with [fallbacks an

The code example below shows how to configure rewrites with fallback using `next.config.js` to send all traffic to the legacy server:

```ts filename="next.config.ts"
import type { NextConfig } from 'next';

const nextConfig: NextConfig = {
  async rewrites() {
    return {
      fallback: [
        {
          source: '/:path*',
          destination: 'https://my-legacy-site.com/:path*',
        },
      ],
    };
  },
};

export default nextConfig;
```

#### Vercel configuration rewrites

Use `vercel.json` for frameworks that do not have rewrite support. See the [how do rewrites work](/docs/rewrites) documentation to learn

#### Edge Config

Use [Edge Config](/docs/edge-config) with [Routing Middleware](/docs/routing-middleware) to rewrite requests on the global network with t

- No need to re-deploy your application when rewrite changes are required
- Immediately switch back to the legacy server if the new feature implementation is broken

Review this [maintenance page example](https://vercel.com/templates/next.js/maintenance-page) to understand the mechanics of this approac

This is an example middleware code for executing the rewrites on the global network:

```ts filename="middleware.ts"
import { get } from '@vercel/edge-config';
import { NextRequest, NextResponse } from 'next/server';

export const config = {
  matcher: '/((?!api|_next/static|favicon.ico).*)',
};

export default async function middleware(request: NextRequest) {
  const url = request.nextUrl;
  const rewrites = await get('rewrites'); // Get rewrites stored in Edge Config

  for (const rewrite of rewrites) {
    if (rewrite.source === url.pathname) {
      url.pathname = rewrite.destination;
      return NextResponse.rewrite(url);
    }
  }

  return NextResponse.next();
}
```

In the above example, you use Edge Config to store one key-value pair for each rewrite. In this case, you should consider [Edge Config Li

### 3. Deploy to production

Connect your [production domain](/docs/getting-started-with-vercel/domains) to your Vercel Project. All your traffic will now be sent to

### 4. Deploy your first iteration

Develop and test the first iteration of your application on Vercel on specific paths.

With the fallback approach such as with the `next.config.js` example above, Next.js will automatically serve content from your Vercel pro

Repeat this process until all the paths are migrated to Vercel and all rewrites are removed.

## Keep your domain on the legacy server

In this approach, once you have tested a specific feature on your new Vercel application, you configure your legacy server or proxy to se

### 1. Deploy your first feature

Use the [framework](/docs/frameworks) of your choice to deploy your application on Vercel and build the first feature that you would like

### 2. Add a rewrite or reverse proxy

Once you have tested the first feature fully on Vercel, add a rewrite or reverse proxy to your existing server to send the traffic on the

For example, if you are using [nginx](https://nginx.org/), you can use the [`proxy_pass`](https://nginx.org/en/docs/http/ngx_http_proxy_m

Let's say you deployed the new feature at the folder `new-feature` of the new Next.js application and set its [`basePath`](https://nextjs

```ts filename="next.config.ts"
import type { NextConfig } from 'next';

const nextConfig: NextConfig = {
  basePath: '/new-feature',
};

export default nextConfig;
```

When deployed, your new feature will be available at `https://my-new-app.vercel.app/`.

You can then use the following nginx configuration to send the traffic for that feature from the legacy server to the new implementation:

```nginx filename="nginx.conf"
server {
    listen 80;
    server_name legacy-server.com www.legacy-server.com;

    location /feature-path-on-legacy-server {
        proxy_pass https://my-new-app.vercel.app/;
    }
}
```

Repeat steps 1 and 2 until all the features have been migrated to Vercel. You can then point your domain to Vercel and remove the legacy

## Troubleshooting

### Maximum number of routes

Vercel has a limit of 1024 routes per deployment for rewrites. If you have more than 1024 routes, you may want to consider creating a cus

### Handling emergencies

If you're facing unexpected outcomes or cannot find an immediate solution for an unexpected behavior with a new feature, you can set up a

For example, with Next.js, you can use the follow [middleware](/docs/edge-middleware) code example:

```ts filename="middleware.ts"
import { NextRequest, NextResponse } from 'next/server';
import { get } from '@vercel/edge-config';

export const config = {
  matcher: ['/'], // URL to match
};

export async function middleware(request: NextRequest) {
  try {
    // Check whether the new version should be shown - isNewVersionActive is a boolean value stored in Edge Config that you can update fr
    const isNewVersionActive = await get<boolean>('isNewVersionActive');

    // If `isNewVersionActive` is false, rewrite to the legacy server URL
    if (!isNewVersionActive) {
      req.nextUrl.pathname = `/legacy-path`;
      return NextResponse.rewrite(req.nextUrl);
    }
  } catch (error) {
    console.error(error);
  }
}
```

[Create an Edge Config](/docs/edge-config/edge-config-dashboard#creating-an-edge-config) and set it to `{ "isNewVersionActive": true }`.

## Session management

When your application is hosted across multiple servers, maintaining [session](https://developer.mozilla.org/docs/Web/HTTP/Session) infor

For example, if your legacy application is served on a different domain than your new application, the HTTP session cookies will not be s

- Using cookies for storing user specific data such as last login time and recent viewed items
- Using cookies for tracking the number of items added to the cart

If you are not currently using a central session store for persisting sessions or are considering moving to one, you can use a [Redis dat

Learn more about [connecting Redis databases through the Marketplace](/docs/redis).

## User group strategies

Minimize risk and perform A/B testing by combining your migration by feature with a user group strategy. You can use [Edge Config](/docs/

- Review our [Edge Config feature flag template](https://vercel.com/templates/next.js/feature-flag-apple-store) for a deeper understandin

- You can also consult our [guide on A/B Testing on Vercel](/kb/guide/ab-testing-on-vercel) for implementing this strategy

## Using functions

Consider using [Vercel Functions](/docs/functions) as you migrate your application.

This allows for the implementation of small, specific, and independent functionality units triggered by events, potentially enhancing fut

## SEO considerations

Prevent the loss of indexed pages, links, and duplicate content when creating rewrites to direct part of your traffic to the new Vercel d

- Write E2E tests to ensure correct setting of canonical tags and robot indexing at each migration step
- Account for existing redirects and rewrites on your legacy server, ensuring they are thoroughly tested during migration
- Maintain the same routes for migrated feature(s) on Vercel

---
title: "Incremental Static Regeneration usage and pricing"
description: "This page outlines information on the limits that are applicable to using Incremental Static Regeneration (ISR), and the co
last_updated: "2026-01-16T02:19:32.222Z"
source: "https://vercel.com/docs/incremental-static-regeneration/limits-and-pricing"
---

# Incremental Static Regeneration usage and pricing

## Pricing

Vercel offers several methods for caching data within Vercel's managed infrastructure. [Incremental Static Regeneration (ISR)](/docs/incr

**ISR Reads and Writes** are priced regionally based on the [Vercel function region(s)](/docs/functions/configuring-functions/region) set

## Usage

The table below shows the metrics for the [**ISR**](/docs/pricing/incremental-static-regeneration) section of the [**Usage** dashboard](/

To view information on managing each resource, select the resource link in the **Metric** column. To jump straight to guidance on optimiz

### Storage

There is no limit on storage for ISR, all the data you write remains cached for the duration you specify. Only you or your team can inval

### Written data

The total amount of Write Units used to durably store new ISR data, measured in 8KB units.

### Read data

The total amount of Read Units used to access the ISR data, measured in 8KB units.

ISR reads and writes are measured in 8 KB units:

- **Read unit**: One read unit equals 8 KB of data read from the ISR cache
- **Write unit**: One write unit equals 8 KB of data written to the ISR cache

## ISR reads and writes price

**ISR Reads and Writes** are priced regionally based on the [Vercel function region(s)](/docs/functions/configuring-functions/region) set

### ISR cache region

The ISR cache region for your deployment is set at build time and is based on the [default Function region](/docs/functions/configuring-f

For best performance, set your default Function region (and hence your ISR cache region) to be close to where your users are. Although th

## Optimizing ISR reads and writes

You are charged based on the volume of data read from and written to the ISR cache, and the regions where reads and writes occur. To optii

- For content that rarely changes, set a longer [time-based revalidation](/docs/incremental-static-regeneration/quickstart#background-rev
- If you have events that trigger data updates, use [on-demand revalidation](/docs/incremental-static-regeneration/quickstart#on-demand-r

When attempting to perform a revalidation, if the content has no changes from the previous version, no ISR write units will be incurred. 

If you are seeing writes, this is because the content has changed. Here's how you can debug unexpected writes:

- Ensure you're not using `new Date()` in the ISR output
- Ensure you're not using `Math.random()` in the ISR output
- Ensure any other code which produces a non-deterministic output is not included in the ISR output

## ISR reads chart

You get charged based on the amount of data read from your ISR cache and the region(s) in which the reads happen.

When viewing your ISR read units chart, you can group by:

- **Projects**: To see the number of read units for each project
- **Region**: To see the number of read units for each region

## ISR writes chart

You get charged based on the amount of ISR write units written to your ISR cache and the region(s) in which the writes happen.

When viewing your ISR writes chart, you can group by sum of units to see a total of all writes across your team's projects.

- **Projects**: To see the number of write units for each project
- **Region**: To see the number of write units for each region

---

title: "Incremental Static Regeneration (ISR)"
description: "Learn how Vercel"
last_updated: "2026-01-16T02:19:32.416Z"
source: "https://vercel.com/docs/incremental-static-regeneration"
---

# Incremental Static Regeneration (ISR)

Incremental Static Regeneration (ISR) allows you to create or update content on your site without redeploying. ISR's main benefits for de

1. **Better Performance:** Static pages can be consistently fast because ISR allows Vercel to cache generated pages in every region on [o
2. **Reduced Backend Load:** ISR helps reduce backend load by using cached content to make fewer requests to your data sources
3. **Faster Builds:** Pages can be generated when requested by a visitor or through an API instead of during the build, speeding up build

ISR is available to applications built with:

- [Next.js](#using-isr-with-next.js)
- [SvelteKit](/docs/frameworks/sveltekit#incremental-static-regeneration-isr)
- [Nuxt](/docs/frameworks/nuxt#incremental-static-regeneration-isr)
- [Astro](/docs/frameworks/astro#incremental-static-regeneration)
- [Gatsby](/docs/frameworks/gatsby#incremental-static-regeneration)
- Or any custom framework solution that implements the [Build Output API](/docs/build-output-api/v3)

## Using ISR with Next.js

> For \['nextjs']:

Next.js will automatically create a Serverless Vercel Function that can revalidate when you use `getStaticProps` with `revalidate`. `getS

> For \['nextjs-app']:

Next.js will automatically create a Serverless Vercel Function that can revalidate when you add `next: { revalidate: 10 }` to the options

The following example demonstrates a Next.js page that uses ISR to render a list of blog posts:

```ts v0="build" filename="pages/blog-posts/index.tsx" framework=nextjs
export async function getStaticProps() {
  const res = await fetch('https://api.vercel.app/blog');
  const posts = await res.json();

  return {
    props: {
      posts,
    },
    revalidate: 10,
  };
}

interface Post {
  title: string;
  id: number;
}

export default function BlogPosts({ posts }: { posts: Post[] }) {
  return (
    <ul>
      {posts.map((post) => (
        <li key={post.id}>{post.title}</li>
      ))}
    </ul>
  );
}
```

```js v0="build" filename="pages/blog-posts/index.jsx" framework=nextjs
export async function getStaticProps() {
  const res = await fetch('https://api.vercel.app/blog');
  const posts = await res.json();

  return {
    props: {
      posts,
    },
    revalidate: 10,
  };
}

export default function BlogPosts({ posts }) {
  return (
    <ul>
      {posts.map((post) => (
        <li key={post.id}>{post.title}</li>
      ))}
    </ul>
  );
}
```

```ts v0="build" filename="app/blog-posts/page.tsx" framework=nextjs-app
export const revalidate = 10; // seconds

interface Post {
  title: string;
  id: number;
}

export default async function Page() {
  const res = await fetch('https://api.vercel.app/blog');
  const posts = (await res.json()) as Post[];
  return (
```

```
    <ul>
      {posts.map((post: Post) => {
        return <li key={post.id}>{post.title}</li>;
      })}
    </ul>
  );
}
```

```js v0="build" filename="app/blog-posts/page.jsx" framework=nextjs-app
export const revalidate = 10; // seconds

export default async function Page() {
  const res = await fetch('https://api.vercel.app/blog');
  const posts = await res.json();

  return (
    <ul>
      {posts.map((post) => {
        return <li key={post.id}>{post.title}</li>;
      })}
    </ul>
  );
}
```

> For \['nextjs-app']:

To learn more about using ISR with Next.js in the App router, such as enabling on-demand revalidation, see [the official Next.js document

> For \['nextjs']:

To learn more about using ISR with Next.js in the Pages router, such as enabling on-demand revalidation, see [the official Next.js docume

## Using ISR with SvelteKit or Nuxt

- See [our dedicated SvelteKit docs](/docs/frameworks/sveltekit#incremental-static-regeneration-isr) to learn how to use ISR with your Sv
- See [our dedicated Nuxt docs](/docs/frameworks/nuxt#incremental-static-regeneration-isr) to use ISR with Nuxt

## Using ISR with the Build Output API

When using the Build Output API, the Serverless Vercel Functions generated for your ISR routes are called [Prerender Functions](/docs/bui

Build Output API Prerender Functions are [Vercel functions](/docs/functions) with accompanying JSON files that describe the Function's ca

## Differences between ISR and `Cache-Control` headers

Both ISR and `Cache-Control` headers help reduce backend load by using cached content to make fewer requests to your data source. However

- **Shared Global Cache:** ISR has **cache shielding** built-in automatically, which helps improve the cache `HIT` ratio. The cache for y
- **300ms Global Purges:** When revalidating (either on-demand or in the background), your ISR route's Vercel Function is re-run, and the
- **Instant Rollbacks:** ISR allows you to roll back instantly and not lose your previously generated pages by persisting them between de
- **Simplified Caching Experience**: ISR abstracts common issues with HTTP-based caching implementations, adds additional features for av

See [our Cache control options docs](/docs/cdn-cache#cache-control-options) to learn more about `Cache-Control` headers.

### ISR vs `Cache-Control` comparison table

## On-demand revalidation limits

On-demand revalidation is scoped to the domain and deployment where it occurs, and doesn't affect sub domains or other deployments.

For example, if you trigger on-demand revalidation for `example-domain.com/example-page`, it won't revalidate the same page served by sub

See [Revalidating across domains](/docs/cdn-cache#revalidating-across-domains) to learn how to get around this limitation.

## Revalidation failure handling

When ISR attempts to revalidate a page, the revalidation request may fail due to network issues, server errors, or invalid responses. Ver

If a revalidation request encounters any of the following conditions, it's considered a failure:

- **Network errors:** Timeouts, connection failures, or other transport-layer issues
- **Invalid HTTP status codes:** Any status code other than 200, 301, 302, 307, 308, 404, or 410
- **Server errors:** Lambda execution failures or runtime errors

When a revalidation failure occurs, Vercel implements a graceful degradation strategy:

1. **Stale content is preserved:** The existing cached version of the page continues to be served to users. Your site remains functional
2. **Short retry window:** The cached page is given a Time-To-Live (TTL) of 30 seconds. This means Vercel will attempt to revalidate the

## ISR pricing

When using ISR with a framework on Vercel, a function is created based on your framework code. This means that you incur usage when the I

- **You incur usage when the function is invoked** – ISR functions are invoked whenever they revalidate in the background or through [on-
- **You incur ISR writes when new content is stored in the ISR cache** – Fresh content returned by ISR functions is persisted to durable
- **You incur Incur ISR reads when content is accessed from the ISR cache** – The content served from the ISR cache when there is an cach
- **You add to your [Fast Origin Transfer](/docs/manage-cdn-usage#fast-origin-transfer) usage**

Explore your [usage top paths](/docs/limits/usage#top-paths) to better understand ISR usage and pricing.

## More resources

- [Quickstart](/docs/incremental-static-regeneration/quickstart)
- [Monitor ISR on Vercel](/docs/observability/monitoring)
- [Next.js Caching](https://nextjs.org/docs/app/building-your-application/data-fetching/caching)

---

title: "Getting started with ISR"
description: "Learn how to use Incremental Static Regeneration (ISR) to regenerate your pages without rebuilding and redeploying your sit
last_updated: "2026-01-16T02:19:32.442Z"
source: "https://vercel.com/docs/incremental-static-regeneration/quickstart"
---

# Getting started with ISR

This guide will help you get started with using Incremental Static Regeneration (ISR) on your project, showing you how to regenerate your

- **Background revalidation** – Regeneration that recurs on an interval
- **On-demand revalidation** – Regeneration that occurs when you send certain API requests to your app

## Background Revalidation

**Background revalidation** allows you to purge the cache for an ISR route automatically on an interval.

> For \["nextjs"]:

When using Next.js with the `pages` router, you can enable ISR by adding a `revalidate` property to the object returned from `getStaticPr

> For \["nextjs-app"]:

When using Next.js with the App Router, you can enable ISR by using the `revalidate` route segment config for a layout or page.

> For \["sveltekit"]:

To deploy a SvelteKit route with ISR, export a config object with an `isr` property. The following example demonstrates a SvelteKit route

> For \["nuxt"]:

To enable ISR in a Nuxt route, add a `routeRules` option to your , as shown in the example below:

```ts filename="apps/example/page.tsx" framework=nextjs-app
export const revalidate = 10; // seconds
```

```js filename="apps/example/page.jsx" framework=nextjs-app
export const revalidate = 10; // seconds
```

```ts filename="pages/example/index.tsx" framework=nextjs
export async function getStaticProps() {
  /* Fetch data here */

  return {
    props: {
      /* Add something to your props */
    },
    revalidate: 10, // Seconds
  };
}
```

```js filename="pages/example/index.jsx" framework=nextjs
export async function getStaticProps() {
  /* Fetch data here */

  return {
    props: {
      /* Add something to your props */
    },
    revalidate: 10, // Seconds
  };
}
```

```ts filename="example-route/+page.server.ts" framework=sveltekit
export const config = {
  isr: {
    expiration: 10,
  },
};
```

```js filename="example-route/+page.server.js" framework=sveltekit
export const config = {
  isr: {
    expiration: 10,
  },
};
```

```ts filename="nuxt.config.ts" framework=nuxt
export default defineNuxtConfig({
  routeRules: {
    // This route will be revalidated
    // every 10 seconds in the background
    '/blog-posts': { isr: 10 },
  },
});
```

```js filename="nuxt.config.js" framework=nuxt
export default defineNuxtConfig({
  routeRules: {
    // This route will be revalidated
    // every 10 seconds in the background
    '/blog-posts': { isr: 10 },
  },
```

});
```

### Example

The following example renders a list of blog posts from a demo site called `jsonplaceholder`, revalidating every 10 seconds or whenever a

> For \['sveltekit']:

First, create a  file that exports your `config` object with `isr` configured and fetches your data:

> For \['sveltekit']:

Then, create a  file that renders the list of blog posts:

> For \['nuxt']:

After enabling ISR in your  file [as described above](#background-revalidation), create an API route that fetches your data:

> For \['nuxt']:

Then, fetch the data and render it in a `.vue` file:

```ts v0="build" filename="pages/blog-posts/index.tsx" framework=nextjs
export async function getStaticProps() {
  const res = await fetch('https://api.vercel.app/blog');
  const posts = await res.json();

  return {
    props: {
      posts,
    },
    revalidate: 10,
  };
}

interface Post {
  title: string;
  id: number;
}

export default function BlogPosts({ posts }: { posts: Post[] }) {
  return (
    <ul>
      {posts.map((post) => (
        <li key={post.id}>{post.title}</li>
      ))}
    </ul>
  );
}
```

```js v0="build" filename="pages/blog-posts/index.jsx" framework=nextjs
export async function getStaticProps() {
  const res = await fetch('https://api.vercel.app/blog');
  const posts = await res.json();

  return {
    props: {
      posts,
    },
    revalidate: 10,
  };
}

export default function BlogPosts({ posts }) {
  return (
    <ul>
      {posts.map((post) => (
        <li key={post.id}>{post.title}</li>
      ))}
    </ul>
  );
}
```

```ts v0="build" filename="app/blog-posts/page.tsx" framework=nextjs-app
export const revalidate = 10; // seconds

interface Post {
  title: string;
  id: number;
}

export default async function Page() {
  const res = await fetch('https://api.vercel.app/blog');
  const posts = (await res.json()) as Post[];
  return (
    <ul>
      {posts.map((post: Post) => {
        return <li key={post.id}>{post.title}</li>;
      })}
    </ul>
  );
}
```

```js v0="build" filename="app/blog-posts/page.jsx" framework=nextjs-app
export const revalidate = 10; // seconds
```
```

```
export default async function Page() {
  const res = await fetch('https://api.vercel.app/blog');
  const posts = await res.json();

  return (
    <ul>
      {posts.map((post) => {
        return <li key={post.id}>{post.title}</li>;
      })}
    </ul>
  );
}
```

To test this code, run the appropriate `dev` command for your framework, and navigate to the `/blog-posts/` route.

You should see a bulleted list of blog posts.

## On-Demand Revalidation

**On-demand revalidation** allows you to purge the cache for an ISR route whenever you want, foregoing the time interval required with ba

> For \['sveltekit']:

To trigger revalidation with SvelteKit:

1. Set an `BYPASS_TOKEN` Environment Variable with a secret value
2. Assign your Environment Variable to the `bypassToken` config option for your route:

3) Send a `GET` or `HEAD` API request to your route with the following header:

```bash
x-prerender-revalidate: bypass_token_here
```

> For \['nuxt']:

To trigger revalidation with Nuxt:

1. Set an `BYPASS_TOKEN` Environment Variable with a secret value
2. Assign your Environment Variable to the `bypassToken` config option in `nitro.config` file:

3) Assign your Environment Variable to the `bypassToken` config option in `nuxt.config` file:

4. Send a `GET` or `HEAD` API request to your route with the following header:

```bash
x-prerender-revalidate: bypass_token_here
```

> For \["nextjs", "nextjs-app"]:

To revalidate a page on demand with Next.js:

1. Create an Environment Variable which will store a revalidation secret
2. Create an API Route that checks for the secret, then triggers revalidation

The following example demonstrates an API route that triggers revalidation if the query paramater `?secret` matches a secret Environment '

```js v0="build" filename="pages/api/revalidate.js" framework=nextjs
export default async function handler(request, response) {
  // Check for secret to confirm this is a valid request
  if (request.query.secret !== process.env.MY_SECRET_TOKEN) {
    return response.status(401).json({ message: 'Invalid token' });
  }

  try {
    // This should be the actual path, not a rewritten path
    // e.g. for "/blog-posts/[slug]" this should be "/blog-posts/1"
    await response.revalidate('/blog-posts');
    return response.json({ revalidated: true });
  } catch (err) {
    // If there was an error, Next.js will continue
    // to show the last successfully generated page
    return response.status(500).send('Error revalidating');
  }
}
```

```ts v0="build" filename="pages/api/revalidate.ts" framework=nextjs
import type { NextApiRequest, NextApiResponse } from 'next';

export default async function handler(
  req: NextApiRequest,
  res: NextApiResponse,
) {
  // Check for secret to confirm this is a valid request
  if (req.query.secret !== process.env.MY_SECRET_TOKEN) {
    return res.status(401).json({ message: 'Invalid token' });
  }

  try {
    // This should be the actual path, not a rewritten path
    // e.g. for "/blog-posts/[slug]" this should be "/blog-posts/1"
    await res.revalidate('/blog-posts');
    return res.json({ revalidated: true });
  } catch (err) {
    // If there was an error, Next.js will continue
    // to show the last successfully generated page
    return res.status(500).send('Error revalidating');
```

  }
}
```

```ts v0="build" filename="app/api/revalidate/route.ts" framework=nextjs-app
import { revalidatePath } from 'next/cache';

export async function GET(request: Request) {
  const { searchParams } = new URL(request.url);
  if (searchParams.get('secret') !== process.env.MY_SECRET_TOKEN) {
    return new Response('Invalid credentials', {
      status: 401,
    });
  }

  revalidatePath('/blog-posts');

  return Response.json({
    revalidated: true,
    now: Date.now(),
  });
}
```

```js v0="build" filename="app/api/revalidate/route.js" framework=nextjs-app
import { revalidatePath } from 'next/cache';

export async function GET(request) {
  const { searchParams } = new URL(request.url);
  if (searchParams.get('secret') !== process.env.MY_SECRET_TOKEN) {
    return new Response('Invalid credentials', {
      status: 401,
    });
  }

  revalidatePath('/blog-posts');

  return Response.json({
    revalidated: true,
    now: Date.now(),
  });
}
```

> For \["nextjs"]:

> For \["nextjs", "nextjs-app", "sveltekit"]:

See the [background revalidation section above](#background-revalidation) for a full ISR example.

## Templates

## Next steps

Now that you have set up ISR, you can explore the following:

- [Explore usage and pricing](/docs/incremental-static-regeneration/limits-and-pricing)
- [Monitor ISR on Vercel through Observability](/docs/observability/monitoring)


---
title: "Performing an Instant Rollback on a Deployment"
description: "Learn how to perform an Instant Rollback on your production deployments and quickly roll back to a previously deployed prod
last_updated: "2026-01-16T02:19:32.304Z"
source: "https://vercel.com/docs/instant-rollback"
---

# Performing an Instant Rollback on a Deployment

Vercel provides Instant Rollback as a way to quickly revert to a previous production deployment. This can be useful in situations that re

- The rolled back deployment is treated as a restored version of a previous deployment
- The configuration used for the rolled back deployment will potentially become stale
- The environment variables will not be updated if you change them in the project settings and will roll back to a previous build
- If the project uses [cron jobs](/docs/cron-jobs), they will be reverted to the state of the rolled back deployment

For teams on a Pro or Enterprise plan, all deployments previously aliased to a production domain are [eligible to roll back](#eligible-de

## How to roll back deployments

To initiate an Instant Rollback from the Vercel dashboard:

- ### Select your project
  On the project's overview page, you will see the [Production Deployment tile](# "Production Deployment tile"). From there, click **Inst

- ### Select the deployment to roll back to
  After selecting Instant Rollback, you'll see an dialog that displays your current production deployment and the eligible deployments th

  If you're on the Pro or Enterprise plans, you can also click the **Choose another deployment** button to display a list of all [eligibl

  Select the deployment that you'd like to roll back to and click **Continue**.

- ### Verify the information
  Once you've selected the deployment to roll back to, verify the roll back information:
  - The names of the domains and sub-domains that will be rolled back
  - There are no change in Environment Variables, and they will remain in their original state
  - A reminder about the changing behavior of external APIs, databases, and CMSes used in the current or previous deployments

- ### Confirm the rollback
  Once you have verified the details, click the **Confirm Rollback** button. At this point, you'll get confirmation details about the suc
```

- ### Successful rollback
  The rollback happens instantaneously and Vercel will point your domain and sub-domain back to the selected deployment. The production d

  When using Instant Rollback, Vercel will turn off [auto-assignment of production domains](/docs/deployments/promoting-a-deployment#stag

  To replace the rolled back deployment, either turn on the **Auto-assign Custom Production Domains** toggle from the [**Production Envir

> **💡 Note:** 

### Accessing Instant Rollback from Deployments tab

You can also roll back from the main **Deployments** tab in your dashboard. Filtering the deployments list by `main` is recommended to vi

Click the vertical ellipses (⋮) next to the deployment row and select the **Instant Rollback** option from the context menu.

## Who can roll back deployments?

- **Hobby** plan: On the hobby plan you can roll back to the previous deployment
- **Pro** and **Enterprise** plan: Owners and Members on these plans can roll back to any [eligible deployment](#eligible-deployments).

## Eligible deployments

Deployments previously aliased to a production domain are eligible for Instant Rollback. Deployments that have never been aliased to prod

## Comparing Instant Rollback and manual promote options

To compare the manual promotion options, see [Manually promoting to Production](/docs/deployments/promoting-a-deployment).

---
title: "Vercel Agility CMS Integration"
description: "Learn how to integrate Agility CMS with Vercel. Follow our tutorial to deploy the Agility CMS template or install the integ
last_updated: "2026-01-16T02:19:32.362Z"
source: "https://vercel.com/docs/integrations/cms/agility-cms"
---

# Vercel Agility CMS Integration

Agility CMS is a headless content management system designed for flexibility and scalability. It allows developers to create and manage d

## Getting started

To get started with the Agility CMS on Vercel deploy the template below:

Or, follow the steps below to install the integration:

- ### Install the Vercel CLI
  To pull in environment variables from  to your Vercel project, you need to install the [Vercel CLI](/docs/cli). Run the following comma
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i vercel
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i vercel
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i vercel
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i vercel
      ```
    </Code>
  </CodeBlock>

- ### Install your CMS integration
  Navigate to the  and follow the steps to install the integration.

- ### Pull in environment variables
  Once you've installed the  integration, you can pull in environment variables from  to your Vercel project. In your terminal, run:
  ```bash
  vercel env pull
  ```

See your installed CMSs documentation for next steps on how to use the integration.

---
title: "Vercel ButterCMS Integration"
description: "Learn how to integrate ButterCMS with Vercel. Follow our tutorial to set up the ButterCMS template on Vercel and manage con
last_updated: "2026-01-16T02:19:32.377Z"
source: "https://vercel.com/docs/integrations/cms/butter-cms"
---

# Vercel ButterCMS Integration

ButterCMS is a headless content management system that enables developers to manage and deliver content through an API.

## Getting started

To get started with the ButterCMS on Vercel deploy the template below:

Or, follow the steps below to install the integration:

- ### Install the Vercel CLI
  To pull in environment variables from  to your Vercel project, you need to install the [Vercel CLI](/docs/cli). Run the following comma
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i vercel
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i vercel
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i vercel
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i vercel
      ```
    </Code>
  </CodeBlock>

- ### Install your CMS integration
  Navigate to the  and follow the steps to install the integration.

- ### Pull in environment variables
  Once you've installed the  integration, you can pull in environment variables from  to your Vercel project. In your terminal, run:
  ```bash
  vercel env pull
  ```

See your installed CMSs documentation for next steps on how to use the integration.


---
title: "Vercel and Contentful Integration"
description: "Integrate Vercel with Contentful to deploy your content."
last_updated: "2026-01-16T02:19:32.334Z"
source: "https://vercel.com/docs/integrations/cms/contentful"
---

# Vercel and Contentful Integration

[Contentful](https://contentful.com/) is a headless CMS that allows you to separate the content management and presentation layers of you

This quickstart guide uses the [Vercel Contentful integration](/integrations/contentful) to allow streamlined access between your Content

If you already have a Vercel deployment and a Contentful account, you should [install the Contentful Integration](/integrations/contentfu

- Getting your [Space ID](#retrieve-your-contentful-space-id) and [Content Management API Token](#create-a-content-management-api-token)
- [Importing your content model](#import-the-content-model)
- [Adding your Contentful environment variables](#add-environment-variables) to your Vercel project

## Getting started

To help you get started, we built a [template](https://vercel.com/templates/next.js/nextjs-blog-preview-mode) using Next.js, Contentful,

You can either deploy the template above to Vercel with one click, or use the steps below to clone it to your machine and deploy it local

- ### Clone the repository
  You can clone the repo using the following command:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>

- ### Create a Contentful Account
  Next, create a new account on [Contentful](https://contentful.com/) and make an empty "space". This is where your content lives. We als

  If you have an existing account and space, you can use it with the rest of these steps.

- ### Retrieve your Contentful Space ID
  The Vercel integration uses your Contentful Space ID to communicate with Contentful. To find this, navigate to your Contentful dashboar

- ### Create a Content Management API token
  You will also need to create a Content Management API token for Vercel to communicate back and forth with the Contentful API. You can g

  Click on **Generate personal token** and a modal will pop up. Give your token a name and click on **Generate**.

  > **💡 Note:** Avoid sharing this token because it allows both read and write access to your
  > Contentful space. Once the token is generated copy the key and save remotely
  > as it will not be accessible later on. If lost, a new one must be created.

- ### Import the Content Model
  Use your Space ID and Content Management Token in the command below to import the pre-made content model into your space using our setu
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>

## Adding Content in Contentful

Now that you've created your space in Contentful, add some content!

- ### Publish Contentful entries
  You'll notice the new author and post entries for the example we've provided. Publish each entry to make this fully live.

- ### Retrieve your Contentful Secrets
  Now, let's save the Space ID and token from earlier to add as Environment Variables for running locally. Create a new `.env.local` file
  ```shell filename="terminal"
  CONTENTFUL_SPACE_ID='your-space-id'
  CONTENTFUL_ACCESS_TOKEN='your-content-api-token'
  ```

- ### Start your application
  You can now start your application with the following comment:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>
  Your project should now be running on `http://localhost:3000`.

## How it works

Next.js is designed to integrate with any data source of your choice, including Content Management Systems. Contentful provides a helpful

```js
async function fetchGraphQL(query) {
  return fetch(
    `https://graphql.contentful.com/content/v1/repos/${process.env.CONTENTFUL_SPACE_ID}`,
    {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
        Authorization: `Bearer ${process.env.CONTENTFUL_ACCESS_TOKEN}`,
      },
      body: JSON.stringify({ query }),
    },
  ).then((response) => response.json());
}
```

This code allows you to fetch data on the server from your Contentful instance. Each space inside Contentful has its own ID (e.g. `CONTEN

This allows you to use secure values you don't want to commit to git, which are only evaluated on the server (e.g. `CONTENTFUL_ACCESS_TOK

## Deploying to Vercel

Now that you have your application wired up to Contentful, you can deploy it to Vercel to get your site online. You can either use the Ve

- ### Publish your code to Git
  Push your code to your git repository (e.g. GitHub, GitLab, or BitBucket).
  ```shell filename="terminal"
  ```

```
  git init
  git add .
  git commit -m "Initial commit"
  git remote add origin
  git push -u origin master
  ```

- ### Import your project into Vercel
  Log in to your Vercel account (or create one) and import your project into Vercel using the [import flow](https://vercel.com/new).

  Vercel will detect that you are using Next.js and will enable the correct settings for your deployment.

- ### Add Environment Variables
  Add the `CONTENTFUL_SPACE_ID` and `CONTENTFUL_ACCESS_TOKEN` Environment Variables from your `.env.local` file by copying and pasting it
  ```shell filename="terminal"
  CONTENTFUL_SPACE_ID='your-space-id'
  CONTENTFUL_ACCESS_TOKEN='your-content-api-token'
  ```
  Click "Deploy" and your application will be live on Vercel!

### Content Link

Content Link enables you to edit content on websites using headless CMSs by providing links on elements that match a content model in the

You can enable Content Link on a preview deployment by selecting  **Edit Mode** in the [Vercel Toolbar](/docs/vercel-toolbar) menu.

The corresponding model in the CMS determines an editable field. You can hover over an element to display a link in the top-right corner

You don't need any additional configuration or code changes on the page to use this feature.

To implement Content Link in your project, follow the steps in [Contentful's documentation](https://www.contentful.com/developers/docs/to


---
title: "Vercel DatoCMS Integration"
description: "Learn how to integrate DatoCMS with Vercel. Follow our step-by-step tutorial to set up and manage your digital content seam
last_updated: "2026-01-16T02:19:32.384Z"
source: "https://vercel.com/docs/integrations/cms/dato-cms"
---

# Vercel DatoCMS Integration

DatoCMS is a headless content management system designed for creating and managing digital content with flexibility. It provides a powerf

## Getting started

To get started with DatoCMS on Vercel, follow the steps below to install the integration:

- ### Install the Vercel CLI
  To pull in environment variables from  to your Vercel project, you need to install the [Vercel CLI](/docs/cli). Run the following comma
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i vercel
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i vercel
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i vercel
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i vercel
      ```
    </Code>
  </CodeBlock>

- ### Install your CMS integration
  Navigate to the  and follow the steps to install the integration.

- ### Pull in environment variables
  Once you've installed the  integration, you can pull in environment variables from  to your Vercel project. In your terminal, run:
  ```bash
  vercel env pull
  ```

See your installed CMSs documentation for next steps on how to use the integration.

### Content Link

Content Link enables you to edit content on websites using headless CMSs by providing links on elements that match a content model in the

You can enable Content Link on a preview deployment by selecting  **Edit Mode** in the [Vercel Toolbar](/docs/vercel-toolbar) menu.

The corresponding model in the CMS determines an editable field. You can hover over an element to display a link in the top-right corner

You don't need any additional configuration or code changes on the page to use this feature.


---
title: "Vercel Formspree Integration"
description: "Learn how to integrate Formspree with Vercel. Follow our tutorial to set up Formspree and manage form submissions on your s
last_updated: "2026-01-16T02:19:32.390Z"
source: "https://vercel.com/docs/integrations/cms/formspree"

---

# Vercel Formspree Integration

Formspree is a form backend platform that handles form submissions on static websites. It allows developers to collect and manage form da

## Getting started

To get started with Formspree on Vercel, follow the steps below to install the integration:

- ### Install the Vercel CLI
  To pull in environment variables from  to your Vercel project, you need to install the [Vercel CLI](/docs/cli). Run the following comma
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i vercel
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i vercel
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i vercel
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i vercel
      ```
    </Code>
  </CodeBlock>

- ### Install your CMS integration
  Navigate to the  and follow the steps to install the integration.

- ### Pull in environment variables
  Once you've installed the  integration, you can pull in environment variables from  to your Vercel project. In your terminal, run:
  ```bash
  vercel env pull
  ```

See your installed CMSs documentation for next steps on how to use the integration.

---
title: "Vercel Makeswift Integration"
description: "Learn how to integrate Makeswift with Vercel. Makeswift is a no-code website builder designed for creating and managing Rea
last_updated: "2026-01-16T02:19:32.396Z"
source: "https://vercel.com/docs/integrations/cms/makeswift"
---

# Vercel Makeswift Integration

Makeswift is a no-code website builder designed for creating and managing React websites. It offers a drag-and-drop interface that allows

## Getting started

To get started with the Makeswift on Vercel deploy the template below:

Or, follow the steps below to install the integration:

- ### Install the Vercel CLI
  To pull in environment variables from  to your Vercel project, you need to install the [Vercel CLI](/docs/cli). Run the following comma
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i vercel
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i vercel
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i vercel
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i vercel
      ```
    </Code>
  </CodeBlock>

- ### Install your CMS integration
  Navigate to the  and follow the steps to install the integration.

- ### Pull in environment variables
  Once you've installed the  integration, you can pull in environment variables from  to your Vercel project. In your terminal, run:
  ```bash
  vercel env pull
  ```

See your installed CMSs documentation for next steps on how to use the integration.

---
title: "Vercel CMS Integrations"
description: "Learn how to integrate Vercel with CMS platforms, including Contentful, Sanity, and Sitecore XM Cloud."
last_updated: "2026-01-16T02:19:32.405Z"
source: "https://vercel.com/docs/integrations/cms"
---

# Vercel CMS Integrations

Vercel Content Management System (CMS) Integrations allow you to connect your projects with CMS platforms, including [Contentful](/docs/i

You can use the following methods to integrate your CMS with Vercel:

- [**Environment variable import**](#environment-variable-import): Quickly setup your Vercel project with environment variables from your
- [**Edit Mode through the Vercel Toolbar**](#edit-mode-with-the-vercel-toolbar): Visualize content from your CMS within a Vercel deploym
  - [**Content Link**](/docs/edit-mode#content-link): Lets you visualize content models from your CMS within a Vercel deployments and edi
- [**Deploy changes from CMS**](#deploy-changes-from-cms): Connect and deploy content from your CMS to your Vercel site

## Environment variable import

The most common way to setup a CMS with Vercel is by installing an integration through the . This method allows you to quickly setup your

Once a CMS has been installed, and a project linked you can pull in environment variables from the CMS to your Vercel project using the ['

- ### Install the Vercel CLI
  To pull in environment variables from  to your Vercel project, you need to install the [Vercel CLI](/docs/cli). Run the following comma
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i vercel
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i vercel
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i vercel
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i vercel
      ```
    </Code>
  </CodeBlock>

- ### Install your CMS integration
  Navigate to the  and follow the steps to install the integration.

- ### Pull in environment variables
  Once you've installed the  integration, you can pull in environment variables from  to your Vercel project. In your terminal, run:
  ```bash
  vercel env pull
  ```

See your installed CMSs documentation for next steps on how to use the integration.

## Edit mode with the Vercel Toolbar

To access Edit Mode:

1. Ensure you're logged into the [Vercel Toolbar](/docs/vercel-toolbar) with your Vercel account.
2. Navigate to a page with editable content. The  **Edit Mode** option will only appear in the [Vercel Toolbar](/docs/vercel-toolbar) men
3. Select the  **Edit Mode** option in the toolbar menu. This will highlight the editable fields as [Content Links](/docs/edit-mode#conte

The following CMS integrations support Content Link:

-
-
-
-
-
-
-
-

See the [Edit Mode documentation](/docs/edit-mode) for information on setup and configuration.

## Draft mode through the Vercel Toolbar

Draft mode allows you to view unpublished content from your CMS within a Vercel preview, and works with Next.js and SvelteKit. See the [D

## Deploy changes from CMS

This method is generally setup through webhooks or APIs that trigger a deployment when content is updated in the CMS. See your CMSs docum

## Featured CMS integrations

- [Agility CMS](/docs/integrations/cms/agility-cms)
- [DatoCMS](/docs/integrations/cms/dato-cms)
- [ButterCMS](/docs/integrations/cms/butter-cms)
- [Formspree](/docs/integrations/cms/formspree)
- [Makeswift](/docs/integrations/cms/makeswift)
- [Sanity](/docs/integrations/cms/sanity)
- [Contentful](/docs/integrations/cms/contentful)
- [Sitecore XM Cloud](/docs/integrations/cms/sitecore)

---
title: "Vercel Sanity Integration"
description: "Learn how to integrate Sanity with Vercel. Follow our tutorial to deploy the Sanity template or install the integration for
last_updated: "2026-01-16T02:19:32.449Z"
source: "https://vercel.com/docs/integrations/cms/sanity"
---

# Vercel Sanity Integration

Sanity is a headless content management system that provides real-time collaboration and structured content management. It offers a highly

## Getting started

To get started with the Sanity on Vercel deploy the template below:

Or, follow the steps below to install the integration:

- ### Install the Vercel CLI
  To pull in environment variables from  to your Vercel project, you need to install the [Vercel CLI](/docs/cli). Run the following comma
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i vercel
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i vercel
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i vercel
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i vercel
      ```
    </Code>
  </CodeBlock>

- ### Install your CMS integration
  Navigate to the  and follow the steps to install the integration.

- ### Pull in environment variables
  Once you've installed the  integration, you can pull in environment variables from  to your Vercel project. In your terminal, run:
  ```bash
  vercel env pull
  ```

See your installed CMSs documentation for next steps on how to use the integration.

### Content Link

Content Link enables you to edit content on websites using headless CMSs by providing links on elements that match a content model in the

You can enable Content Link on a preview deployment by selecting  **Edit Mode** in the [Vercel Toolbar](/docs/vercel-toolbar) menu.

The corresponding model in the CMS determines an editable field. You can hover over an element to display a link in the top-right corner o

You don't need any additional configuration or code changes on the page to use this feature.


---
title: "Vercel and Sitecore XM Cloud Integration"
description: "Integrate Vercel with Sitecore XM Cloud to deploy your content."
last_updated: "2026-01-16T02:19:32.483Z"
source: "https://vercel.com/docs/integrations/cms/sitecore"
---

# Vercel and Sitecore XM Cloud Integration

[Sitecore XM Cloud](https://www.sitecore.com/products/xm-cloud) is a CMS platform designed for both developers and marketers. It utilizes

This guide outlines the steps to integrate a headless JavaScript application on Vercel with Sitecore XM Cloud. In this guide, you will le

The key parts you will learn from this guide are:

1. Configuring the GraphQL endpoint for content retrieval from Sitecore XM Cloud
2. Utilizing the Sitecore Next.js for JSS library for content integration
3. Setting up environment variables in Vercel for Sitecore API key, GraphQL endpoint, and JSS app name

## Setting up an XM Cloud project, environment, and website

- ### Access XM Cloud Deploy app
  Log in to your XM Cloud Deploy app account.

- ### Initiate project creation
  Navigate to the **Projects** page and select **Create project**.

- ### Select project foundation
  In the **Create new project** dialog, select **Start from the XM Cloud starter foundation**. Proceed by selecting **Next**.

- ### Select starter template
  Select the XM Cloud Foundation starter template and select **Next**.

- ### Name your project
  Provide a name for your project in the **Project name** field and select **Next**.

- ### Select source control provider
  Choose your source control provider and select **Next**.

- ### Set up source control connection
  If you haven't already set up a connection, create a new source control connection and follow the instructions provided by your source

- ### Specify repository name
  In the **Repository name** field, provide a unique name for your new repository and select **Next**.

- ### Configure environment details
  - Specify the environment name in the **Environment name** field
  - Determine if the environment is a production environment using the **Production environment** drop-down menu
  - Decide if you want automatic deployments upon commits to the linked repository branch using the **Trigger deployment on commit to bra

- ### Finalize setup
  Select **Create and deploy**.

- ### Create a new website
  - When the deployment finishes, select **Go to XM Cloud**
  * Under Sites, select **Create Website**
  - Select **Basic Site**
  * Enter a name for your site in the **Site name** field
  * Select **Create website**

- ### Publish the site
  - Select the **Open in Pages** option on the newly created website
  * Select **Publish** > **Publish item with all sub-items**

## Creating a Next.js JSS application

To help get you started, we built a [template](https://vercel.com/templates/next.js/sitecore-starter) using Sitecore JSS for Next.js with

Sitecore also offers a [JSS app initializer](https://doc.sitecore.com/xmc/en/developers/xm-cloud/the-jss-app-initializer.html) and templa

You can either deploy the template above to Vercel with one-click, or use the steps below to clone it to your machine and deploy it local

- ### Clone the repository
  You can clone the repo using the following command:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>

- ### Retrieve your API key, GraphQL endpoint, and JSS app name
  Next, navigate to your newly created XM Cloud site under **Sites** and select **Settings**.

  Under the **Developer Settings** tab select **Generate API Key**.

  Save the `SITECORE_API_KEY`, `JSS_APP_NAME`, and `GRAPH_QL_ENDPOINT` values – you'll need it for the next step.

- ### Configure your Next.js JSS application
  Next, add the `JSS_APP_NAME`, `GRAPH_QL_ENDPOINT` , `SITECORE_API_KEY`, and `SITECORE_API_HOST` values as environment variables for run
  ```shell filename=".env.local"
  JSS_APP_NAME='your-jss-app-name'
  GRAPH_QL_ENDPOINT='your-graphql-endpoint'
  SITECORE_API_KEY='your-sitecore-api-key'
  SITECORE_API_HOST='host-from-endpoint'
  ```

- ### Start your application
  You can now start your application with the following command:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```

    </Code>
  </CodeBlock>

## How it works

Sitecore XM Cloud offers a GraphQL endpoint for its sites, serving as the primary mechanism for both retrieving and updating content. The

Through this integration, content editors can log into XM Cloud to not only modify content but also adjust the composition of pages.

The frontend application hosted on Vercel establishes a connection to Sitecore XM Cloud using the `GRAPH_QL_ENDPOINT` to determine the da

With these components in place, developers can seamlessly integrate content from Sitecore XM Cloud into a Next.js application on Vercel.

> **💡 Note:** Vercel Deployment Protection is enabled for new projects by
> [default](/changelog/deployment-protection-is-now-enabled-by-default-for-new-projects)
> which limits access to preview and production URLs. This may impact Sitecore
> Experience Editor and Pages functionality. Refer to Deployment Protection
> [documentation](/docs/security/deployment-protection) and Sitecore XM Cloud
> [documentation](https://doc.sitecore.com/xmc/en/developers/xm-cloud/use-vercel-s-deployment-protection-feature-with-jss-apps.html)
> for more details and integration steps.

## Deploying to Vercel

- ### Push to Git
  Ensure your integrated application code is pushed to your git repository.
  ```shell filename="terminal"
  git init
  git add .
  git commit -m "Initial commit"
  git remote add origin [repository url]
  git push -u origin main
  ```

- ### Import to Vercel
  Log in to your Vercel account (or create one) and import your project into Vercel using the [import flow](https://vercel.com/new).

- ### Configure environment variables
  Add the `FETCH_WITH`, `JSS_APP_NAME`, `GRAPH_QL_ENDPOINT` , `SITECORE_API_KEY`, and `SITECORE_API_HOST` environment variables to the **
  ```shell filename=".env.local"
  JSS_APP_NAME='your-jss-app-name'
  GRAPH_QL_ENDPOINT='your-graphql-endpoint'
  SITECORE_API_KEY='your-sitecore-api-key'
  SITECORE_API_HOST='host-from-endpoint'
  FETCH_WITH='GraphQL'
  ```
  Select "Deploy" and your application will be live on Vercel!

---
title: "Integration Approval Checklist"
description: "The integration approval checklist is used ensure all necessary steps have been taken for a great integration experience."
last_updated: "2026-01-16T02:19:32.498Z"
source: "https://vercel.com/docs/integrations/create-integration/approval-checklist"
---

# Integration Approval Checklist

Use this checklist to ensure all necessary steps have been taken for a great integration experience to get listed on the .
Make sure you read the  before you start.

## Marketplace listing

Navigate to `/integrations/:slug` to view the listing for the integration.

**Examples:**

-
-

## Overview and instructions

## Installation flow

From clicking the install button, a wizard should pop up, guiding you through the setup process.

### Deploy button flow

Using  allows users to install an integration together with an example repository on GitHub.

## Integration is installed successfully

After we have installed an integration (through the Marketplace), you should be presented with the details of your installation.

---
title: "Manage Billing and Refunds for Integrations"
description: "Learn how billing works for native integrations, including invoice lifecycle, pricing models, and refunds."
last_updated: "2026-01-16T02:19:32.517Z"
source: "https://vercel.com/docs/integrations/create-integration/billing"
---

# Manage Billing and Refunds for Integrations

When a Vercel user installs your native integration, you manage billing through the [Vercel API billing endpoints](/docs/integrations/cre

## Billing models

You can choose between two billing models:

- **Installation-level billing**: Charges apply to the entire installation. A single billing plan covers all resources provisioned under

- **Resource-level billing**: Charges are scoped to individual products or resources. Each resource can have its own billing plan.

You determine which model to use. You can only submit one invoice per resource per billing period, but a single invoice can include multi

## Billing periods and cycles

You control the billing cycle through the `period` field in your API calls. There's no required day of the month for billing cycles to al

Vercel users can configure a different payment method for each integration installation, independent of their Vercel plan payment method

## Invoice lifecycle

Invoices move through several states as they're processed:

### Invoice states

| State | Description |
| --- | --- |
| **pending** | Default state after you submit an invoice. Vercel queues it for immediate processing. |
| **scheduled** | Queued for future processing based on the billing plan's timing (at signup, period start, or period end). |
| **invoiced** | Vercel processed and sent the invoice to the Vercel user. |
| **paid** | Vercel received payment successfully. |
| **notpaid** | Payment failed on first attempt. Vercel continues retrying up to 9 times while the invoice remains in this state. |
| **refunded** | Vercel fully or partially refunded the invoice. |

> **💡 Note:** When an invoice enters `notpaid` status, Vercel does not automatically
> restrict access to deployments, teams, or products. The
> `marketplace.invoice.notpaid` webhook fires on each failed payment attempt,
> not just the final one. Since Vercel retries payment up to 9 times, you may
> receive multiple webhooks before payment eventually succeeds. Wait at least 15
> days before taking any destructive actions like deleting databases. In the
> meantime, you may choose to degrade service or pause fulfillment (for example,
> stop issuing tokens) until payment succeeds.

## Line items and pricing structures

You have flexibility in how you structure charges. A single invoice can include multiple line items covering:

- **Flat fees**: Fixed monthly or periodic charges
- **Usage-based charges**: Costs calculated from actual resource consumption
- **Tiered pricing**: Different rate tiers (for example, tier 1 usage at one rate, tier 2 at another)

Each line item can specify a unit, quantity, rate, and detailed description. This gives Vercel users a clear breakdown of charges.

We recommend consolidating all resource billing under a single invoice and keeping resources on the same billing cycle. This reduces the

## Technical requirements

When working with billing data:

- **Decimal precision**: All monetary values use 2 decimal places
- **Minimum threshold**: Vercel won't send invoices totaling less than $0.50. You should still submit billing data for transparency so Ve

## Submitting invoices

To bill customers, call the [Vercel billing API endpoints](/docs/integrations/create-integration/marketplace-api/reference#marketplace). A

### Send interim billing data

Call the [Submit Billing Data](/docs/integrations/create-integration/marketplace-api/reference#vercel/submit-billing-data) endpoint (`POST

This data is for display purposes only, helping Vercel users understand their expected charges throughout the billing period. Vercel does

The following example shows a request with billing items and usage metrics:

```bash
curl -X POST "https://api.vercel.com/v1/installations/{integrationConfigurationId}/billing" \
  -H "Authorization: Bearer {access_token}" \
  -H "Content-Type: application/json" \
  -d '{
    "timestamp": "2025-01-15T12:00:00Z",
    "eod": "2025-01-15T00:00:00Z",
    "period": {
      "start": "2025-01-01T00:00:00Z",
      "end": "2025-02-01T00:00:00Z"
    },
    "billing": {
      "items": [
        {
          "billingPlanId": "plan_pro",
          "resourceId": "db_abc123",
          "name": "Pro Plan",
          "price": "29.00",
          "quantity": 1,
          "units": "month",
          "total": "29.00"
        }
      ]
    },
    "usage": [
      {
        "resourceId": "db_abc123",
        "name": "Storage",
        "type": "total",
        "units": "GB",
        "dayValue": 5.2,
        "periodValue": 5.2
      }
    ]
  }'
```

```

> **💡 Note:** * **period.start / period.end**: The full billing period (for example, `2025-01-01` to `2025-02-01` for a monthly cycle)
> * **eod**: The end-of-day timestamp for this data snapshot, representing a single day within the billing period
> * **usage values**: Submit running totals for the entire period, not incremental usage since your last report. Vercel uses the latest v

### Submit an invoice

At the end of a billing period, call the [Submit Invoice endpoint](/docs/integrations/create-integration/marketplace-api/reference#vercel

The following example shows a request with multiple line items:

```bash
curl -X POST "https://api.vercel.com/v1/installations/{integrationConfigurationId}/billing/invoices" \
  -H "Authorization: Bearer {access_token}" \
  -H "Content-Type: application/json" \
  -d '{
    "externalId": "inv_2025_01_abc123",
    "invoiceDate": "2025-02-01T00:00:00Z",
    "period": {
      "start": "2025-01-01T00:00:00Z",
      "end": "2025-02-01T00:00:00Z"
    },
    "items": [
      {
        "billingPlanId": "plan_pro",
        "resourceId": "db_abc123",
        "name": "Pro Plan - January 2025",
        "price": "29.00",
        "quantity": 1,
        "units": "month",
        "total": "29.00"
      },
      {
        "billingPlanId": "plan_pro",
        "resourceId": "db_abc123",
        "name": "Additional Storage",
        "details": "5.2 GB over included 1 GB",
        "price": "0.50",
        "quantity": 4.2,
        "units": "GB",
        "total": "2.10"
      }
    ]
  }'
```

We recommend including an `externalId` in your invoice requests. This lets you tie invoices to your internal billing records for easier r

The response includes an `invoiceId` you can use to track status or request refunds.

### Track invoice status

To check invoice status, call the Get Invoice endpoint (`GET /v1/installations/{integrationConfigurationId}/billing/invoices/{invoiceId}`

## Testing with test mode

You can use test mode to validate your billing integration before going live. Test mode uses the `test` object in the Submit Invoice API v

- `validate: true`: Runs full validation including date checks, item validation, discount validation, and duplicate detection
- `validate: false`: Skips these validations

Outside of test mode, Vercel always runs validation and you cannot override it.

> **💡 Note:** Test-mode invoices don't appear in the Integration Console or Dashboard. This
> is because test invoices bypass the backend billing processes where invoices
> are normally retrieved for display.

To test with live payment methods during the pre-launch phase:

1. Remove the `test` object from your Submit Invoice calls
2. Submit the invoice
3. Wait for the `marketplace.invoice.created` and `marketplace.invoice.paid` webhooks
4. Issue a refund using the Invoice Actions API

## Refunds and credit notes

When you request a refund, Vercel handles it as follows:

1. If Vercel hasn't charged the invoice yet, it cancels the invoice
2. If Vercel already charged the invoice, it attempts to refund the original payment method
3. If the payment method isn't working, Vercel creates a support ticket
4. If anything goes wrong with the refund attempt, Vercel creates a support ticket

For invoices in `notpaid` status, a refund request will succeed and move the status to `refund_requested`, then to `refunded` once the fu

### Refunds after installation deletion

With installation-level billing, the installation goes through finalization after deletion. This gives you time to calculate any remainin

1. **Open invoices exist**: Vercel blocks finalization until invoices are settled. You can refund these invoices during this time.
2. **Finalization window**: By default, you have 24 hours after deletion to submit any final invoices. If you submit invoices during this
3. **Installation finalized**: Refunds must be processed manually through Vercel customer support.

## Tax and VAT

Vercel handles all taxation since Vercel issues the invoices. You only submit raw service charges to the billing APIs. You don't need to

## Invoice visibility and access
```

Only Vercel users with **Owner** or **Billing** roles can view invoices for your integration. They can view their invoices by:

1. Going to the **Integrations** tab in their Vercel [dashboard](/dashboard)
2. Selecting **Manage** next to your integration
3. Navigating to the **Invoices** section

---
title: "Deployment integration actions"
description: "These actions allow integration providers to set up automated tasks with Vercel deployments."
last_updated: "2026-01-16T02:19:32.526Z"
source: "https://vercel.com/docs/integrations/create-integration/deployment-integration-action"
---

# Deployment integration actions

With deployment integration actions, integration providers can enable [integration resource](/docs/integrations/create-integration/native

For example, you can use deployment integration actions with the checks API to [create integrations](/docs/checks#build-your-checks-integ

## How deployment actions work

1. Action declaration:
   - An integration [product](/docs/integrations/create-integration/native-integration#resources) declares deployment actions with an ID,
   - Actions can specify configuration options that integration users can modify.
   - Actions can include suggestions for default actions to run such as "this action should be run on previews".

2. Project configuration:
   - When a resource is connected to a project, integration users select which actions should be triggered during deployments.
   - Integration users are also presented with suggestions on what actions to run if these were configured in the action declaration.

3. Deployment execution:
   - When a deployment is created, the configured actions are registered on the deployment.
   - The registered actions trigger the `deployment.integration.action.start` webhook.
   - If a deployment is canceled, the `deployment.integration.action.cancel` webhook is triggered.

4. Resource-side processing:
   - The integration provider processes the webhook, executing the necessary resource-side actions such as creating a database branch.
   - During the processing of these actions, the build is blocked and the deployment set in a provisioning state.
   - Once complete, the integration provider updates the action status.

5. Deployment unblock:
   - Vercel validates the completed action, updates environment variables, and unblocks the deployment.

## Creating deployment actions

As an integration provider, to allow your integration users to add deployment actions to an installed native integration, follow these st

- ### Declare deployment actions
  Declare the deployment actions for your native integration product.
  1. Open the Integration Console.
  2. Select your Marketplace integration and click **Manage**.
  3. Edit an existing product or create a new one.
  4. Go to **Deployment Actions** in the left-side menu.
  5. Create an action by assigning it a slug and a name.
  Next, handle webhook events and perform API actions in your [integration server](/docs/integrations/marketplace-product#deploy-the-inte

- ### Handle the deployment start
  Handle the `deployment.integration.action.start` webhook. This webhook triggers when a deployment starts an action.

  This is a webhook payload example:
  ```json
  {
    "installationId": "icfg_1234567",
    "action": "branch",
    "resourceId": "abc-def-1334",
    "deployment": { "id": "dpl_568301234" }
  }
  ```
  This payload provides IDs for the installation, action, resource, and deployment.

- ### Use the Get Deployment API
  You can retrieve additional deployment details using the [Get a deployment by ID or URL](https://vercel.com/docs/rest-api/endpoints#tag
  ```bash
  curl https://api.vercel.com/v13/deployments/dpl_568301234 \
    -H "Authorization: {access_token}"
  ```
  You can create your `access_token` from [Vercel's account settings](/docs/rest-api#creating-an-access-token).

  Review the [full code](https://github.com/vercel/example-marketplace-integration/blob/6d2372b8afdab36a0c7f42e1c5a4f0deb2c496c1/app/dash

- ### Complete a deployment action
  Once an action is processed, update its status using the [Update Deployment Integration Action](/docs/rest-api/reference/endpoints/depl

  Example request to this endpoint:
  ```bash
  PATCH https://api.vercel.com/v1/deployments/{deploymentId}/integrations/{installationId}/resources/{resourceId}/actions/{action}
  ```
  Example request body to send that includes the resulting updated resource secrets:
  ```json
  {
    "status": "succeeded",
    "outcomes": [
      {
        "kind": "resource-secrets",
        "secrets": [{ "name": "TOP_SECRET", "value": "****" }]
      }
    ]
  }
  ```

- ### Handle deployment cancellation
  When a deployment is canceled, the `deployment.integration.action.cancel` webhook is triggered. You should handle this action to clean

  Use the `deployment.integration.action.cleanup` webhook to clean up any persistent state linked to the deployment. It's triggered when


---
title: "Integration Image Guidelines"
description: "Guidelines for creating images for integrations, including layout, content, visual assets, descriptions, and design standar
last_updated: "2026-01-16T02:19:32.535Z"
source: "https://vercel.com/docs/integrations/create-integration/integration-image-guidelines"
---

# Integration Image Guidelines

These guidelines help ensure consistent, high-quality previews for integrations across the Vercel platform.

See [Clerk's Integration](https://vercel.com/marketplace/clerk) for a strong example.

## 1. Rules on image layout

a. Images must use a 16:9 layout (1920 × 1080 minimum).

b. Layouts must have symmetrical margins and a reasonable safe area.

c. All images must have both a central visual asset and a description.

## 2. Rules on central visual assets

a. Central visual assets must offer a real glimpse into the product.

b. Central visual assets shouldn't be full window screenshots. Instead, you should showcase product components.

c. Products with GUIs must have at least one central visual asset displaying a component of the GUI.

d. You can include additional decor as long as it does not overpower the central visual asset.

## 3. Rules on descriptions

a. Descriptions must explain the paired visual asset.

b. Descriptions must be clear and concise.

c. Descriptions must follow proper grammar.

## 4. Rules on image design

a. Images must meet a baseline design standard and maintain a consistent visual style across all assets.

b. Images must be accessible and legible. You should ensure good contrast and type size.

c. Avoid unnecessary clutter on images and focus on clarity.

d. All images must be high-resolution to prevent any pixelation.

e. Images should clearly highlight the most compelling parts of the UI and showcase features that are valuable to customers.


---
title: "Using the Integrations REST API"
description: "Learn how to authenticate and use the Integrations REST API to build your integration server."
last_updated: "2026-01-16T02:19:32.654Z"
source: "https://vercel.com/docs/integrations/create-integration/marketplace-api"
---

# Using the Integrations REST API

Learn how to authenticate and use the [Integrations REST API](/docs/integrations/create-integration/marketplace-api/reference) to build y

## How it works

When a  uses your integration, the following two
APIs are used for interaction and communication between the user,
Vercel and the provider integration:

- **Vercel calls the provider API**: You implement the [Vercel Marketplace Partner API](/docs/integrations/create-integration/marketplace
  handle installations, and process billing.
- **The provider calls the Vercel API**: Vercel provides [these endpoints](/docs/integrations/create-integration/marketplace-api/referenc

When building your integration, you'll implement the partner endpoints
and call the Vercel endpoints as needed.

See the [Native Integration Flows](/docs/integrations/create-integration/marketplace-flows) to understand how these endpoints work togeth

## Authentication

The authentication method depends on whether Vercel is calling the integration provider's API or the provider is calling Vercel's API.

### Provider API authentication

There are two authentication methods available:

- **User authentication**: The user initiates the action. You receive a JWT token that identifies the user making the request.
- **System authentication**: Your integration performs the action automatically. You use account-level OpenID Connect (OIDC) credentials

System authentication uses OIDC tokens that represent your integration account, not a specific user. This lets you make API calls to Verc

#### When to use system authentication

- Automatic balance top-ups for prepayment plans
- Background synchronization tasks
- Automated resource management
- Any operation that should run without user action
- Installation cleanup operations when the Vercel account is deleted

#### When to use user authentication

- User-initiated actions
- Operations that require user consent
- Actions tied to a specific user's context

#### Security best practices

- Verify the OIDC token signature and claims: Always validate the token signature using Vercel's [OIDC configuration](https://marketplace
- For user authentication always validate the user's role.

Review the [user authentication](/docs/integrations/create-integration/marketplace-api/reference#user-authentication) and [system authent

### Vercel API authentication

When your integration calls Vercel's API, you authenticate using an access token. You receive this token during the installation process

You can also use [OAuth2](/docs/integrations/create-integration/marketplace-api#oauth2) to obtain access tokens for user-specific operati

### Authentication with SSO

#### Vercel initiated SSO

Vercel initiates SSO as part of the [**Open in Provider** flow](/docs/integrations/marketplace-flows#open-in-provider-button-flow).

1. Vercel sends the user to the provider [redirectLoginUrl](/docs/integrations/create-integration/submit-integration#redirect-login-url),
2. The provider calls the [SSO Token Exchange](/docs/integrations/create-integration/marketplace-api/reference#vercel/sso-token-exchange)
3. The user gains authenticated access to the requested resource.

**Parameters:**

The SSO request to the [redirectLoginUrl](/docs/integrations/create-integration/submit-integration#redirect-login-url) will include the f

- `mode`. The mode of the OAuth authorization is always set to `sso`.
- `code`: The OAuth authorization code.
- `state`: The state parameter that was passed in the OAuth authorization request.

The `code` and `state` parameters will be passed back to Vercel in the [SSO Token Exchange](/docs/integrations/create-integration/marketp

Additionally, the SSO request to the [redirectLoginUrl](/docs/integrations/create-integration/submit-integration#redirect-login-url) may

- `product_id`: The ID of the provider's product
- `resource_id`: The ID of the provider's resource
- `check_id`: The ID of the deployment check, when the resource is associated with a deployment check. Example: "chk_abc123".
- `project_id`: The ID of the Vercel project, for instance, when the resource is connected to the Vercel project. Example: "prj_ff7777b9
- `experimentation_item_id`: See [Experimentation flow](/docs/integrations/create-integration/marketplace-flows#experimentation-flow).
- `invoice_id`: The ID of the provider's invoice
- `pr`: The URL of the pull request in the Vercel project, when known in the context. Example: `https://github.com/owner1/repo1/pull/123`
- `path`: Indicates the area where the user should be redirected to after SSO. The possible values are: "billing", "usage", "onboarding",
- `url`: The provider-specific URL to redirect the user to after SSO. Must be validated by the provider for validity. The data fields tha

The provider should match the most appropriate part of their dashboard to the user's context.

**Using SSO with API responses:**

You can trigger SSO by using `sso:` URLs in your API responses. When users click these links, Vercel initiates the SSO flow before redire

**Format:**

```
sso:https://your-integration.com/resource-page
```

When a user clicks a link with an `sso:` URL:

1. Vercel initiates the SSO flow
2. Your provider validates the SSO request via the [SSO Token Exchange](/docs/integrations/create-integration/marketplace-api/reference#v
3. The user is redirected to the target URL with authenticated access

**Example with notifications:**

```ts filename="upsert-installation-with-sso.ts"
// When creating or updating an installation, include an sso: URL
const response = await fetch(
  `https://api.vercel.com/v1/installations/${installationId}`,
  {
    method: 'PATCH',
    headers: {
      Authorization: `Bearer ${vercelToken}`,
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      notification: {
        title: 'Review your usage',
        message: 'Your monthly usage report is ready',
        href: 'sso:https://your-integration.com/dashboard/usage',
        type: 'info',
      },
    }),
  },
);
```

#### Provider initiated SSO

The integration provider can initiate the SSO process from their side. This helps to streamline the authentication process for users comi

To initiate SSO, an integration provider needs to construct a URL using the following format:

```
https://vercel.com/sso/integrations/{URLSlug}/{installationId}?{query}
```

- [`URLSlug`](/docs/integrations/create-integration/submit-integration#url-slug): The unique identifier for your integration in the Verce
- [`installationId`](/docs/integrations/marketplace-api#installations): The ID of the specific installation for the user
- `query`: Optional query parameters to include additional information

**Example:**

Let's say you have an AWS integration with the following details:

- `URLSlug`: `aws-marketplace-integration-demo`
- `installationId`: `icfg_PSFtkFqr5djKRtOkNtOHIfSd`
- Additional parameter: `resource_id=123456`

The constructed URL would look like this:

```
https://vercel.com/sso/integrations/aws-marketplace-integration-demo/icfg_PSFtkFqr5djKRtOkNtOHIfSd?resource_id=123456
```

**Flow:**

1. The provider constructs and redirects the user to the SSO URL
2. Vercel validates the SSO request and confirms user access
3. After successfully validating the request, Vercel redirects the user back to the provider using the same flow described in the [Vercel
4. The user gains authenticated access to the requested resource

## Working with member information

Get details about team members who have access to an installation. Use this endpoint to retrieve member information for access control, a

To retrieve information about a specific team member associated with an installation, use the [`/v1/installations/{installationId}/member

### Member information request parameters

- `installationId` - The installation ID
- `memberId` - The member ID

### Member information request

```ts filename="get-member-info.ts"
async function getMemberInfo(
  installationId: string,
  memberId: string
): Promise<MemberInfo> {
  const response = await fetch(
    `https://api.vercel.com/v1/installations/${installationId}/member/${memberId}`,
    {
      headers: {
        'Authorization': `Bearer ${token}`,
      },
    }
  );

  if (!response.ok) {
    throw new Error(`Failed to get member info: ${response.statusText}`);
  }

  return response.json();
}
```

### Member information response

```json filename="get-member-info-response.json"
{
  "id": "member_abc123",
  "name": "John Doe",
  "email": "john@example.com",
  "role": "ADMIN",
  "avatar": "https://example.com/avatar.jpg",
  "createdAt": "2025-01-15T10:00:00Z"
}
```

### Member roles

Members can have the following roles:

- `ADMIN` - Full access to the installation and its resources
- `USER` - Limited access, can use resources but can't modify settings

Check the member's role to determine what actions they can perform below.

## Working with installation notifications

Installation notifications appear in the Vercel dashboard to alert users about important information or actions needed for their installa

### Update installation notification

Update the notification field using the [`PATCH /v1/installations/{installationId}`](/docs/integrations/create-integration/marketplace-ap

```ts filename="update-installation-notification.ts"
interface Notification {
  title: string;
  message: string;
  href?: string;
  type?: 'info' | 'warning' | 'error';
}

async function updateInstallationNotification(
  installationId: string,
  notification: Notification
) {
  const response = await fetch(
    `https://api.vercel.com/v1/installations/${installationId}`,
    {
      method: 'PATCH',
      headers: {
        'Authorization': `Bearer ${token}`,
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({ notification }),
    }
  );

  if (!response.ok) {
    throw new Error(`Failed to update notification: ${response.statusText}`);
  }

  return response.json();
}

// Example usage with regular URL:
await updateInstallationNotification('icfg_abc123', {
  title: 'Action Required',
  message: 'Please complete your account setup',
  href: 'https://your-integration.com/setup',
  type: 'warning',
});

// Or with SSO-enabled URL for authenticated access:
// href: 'sso:https://your-integration.com/setup',
```

```json filename="update-installation-notification-response.json"
{
  "id": "icfg_abc123",
  "notification": {
    "title": "Action Required",
    "message": "Please complete your account setup",
    "href": "https://your-integration.com/setup",
    "type": "warning"
  }
}
```

### Notification types

Use different notification types to indicate severity:

- `info` - Informational message (default)
- `warning` - Warning that requires attention
- `error` - Error that needs immediate action

### SSO-enabled notification links

The notification `href` field supports special `sso:` URLs that trigger Single Sign-On before redirecting users to your destination. This

**Format:**

```
sso:https://your-integration.com/resource-page
```

When a user clicks a notification link with an `sso:` URL:

1. Vercel initiates the SSO flow (as described in [Vercel initiated SSO](/docs/integrations/create-integration/marketplace-api#vercel-ini
2. Your provider validates the SSO request via the [SSO Token Exchange](/docs/integrations/create-integration/marketplace-api/reference#v
3. The user is redirected to the target URL with authenticated access

**Example:**

```ts filename="notification-with-sso.ts"
await updateInstallationNotification('icfg_abc123', {
  title: 'Review your usage',
  message: 'Your monthly usage report is ready',
  href: 'sso:https://your-integration.com/dashboard/usage',
  type: 'info',
});
```

Use `sso:` URLs in notification links when they point to resources that require authentication on your platform. For public pages or gene

### Clear notifications

Remove a notification by setting it to `null`:

```ts filename="clear-installation-notification.ts"
async function clearInstallationNotification(installationId: string) {
  const response = await fetch(
    `https://api.vercel.com/v1/installations/${installationId}`,
```

```
    {
      method: 'PATCH',
      headers: {
        'Authorization': `Bearer ${token}`,
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({ notification: null }),
    }
  );

  return response.json();
}
```

### Get installation with notification

You can find the value of the notification field by calling the [`/v1/installations/{installationId}`](/docs/integrations/create-integrat

```ts filename="get-installation-with-notification.ts"
async function getInstallation(installationId: string) {
  const response = await fetch(
    `https://api.vercel.com/v1/installations/${installationId}`,
    {
      headers: {
        'Authorization': `Bearer ${token}`,
      },
    }
  );

  const installation = await response.json();

  if (installation.notification) {
    console.log(`Notification: ${installation.notification.title}`);
    console.log(`Message: ${installation.notification.message}`);
  }

  return installation;
}
```

## Secrets rotation

When your integration provisions resources with credentials, you should implement secrets rotation to allow users to update credentials s

## Working with billing events through webhooks

You can receive billing events with [webhooks](/docs/webhooks) to stay informed about invoice status changes and take appropriate actions

You can receive the following events:

- [`marketplace.invoice.created`](/docs/webhooks/webhooks-api#marketplace.invoice.created): The invoice was created and sent to the custo
- [`marketplace.invoice.paid`](/docs/webhooks/webhooks-api#marketplace.invoice.paid): The invoice was paid
- [`marketplace.invoice.notpaid`](/docs/webhooks/webhooks-api#marketplace.invoice.notpaid): The invoice was not paid after a grace period
- [`marketplace.invoice.refunded`](/docs/webhooks/webhooks-api#marketplace.invoice.refunded): The invoice was refunded

### Webhook security

You should verify webhook signatures to ensure requests come from Vercel. Integration webhooks use your **Integration Secret** (also call

### Billing webhook handlers

You can implement handlers for each billing event type to manage invoice lifecycle and resource access.

#### Handle invoice created

When an invoice is created, you can prepare your systems for billing or send notifications.

**Event:** `marketplace.invoice.created`

```ts filename="handle-invoice-created.ts"
async function handleInvoiceCreated(webhook: WebhookPayload) {
  const { invoice } = webhook.payload;

  // Log invoice creation
  console.log(`Invoice ${invoice.id} created for installation ${invoice.installationId}`);

  // Update your internal records
  await updateInvoiceRecord(invoice.id, {
    status: 'created',
    amount: invoice.amount,
    currency: invoice.currency,
    createdAt: invoice.createdAt,
  });

  // Optional: Send notification to customer
  await sendInvoiceNotification(invoice.customerId, invoice.id);
}
```

```ts filename="invoice-created-webhook-payload.json"
{
  "id": "evt_abc123",
  "type": "marketplace.invoice.created",
  "payload": {
    "invoice": {
      "id": "inv_xyz789",
      "installationId": "icfg_def456",
      "amount": 29.99,
      "currency": "USD",
      "createdAt": "2025-01-15T10:00:00Z"
```

```
    }
  }
}
```

### Handle invoice paid

When an invoice is paid, activate resources or update billing status.

**Event:** `marketplace.invoice.paid`

```ts filename="handle-invoice-paid.ts"
async function handleInvoicePaid(webhook: WebhookPayload) {
  const { invoice } = webhook.payload;

  console.log(`Invoice ${invoice.id} paid`);

  // Update invoice status
  await updateInvoiceRecord(invoice.id, {
    status: 'paid',
    paidAt: invoice.paidAt,
  });

  // Activate resources if they were suspended
  const resources = await getResourcesForInstallation(invoice.installationId);
  for (const resource of resources) {
    if (resource.status === 'suspended') {
      await activateResource(resource.id);
    }
  }

  // Update billing plan if needed
  await updateBillingPlan(invoice.installationId, invoice.billingPlanId);
}
```

```ts filename="invoice-paid-webhook-payload.json"
{
  "id": "evt_def456",
  "type": "marketplace.invoice.paid",
  "payload": {
    "invoice": {
      "id": "inv_xyz789",
      "installationId": "icfg_def456",
      "amount": 29.99,
      "currency": "USD",
      "paidAt": "2025-01-15T11:00:00Z"
    }
  }
}
```

### Handle invoice not paid

When an invoice isn't paid after the grace period, suspend resources or take other actions.

**Event:** `marketplace.invoice.notpaid`

> **💡 Note:** The current webhook payload doesn't include retry attempt information. You'll need to track retry attempts in your system

```ts filename="handle-invoice-not-paid.ts"
async function handleInvoiceNotPaid(webhook: WebhookPayload) {
  const { invoice } = webhook.payload;

  console.log(`Invoice ${invoice.id} not paid`);

  // Update invoice status
  await updateInvoiceRecord(invoice.id, {
    status: 'not_paid',
    notPaidAt: invoice.notPaidAt,
  });

  // Check if this is the final attempt (you may need to query invoice status)
  const invoiceDetails = await getInvoiceDetails(invoice.id);
  const isFinalAttempt = invoiceDetails.retryAttempts >= invoiceDetails.maxRetries;

  if (isFinalAttempt) {
    // Suspend resources after final payment failure
    const resources = await getResourcesForInstallation(invoice.installationId);
    for (const resource of resources) {
      await suspendResource(resource.id, {
        reason: 'payment_failed',
        invoiceId: invoice.id,
      });
    }

    // Notify customer
    await sendPaymentFailureNotification(invoice.customerId, invoice.id);
  } else {
    // Schedule retry or send reminder
    await schedulePaymentRetry(invoice.id, invoiceDetails.nextRetryAt);
  }
}
```

```ts filename="invoice-not-paid-webhook-payload.json"
{
  "id": "evt_ghi789",
  "type": "marketplace.invoice.notpaid",
  "payload": {
```

```
    "invoice": {
      "id": "inv_xyz789",
      "installationId": "icfg_def456",
      "amount": 29.99,
      "currency": "USD",
      "notPaidAt": "2025-01-20T10:00:00Z"
    }
  }
}
```

### Handle invoice refunded

When an invoice is refunded, update records and handle resource access accordingly.

**Event:** `marketplace.invoice.refunded`

```ts filename="handle-invoice-refunded.ts"
async function handleInvoiceRefunded(webhook: WebhookPayload) {
  const { invoice } = webhook.payload;

  console.log(`Invoice ${invoice.id} refunded`);

  // Update invoice status
  await updateInvoiceRecord(invoice.id, {
    status: 'refunded',
    refundedAt: invoice.refundedAt,
    refundAmount: invoice.refundAmount,
  });

  // Adjust billing records
  await adjustBillingRecords(invoice.installationId, {
    type: 'refund',
    amount: invoice.refundAmount,
    invoiceId: invoice.id,
  });

  // Optional: Notify customer
  await sendRefundNotification(invoice.customerId, invoice.id);
}
```

```ts filename="invoice-refunded-webhook-payload.json"
{
  "id": "evt_jkl012",
  "type": "marketplace.invoice.refunded",
  "payload": {
    "invoice": {
      "id": "inv_xyz789",
      "installationId": "icfg_def456",
      "amount": 29.99,
      "currency": "USD",
      "refundAmount": 29.99,
      "refundedAt": "2025-01-21T10:00:00Z"
    }
  }
}
```

---
title: "Implementing secrets rotation"
description: "Learn how to implement secrets rotation in your integration to allow users to rotate credentials securely."
last_updated: "2026-01-16T02:19:32.563Z"
source: "https://vercel.com/docs/integrations/create-integration/marketplace-api/secrets-rotation"
---

# Implementing secrets rotation

When your integration provisions resources with credentials (like API keys, database passwords, or access tokens), you must implement sec

> **⚠️ Warning:** This functionality must be turned on by Vercel for your integration. Contact your partner support team in Slack to have

## How it works

Vercel calls your partner API to trigger a rotation. This happens when a user or admin requests secret rotation for a resource and may al

1. The customer clicks "rotate secret" in the Vercel dashboard for a resource you manage
2. Vercel makes a `POST` request to your `/v1/installations/{installationId}/resources/{resourceId}/secrets/rotate` endpoint
3. Your backend either generates new secrets for the resource and returns them in the response or returns `sync: false` and performs the
4. Once Vercel has the new secrets for the resource, the customer's linked projects will be redeployed to pick up the new secrets.
5. After the period of time specified in `delayOldSecretsExpirationHours`, the old secrets should stop working and be deleted by your cod

> **⚠️ Warning:** It's critical that you keep the old secrets active for the amount of time specified in the request to your rotate secre

## Endpoint specification

Vercel calls this endpoint on your partner API to request secret rotation:

```http
POST /v1/installations/{installationId}/resources/{resourceId}/secrets/rotate
Authorization: Bearer <oidc-token>
```

**Authentication:**

Vercel includes an OIDC token in the `Authorization` header using either user or system authentication. You must verify this token before

When using user authentication, the token contains claims about the user who initiated the rotation, including their role (which may be `

**Path parameters:**

- `installationId`: The Vercel installation ID (e.g., `icfg_9bceb8ccT32d3U417ezb5c8p`)
- `resourceId`: Your external resource ID that you provided when provisioning the resource

**Request body:**

```json filename="Request body schema"
{
  "reason": "Security audit requirement",
  "delayOldSecretsExpirationHours": 3
}
```

- `reason` (optional): A string explaining why the rotation was requested
- `delayOldSecretsExpirationHours` (optional): Number of hours (0-720, max 30 days) before old secrets expire. Can be a decimal amount (e

Once you receive this request, you should rotate the secrets for this resource and keep the old ones live for the specified amount of tim

> **💡 Note:** Discuss with Vercel partner support what values should be sent to your backend for `delayOldSecretsExpirationHours`.

## Response options

You can respond in two ways depending on your implementation:

### Synchronous rotation (HTTP 200)

Return the rotated secrets immediately:

```json filename="Synchronous response"
{
  "sync": true,
  "secrets": [
    {
      "name": "DATABASE_URL",
      "value": "postgresql://user:newpass@host:5432/db"
    },
    {
      "name": "API_KEY",
      "value": "rotated-key-value"
    }
  ],
  "partial": false
}
```

- `sync: true`: Indicates you've completed rotation immediately
- `secrets`: Array of rotated secrets with `name` and `value`
- `partial` (optional): Set to `true` if only a subset of secrets are included in the response (the default is `false` indicating your re

> **💡 Note:** When you return secrets synchronously, Vercel automatically updates the environment variables and tracks the rotation as c

### Asynchronous rotation (HTTP 202)

Indicate that rotation will happen later:

```json filename="Asynchronous response"
{
  "sync": false
}
```

When you return `sync: false`, you must call Vercel's API later to complete the rotation using the [Update Resource Secrets endpoint](/do

```http
PUT https://api.vercel.com/v1/installations/{installationId}/resources/{resourceId}/secrets
```

```json filename="Complete rotation request"
{
  "secrets": [
    {
      "name": "DATABASE_URL",
      "value": "postgresql://user:newpass@host:5432/db"
    }
  ],
  "partial": false
}
```

Use the access token you received during installation to authenticate this request.

## Implementation example

Here's a complete example of implementing the rotation endpoint:

```ts filename="handle-secrets-rotation.ts"
import { verifyOIDCToken } from './auth';

async function handleSecretsRotation(req, res) {
  const { installationId, resourceId } = req.params;
  const { reason, delayOldSecretsExpirationHours = 0 } = req.body;

  // Verify authentication - Vercel sends an OIDC token (user or system authentication)
  const token = req.headers.authorization?.replace('Bearer ', '');
  const claims = await verifyOIDCToken(token);

  if (!claims || (claims.user_role && claims.user_role !== 'ADMIN')) {
    return res.status(401).json({ error: 'Invalid token' });
  }
```

```
  // Get resource from your database
  const resource = await getResource(resourceId);
  if (!resource) {
    return res.status(404).json({ error: 'Resource not found' });
  }

  // Rotate credentials in your system
  const newCredentials = await rotateResourceCredentials(resourceId);

  // Schedule old credentials expiration
  if (delayOldSecretsExpirationHours > 0) {
    await scheduleCredentialExpiration(
      resource.oldCredentials,
      delayOldSecretsExpirationHours
    );
  } else {
    // Expire old credentials immediately
    await expireCredentials(resource.oldCredentials);
  }

  // Return new secrets immediately
  return res.status(200).json({
    sync: true,
    secrets: [
      {
        name: 'DATABASE_URL',
        value: newCredentials.connectionString,
      },
      {
        name: 'DATABASE_PASSWORD',
        value: newCredentials.password,
      },
    ],
    partial: false
  });
}
```

## Error handling

Return appropriate HTTP status codes for error cases:

```ts filename="error-responses.ts"
// Resource not found
res.status(404).json({ error: 'Resource not found' });

// Invalid request body
res.status(400).json({ error: 'Invalid delayOldSecretsExpirationHours' });

// Insufficient permissions
res.status(403).json({ error: 'User lacks permission to rotate secrets' });

// Rotation temporarily unavailable
res.status(503).json({ error: 'Rotation service unavailable, try again later' });

// Internal error during rotation
res.status(500).json({ error: 'Failed to rotate credentials' });
```

## Testing rotation

When testing your implementation:

1. Provision a test resource through your integration
2. Navigate to the resource in the Vercel dashboard
3. Click "Rotate Secrets" or similar action
4. Verify your endpoint receives the request with correct parameters
5. For synchronous rotation, confirm Vercel receives and updates the secrets
6. For asynchronous rotation, verify your background job completes and calls Vercel's API
7. Confirm the resource now displays the correct environment variables on the resource page in the Vercel dashboard
8. Confirm old credentials expire at the correct time

## Best practices

- **Always verify authentication**: Validate the OIDC token from the `Authorization` header before processing any rotation request. Verce
- **Validate all inputs**: Check that `delayOldSecretsExpirationHours` doesn't exceed your `maxDelayHours`
- **Audit all rotations**: Log who or what requested rotation, when, and why (the OIDC token claims contain either user information or sy
- **Handle failures gracefully**: If rotation fails, maintain old credentials and return an error
- **Test credential expiration**: Ensure old credentials are properly revoked after the delay period
- **Support partial rotation**: If you can't rotate all secrets, return `partial: true` with the secrets you did rotate
- **Implement idempotency**: Handle duplicate rotation requests gracefully
- **Monitor rotation requests**: Track rotation frequency to detect unusual patterns

---
title: "Native Integration Flows"
description: "Learn how information flows between the integration user, Vercel, and the integration provider for Vercel native integratio
last_updated: "2026-01-16T02:19:32.603Z"
source: "https://vercel.com/docs/integrations/create-integration/marketplace-flows"
---

# Native Integration Flows

As a Vercel integration provider, when you [create a native product integration](/docs/integrations/marketplace-product), you need to set

The following diagrams help you understand how information flows in both directions between the integration user, Vercel and your native

## Create a storage product flow

When a Vercel user, who wants to  a provider native integration, selects the **Storage** tab of the Vercel dashboard, followed by **Creat

After reviewing the flow diagram below, explore the sequence for each step:

- [Select storage product](#select-storage-product)
- [Select billing plan](#select-billing-plan)
- [Submit store creation](#submit-store-creation)

Understanding the details of each step will help you set up the installation section of the [integration server](https://github.com/verce

### Select storage product

When the integration user selects a storage provider product, an account is created for this user on the provider's side if the account d

### Select billing plan

Using the installation id for this product and integration user, the Vercel dashboard presents available billing plans for the product. T

### Submit store creation

After confirming the plan selection, the integration user is presented with information fields that the integration provider specified in

## Connections between Vercel and the provider

### Open in Provider button flow

When an integration user selects the **Manage** button for a product integration from the Vercel dashboard's **Integrations** tab, they a

### Provider to Vercel data sync flow

This flow happens when a provider edits information about a resource in the provider's system.

### Vercel to Provider data sync flow

This flow happens when a user who has installed the product integration edits information about it on the Vercel dashboard.

### Rotate credentials in provider flow

This flow happens when a provider rotates the credentials of a resource in the provider system.

> **💡 Note:** Vercel will update the environment variables of projects connected to the
> resource but will not automatically redeploy the projects. The user must
> redeploy them manually.

## Flows for the Experimentation category

### Experimentation flow

This flow applies to the products in the **Experimentation** category, enabling providers to display [feature flags](/docs/feature-flags)

### Experimentation Edge Config Syncing

This flow applies to integration products in the **Experimentation** category. It enables providers to push the necessary configuration d

Edge Config Syncing is an optional feature that providers can enable for their integration. Users can opt in by enabling it for their ins

Users can enable this setting either during the integration's installation or later through the installation's settings page. Providers m

The presence of `protocolSettings.experimentation.edgeConfigId` in the payload indicates that the user has enabled the setting and expect

Afterward, providers can use the [Edge Config Syncing](/docs/integrations/create-integration/marketplace-api#push-data-into-a-user-provid

Once the data is available, users can connect the resource to a Vercel project. Doing so will add an `EXPERIMENTATION_CONFIG` environment

Users can then use the appropriate [adapter provided by the Flags SDK](https://flags-sdk.dev/providers), which will utilize the Edge Conf

## Resources with Claim Deployments

When a Vercel user claims deployment ownership with the [Claim Deployments feature](/docs/deployments/claim-deployments), storage integra

### Ownership transfer requirements

Vercel users can transfer ownership of an integration installation if they meet these requirements:

- They must have DELETE permissions on the source team (Owner role)
- They must also be a valid owner or member of the destination team

This ensures only authorized users can transfer billing responsibility between teams.

### Provision flow

This flow describes how a claims generator (e.g. AI agent) provisions a provider resource and connects it to a Vercel project. Before the

### Transfer request creation flow

This flow describes how a claims generator initiates a request to transfer provider resources, with Vercel as an intermediary. The flow r

Example for `CreateResourceTransfer` request (Vercel API):

```bash filename="terminal"
curl --request POST \
  --url https://api.vercel.com/projects/<project_id>/transfer-request\?teamId\=<team_id> \
  --header 'Authorization: Bearer <token>' \
  --header 'Content-Type: application/json' \
  --data '{}'
```

`CreateResourceTransfer` response with a claim code:

```json filename="terminal"
```

{ "code": "c7a9f0b4-4d4a-45bf-b550-2bfa34de1c0d" }
```

### Transfer request accept flow

This flow describes how a Vercel user accepts a resource transfer request when they visit a Vercel URL sent by the claims generator. The

Vercel calls your integration server twice during the accept flow:

**Step 1: Verify the transfer**

**Endpoint:** `GET /v1/installations/{installationId}/resource-transfer-requests/{providerClaimId}/verify`

Verify that the transfer is still valid. Check that:

- The provider claim ID exists and hasn't expired
- The resources still exist
- The transfer hasn't already been completed

**Response:**

```json
{
  "valid": true,
  "billingPlan": {
    "id": "plan_xyz",
    "cost": 10.00
  }
}
```

If the transfer requires a new billing plan for the target team, include it in the response.

**Step 2: Accept the transfer**

**Endpoint:** `POST /v1/installations/{installationId}/resource-transfer-requests/{providerClaimId}/accept`

Complete the transfer by:

- Updating resource ownership from the claims generator to the target user
- Linking resources to the target installation
- Invalidating the provider claim ID

**Request body:**

```json
{
  "targetInstallationId": "icfg_target123",
  "targetTeamId": "team_target456"
}
```

**Response:**

```json
{
  "success": true
}
```

### Troubleshooting resource transfers

If transfers fail, check these common issues:

- **Invalid provider claim ID**: The claim ID might have expired or already been used. Generate a new transfer request.
- **Missing installation**: The target team must have your integration installed. Prompt the user to install it first.
- **Billing plan conflicts**: If the transfer requires a billing plan change, ensure the target team can accept it.
- **Resource ownership**: Verify that resources belong to the source installation before transferring.

---
title: "Create a Native Integration"
description: "Learn how to create a product for your Vercel native integration"
last_updated: "2026-01-16T02:19:32.618Z"
source: "https://vercel.com/docs/integrations/create-integration/marketplace-product"
---

# Create a Native Integration

With a , you allow a Vercel customer who has  your integration to use specific features of your integration **without** having them leave

## Requirements

To create and list your products as a Vercel provider, you need to:

- Use a Vercel Team on a [Pro plan](/docs/plans/pro-plan).
- Provide a **Base URL** in the product specification for a native integration server that you will create based on:
  - The [sample integration server repository](https://github.com/vercel/example-marketplace-integration).
  - The [native integrations API endpoints](/docs/integrations/marketplace-api).
- Be an approved provider so that your product is available in the Vercel Marketplace. To do so, [submit your application](https://vercel

## Create your product

In this tutorial, you create a storage  for your native integration through the following steps:

- ### Set up the integration
  Before you can create a product, you must have an existing integration. [Create a new Native Integration](/docs/integrations/create-int

- ### Deploy the integration server
  In order to deploy the integration server, you should update your integration configuration to set the **base URL** to the integration
```

  1. Select the team you would like to use from the scope selector.
  2. From your dashboard, select the **Integrations** tab and then select the **Integrations Console** button.
  3. Select the integration you would like to use for the product.
  4. Find the **base URL** field in the **Product** section and set it to the integration server URL.
  5. Select **Update**.
  You can use this [example Next.js application](https://github.com/vercel/example-marketplace-integration) as a guide to create your&#x2

- ### Add a new product
  1. Select the integration you would like to use for the product from the Integrations Console
  2. Select **Create Product** from the **Products** card of the **Product** section

- ### Complete the fields and save
  You should now see the **Create Product** form. Fill in the following fields:
  1. Complete the **Name**, **URL Slug**, **Visibility** and **Short Description** fields
  2. Optionally update the following in the [Metadata Schema](#metadata-schema) field:
  - Edit the `properties` of the JSON schema to match the options that you are making available through the .
  - Edit and check that the attributes of each property such as `type` matches your requirements.
  - Include the billing plan options that Vercel will send to your integration server when requesting the list of billing plans.
  - Use the **** section to check your JSON schema as you update it.
  Review the data collection process shown in the [submit store creation flow](/docs/integrations/create-integration/marketplace-flows#su
  3. Select **Apply Changes**

- ### Update your integration server
  Add or update the [Billing](/docs/integrations/marketplace-api#billing) endpoints in your integration server so that the appropriate pl

  Your integration server needs to handle the [billing plan selection flow](/docs/integrations/create-integration/marketplace-flows#selec

- ### Publish your product
  To publish your product, you'll need to request for the new product to be approved:
  1. Check that your product integration follows our [review guidelines](/docs/integrations/create-integration/approval-checklist)
  2. Email integrations@vercel.com with your request to be reviewed for listing
  Once approved, Vercel customers can now add your product with the integration and select a billing plan.

## Reference

### Metadata schema

When you first create your , you will see a [JSON schema](https://json-schema.org/) in the **Metadata Schema** field of the product confi

When the customer installs your product, Vercel collects data from this customer and sends it to your  based on the Metadata schema you p

As an example, use the following configuration to only show the name of the product:

```json
{
  "type": "object",
  "properties": {},
  "additionalProperties": false,
  "required": []
}
```

See the endpoints for [Provision](/docs/integrations/marketplace-api#provision-resource) or [Update](/docs/integrations/marketplace-api#u

| Property `ui:control` | Property `type` | Notes
| --- | --- | ---
| `input` | `number` | Number input
| `input` | `string` | Text input
| `toggle` | `boolean` | Toggle input
| `slider` | `array` | Slider input. The `items` property of your array must have a type of number
| `select` | `string` | Dropdown input
| `multi-select` | `array` | Dropdown with multi-select input. The items property of your array must have a type of string
| `vercel-region` | `string` | Vercel Region dropdown input. You can restrict the list of available regions by settings the
| `multi-vercel-region` | `array` | Vercel Region dropdown with multi-select input. You can restrict the list of available region
| `domain` | `string` | Domain name input
| `git-namespace` | `string` | Git namespace selector

> **💡 Note:** See the [full JSON
> schema](https://vercel.com/api/v1/integrations/marketplace/metadata-schema)
> for the Metadata Schema. You can add it to your code editor for autocomplete
> and validation.

You can add it to your editor configuration as follows:

```json
{
  "$schema": "https://vercel.com/api/v1/integrations/marketplace/metadata-schema"
}
```

## More resources

- [Native integrations API reference](/docs/integrations/create-integration/marketplace-api)
- [Native integration server Github code sample](https://github.com/vercel/example-marketplace-integration)
- [Native Integration Flows](/docs/integrations/create-integration/marketplace-flows)


---
title: "Native integration concepts"
description: "As an integration provider, understanding how your service interacts with Vercel"
last_updated: "2026-01-16T02:19:32.667Z"
source: "https://vercel.com/docs/integrations/create-integration/native-integration"
---

# Native integration concepts

Native integrations allow a two-way connection between Vercel and third-party providers. This enables providers to embed their services i

- They **do not** need to create an account on your site.
- They can choose suitable billing plans for each product through the Vercel dashboard.

- Billing is managed through their Vercel account.

This document outlines core concepts, structure, and best practices for creating robust, scalable integrations that align with Vercel's e

## Team installations

Team installations are the foundation of native integrations, providing a secure and organized way to connect user teams with specific in

Installations represent a connection between a Vercel team and your system. They are **team-scoped, not user-scoped**, meaning they belon

Because installations are tied to teams and not individual users, use the [Get Account Information endpoint](/docs/integrations/create-in

| Concept | Definition |
| --- | --- |
| Team installation | The primary connection between a user's team and a specific inte |
| [`installationId`](/docs/integrations/marketplace-api#installations) | The main partition key connecting the user's team to the integra |

### Reinstallation behavior

If a team uninstalls and then reinstalls your integration, Vercel creates a new `installationId`. Treat this as a completely new installa

### Limits

Understanding the limits of team installation instances for all types of integrations can help you design a better integration architectu

A Vercel team can only have one native integration installation at a time. If a team wants to install the integration again, they need to

| Metric | Limit |
| --- | --- |
| [Native integration](/docs/integrations#native-integrations) installation | A maximum of o |
| [Connectable account integration](/docs/integrations/create-integration#connectable-account-integrations) installation | A maximum of o |

A team can have both a native integration installation and a connectable account integration installation for the same integration if you

## Products

Products represent the offerings available within an integration, allowing integration users to select and customize an asset such as "AC

| Concept | Definition |
| --- | --- |
| Product | An offering that integration users can add to their native integration installation. A provider ca |
| [Billing plan](#billing-and-usage) | Each product has an associated pricing structure that the provider specifies when creating product |

## Resources

Resources are the actual instances of products that integration users provision and utilize. They represent instances of products in your

Resources track usage and billing at the individual resource level, giving you the ability to monitor and charge for each provisioned ins

| Concept | Definition |
| --- | --- |
| Resource | A specific instance of a product provisioned in an installation. |
| Provisioning | Explicit creation and removal (de-provisioning) of resources by users. |
| Keysets | Independent sets of secrets for each resource. |
| Project connection | Ability to link resources to Vercel projects independently. |

### Working with installation and team information

When working with resources, you'll use the `installationId` as the main identifier for connecting resources to a team's installation. No

### Resource usage patterns

Integration users can add and manage resources in various ways. For example:

- Single resource: Using one resource such as one database for all projects.
- Per-project resources: Dedicating separate resources for each project.
- Environment-specific resources: Using separate resources for different environments (development, preview, production) within a project

## Relationships

The diagram below illustrates the relationships between team installations, products, and resources:

- One installation can host multiple products and resources.
- One product can have multiple resource instances.
- Resources can be connected to multiple projects independently.

## Billing and usage

Billing and usage tracking are crucial aspects of native integrations that are designed to help you create a system of transparent billin

| Concept | Definition |
| --- | --- |
| Resource-level billing | Billing and u |
| [Installation-level billing](/docs/integrations/create-integration/submit-integration#installation-level-billing-plans) | Billing and u |
| Billing plan and payment | A plan can be |

We recommend you implement resource-level billing, which is the default, to provide users with detailed cost breakdowns and enable more f

## More resources

To successfully implement your native integration, you'll need to handle several key flows:

- [Storage product creation flow](/docs/integrations/create-integration/marketplace-flows#create-a-storage-product-flow)
- [Data synchronization flows between Vercel and the provider](/docs/integrations/create-integration/marketplace-flows#connections-betwee
- [Provider dashboard access](/docs/integrations/create-integration/marketplace-flows#open-in-provider-button-flow)
- [Credential management](/docs/integrations/create-integration/marketplace-flows#rotate-credentials-in-provider-flow)
- [Experimentation integrations flows](/docs/integrations/create-integration/marketplace-flows#flows-for-the-experimentation-category)
- [Flows for resource handling with claim deployments](/docs/integrations/create-integration/marketplace-flows#resources-with-claim-deplo

---
title: "Integrate with Vercel"
description: "Learn how to create and manage your own integration for internal or public use with Vercel."
last_updated: "2026-01-16T02:19:32.718Z"
source: "https://vercel.com/docs/integrations/create-integration"
---

# Integrate with Vercel

Learn the process of creating and managing integrations on Vercel, helping you extend the capabilities of Vercel projects by connecting t

1. Submit a [create integration form](#creating-an-integration) request to Vercel
2. If you are creating a native integration, submit the [create product form](#native-integration-product-creation) as well
3. Once your integration is approved, you can share it for users to install if it's a [connectable account integration](/docs/integration
4. For a [native integration](/docs/integrations#native-integrations), you need to [create a product](/docs/integrations/create-integrati
5. [Publish your native integration](/docs/integrations/create-integration/marketplace-product#publish-your-product) for users to install

## Creating an integration

Integrations can be created by filling out the **Create Integration** form. To access the form:

1. From your Vercel [dashboard](/dashboard), select your account/team from the [scope selector](/docs/dashboard-features#scope-selector)
2. Select the **Integrations** tab to see the Integrations overview
3. Then, select the [**Integrations Console**](/dashboard/integrations/console) button and then select **Create**
4. Fill out all the entries in the [Create integration form](#create-integration-form-details) as necessary
5. At the end of the form, depending on the type of integration you are creating, you **must** accept the terms provided by Vercel so tha
6. If you are creating a native integration, continue to the [Native integration product creation](#native-integration-product-creation)

### Native integration product creation

> **💡 Note:** In order to create native integrations, please share your `team_id` and
> Integration's [URL
> Slug](/docs/integrations/create-integration/submit-integration#url-slug) with
> Vercel in your shared Slack channel (`#shared-mycompanyname`). You can sign up
> to be a native integration provider [here](/marketplace/program).

You can create your product(s) using the [Create product form](#create-product-form-details) after you have submitted the integration for

### Create Integration form details

The **Create Integration** form must be completed in full before you can submit your integration for review. The form has the following f

| Field
| :----------------------------------------------------------------------------------------------------------------------------------------------------------------------
| [Name](/docs/integrations/create-integration/submit-integration#integration-name)
| [URL Slug](/docs/integrations/create-integration/submit-integration#url-slug)
| [Developer](/docs/integrations/create-integration/submit-integration#developer)
| [Contact Email](/docs/integrations/create-integration/submit-integration#email)
| [Support Contact Email](/docs/integrations/create-integration/submit-integration#email)
| [Short Description](/docs/integrations/create-integration/submit-integration#short-description)
| [Logo](/docs/integrations/create-integration/submit-integration#logo)
| [Category](/docs/integrations/create-integration/submit-integration#category)
| [Website](/docs/integrations/create-integration/submit-integration#urls)
| [Documentation URL](/docs/integrations/create-integration/submit-integration#urls)
| [EULA URL](/docs/integrations/create-integration/submit-integration#urls)
| [Privacy Policy URL](/docs/integrations/create-integration/submit-integration#urls)
| [Overview](/docs/integrations/create-integration/submit-integration#overview)
| [Additional Information](/docs/integrations/create-integration/submit-integration#additional-information)
| [Feature Media](/docs/integrations/create-integration/submit-integration#feature-media)
| [Redirect URL](/docs/integrations/create-integration/submit-integration#redirect-url)
| [API Scopes](/docs/integrations/create-integration/submit-integration#api-scopes)
| [Webhook URL](/docs/integrations/create-integration/submit-integration#webhook-url)
| [Configuration URL](/docs/integrations/create-integration/submit-integration#configuration-url)
| [Base URL](/docs/integrations/create-integration/submit-integration#base-url) (Native integration)
| [Redirect Login URL](/docs/integrations/create-integration/submit-integration#redirect-login-url) (Native integration)
| [Installation-level Billing Plans](/docs/integrations/create-integration/submit-integration#installation-level-billing-plans) (Native i
| [Integrations Agreement](/docs/integrations/create-integration/submit-integration#integrations-agreement)

### Create Product form details

The **Create Product** form must be completed in full for at least one product before you can submit your product for review. The form ha

| Field | D
| :---------------------------------------------------------------------------------------------------------------------------------------------------------------------- | :
| [Name](/docs/integrations/create-integration/submit-integration#product-name) | T
| [URL Slug](/docs/integrations/create-integration/submit-integration#product-url-slug) | T
| [Short Description](/docs/integrations/create-integration/submit-integration#product-short-description) | A
| [Short Billing Plans Description](/docs/integrations/create-integration/submit-integration#product-short-billing-plans-description) | A
| [Metadata Schema](/docs/integrations/create-integration/submit-integration#product-metadata-schema) | T
| [Logo](/docs/integrations/create-integration/submit-integration#product-logo) | T
| [Tags](/docs/integrations/create-integration/submit-integration#product-tags) | T
| [Guides](/docs/integrations/create-integration/submit-integration#product-guides) | G
| [Resource Links](/docs/integrations/create-integration/submit-integration#product-resource-links) | R
| [Snippets](/docs/integrations/create-integration/submit-integration#product-snippets) | A
| [Edge Config Support](/docs/integrations/create-integration/submit-integration#edge-config-support) | E
| [Log Drain Settings](/docs/integrations/create-integration/submit-integration#log-drain-settings) | C
| [Checks API](/docs/integrations/create-integration/submit-integration#checks-api) | E

## After integration creation

### Native integrations

To create a  for your [native integration](/docs/integrations#native-integrations), follow the steps in [Create a product for a native in

### Connectable account integrations

Once you have created your [connectable account integration](/docs/integrations#connectable-accounts), it will be assigned the [**Communi

If you are interested in having your integration listed on the public [Integrations](/integrations) page:

- The integration must have at least 500 active installations (500 accounts that have the integration installed).
- The integration follows our [review guidelines](/docs/integrations/create-integration/approval-checklist).
- Once you've reached this minimum install requirement, please email integrations@vercel.com with your request to be reviewed for listing

### View created integration

You can view all integrations that you have created on the [**Integrations Console**](/dashboard/integrations/console).

To preview an integration's live URL, click **View Integration**. This URL can be shared for installation based on the integration's visi

The live URL has the following format:

```javascript filename="example-url"
https://vercel.com/integrations/<slug>
```

Where, `<slug>` is the name you specified in the **URL Slug** field during the integration creation process.

### View logs

To help troubleshoot errors with your integration, select the **View Logs** button on the **Edit Integration** page. You will see a list

### Community badge

In the [**Integrations Console**](/dashboard/integrations/console), a **Community** badge will appear under your new integration's title

Community integrations are developed by third parties and are supported solely by the developers. Before installing, review the developer

## Installation flow

The installation of the integration is a critical component of the developer experience that must cater to all types of developers. While

- New user flow: Developers should be able to create an account on your service while installing the integration
- Existing user flow: With existing accounts, developers should sign in as they install the integration. Also, make sure the forgotten pa
- Strong defaults: The installation flow should have minimal steps and have set defaults whenever possible
- Advanced settings: Provide developers with the ability to override or expand settings when installing the integration

For the installation flow, you should consider adding the following specs:

| Spec Name | Required | Spec Notes |
| --- | --- | --- |
| Documentation | Yes | Explain the integration and how to use it. Also explain the defaults and how to override them. |
| Deploy Button | No | Create a [Deploy Button](/docs/deploy-button) for projects based on a Git repository. |

## Integrations console

You can view all the integrations that you created for a team on the [**Integrations Console**](https://vercel.com/d?to=%2F%5Bteam%5D%2F%

### Integration credentials

When you create an integration, you are assigned a client (integration) ID and secret which you will use to authenticate your webhooks as

## Integration support

As an integration creator, you are solely responsible for the support of your integration developed and listed on Vercel. When providing

When submitting an integration, you'll enter a [support email](/docs/integrations/create-integration/submit-integration#email), which wil

### Compliance and sanctions

Vercel complies with applicable laws and regulations, including sanctions administered by the Office of Foreign Assets Control (OFAC). Ou

Vercel does not perform OFAC checks on behalf of its customers or their end users. As an integration provider, you are solely responsible

---
title: "Requirements for listing an Integration"
description: "Learn about all the requirements and guidelines needed when creating your Integration."
last_updated: "2026-01-16T02:19:32.880Z"
source: "https://vercel.com/docs/integrations/create-integration/submit-integration"
---

# Requirements for listing an Integration

Defining the content specs helps you create the main cover page of your integration. On the marketplace listing, the cover page looks lik

The following requirements are located in the integrations console, separated in logical sections.

## Profile

## Integration Name

- **Character Limit**: 64
- **Required**: Yes

This is the integration title which appears on Integration overview. This title should be unique.

## URL Slug

- **Character Limit**: 32
- **Required**: Yes

This will create the URL for your integration. It will be located at:

```javascript filename="example-url"
https://vercel.com/integrations/<slug>
```

## Developer

- **Character Limit**: 64
- **Required**: Yes

The name of the integration owner, generally a legal name.

## Email

- **Required**: Yes

There are two types of email that you must provide:

- **Contact email**: This is the contact email for the owner of the integration. It will not be publicly visible and will only be used by
- **Support contact email**: The support email for the integration. This email will be publicly listed and used by developers to contact

> **💡 Note:** As an integration creator, you are responsible for the support of integration
> developed and listed on Vercel. For more information, refer to [Section 3.2 of
> Vercel Integrations Marketplace
> Agreement](/legal/integrations-marketplace-agreement). You are also solely
> responsible for your own compliance with applicable laws and regulations,
> including sanctions and export controls. See [Compliance and
> sanctions](/docs/integrations/create-integration#compliance-and-sanctions) for
> more details.

## Short Description

- **Character Limit**: 40
- **Required**: Yes

The integration tagline on the Marketplace card, and the Integrations overview in the dashboard.

## Logo

- **Required**: Yes

The image displayed in a circle, that appears throughout the dashboard and marketing pages. Like all assets, it will appear in both light

You must make sure that the images adhere to the following dimensions and aspect ratios:

| Spec Name | Ratio | Size | Notes |
| --- | --- | --- | --- |
| Icon | 1:1 | 20-80px | High resolution bitmap image, non-transparent PNG, minimum 256px |

## Category

- **Required**: Yes

The category of your integration is used to help developers find your integration in the marketplace. You can choose from the following c

- Commerce
- Logging
- Databases
- CMS
- Monitoring
- Dev Tools
- Performance
- Analytics
- Experiments
- Security
- Searching
- Messaging
- Productivity
- Testing
- Observability
- Checks

## URLs

The following URLs must be submitted as part of your application:

- **Website**: A URL to the website related to your integration.
- **Documentation URL**: A URL for users to learn how to use your integration.
- **EULA URL**: The URL to your End User License Agreement (EULA) for your integration. For more information about your required EULA, se
- **Privacy Policy URL**: The URL to your Privacy Policy for your integration. For more information about your required privacy policy, s
- **Support URL**: The URL for your Integration's support page.

They are displayed in the Details section of the Marketplace integration page that Vercel users view before they install the integration.

## Overview

- **Character Limit**: 768
- **Required**: Yes

This is a long description about the integration. It should describe why and when a user may want to use this integration. Markdown is su

## Additional Information

- **Character Limit**: 1024
- **Required**: No

Additional steps to install or configure your integrations. Include environment variables and their purpose. Markdown is supported.

## Feature media

- **Required**: Yes

These are a collection of images displayed on the carousel at the top of your marketplace listing. We require at least 1 image, but you c

These gallery images will appear in both light and dark mode. Avoid long text, as it may not be legible on smaller screens.

Also consider the 20% safe zone around the edges of the image by placing the most important content of your images within the bounds. Thi

Your media should adhere to the following dimensions and aspect ratios:

| Spec Name | Ratio | Size | Notes |
| --- | --- | --- | --- |
| Gallery Images | 3:2 | 1440x960px | High resolution bitmap image, non-transparent PNG. Minimum 3 images, up to 5 can be uploaded. You

## External Integration Settings

## Redirect URL

- **Required**: Yes

The Redirect URL is an HTTP endpoint that handles the installation process by exchanging a code for an API token, serving a user interfac

- **Token Exchange**: Exchanges a provided code for a [Vercel REST API access token](/docs/rest-api/vercel-api-integrations#exchange-code
- **User Interface**: Displays a responsive UI in a popup window during the installation
- **Project Provisioning**: Allows users to create new projects or connect existing ones in your system to their Vercel Projects
- **Completion**: Redirects the user back to Vercel upon successful installation

**Important considerations**:

- If your application uses the `Cross-Origin-Opener-Policy` header, use the value `unsafe-none` to allow the Vercel dashboard to monitor
  dashboard to monitor the popup's closed state.
- For local development and testing, you can specify a URL on `localhost`.

## API Scopes

- **Required**: No

API Scopes define the level of access your integration will have to the Vercel REST API. When setting up a new integration, you need to:

- Select only the API Scopes that are essential for your integration to function
- Choose the appropriate permission level for each scope: `None`, `Read`, or `Read/Write`

After activation, your integration may collect specific user data based on the selected scopes. You are accountable for:

- The privacy, security, and integrity of this user data
- Compliance with [Vercel's Shared Responsibility Model](/docs/security/shared-responsibility#shared-responsibilities)

Learn more about API scope permissions in the [Extending Vercel](/docs/integrations/install-an-integration/manage-integrations-reference)

## Webhook URL

- **Required**: No

With your integration, you can listen for events on the Vercel platform through Webhooks. The following events are available:

### Deployment events

The following events are available for deployments:

- [`deployment.created`](/docs/webhooks/webhooks-api#deployment.created)
- [`deployment.error`](/docs/webhooks/webhooks-api#deployment.error)
- [`deployment.canceled`](/docs/webhooks/webhooks-api#deployment.canceled)
- [`deployment.succeeded`](/docs/webhooks/webhooks-api#deployment.succeeded)

### Configuration events

The following events are available for configurations:

- [`integration-configuration.permission-upgraded`](/docs/webhooks/webhooks-api#integration-configuration.permission-upgraded)
- [`integration-configuration.removed`](/docs/webhooks/webhooks-api#integration-configuration.removed)
- [`integration-configuration.scope-change-confirmed`](/docs/webhooks/webhooks-api#integration-configuration.scope-change-confirmed)
- [`integration-configuration.transferred`](/docs/webhooks/webhooks-api#integration-configuration.transferred)

### Domain events

The following events are available for domains:

- [`domain.created`](/docs/webhooks/webhooks-api#domain.created)

### Project events

The following events are available for projects:

- [`project.created`](/docs/webhooks/webhooks-api#project.created)
- [`project.removed`](/docs/webhooks/webhooks-api#project.removed)

### Check events

The following events are available for checks:

- [`deployment.ready`](/docs/webhooks/webhooks-api#deployment-ready)
- [`deployment.check-rerequested`](/docs/webhooks/webhooks-api#deployment-check-rerequested)

See the [Webhooks](/docs/webhooks) documentation to learn more.

## Configuration URL

- **Required**: No

To allow the developer to configure an installed integration, you can specify a **Configuration URL**. This URL is used for the **Configu

If you leave the **Configuration URL** field empty, the **Configure** button will default to a **Website** button that links to the websi

## Marketplace Integration Settings

## Base URL

- **Required: If it's a **

The URL that points to the provider's integration server that implements the [Marketplace Provider API](/docs/integrations/marketplace-ap

For example, if the base url is `https://foo.bar.com/vercel-integration-server`, Vercel makes a `POST` request to something like `https:/

## Redirect Login URL

- **Required: If it's a **

The URL where Vercel redirect users of the integration in the following situations:

- They open the link to the integration provider's dashboard from the Vercel dashboard as explained in the [Open in Provider button flow]
- They open a specific resource on the Vercel dashboard

This allows providers to automatically log users into their dashboard without asking them to log in.

## Installation-level Billing Plans

- **Required**: No (It's a toggle which is disabled by default)
- Applies to a 

When enabled, it allows the integration user to select a billing plan for their installation. The default installation-level billing plan

### Usage

If the billing for your integration happens at the team, organization or account level, enable this toggle to allow Vercel to fetch the i

The user can update this installation-level plan at any time from the installation detail page of the Vercel dashboard.

## Terms of Service

## Integrations Agreement

- **Required**:
  - **Yes**: If it's a connectable account integration or this is the first time you are creating a native integration
  - **No**: If you are adding a product to the integration. A different agreement may be needed for the first added product

You must agree to the Vercel terms before your integration can be published. The terms may differ depending the type of integration, [con

### Marketplace installation flow

**Usage Scenario**: For installations initiated from the [Vercel Marketplace](/integrations).

- **Post-Installation**: After installation, the user is redirected to a page on your side to complete the setup
- **Completion**: Redirect the user to the provided next URL to close the popup and continue

#### Query parameters for marketplace

| Name | Definition | Example |
| --- | --- | --- |
| **code** | The code you received. | `jMIukZ1DBCKXHje3X14BCkU0` |
| **teamId** | The ID of the team (only if a team is selected). | `team_LLHUOMOoDlqOp8wPE4kFo |
| **configurationId** | The ID of the configuration. | `icfg_6uKSUQ359QCbPfECTAY9m |
| **next** | Encoded URL to redirect to, once the installation process on your side is finished. | `https%3A%2F%2Fvercel.com%2 |
| **source** | Source defines where the integration was installed from. | `marketplace` |

### External installation flow

**Usage Scenario**: When you're initiating the installation from your application.

- **Starting Point**: Use this URL to start the process: `https://vercel.com/integrations/:slug/new` - `:slug` is the name you added in t

#### Query parameters for external flow

| Name | Definition | Example |
| --- | --- | --- |
| **code** | The code you received. | `jMIukZ1DBCKXHje3X |
| **teamId** | The ID of the team (only if a team is selected). | `team_LLHUOMOoDlqO |
| **configurationId** | The ID of the configuration. | `icfg_6uKSUQ359QCb |
| **next** | Encoded URL to redirect to, once the installation process on your side is finished. | `https%3A%2F%2Fver |
| **state** | Random string to be passed back upon completion. It is used to protect against CSRF attacks. | `xyzABC123` |
| **source** | Source defines where the integration was installed from. | `external` |

### Deploy button installation flow

**Usage Scenario**: For installations using the [Vercel deploy button](/docs/deploy-button).

- **Post-Installation**: The user will complete the setup on your side
- **Completion**: Redirect the user to the provided next URL to proceed

#### Query Parameters for Deploy Button

| Name | Definition | Exampl |
| --- | --- | --- |
| **code** | The code you received. | `jMIuk |
| **teamId** | The ID of the team (only if a team is selected). | `team_ |
| **configurationId** | The ID of the configuration. | `icfg_ |
| **next** | Encoded URL to redirect to, once the installation process on your side is finished. | `https |
| **currentProjectId** | The ID of the created project. | `QmXGT |
| **external-id** | Reference of your choice. See [External ID](/docs/deploy-button/callback#external-id) for more details. | `12842 |
| **source** | Source defines where the integration was installed from. | `deplo |

If the integration is already installed in the selected scope during the deploy button flow, the redirect URL will be called with the mos

Make sure to store `configurationId` along with an access token such that if an existing `configurationId` was passed, you could retrieve

## Product form fields

### Product Name

It's used as the product card title in the **Products** section of the marketplace integration page.

### Product URL Slug

It's used in the integration console for the url slug of the product's detail page.

### Product Short Description

It's used as the product card description in the **Products** section of the marketplace integration page.

### Product Short Billing Plans Description

It's used as the product card footer description in the **Products** section of the marketplace integration page and should be less than 1

### Product Metadata Schema

The [metadata schema](/docs/integrations/marketplace-product#metadata-schema) controls the product features such as available regions and

### Product Logo

It's used as the product logo at the top of the Product settings page once the integration user installs this product. If this is not set

### Product Tags

It's used to help integration users filter and group their installed products on the installed integration page.

### Product Guides

You are recommended to include links to get started guides for using your product with specific frameworks. Once your product is added by

### Product Resource Links

These links appear under the **Resources** left side bar on the product's detail page of the user's Vercel dashboard.

### Support link

Under the **Resources** section, Vercel automatically adds a **Support** link that is a deep link to the provider's dashboard with a query

### Product Snippets

These code snippets are designed to be quick starts for the integration user to connect with the installed product with tools such as `cU

You can add up to 6 code snippets to help users get started with your product. These appear at the top of the product's detail page under

You can include secrets in the following way:

```typescript
import { createClient } from 'acme-sdk';

const client = createClient('https://your-project.acme.com', '{{YOUR_SECRET}}');
```

When integration users view your snippet in the Vercel dashboard, `{{YOUR_SECRET}}` is replaced with a `*` accompanied by a **Show Secret

If you're using TypeScript or JavaScript snippets, you can use `{{process.env.YOUR_SECRET}}`. In this case, the snippet view in the Verce

### Edge Config Support

When enabled, integration users can choose an [Edge Config](/docs/edge-config) to access experimentation feature flag data.

### Log Drain Settings

When enabled, the integration user can configure a Log Drain for the Native integration. Once the `Delivery Format` is chosen, the integr

### Checks API

When enabled, the integration can use the [Checks API](/docs/checks)

---
title: "Upgrade an Integration"
description: "Lean more about when you may need to upgrade your Integration."
last_updated: "2026-01-16T02:19:32.889Z"
source: "https://vercel.com/docs/integrations/create-integration/upgrade-integration"
---

# Upgrade an Integration

You should upgrade your integration if you are using any of the following scenarios.

## Upgrading your Integration

If your Integration is using outdated features on the Vercel Platform, [follow these guidelines](/docs/integrations/create-integration/up

Once ready, make sure to [submit your Integration](/docs/integrations/create-integration/submit-integration) for review after you upgrade

## Use generic Webhooks

You can now specify a generic Webhook URL in your Integration settings. Use generic Webhooks instead of Webhooks APIs and Delete Hooks.

The Vercel REST API to list, create, and delete Webhooks [has been removed](https://vercel.com/changelog/sunsetting-ui-hooks-and-legacy-w

## Use External Flow

If your Integration is using the OAuth2 installation flow, you should use the [External installation flow](/docs/integrations/create-inte

## Use your own UI

UI Hooks is a deprecated feature that allowed you to create custom configuration UI for your Integration inside the Vercel dashboard. If

## Legacy Integrations

Integration that use UI Hooks are now [fully deprecated](https://vercel.com/changelog/sunsetting-ui-hooks-and-legacy-webhooks). Users are

If you are using a Legacy Integrations, it's recommended finding an updated Integration on the [Integrations Marketplace](https://vercel.
If adequate replacement is not available, contact the integration developer for more information.

## `currentProjectId` in Deploy Button

If your Integration is not using `currentProjectId` to determine the target project for the Deploy Button flow, please use it. [Here's th

## Single installation per scope

If your Integration assumes that it can be installed multiple times in a Vercel scope (Hobby team or team), read the following so that it

- [Marketplace flow](/docs/integrations/create-integration/marketplace-product)
- [External flow](/docs/integrations/create-integration/submit-integration#external-installation-flow)
- [Deploy Button flow](/docs/deploy-button)

## Latest API for Environment Variables

If your Integration is setting Environment Variables, please make sure to use `type=encrypted` with the latest version (v7) of the API wh

> **💡 Note:** Creating project secrets is not required anymore and will be deprecated in the
> near future.

---
title: "Vercel and BigCommerce Integration"
description: "Integrate Vercel with BigCommerce to deploy your headless storefront."
last_updated: "2026-01-16T02:19:32.744Z"
source: "https://vercel.com/docs/integrations/ecommerce/bigcommerce"
---

# Vercel and BigCommerce Integration

[BigCommerce](https://www.bigcommerce.com/) is an ecommerce platform for building and managing online storefronts. This guide explains ho

## Overview

This guide uses [Catalyst](/templates/next.js/catalyst-by-bigcommerce) by BigCommerce to connect your BigCommerce store to a Vercel deplo

> **💡 Note:** You can use this guide as a reference for creating a custom headless
> BigCommerce storefront, even if you're not using Catalyst by BigCommerce.

## Getting Started

You can either deploy the template below to **Vercel** or use the following steps to fork and clone it to your machine and deploy it loca

## Configure BigCommerce

- ### Set up a BigCommerce account and storefront
  You can use an existing BigCommerce account and storefront, or get started with one of the options below:
  - [Start a free trial](https://www.bigcommerce.com/start-your-trial/)
  - [Create a developer sandbox](https://start.bigcommerce.com/developer-sandbox/)

- ### Fork and clone the Catalyst repository
  1. [Fork the Catalyst repository on GitHub](https://github.com/bigcommerce/catalyst/fork). You can name your fork as you prefer. This g
  2. Clone your forked repository to your local machine using the following command:
  ```bash filename="Terminal"
  git clone https://github.com/<YOUR_GITHUB_USERNAME>/<YOUR_FORK_NAME>.git
  cd <YOUR_FORK_NAME>
  ```
  > **💡 Note:** Replace `<YOUR_GITHUB_USERNAME>` with your GitHub username and `<YOUR_FORK_NAME>` with the name you chose for your fork.

- ### Add the upstream Catalyst repository
  To automatically sync updates, add the BigCommerce Catalyst repository as a remote named "upstream" using the following command:
  ```bash filename="Terminal"
  git remote add upstream git@github.com:bigcommerce/catalyst.git
  ```
  Verify the local repository is set up with the remote repositories using the following command:
  ```bash filename="Terminal"
  git remote -v
  ```
  The output should look similar to this:
  ```bash filename="Terminal"
  origin    git@github.com:<YOUR_GITHUB_USERNAME><YOUR_FORK_NAME>.git (fetch)
  origin    git@github.com:<YOUR_GITHUB_USERNAME>/<YOUR_FORK_NAME>.git (push)
  upstream  git@github.com:bigcommerce/catalyst.git (fetch)
  upstream  git@github.com:bigcommerce/catalyst.git (push)
  ```
  Learn more about [syncing a fork](https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/working-with-forks/syncing-

- ### Enable Corepack and install dependencies
  Catalyst requires pnpm as the Node.js package manager. [Corepack](https://github.com/nodejs/corepack#readme) is a tool that helps manag
  ```bash filename="Terminal"
  corepack enable pnpm && pnpm install
  ```

- ### Run the Catalyst CLI command
  The Catalyst CLI (Command Line Interface) is a tool that helps set up and configure your Catalyst project. When run, it will:
  1. Guide you through logging into your BigCommerce store
  2. Help you create a new or select an existing Catalyst storefront Channel
  3. Automatically create an `.env.local` file in your project root
  To start this process, run the following command:
  ```bash filename="Terminal"
  pnpm create @bigcommerce/catalyst@latest init
  ```
  Follow the CLI prompts to complete the setup.

- ### Start the development server
  After setting up your Catalyst project and configuring the environment variables, you can start the development server. From your proje
  ```bash filename="Terminal"
  pnpm dev
  ```
  Your local storefront should now be accessible at `http://localhost:3000`.

## Deploy to Vercel

Now that your Catalyst storefront is configured, let's deploy your project to Vercel.

- ### Create a new Vercel project
  Visit https://vercel.com/new to create a new project. You may be prompted to sign in or create a new account.
  1. Find your forked repository in the list.
  2. Click the **Import** button next to your repository.
  3. In the **Root Directory** section, click the **Edit** button.
  4. Select the `core` directory from file tree. Click **Continue** to confirm your selection.
  5. Verify that the Framework preset is set to Next.js. If it isn't, select it from the dropdown menu.
  6. Open the **Environment Variables** dropdown and paste the contents of your `.env.local` into the form.
  7. Click the **Deploy** button to start the deployment process.

- ### Link your Vercel project
  To ensure seamless management of deployments and project settings you can link your local development environment with your Vercel proj

  If you haven't already, install the Vercel CLI globally with the following command:
  ```bash filename="Terminal"
  pnpm i -g vercel
  ```
  This command will prompt you to log in to your Vercel account and link your local project to your existing Vercel project:
  ```bash filename="Terminal"
  vercel link
  ```
  Learn more about the [Vercel CLI](/docs/cli).

## Enable Vercel Remote Cache

Vercel Remote Cache optimizes your build process by sharing build outputs across your Vercel team, eliminating redundant tasks. Follow th

- ### Authenticate with Turborepo
  Run the following command to authenticate the Turborepo CLI with your Vercel account:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>
  For SSO-enabled Vercel teams, include your team slug:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>

- ### Link your Remote Cache
  To link your project to a team scope and specify who the cache should be shared with, run the following command:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>

```
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>
  > **⚠️ Warning:** If you run these commands but the owner has [disabled Remote
  > Caching](#enabling-and-disabling-remote-caching-for-your-team) for your team,
  > Turborepo will present you with an error message: "Please contact your account
  > owner to enable Remote Caching on Vercel."

- ### Add Remote Cache Signature Key
  To securely sign artifacts before uploading them to the Remote Cache, use the following command to add the `TURBO_REMOTE_CACHE_SIGNATUR
  ```bash filename="Terminal"
  vercel env add TURBO_REMOTE_CACHE_SIGNATURE_KEY
  ```
  When prompted, add the environment variable to Production, Preview, and Development environments. Set the environment variable to a sec

  Once finished, pull the new environment variable into your local project with the following command:
  ```bash filename="Terminal"
  vercel env pull
  ```
  Learn more about [Vercel Remote Cache](/docs/monorepos/remote-caching#vercel-remote-cache).

## Enable Web Analytics and Speed Insights

The Catalyst monorepo comes pre-configured with Vercel Web Analytics and Speed Insights, offering you powerful tools to understand and op

Web Analytics provides real-time insights into your site's traffic and user behavior,
helping you make data-driven decisions to improve your storefront's performance:

Speed Insights offers detailed performance metrics and suggestions to optimize your
site's loading speed and overall user experience:

For more advanced configurations or to learn more about BigCommerce Catalyst, refer to the [BigCommerce Catalyst documentation](https://c


---
title: "Vercel Ecommerce Integrations"
description: "Learn how to integrate Vercel with ecommerce platforms, including BigCommerce and Shopify."
last_updated: "2026-01-16T02:19:32.756Z"
source: "https://vercel.com/docs/integrations/ecommerce"
---

# Vercel Ecommerce Integrations

Vercel Ecommerce Integrations allow you to connect your projects with ecommerce platforms, including [BigCommerce](/docs/integrations/eco

## Featured Ecommerce integrations

- [**BigCommerce**](/docs/integrations/ecommerce/bigcommerce)
- [**Shopify**](/docs/integrations/ecommerce/shopify)


---
title: "Vercel and Shopify Integration"
description: "Integrate Vercel with Shopify to deploy your headless storefront."
last_updated: "2026-01-16T02:19:32.845Z"
source: "https://vercel.com/docs/integrations/ecommerce/shopify"
---

# Vercel and Shopify Integration

[Shopify](https://www.shopify.com/) is an ecommerce platform that allows you to build and manage online storefronts. Shopify does offer t

This guide uses the [Next.js Commerce template](/templates/ecommerce/nextjs-commerce) to connect your Shopify store to a Vercel deploymen

To finish, the important parts that you need to know are:

- [Configure Shopify for use as a headless CMS](#configure-shopify)
- [Deploy your headless storefront on Vercel](#deploy-to-vercel)
- [Configure environment variables](#configure-environment-variables)

> **💡 Note:** Even if you are not using Next.js Commerce, you can still use this guide as a
> roadmap to create your own headless Shopify theme.

## Getting started

To help you get started, we built a [template](/templates/ecommerce/nextjs-commerce) using Next.js, Shopify, and Tailwind CSS.

You can either deploy the template above to Vercel or use the steps below to clone it to your machine and deploy it locally.

## Configure Shopify

- ### Create a Shopify account and storefront
  If you have an existing Shopify account and storefront, you can use it with the rest of these steps.

  If you do not have an existing Shopify account and storefront, you'll need to [create one](https://www.shopify.com/signup).
  > **💡 Note:** Next.js Commerce will not work with a Shopify Starter plan as it does not
  > allow installation of custom themes, which is required to run as a headless
  > storefront.

- ### Install the Shopify Headless theme
  To use Next.js Commerce as your headless Shopify theme, you need to install the [Shopify Headless theme](https://github.com/instantcomm
```

  Download [Shopify Headless Theme](https://github.com/instantcommerce/shopify-headless-theme).

  Navigate to `https://[your-shopify-store-subdomain].myshopify.com/admin/themes`, click `Add theme`, and then `Upload zip file`.

  Select the downloaded zip file from above, and click the green `Upload file` button.

  Click `Customize`.

  Click `Theme settings` (the paintbrush icon), expand the `STOREFRONT` section, enter your headless store domain, click the gray `Publis

  Confirm the theme change by clicking the green `Save and publish` button.

  The headless theme should now be your current active theme.

- ### Install the Shopify Headless app
  Shopify provides a [Storefront API](https://shopify.dev/docs/api/storefront) which allows you to fetch products, collections, pages, an

  Navigate to `https://[your-shopify-store-subdomain].myshopify.com/admin/settings/apps` and click the green `Shopify App Store` button.

  Search for `Headless` and click on the `Headless` app.

  Click the black `Add app` button.

  Click the green `Add sales channel` button.

  Click the green `Create storefront` button.

  Copy the public access token as it will be used when we [configure environment variables](#configure-environment-variables).

  If you need to reference the public access token again, you can navigate to `https://[your-shopify-store-subdomain].myshopify.com/admin

- ### Configure your Shopify branding and design
  Even though you're creating a headless store, there are still a few aspects Shopify will control.
  - Checkout
  - Emails
  - Order status
  - Order history
  - Favicon (for any Shopify controlled pages)
  You can use Shopify's admin to customize these pages to match your brand and design.

- ### Customize checkout, order status, and order history
  Navigate to `https://[your-shopify-store-subdomain].myshopify.com/admin/settings/checkout` and click the green `Customize` button.

  Click `Branding` (the paintbrush icon) and customize your brand.
  > **💡 Note:** There are three steps / pages to the checkout flow. Use the dropdown to change
  > pages and adjust branding as needed on each page. Click `Save` when you are
  > done.
  Navigate to `https://[your-shopify-store-subdomain].myshopify.com/admin/settings/branding` and customize settings to match your brand.

- ### Customize emails
  Navigate to `https://[your-shopify-store-subdomain].myshopify.com/admin/settings/email_settings` and customize settings to match your b

- ### Customize favicon
  Navigate to `https://[your-shopify-store-subdomain].myshopify.com/admin/themes` and click the green `Customize` button.

  Click `Theme settings` (the paintbrush icon), expand the `FAVICON` section, upload favicon, then click the `Save` button.

- ### Configure Shopify webhooks
  Utilizing [Shopify's webhooks](https://shopify.dev/docs/apps/webhooks), and listening for select [Shopify webhook event topics](https:/

  Next.js Commerce is pre-configured to listen for the following Shopify webhook events and automatically revalidate fetches.
  - `collections/create`
  - `collections/delete`
  - `collections/update`
  - `products/create`
  - `products/delete`
  - `products/update` (this includes when variants are added, updated, and removed as well as when products are purchased so inventory an

- ### Create a secret for secure revalidation
  Create your own secret or [generate a random UUID](https://www.uuidgenerator.net/guid).

  This secret value will be used when we [configure environment variables](#configure-environment-variables).

- ### Configure Shopify webhooks in the Shopify admin
  Navigate to `https://[your-shopify-store-subdomain].myshopify.com/admin/settings/notifications` and add webhooks for all six event topi

  You can add more sets for other preview urls, environments, or local development. Append `?secret=[your-secret]` to each url, where `[y

- ### Testing webhooks during local development
  [ngrok](https://ngrok.com) is the easiest way to test webhooks while developing locally.
  - [Install and configure ngrok](https://ngrok.com/download) (you will need to create an account).
  - Run your app locally, `npm run dev`.
  - In a separate terminal session, run `ngrok http 3000`.
  - Use the url generated by ngrok and add or update your webhook urls in Shopify.
  You can now make changes to your store and your local app should receive updates. You can also use the `Send test notification` button

### Using Shopify as a full-featured CMS

Next.js Commerce is fully powered by Shopify in every way. All products, collections, pages header and footer menus, and SEO are controll

#### Products

Navigate to `https://[your-shopify-store-subdomain].myshopify.com/admin/products` to mange your products.

- Only `Active` products are shown. `Draft` products will not be shown until they are marked as `Active`.
- `Active` products can still be hidden and not seen by navigating the site, by adding a `nextjs-frontend-hidden` tag on the product. Thi
- Product options and option combinations are driven from Shopify options and variants. When selecting options on the product detail page
- Products that are `Active` but no quantity remaining will still be displayed on the site, but will be marked as "out of stock". The abi

#### Collections

Navigate to `https://[your-shopify-store-subdomain].myshopify.com/admin/collections` to manage your collections.

All available collections will show on the search page as filters on the left, with one exception.

Any collection names that start with the word `hidden` will not show up on the headless front end. Next.js Commerce comes pre-configured

Create the following collections:

- `Hidden: Homepage Featured Items` — Products in this collection are displayed in the three featured blocks on the homepage.
- `Hidden: Homepage Carousel` — Products in this collection are displayed in the auto-scrolling carousel section on the homepage.

#### Pages

Navigate to `https://[your-shopify-store-subdomain].myshopify.com/admin/pages` to manage your pages.

Next.js Commerce contains a dynamic `[page]` route. It will use the value to look for a corresponding page in Shopify.

- If a page is found, it will display its rich content using [Tailwind's typography plugin](https://tailwindcss.com/docs/typography-plugi
- If a page is not found, a `404` page is displayed.

#### Navigation menus

`https://[your-shopify-store-subdomain].myshopify.com/admin/menus`

Next.js Commerce's header and footer navigation is pre-configured to be controlled by Shopify navigation menus. They can be to collection

Create the following navigation menus:

- `Next.js Frontend Header Menu` — Menu items to be shown in the headless frontend header.
- `Next.js Frontend Footer Menu` — Menu items to be shown in the headless frontend footer.

#### SEO

Shopify's products, collections, pages, etc. allow you to create custom SEO titles and descriptions. Next.js Commerce is pre-configured t

## Deploy to Vercel

Now that your Shopify store is configured, you can deploy your code to Vercel.

### Clone the repository

You can clone the repo using the following command:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i
    ```
  </Code>
</CodeBlock>

### Publish your code

Publish your code to a Git provider like GitHub.

```shell
git init
git add .
git commit -m "Initial commit"
git remote add origin https://github.com/your-account/your-repo
git push -u origin main
```

### Import your project

Import the repository into a [new Vercel project](/new).

Vercel will automatically detect you are using Next.js and configure the optimal build settings.

### Configure environment variables

Create [Vercel Environment Variables](/docs/environment-variables) with the following names and values.

- `COMPANY_NAME` *(optional)* — Displayed in the footer next to the copyright in the event the company is different from the site name, f
- `SHOPIFY_STORE_DOMAIN` — Used to connect to your Shopify storefront, for example `[your-shopify-store-subdomain].myshopify.com`
- `SHOPIFY_STOREFRONT_ACCESS_TOKEN` — Used to secure API requests between Shopify and your headless site, which was created when you [ins
- `SHOPIFY_REVALIDATION_SECRET` — Used to secure data revalidation requests between Shopify and your headless site, which was created whe
- `SITE_NAME` — Displayed in the header and footer navigation next to the logo, for example `Acme Store`
- `TWITTER_CREATOR` — Used in Twitter OG metadata, for example `@nextjs`
- `TWITTER_SITE` — Used in Twitter OG metadata, for example `https://nextjs.org`

You can [use the Vercel CLI to setup your local development environment variables](/docs/environment-variables#development-environment-va

---

title: "Integrating Vercel and Kubernetes"
description: "Deploy your frontend on Vercel alongside your existing Kubernetes infrastructure."
last_updated: "2026-01-16T02:19:32.787Z"
source: "https://vercel.com/docs/integrations/external-platforms/kubernetes"
---

# Integrating Vercel and Kubernetes

Kubernetes (K8s) is an open-source system for automating deployment, scaling, and management of containerized applications. It has become

You can integrate Vercel with your existing Kubernetes infrastructure to optimize the delivery of your frontend applications—reducing the

Let’s look at key Kubernetes concepts and how Vercel’s [managed infrastructure](/products/managed-infrastructure) handles them:



## Server management and provisioning

With Kubernetes, you must define and configure a web server (e.g. Nginx), resources (CPU, memory), and networking (ingress, API Gateway,

Vercel manages server provisioning for you. Through [framework-defined infrastructure](/blog/framework-defined-infrastructure) and suppor

## Scaling and redundancy

In a self-managed Kubernetes setup, you manually configure your Kubernetes cluster to scale horizontally (replicas) or vertically (resour

In addition to scaling, you may need to deploy your Kubernetes clusters to multiple regions to improve the availability, disaster recover

Vercel automatically scales your applications based on end-user traffic. Vercel deploys your application globally on our [CDN](/docs/cdn)

## Managing environments and deployments

Managing the container lifecycle and promoting environments in a self-managed ecosystem typically involves three parts:

- **Containerization (Docker)**: Packages applications and their dependencies into containers to ensure consistent environments across de
- **Container orchestration (Kubernetes)**: Manages containers (often Docker containers) at scale. Handles deployment, scaling, and netwo
- **Infrastructure as Code (IaC) tool (Terraform)**: Provisions and manages the infrastructure (cloud, on-premises, or hybrid) in a consi

These parts work together by Docker packaging applications into containers, Kubernetes deploying and managing these containers across a c

Vercel knows how to automatically configure your environment through our [framework-defined infrastructure](/blog/framework-defined-infra

Once you connect a Vercel project to a Git repository, every push to a branch automatically creates a new deployment of your application v

Every deploy is immutable, and these generated domains act as pointers. Reverting and deploying is an atomic swap operation. These infras

## Managing access and security

In a Kubernetes environment, you need to implement security measures such as Role-Based Access Control (RBAC), network policies, secrets m

With Vercel, you can securely configure [environment variables](/docs/environment-variables) and manage [user access, roles, and permissi

## Observability

A Kubernetes setup typically uses observability solutions to aid in troubleshooting, alerting, and monitoring of your applications. You c

Vercel provides built-in logging and monitoring capabilities through our [observability products](/docs/observability) with real-time log

## Integrating Vercel with your Kubernetes backend

If you’re running backend services on Kubernetes (e.g., APIs, RPC layers, data processing jobs), you can continue doing so while offloadi

- **Networking**: Vercel can securely connect to your Kubernetes-hosted backend services. You can keep your APIs behind load balancers or
- **Environment Variables and Secrets**: Your application’s environment variables (e.g., API keys, database credentials) can be configure
- **Observability**: You can maintain your existing observability setup for Kubernetes (Grafana, DataDog, etc.) while also leveraging Ver

## Before/after comparison: Kubernetes vs. Vercel

Here's how managing frontend infrastructure compares between traditional, self-managed Kubernetes and Vercel's fully managed frontend sol

| **Capability** | **Kubernetes (Self-managed)** | **Ve |
| --- | --- | --- |
| **Server Provisioning** | Manual setup of Nginx, Node.js pods, ingress, load balancing, and networking policies | Auto |
| **Autoscaling** | Manual configuration required (horizontal/vertical scaling policies) | Fully |
| **Availability (Multi-region)** | Manually set up multi-region clusters for redundancy and latency | Buil |
| **Deployment & Rollbacks** | Rolling updates can cause downtime (version skew) | Zero |
| **Runtime & OS Security Patches** | Manual and ongoing maintenance | Auto |
| **Multi-region Deployment & Failover** | Manual setup, configuration, and management | Auto |
| **Version Skew Protection** | Manual rolling deployments (possible downtime) | Buil |
| **Observability & Logging** | Requires third-party setup (Grafana, Splunk, DataDog) | Buil |
| **CI/CD & Deployment Management** | Requires integration of multiple tools (Docker, Kubernetes, Terraform, CI/CD pipelines) | Buil |

By migrating just your frontend to Vercel, you drastically reduce the operational overhead of managing and scaling web servers, pods, loa

## Migrating from Kubernetes to Vercel

To incrementally move your frontend applications to Vercel:

- ### Create a Vercel account and team
  Start by [creating a Vercel account](/signup) and [team](/docs/accounts/create-a-team), if needed.

- ### Create two versions of your frontend codebase
  Keep your current frontend running in Kubernetes for now. Create a fork or a branch of your frontend codebase and connect it to a [new

  Once connected, Vercel will automatically build and deploy your application. It’s okay if the first deployment fails. [View the build l
  - Adjustments to build scripts
  - Changes to the [project configuration](/docs/project-configuration)
  - Missing [environment variables](/docs/environment-variables)
  Continue addressing errors until you get a successful Preview Deployment.

  Depending on how you have your Kubernetes environment configured, you may need to adjust firewall and security policies to allow the ap

  The goal is to use the Preview Deployment to test the integration with your Kubernetes-hosted backends, ensuring that API calls and dat

- ### Set up users and integrations
  Use [Vercel’s dashboard](/dashboard) to securely manage [user access, roles, and permissions](/docs/accounts/team-members-and-roles), s
  - [Add team members and assign roles](/docs/rbac/managing-team-members#adding-team-members-and-assigning-roles) ([SAML SSO](/docs/saml)
  - [Add integrations](/integrations) to any existing services and tools your team uses

- ### Begin a full or gradual rollout
  Once your preview deployment is passing all tests, and your team is happy with it, you can start to roll it out.

  We recommend following our [incremental migration guide](/docs/incremental-migration/migration-guide) or our [Vercel Adoption](/resourc

  Some other tools or strategies you may want to use:
  - [Feature Flags on Vercel](/docs/feature-flags)
  - [A/B Testing on Vercel](/kb/guide/ab-testing-on-vercel)
  - [Implementing Blue-Green Deployments on Vercel](/kb/guide/blue_green_deployments_on_vercel)
  - [Transferring Domains to Vercel](/kb/guide/transferring-domains-to-vercel)
  - [How to migrate a site to Vercel without downtime](/kb/guide/zero-downtime-migration)

- ### Maintain the backend on Kubernetes
  Continue running your backend services on Kubernetes, taking advantage of its strengths in container orchestration for applications you
  - APIs
  - Remote Procedure Calls (RPC)
  - Change Data Captures (CDC)
  - Extract Transfer Loads (ETL)
  Over time, you can evaluate whether specific backend services could also benefit from a serverless architecture and be migrated to Verc

- ### Accelerate frontend iteration velocity on Vercel
  With Vercel, your development processes become simpler and faster. Vercel combines all the tools you need for CI/CD, staging, testing,

  A [recent study](/roi) found Vercel customers see:
  - Up to 90% increase in site performance
  - Up to 80% reduction in time spent deploying
  - Up to 4x faster time to market


---
title: "Add a Connectable Account"
description: "Learn how to connect Vercel to your third-party account."
last_updated: "2026-01-16T02:19:32.792Z"
source: "https://vercel.com/docs/integrations/install-an-integration/add-a-connectable-account"
---

# Add a Connectable Account

## Add a connectable account

1. From the [Vercel dashboard](/dashboard), select the **Integrations** tab and then the **Browse Marketplace** button. You can also go d
2. Under the **Connectable Accounts** section, select an integration that you would like to install. The integration page provides inform
3. From the integration's detail page, select **Connect Account**.
4. From the dialog that appears, select which projects the integration will have access to. Select **Install**.
5. Follow the prompts to sign-in to your third-party account and authorize the connection to Vercel. Depending on the integration, you ma

## Manage connectable accounts

Once installed, you can manage the following aspect of the integration:

- [View all the permissions](/docs/integrations/install-an-integration/manage-integrations-reference)
- [Manage access to your projects](/docs/integrations/install-an-integration/manage-integrations-reference#manage-project-access)
- [Uninstall the integration](/docs/integrations/install-an-integration/add-a-connectable-account#uninstall-a-connectable-account)

To manage the installed integration:

1. From your Vercel Dashboard, select the [**Integrations tab**](/dashboard/integrations).
2. Click the **Manage** button next to the installed Integration.
3. This will take you to the Integration page from where you can see permissions, access, and uninstall the integration.

If you need addition configurations, you can also select the **Configure** button on the integration page to go to the third-party servic

### Uninstall a connectable account

To uninstall an integration:

1. From your Vercel [dashboard](/dashboard), go to the **Integrations** tab
2. Next to the integration, select the **Manage** button
3. On the integrations page, select **Settings**, then select **Uninstall Integration** and follow the steps to uninstall.


---
title: "Interact with Integrations using Agent Tools"
description: "Use Agent Tools to query, debug, and manage your installed integrations through a chat interface with natural language."
last_updated: "2026-01-16T02:19:32.799Z"
source: "https://vercel.com/docs/integrations/install-an-integration/agent-tools"
---

# Interact with Integrations using Agent Tools

With Agent Tools, you can interact with your installed integrations through a chat interface in the Vercel Dashboard. Instead of navigati

When you install an integration from the Marketplace, any tools that the provider has enabled via MCP (Model Context Protocol) become ava

## What you can do with Agent Tools

You can use the chat interface to:

- Query databases and view table structures
- Run SQL queries on your data
- Inspect cache contents and performance metrics
- Fetch logs for debugging
- Trigger test events in your services
- Manage media assets and check processing status

This works with installed native integrations that provide tools through the MCP standard, including Neon, Prisma, Supabase, Dash0, Strip

## Access Agent Tools

To use Agent Tools:

1. From the [Vercel Dashboard](/dashboard), make sure you have at least one native integration installed. See [Add a Native Integration](
2. Navigate to the **Integrations** tab in your dashboard.
3. Select an integration that supports Agent Tools.
4. Click on **Agent Tools** in the left navigation to open the chat interface.
5. Your installed integration's tools load automatically and are ready to use.

## Read-Only Mode

Agent Tools includes a **Read-Only Mode** toggle that is enabled by default. When enabled, you can query and view data, but cannot perfor

This is useful for:

- Safely exploring your data without risk of accidental changes
- Allowing team members to investigate issues without write access
- Demonstrating integrations without modifying production data

To disable Read-Only Mode, click the toggle at the bottom of the Agent Tools interface. Be aware that this will allow the agent to create

## Interact with your integrations

Type natural language questions or commands in the chat interface. The agent understands what you're trying to do and routes your request

Here are some examples of queries you can try:

- "Show me all my tables in this Neon database"
- "Run my Supabase SQL query"
- "Fetch my Dash0 logs"
- "Trigger a Stripe test event"

The specific tools and capabilities available depend on what each provider has enabled. You can ask questions about your data, run querie

## Supported integrations

Agent Tools is currently enabled for the following integrations: [Neon](https://vercel.com/marketplace/neon), [Prisma](https://vercel.com

## Next steps

- [Learn how to add a native integration](/docs/integrations/install-an-integration/product-integration) to your project

---
title: "Permissions and Access"
description: "Learn how to manage project access and added products for your integrations."
last_updated: "2026-01-16T02:19:32.902Z"
source: "https://vercel.com/docs/integrations/install-an-integration/manage-integrations-reference"
---

# Permissions and Access

## View an integration's permissions

To view an integration's permissions:

1. From your Vercel [dashboard](/dashboard), go to the **Integrations** tab.
2. Next to the integration, select the **Manage** button.
3. On the Integrations detail page, scroll to **Permissions** section at the bottom of the page.

## Permission Types

Integration permissions restrict how much of the API the integration is allowed to access. When you install an integration, you will see

| **Permission Type** | **Read Access** |
| --- | --- |
| **Installation** | Reads whether the integration is installed for the hobby or team account |
| **Deployment** | Retrieves deployments for the hobby or team account. Includes build logs, a list of files an |
| **Deployment Checks** | N/A |
| **Project** | Retrieves projects for the hobby or team account. Also includes retrieving all domains for a |
| **Project Environment Variables** | N/A |
| **Global Project Environment Variables** | N/A |
| **Team** | Accesses team details for the account. Includes listing team members |
| **Current User** | Accesses information about the Hobby team on which the integration is installed |
| **Log Drains** | N/A |
| **Domain** | Retrieves all domains for the hobby or team account. Includes reading its status and configu |

## Confirming Permission Changes

Integrations can request more permissions over time.
Individual users and team owners are [notified](/docs/notifications#notification-details) by Vercel when an integration installation has

## Manage project access

To manage which projects the installed integration has access to:

1. From your Vercel [dashboard](/dashboard), go to the **Integrations** tab.
2. Next to the integration, select the **Manage** button.
3. On the Integrations page, under **Access**, select the **Manage Access** button.
4. From the dialog, select the option to manage which projects have access.

### Disabled integrations

Every integration installed for a team creates an access token that is associated with the developer who originally installed it. If the

When an integration is disabled, team owners must take action by clicking **Manage** and either changing ownership or removing the integr

> **💡 Note:** If a disabled integration is not re-enabled, it will be automatically removed
> after 30 days. Any environment variables that were created by that integration
> will also be removed - this may prevent new deployments from working.

When an integration is `disabled`:

- The integration will no longer have API access to your team or account
- If the integration has set up log drains, then logs will cease to flow
- The integration will no longer receive the majority of webhooks, other than those essential to its operation (`project.created`, `proje

If you are an integrator, see the [disabled integration configurations](/docs/rest-api/vercel-api-integrations#disabled-integration-confi

## Invoice access

Only users with **Owner** or **Billing** roles can view invoices for native integrations. See [Billing](/docs/integrations/create-integra

---
title: "Extend your Vercel Workflow"
description: "Learn how to pair Vercel"
last_updated: "2026-01-16T02:19:32.908Z"
source: "https://vercel.com/docs/integrations/install-an-integration"
---

# Extend your Vercel Workflow

## Installing an integration

Using Vercel doesn't stop at the products and features that we provide. Through integrations, you can use third-party platforms or servic

- Connecting your Vercel account and project with a third-party service. See [Add a connectable account](/docs/integrations/install-an-in
- Buying or subscribing to a product with a third-party service that you will use with your Vercel project. see [Add a Native Integration
- Interacting with your installed integrations through a chat interface. See [Agent Tools](/docs/integrations/install-an-integration/agen

## Find integrations

You can extend the Vercel platform through the [Marketplace](#marketplace), [templates](#templates), or [third-party site](#third-party-s

### Marketplace

The [Integrations Marketplace](https://vercel.com/integrations) is the best way to find suitable integrations that fit into a variety of

You have access to two types of integrations:

- **Native integrations** that include  that you can buy and use in your Vercel project after you installed the integration
- **Connectable accounts** that allow you to connect third-party services to your Vercel project

Once installed, you can interact with native integrations through [Agent Tools](/docs/integrations/install-an-integration/agent-tools).

- [Permissions and Access](/docs/integrations/install-an-integration/manage-integrations-reference)
- [Add a Native Integration](/docs/integrations/install-an-integration/product-integration)
- [Billing](/docs/integrations/create-integration/billing)
- [Agent Tools](/docs/integrations/install-an-integration/agent-tools)

### Templates

You can use one of our verified and pre-built [templates](/templates) to learn more about integrating your favorite tools and get a quick

### Third-party site

Integration creators can prompt you to install their Vercel Integration through their app or website.

When installing or using an integration, your data may be collected or
disclosed to Vercel. Your information may also be sent to the integration
creator per our [Privacy Notice](/legal/privacy-policy). Third party
integrations are available "as is" and not operated or controlled by Vercel.
We suggest reviewing the terms and policies for the integration and/or
contacting the integration creator directly for further information on their
privacy practices.

---
title: "Add a Native Integration"
description: "Learn how you can add a product to your Vercel project through a native integration."
last_updated: "2026-01-16T02:19:32.918Z"
source: "https://vercel.com/docs/integrations/install-an-integration/product-integration"
---

# Add a Native Integration

## Add a product

1. From the [Vercel dashboard](/dashboard), select the **Integrations** tab and then the **Browse Marketplace** button. You can also go d
2. Under the **Native Integrations** section, select an integration that you would like to install. You can see the details of the integr
3. From the integration's detail page, select **Install**.
4. Review the dialog showing the products available for this integration and a summary of the billing plans for each. Select **Install**.
5. Then, select a pricing plan option and select **Continue**. The specific options available in this step depend on the type of product
6. Provide additional information in the next step like **Database Name**. Review the details and select **Create**. Once the integration

## Manage native integrations

Once installed, you can manage the following aspect of the native integration:

- View the installed resources (instances of products) and then manage each resource.
- Connect project(s) to a provisioned resource. For products supporting Log Drains, you can enable them and configure which log sources t
- View the invoices and usage for each of your provisioned resources in that installation. See [Billing](/docs/integrations/create-integr
- [Uninstall the integration](/docs/integrations/install-an-integration/product-integration#uninstall-an-integration)

### Manage products

To manage products inside the installed integration:

1. From your Vercel [dashboard](/dashboard), go to the **Integrations** tab.
2. Next to the integration, select the **Manage** button. Native integrations appear with a `billable` badge.
3. On the Integrations page, under **Installed Products**, select the card for the product you would like to update to be taken to the pr

#### Projects

By selecting the **Projects** link on the left navigation, you can:

- Connect a project to the product
- View a list of existing connections and manage them

#### Settings

By selecting the **Settings** link on the left navigation, you can update the following:

- Product name
- Manage funds: if you selected a prepaid plan for the product, you can **Add funds** and manage auto recharge settings
- Delete the product

#### Getting Started

By selecting the **Getting Started** link on the left navigation, you can view quick steps with sample code on how to use the product in

#### Usage

By selecting the **Usage** link on the left navigation, you can view a graph of the funds used over time by this product in all the proje

#### Resources

Under **Resources** on the left navigation, you can view a list of links which vary depending on the provider for support, guides and add

### Add more products

To add more products to this integration:

1. From your Vercel [dashboard](/dashboard), go to the **Integrations** tab.
2. Next to the integration, select the **Manage** button. Native integrations appear with a `billable` badge.
3. On the Integrations page, under **More Products**, select the **Install** button for the any additional products in that integration t

### Uninstall an integration

Uninstalling an integration automatically removes all associated products and their data.

1. From your Vercel [dashboard](/dashboard), go to the **Integrations** tab.
2. Next to the integration, select the **Manage** button.
3. At the bottom of the integrations page, under **Uninstall**, select **Uninstall Integration** and follow the steps to uninstall.

## Use deployment integration actions

If available in the integration you want to install, [deployment integration actions](/docs/integrations/create-integration/deployment-in

1. Navigate to the integration and use **Install Product** or use an existing provisioned resource.
2. Open the **Projects** tab for the provisioned resource, click **Connect Project** and select the project for which to configure deploy
3. When you create a deployment (with a Git pull request or the Vercel CLI), the configured actions will execute automatically.

## Best practices

- Plan your product strategy: Decide whether you need separate products for different projects or environments:
  - Single resource strategy: For example, a small startup can use a single storage instance for all their Vercel projects to simplify ma
  - Per-project resources strategy: For example, an enterprise with multiple product lines can use separate storage instances for each pr
  - Environment-specific resources strategy: For example, a company can use different storage instances for each environment to ensure da
- Monitor Usage: Take advantage of per-product usage tracking to optimize costs and performance by using the **Usage** and **Invoices** t

---
title: "Vercel Integrations"
description: "Learn how to extend Vercel"
last_updated: "2026-01-16T02:19:32.928Z"
source: "https://vercel.com/docs/integrations"
---

# Vercel Integrations

Integrations allow you to extend the capabilities of Vercel by connecting with third-party platforms or services to do things like:

- Work with [storage](/docs/storage) products from third-party solutions
- Connect with external [AI](/docs/ai) services
- Send logs to services
- Integrate with testing tools
- Connect your CMS and ecommerce platform

To extend and automate your workflow, the [Vercel Marketplace](https://vercel.com/marketplace) page provides you with two types of integr

- [Native integrations](/docs/integrations#native-integrations)
- [Connectable accounts](/docs/integrations#connectable-accounts)

## Native integrations

Native integrations allow a two-way connection between Vercel and third-parties Vercel has partnered with. These native integrations prov

Native integrations provide the following benefits:

- You **don't** have to create an account on the integration provider's site.
- For each available , you can choose the billing plan suitable for your needs through the Vercel dashboard.
- The billing is managed through your Vercel account.

### Get started with native integrations

As a Vercel customer:

- [**Extend your Vercel workflow**](/docs/integrations/install-an-integration/product-integration): You can install an integration from t
- View the [list of available native integrations](#native-integrations-list).
- [**Add an AI provider**](/docs/ai/adding-a-provider): You can add a provider to your Vercel workflow.
- [**Add an AI model**](/docs/ai/adding-a-model): You can add a model to your Vercel workflow.

As a Vercel provider:

- [**Integrate with Vercel**](/docs/integrations/create-integration/native-integration): You can create an integration and make different

## Connectable accounts

These integrations allow you to connect Vercel with an existing account on a third-party platform or service and provide you with feature

When you add a connectable account integration through the Vercel dashboard, you are prompted to log in to your account on the third-part

### Get started with connectable account integrations

- [**Add a connectable account**](/docs/integrations/install-an-integration/add-a-connectable-account): As a Vercel customer, you can int
- [**Integrate with Vercel**](/docs/integrations/create-integration): You can extend the Vercel platform through traditional integrations
- View the [list of available connectable account integrations](#connectable-account-integrations-list).

## Native integrations list

## Connectable account integrations list

## Integrations guides

- [Contentful](/docs/integrations/cms/contentful)
- [Sanity](/docs/integrations/cms/sanity)
- [Sitecore XM Cloud](/docs/integrations/cms/sitecore)
- [Shopify](/docs/integrations/ecommerce/shopify)
- [Kubernetes](/docs/integrations/external-platforms/kubernetes)

---
title: "Building Integrations with Vercel REST API"
description: "Learn how to use Vercel REST API to build your integrations and work with redirect URLs."
last_updated: "2026-01-16T02:19:32.951Z"
source: "https://vercel.com/docs/integrations/vercel-api-integrations"
---

# Building Integrations with Vercel REST API

## Using the Vercel REST API

See the following API reference documentation for how to use Vercel REST API to create integrations:

- [Creating a Project Environment Variable](/docs/rest-api/reference/endpoints/projects/create-one-or-more-environment-variables)
- [Forwarding Logs using Log Drains](/docs/drains/reference/logs)
- [Create an Access Token](/docs/rest-api/vercel-api-integrations#create-an-access-token)
- [Interacting with Teams](/docs/rest-api/vercel-api-integrations#interacting-with-teams)
- [Interacting with Configurations](/docs/rest-api/vercel-api-integrations#interacting-with-configurations)
- [Interacting with Vercel Projects](/docs/rest-api/vercel-api-integrations#interacting-with-vercel-projects)

### Create an Access Token

To use Vercel REST API, you need to authenticate with an [access token](/docs/rest-api/reference/welcome#authentication) that contains th

#### Exchange `code` for Access Token

When you create an integration, you define a [redirect URL](/docs/integrations/create-integration/submit-integration#redirect-url) that c

One of these parameters is the `code` parameter. This short-lived parameter is valid for **30 minutes** and can be exchanged **once** for

```bash filename="terminal"
{`POST https://api.vercel.com/v2/oauth/access_token`}
```

Pass the following values to the request body in the form of `application/x-www-form-urlencoded`.

| Key | | Required | Description | |
| --- | --- | --- | --- | --- |
| **client_id** | | Yes | ID of your application. | |
| **client_secret** | | Yes | Secret of your application. | |
| **code** | | Yes | The code you received. | |
| **redirect_uri** | | Yes | The Redirect URL you configured on the Integration Console. | |

#### Example Request

### Interacting with Teams

The response of your `code` exchange request includes a `team_id` property. If `team_id` is not null, you know that this integration was

If your integration is installed on a team, append the `teamId` query parameter to each API request. See [Accessing Resources Owned by a

### Interacting with Configurations

Each installation of your integration is stored and tracked as a configuration.

Sometimes it makes sense to fetch the configuration in order to get more insights about the current scope or the projects your integratio

To see which endpoints are available, see the [Configurations](/docs/project-configuration) documentation for more details.

#### Disabled Integration Configurations

When integration configurations are disabled:

- Any API requests will fail with a `403` HTTP status code and a `code` of `integration_configuration_disabled`
- We continue to send `project.created`, `project.removed` and `integration-configuration.removed` webhooks, as these will allow the inte
- Log drains will not receive any logs

### Interacting with Vercel Projects

Deployments made with Vercel are grouped into Projects. This means that each deployment is assigned a name and is grouped into a project

Using the Vercel REST API, you can modify Projects that the Integration has access to. Here are some examples:

### Modifying Environment Variables on a Project

When building a Vercel Integration, you may want to expose an API token or a configuration URL for deployments within a [Project](/docs/p

You can do so by [Creating a Project Environment Variable](/docs/rest-api/reference/endpoints/projects/create-one-or-more-environment-var

> **💡 Note:** Environment Variables created by an Integration will.

## Scopes

When creating integrations the following scopes can be updated within the Integration Console:

> **💡 Note:** Write permissions are required for both
>  and  when
> updating the domain of a project.

| Scope | Description |
| --- | --- |
| integration-configuration | Interact with the installation of your integration |
| deployment | Interact with deployments |
| deployment-check | Verify deployments with Checks |
| edge-config | Create and manage Edge Configs and their tokens |
| project | Access project details and settings |
| project-env-vars | Create and manage integration-owned project environment variables |
| global-project-env-vars | Create and manage all account project environment variables |
| team | Access team details |
| user | Get information about the current user |
| log-drain | Create and manage log drains to forward logs |
| domain | Manage and interact with domains and certificates. Write permissions are required for both  and  when updat |

### Updating Scopes

As the Vercel REST API evolves, you'll need to update your scopes based on your integration's endpoint usage.

Additions and upgrades always require a review and confirmation. To ensure this, every affected user and team owner will be informed thro
Please make sure you provide a meaningful, short, and descriptive note for your changes.

Scope removals and downgrades won't require user confirmation and will be applied **immediately** to confirmed scopes and pending request

### Confirmed Scope Changes

User and Teams will always confirm **all pending changes** with one confirmation.
That means that if you have requested new scopes multiple times over the past year, the users will see a summary of all pending changes w

Once a user confirms these changes, scopes get directly applied to the installation. You will also get notified through the new `integrat

## Common Errors

When using the Vercel REST API with Integrations, you might come across some errors which you can address immediately.

### CORS issues

To avoid CORS issues, make sure you only interact with the Vercel REST API on the **server side**.

Since the token grants access to resources of the Team or Personal Account, you should never expose it on the client side.

For more information on using CORS with Vercel, see [How can I enable CORS on Vercel?](/kb/guide/how-to-enable-cors).

### 403 Forbidden responses

Ensure you are not missing the `teamId` [query parameter](/docs/integrations/create-integration/submit-integration#redirect-url). `teamId
Ensure the Scope of Your [Access Token](/docs/rest-api/vercel-api-integrations#using-the-vercel-api/scopes/teams) is properly set.

---
title: "Fair use Guidelines"
description: "Learn about all subscription plans included usage that is subject to Vercel"
last_updated: "2026-01-16T02:19:32.990Z"
source: "https://vercel.com/docs/limits/fair-use-guidelines"
---

# Fair use Guidelines

All subscription plans include usage that is subject to these fair use guidelines. Below is a rule-of-thumb for determining which project

### Examples of fair use

### Never fair use

## Usage guidelines

As a guideline for our community, we expect most users to fall within the below ranges for each plan. We will notify you if your usage is

### Typical monthly usage guidelines

| | Hobby |
| --- | --- |
| Fast Data Transfer | Up to 100 GB |
| Fast Origin Transfer | Up to 10 GB |
| Function Execution | Up to 100 GB-Hrs |
| Build Execution | Up to 100 Hrs |
| [Image transformations](/docs/image-optimization/limits-and-pricing#image-transformations) | Up to 5K transformations/month |
| [Image cache reads](/docs/image-optimization/limits-and-pricing#image-cache-reads) | Up to 300K reads/month |
| [Image cache writes](/docs/image-optimization/limits-and-pricing#image-cache-writes) | Up to 100K writes/month |
| Storage | [Edge Config](/docs/edge-config/edge-confi |

For Teams on the Pro plan, you can pay for [additional usage](/docs/limits/fair-use-guidelines#additional-resources) as you go.

### Other guidelines

**Middleware with the `edge` runtime configured CPU Limits** - Middleware with the `edge` runtime configured can use no more than **50ms

For [on-demand concurrent builds](/docs/builds/managing-builds#on-demand-concurrent-builds), there is a fair usage limit of 500 concurren

### Additional resources

For members of our **Pro** plan, we offer a pay-as-you-go model for additional usage, giving you greater flexibility and control over you

| | Pro |
| --- | --- |
| Fast Data Transfer | [Regionally priced](/docs/pricing/regional- |
| Fast Origin Transfer | [Regionally priced](/docs/pricing/regional- |
| Function Execution | $0.60 per 1 GB-Hrs increment |
| [Image Optimization Source Images](/docs/image-optimization/legacy-pricing#source-images) | $5 per 1000 increment |

### Commercial usage

**Hobby teams** are restricted to non-commercial personal use only. All commercial usage of the platform requires either a Pro or Enterpr

Commercial usage is defined as any [Deployment](/docs/deployments) that is used for the purpose of financial gain of **anyone** involved

- Any method of requesting or processing payment from visitors of the site
- Advertising the sale of a product or service
- Receiving payment to create, update, or host the site
- Affiliate linking is the primary purpose of the site
- The inclusion of advertisements, including but not limited to online advertising platforms like Google AdSense

> **💡 Note:** Asking for Donations  fall under commercial usage.

If you are unsure whether or not your site would be defined as commercial usage, please [contact the Vercel Support team](/help#issues).

### General Limits

[**Take a look at our Limits documentation**](/docs/limits#general-limits) for the limits we apply to all accounts.

### Learn More

Circumventing or otherwise misusing Vercel's limits or usage guidelines is a violation of our fair use guidelines.

For further information regarding these guidelines and acceptable use of our services, refer to our [Terms of Service](/legal/terms#fair-

---
title: "Limits"
description: "This reference covers a list of all the limits and limitations that apply on Vercel."
last_updated: "2026-01-16T02:19:33.219Z"
source: "https://vercel.com/docs/limits"
---

# Limits

## General limits

To prevent abuse of our platform, we apply the following limits to all accounts.

| | Hobby |
| --- | --- |
| Projects | 200 |
| Deployments Created per Day | 100 |
| Serverless Functions Created per Deployment | [Framework-dependent*](/docs/functions/run |
| [Proxied Request Timeout](#proxied-request-timeout) (Seconds) | 120 |
| Deployments Created from CLI per Week | 2000 |
| [Vercel Projects Connected per Git Repository](#connecting-a-project-to-a-git-repository) | 10 |
| [Routes created per Deployment](#routes-created-per-deployment) | 2048 |
| [Build Time per Deployment](#build-time-per-deployment) (Minutes) | 45 |
| [Static File uploads](#static-file-uploads) | 100 MB |
| [Concurrent Builds](/docs/deployments/concurrent-builds) | 1 |
| Disk Size (GB) | 23 |
| Cron Jobs | [2*](/docs/cron-jobs/usage-and-pricing) |

## Included usage

| | Hobby | Pro |
| --- | --- | --- |
| Active CPU | 4 CPU-hrs | N/A |
| Provisioned Memory | 360 GB-hrs | N/A |
| Invocations | 1 million | N/A |
| Fast Data Transfer | 100 GB | 1 TB |
| Fast Origin Transfer | Up to 10 GB | N/A |
| Build Execution | 100 Hrs | N/A |

| [Image Optimization Source Images](/docs/image-optimization/legacy-pricing#source-images) | 1000 Images | N/A  |

For Teams on the Pro plan, you can pay for [usage](/docs/limits#additional-resources) on-demand.

## On-demand resources for Pro

For members of our Pro plan, we offer an included credit that can be used across all resources and a pay-as-you-go model for additional c

## Pro trial limits

See the [Pro trial limitations](/docs/plans/pro-plan/trials#trial-limitations) section for information on the limits that apply to Pro tr

## Routes created per deployment

The limit of "Routes created per Deployment" encapsulates several options that can be configured on Vercel:

- If you are using a `vercel.json` configuration file, each [rewrite](/docs/project-configuration#rewrites), [redirect](/docs/project-con
- If you are using the [Build Output API](/docs/build-output-api/v3), you might configure [routes](/docs/build-output-api/v3/configuratio

Note that most frameworks will create Routes automatically for you. For example, Next.js will create a set of Routes corresponding to you

## Build time per deployment

The maximum duration of the [Build Step](/docs/deployments/configure-a-build) is 45 minutes.
When the limit is reached, the Build Step will be interrupted and the Deployment will fail.

### Build container resources

Every Build is provided with the following resources:

|  | Hobby | Pro | Enterprise |
| --- | --- | --- | --- |
| Memory | 8192 MB | 8192 MB | Custom |
| Disk space | 23 GB | 23 GB | Custom |
| CPUs | 2 | 4 | Custom |

The limit for static file uploads in the build container is 1 GB.
Pro and Enterprise customers can purchase [Enhanced or Turbo build machines](/docs/builds/managing-builds#build-machine-types) with up to

For more information on troubleshooting these, see [Build container resources](/docs/deployments/troubleshoot-a-build#build-container-res

## Static file uploads

When using the CLI to deploy, the maximum size of the source files that can be uploaded is limited to 100 MB for Hobby and 1 GB for Pro.

### Build cache maximum size

The maximum size of the Build's cache is 1 GB. It is retained for one month and it applies at the level of each [Build cache key](/docs/d

## Monitoring

Check out [the limits and pricing section](/docs/observability/monitoring/limits-and-pricing) for more details about the limits of the [M

## Logs

There are two types of logs: **build logs** and **runtime logs**. Both have different behaviors when storing logs.

[Build logs](/docs/deployments/logs) are stored indefinitely for each deployment.

[Runtime logs](/docs/runtime-logs) are stored for **1 hour** on Hobby, **1 day** on Pro, and for **3 days** on Enterprise accounts. To le

## Environment variables

The maximum number of [Environment Variables](/docs/environment-variables) per environment per [Project](/docs/projects/overview)
is `1000`. For example, you cannot have more than `1000` Production Environment Variables.

The total size of your Environment Variables, names and values, is limited to **64KB** for projects using Node.js, Python, Ruby, Go, Java

If you are using [System Environment Variables](/docs/environment-variables/system-environment-variables), the framework-specific ones (i

## Domains

|  | Hobby | Pro | Enterprise |
| --- | --- | --- | --- |
| Domains per Project | 50 | Unlimited* | Unlimited* |

- To prevent abuse, Vercel implements soft limits of 100,000 domains per project for the Pro plan and 1,000,000 domains for the Enterpris

## Files

The maximum number of files that can be uploaded when creating a CLI [Deployment](/docs/deployments) is `15,000` for source files. Deploy

Although there is no upper limit for output files created during a build, you can expect longer build times as a result of having many th

We recommend using [Incremental Static Regeneration](/docs/incremental-static-regeneration) (ISR) to help reduce build time. Using ISR wi

## Proxied request timeout

The amount of time (in seconds) that a proxied request (`rewrites` or `routes` with an external destination) is allowed to process an HTT
If the external server does not reply until the maximum timeout is reached, an error with the message `ROUTER_EXTERNAL_TARGET_ERROR` will

## WebSockets

[Vercel Functions](/docs/functions) do not support acting as a WebSocket server.

We recommend third-party [solutions](/kb/guide/publish-and-subscribe-to-realtime-data-on-vercel) to enable realtime communication for [De

## Web Analytics

Check out the [Limits and Pricing section](/docs/analytics/limits-and-pricing) for more details about the limits of Vercel Web Analytics.

## Speed Insights

Check out the [Limits and Pricing](/docs/speed-insights/limits-and-pricing) doc for more details about the limits of the Speed Insights f

## Cron Jobs

Check out the Cron Jobs [limits](/docs/cron-jobs/usage-and-pricing) section for more information about the limits of Cron Jobs on Vercel.

## Vercel Functions

The limits of Vercel functions are based on the [runtime](/docs/functions/runtimes) that you use.

For example, different runtimes allow for different [bundle sizes](/docs/functions/runtimes#bundle-size-limits), [maximum duration](/docs

## Connecting a project to a Git repository

Vercel does not support connecting a project on your Hobby team to Git repositories owned by Git organizations. You can either switch to

The same limitation applies in the Project creation flow when importing an existing Git repository or when cloning a Vercel template to a

## Reserved variables

See the [Reserved Environment Variables](/docs/environment-variables/reserved-environment-variables) docs for more information.

## Rate limits

**Rate limits** are hard limits that apply to the platform when performing actions that require a response from our [API](/docs/rest-api#

The **rate limits** table consists of the following four columns:

- **Description** - A brief summary of the limit which, where relevant, will advise what type of plan it applies to.
- **Limit** - The amount of actions permitted within the amount of time (**Duration**) specified.
- **Duration** - The amount of time (seconds) in which you can perform the specified amount of actions. Once a rate limit is hit, it will
- **Scope** - How the rate limit is applied:
  - `owner` - Rate limit applies to the team or to an individual user, depending on the resource.
  - `user` - Rate limit applies to an individual user.
  - `team` - Rate limit applies to the team.

### Rate limit examples

Below are five examples that provide further information on how rate limits work.

#### Domain deletion

You are able to delete up to `60` domains every `60` seconds (1 minute). Should you hit the rate limit, you will need to wait another min

#### Team deletion

You are able to delete up to `20` teams every `3600` seconds (1 hour). Should you hit the rate limit, you will need to wait another hour

#### Username change

You are able to change your username up to `6` times every `604800` seconds (1 week). Should you hit the rate limit, you will need to wai

#### Builds per hour (Hobby)

You are able to build `32` [Deployments](/docs/deployments) every `3600` seconds (1 hour). Should you hit the rate limit, you will need t

> **💡 Note:** Using Next.js or any similar framework to build your deployment is classed as
> a build. Each Vercel Function is also classed as a build. Hosting static files
> such as an index.html file is not classed as a build.

#### Deployments per day (Hobby)

You are able to deploy `100` times every `86400` seconds (1 day). Should you hit the rate limit, you will need to wait another day before

***

| Description | Limit | Duration (Seconds) | Scope |
|---|---|---|---|
| Abuse report creation per minute. | 200 | 60 | `owner` |
| Artifacts requests per minute (Free). | 100 | 60 | `owner` |
| Requests per minute to fetch the microfrontends groups for a team. | 30 | 60 | `owner` |
| Requests per minute to fetch the microfrontends config for a team. | 30 | 60 | `owner` |
| Requests per minute to fetch the deployment of the best default app. | 30 | 60 | `owner` |
| Artifacts requests per minute (Paid). | 10000 | 60 | `owner` |
| Project production deployment per minute. | 500 | 60 | `user` |
| Project expiration updates per minute. | 100 | 60 | `owner` |
| Project release configuration updates per minute. | 100 | 60 | `owner` |
| Project domains get per minute. | 500 | 60 | `user` |
| Get project domains count per minute. | 100 | 60 | `user` |
| Project domains verification per minute. | 100 | 60 | `user` |
| Project branches get per minute. | 100 | 60 | `user` |
| Project branches get search per minute. | 500 | 60 | `user` |
| Project domain creation, update, or remove per minute. | 100 | 60 | `owner` |
| Project protection bypass creation, update, or remove per minute. | 100 | 60 | `owner` |
| Listing Deployment Protection Exceptions per minute | 250 | 60 | `owner` |
| Project environment variable retrieval per minute. | 500 | 60 | `owner` |
| Project environment variable updates per minute. | 120 | 60 | `owner` |
| Team enable new standard protection for all projects updates per minute. | 10 | 60 | `owner` |
| Project environment variable creation per minute. | 120 | 60 | `owner` |
| Project environment variable deletions per minute. | 60 | 60 | `owner` |
| Project client certificate uploads per minute. | 5 | 60 | `owner` |
| Project client certificate deletions per minute. | 5 | 60 | `owner` |
| Project client certificate retrievals per minute. | 300 | 60 | `owner` |
| Project environment variable batch deletions per minute. | 60 | 60 | `owner` |
| Project environment variable pulls per minute. | 500 | 60 | `owner` |
| Custom deployment suffix changes per hour. | 5 | 3600 | `owner` |

| Deploy hook triggers per hour. | 60 | 3600 | `owner` |
| Deployments retrieval per minute. | 500 | 60 | `user` |
| Deployments retrieval per minute (Enterprise). | 2000 | 60 | `user` |
| Deployments per day (Free). | 100 | 86400 | `owner` |
| Deployments per day (Pro). | 6000 | 86400 | `owner` |
| Deployments per day (Enterprise). | 24000 | 86400 | `owner` |
| Deployments per hour (Free). | 100 | 3600 | `owner` |
| Deployments per hour (Pro). | 450 | 3600 | `owner` |
| Deployments per hour (Enterprise). | 1800 | 3600 | `owner` |
| Deployment user access check per minute. | 100 | 60 | `user` |
| Deployment undeletes per minute. | 100 | 60 | `owner` |
| Skipped deployments per minute. | 100 | 60 | `user` |
| AI domain search per minute. | 20 | 60 | `user` |
| Domains deletion per minute. | 100 | 60 | `owner` |
| Domain price per minute. | 100 | 60 | `user` |
| Domains retrieval per minute. | 200 | 60 | `user` |
| Domains retrieval per minute. | 500 | 60 | `user` |
| Domain's transfer auth code. | 50 | 60 | `user` |
| Domain's transfer auth code. | 10 | 60 | `user` |
| Domain contact verification status retrieval per minute. | 20 | 60 | `user` |
| Domains dns config retrieval per minute. | 500 | 60 | `user` |
| Domains update per minute. | 60 | 60 | `owner` |
| Domains creation per hour. | 120 | 3600 | `owner` |
| Domain delegation requests per day. | 20 | 86400 | `owner` |
| Automatic domain delegation requests per minute. | 10 | 60 | `owner` |
| Enterprise domain delegation requests per minute. | 10 | 60 | `owner` |
| Domains record update per minute. | 50 | 60 | `owner` |
| Domains record creation per hour. | 100 | 3600 | `owner` |
| Domains status retrieval per minute. | 120 | 60 | `owner` |
| Domains availability retrieval per minute. | 20 | 60 | `user` |
| Domain verification record retrieval per minute. | 60 | 60 | `owner` |
| Domain ownership claim attempts per minute. | 10 | 60 | `owner` |
| Domain save attempts per minute. | 20 | 60 | `user` |
| Domain unsave attempts per minute. | 20 | 60 | `user` |
| Events retrieval per minute. | 60 | 60 | `user` |
| Events retrieval per minute. | 10 | 60 | `user` |
| Download Audit Log exports per minute. | 5 | 60 | `user` |
| Setup up Audit Log Stream per minute | 10 | 60 | `user` |
| Plan retrieval per minute. | 120 | 60 | `owner` |
| Plan update per hour. | 60 | 3600 | `owner` |
| Requests to self-unblock per hour. | 5 | 3600 | `owner` |
| Team deletion per hour. | 20 | 3600 | `user` |
| Team retrieval per minute. | 600 | 60 | `user` |
| Team retrieval per minute. | 600 | 60 | `user` |
| Team update per hour. | 100 | 3600 | `user` |
| Requests per minute to patch the microfrontends groups for a team. | 10 | 60 | `user` |
| Team SSO configuration per hour. | 100 | 3600 | `user` |
| Team creation per day (Free). | 5 | 86400 | `user` |
| Team creation per day (Paid). | 25 | 86400 | `user` |
| Team slug creation per hour. | 200 | 3600 | `user` |
| Team slug update per week. | 6 | 604800 | `owner` |
| Team exclusivity creation per team per hour. | 10 | 3600 | `owner` |
| Team exclusivity update per team per hour. | 10 | 3600 | `owner` |
| Team exclusivity delete per team per hour. | 10 | 3600 | `owner` |
| Team exclusivity list per user per minute. | 120 | 60 | `user` |
| Git exclusivity get per user per minute. | 120 | 60 | `user` |
| Preview Deployment Suffix updates per day. | 10 | 86400 | `owner` |
| Team member deletion per ten minutes. | 500 | 600 | `owner` |
| Team member retrieval per minute. | 120 | 60 | `owner` |
| Team member update per ten minutes. | 40 | 600 | `owner` |
| Team member creation per hour (Free). | 50 | 3600 | `owner` |
| Team member creation per hour (Paid). | 150 | 3600 | `owner` |
| Team member creation per hour (Enterprise). | 300 | 3600 | `owner` |
| Team member creation (batch) | 1 | 1 | `owner` |
| Team invite requests per hour. | 10 | 3600 | `user` |
| Team invite retrieval per minute. | 120 | 60 | `owner` |
| Requests to bulk update project retention per minute. | 1 | 60 | `owner` |
| Requests to list teams eligible for merge per minute. | 60 | 60 | `user` |
| Requests to get the status of a merge per minute. | 120 | 60 | `user` |
| Requests to create merge plans per minute. | 20 | 60 | `user` |
| Requests to create merge plans per minute. | 20 | 60 | `user` |
| Organizations retrieval per minute. | 120 | 60 | `user` |
| User retrieval per minute. | 500 | 60 | `owner` |
| User update per minute. | 60 | 60 | `owner` |
| Username update per week. | 6 | 604800 | `owner` |
| Uploads per day (Free). | 5000 | 86400 | `owner` |
| Uploads per day (Pro). | 40000 | 86400 | `owner` |
| Uploads per day (Enterprise). | 80000 | 86400 | `owner` |
| Token retrieval per minute. | 120 | 60 | `owner` |
| Token creation per hour. | 32 | 3600 | `owner` |
| Token deletion per five minutes. | 50 | 300 | `owner` |
| Payment method update per day. | 10 | 86400 | `owner` |
| Payment method setup per hour | 10 | 3600 | `owner` |
| Balance due retrieval per minute. | 70 | 60 | `owner` |
| Upcoming invoice retrieval per minute. | 70 | 60 | `owner` |
| Invoice Settings updates per ten minutes. | 10 | 600 | `owner` |
| Concurrent Builds updates per ten minutes. | 10 | 600 | `owner` |
| Monitoring updates per ten minutes. | 10 | 600 | `owner` |
| Web Analytics updates per ten minutes. | 10 | 600 | `owner` |
| Preview Deployment Suffix updates per ten minutes. | 10 | 600 | `owner` |
| Advanced Deployment Protection updates per ten minutes. | 10 | 600 | `owner` |
| Retry payment per ten minutes. | 25 | 600 | `owner` |
| Alias retrieval per ten minutes. | 300 | 600 | `user` |
| Alias creation per ten minutes. | 120 | 600 | `owner` |
| Aliases list per minute. | 500 | 60 | `user` |
| Aliases deletion per minute. | 100 | 60 | `owner` |
| Certificate deletion per ten minutes. | 60 | 600 | `owner` |
| Certificate retrieval per minute. | 500 | 60 | `user` |
| Certificate update per hour. | 30 | 3600 | `owner` |
| Certificate creation per hour. | 30 | 3600 | `owner` |

| User supplied certificate update per hour. | 30 | 60 | `owner` |
| Deployments list per minute. | 1000 | 60 | `user` |
| Deployments configuration list per minute. | 100 | 60 | `owner` |
| Deployments deletion per ten minutes. | 200 | 600 | `owner` |
| Integration job creation per five minutes. | 100 | 300 | `owner` |
| Integration retrieval per minute (All). | 100 | 60 | `user` |
| Integration retrieval per minute (Single). | 100 | 60 | `user` |
| Integration creation per minute. | 120 | 3600 | `user` |
| Integration update per minute. | 120 | 3600 | `user` |
| Integration deletion per minute. | 120 | 3600 | `user` |
| Integration deployment action updates per minute. | 100 | 60 | `user` |
| Marketplace integration installations per minute. | 120 | 3600 | `user` |
| Marketplace integration uninstallations per minute. | 120 | 3600 | `user` |
| Marketplace integration secrets rotation requests per minute. | 120 | 60 | `user` |
| Marketplace integration transfers per minute. | 120 | 3600 | `user` |
| Marketplace purchase provisions per minute. | 120 | 3600 | `user` |
| Resource drains retrieval per minute. | 100 | 60 | `user` |
| Marketplace config retrieval per minute. | 100 | 60 | `ip` |
| Marketplace config updates per minute. | 20 | 60 | `owner` |
| Marketplace featured image uploads per minute. | 10 | 60 | `user` |
| Integration product get per minute. | 120 | 60 | `user` |
| Integration products get per minute. | 120 | 60 | `user` |
| Integration product delete per minute. | 120 | 3600 | `user` |
| Integration product create per minute. | 120 | 3600 | `user` |
| Integration product create per minute. | 120 | 3600 | `user` |
| Integration product billing plans retrieval per minute. | 120 | 3600 | `user` |
| Integration installation billing plans retrieval per minute. | 120 | 3600 | `user` |
| Integration resource billing plans retrieval per minute. | 120 | 3600 | `user` |
| Integration resource usage retrieval per minute. | 120 | 3600 | `user` |
| Store-to-project connection per minute. | 120 | 3600 | `user` |
| Integration SSO redirect URI create per minute. | 20 | 60 | `user` |
| Integration MCP access token requests. | 2 | 60 | `user` |
| Integration MCP access token requests when cached. | 200 | 60 | `user` |
| MCP domain search requests per minute per IP. | 100 | 60 | `user` |
| Installation Resource secrets update per minute. | 240 | 60 | `user` |
| Installation Resource import per minute. | 100 | 60 | `user` |
| Installation account info retrieval per minute. | 60 | 60 | `user` |
| Installation event create per minute. | 60 | 60 | `user` |
| Integration favorite retrieval per minute. | 100 | 60 | `user` |
| Integration favorite update per minute. | 120 | 3600 | `user` |
| Integration configuration creation per minute. | 120 | 3600 | `owner` |
| Integration authorization creation per minute. | 120 | 3600 | `user` |
| Integration configuration retrieval per minute (All). | 200 | 60 | `user` |
| Integration configuration retrieval per minute (Single). | 120 | 60 | `user` |
| Most recent integration configuration retrieval per minute (Single). | 60 | 60 | `user` |
| Integration configuration permissions retrieval per minute (All). | 60 | 60 | `user` |
| Integration configuration update per minute. | 120 | 3600 | `owner` |
| Integration associated user transfers per minute. | 120 | 3600 | `user` |
| Integration configuration deletion per minute. | 120 | 3600 | `owner` |
| Integration metadata retrieval per minute. | 300 | 60 | `user` |
| Integration metadata creation per minute. | 300 | 60 | `user` |
| Integration metadata deletion per minute. | 60 | 60 | `user` |
| Integration logs retrieval per minute. | 100 | 60 | `user` |
| Integration logs creation per minute. | 20 | 60 | `user` |
| Integration logs deletion per minute. | 60 | 60 | `user` |
| Integration webhooks retrieval per minute. | 100 | 60 | `user` |
| Integration webhooks retrieval per minute. | 100 | 60 | `user` |
| Integration webhooks retrieval per minute. | 100 | 60 | `user` |
| Integration webhooks creation per minute. | 20 | 60 | `user` |
| Integration webhooks deletion per minute. | 60 | 60 | `user` |
| Integration app install status retrieval per minute. | 60 | 60 | `user` |
| Membership info retrievals per minute for an installation. | 1000 | 60 | `owner` |
| Membership info retrievals per minute for a user. | 60 | 60 | `user` |
| List of memberships retrieval per minute for a user. | 60 | 60 | `user` |
| Integration resource usage retrieval per minute. | 120 | 60 | `user` |
| Installation prepayment balance submissions per minute. | 10 | 60 | `user` |
| Installation billing data submissions per minute. | 10 | 60 | `user` |
| Installation invoice submissions per minute. | 10 | 60 | `user` |
| Installation resources retrieval per minute. | 1000 | 60 | `user` |
| Installation resource deletion per minute. | 100 | 60 | `user` |
| Installation invoice retrieval per minute. | 60 | 60 | `user` |
| Integration resource retrieval per minute. | 1000 | 60 | `user` |
| Integration payment method retrieval per minute. | 60 | 60 | `user` |
| Integration payment method update per minute. | 60 | 60 | `user` |
| Admin users for the installation. | 60 | 60 | `user` |
| Update admin users for the installation. | 60 | 60 | `user` |
| Create authorization for a marketplace purchase. | 30 | 60 | `user` |
| Check marketplace authorization state. | 500 | 60 | `user` |
| Get installation statistics for a marketplace integration. | 500 | 60 | `user` |
| Get installation statistics for a marketplace integration. | 500 | 60 | `user` |
| Get billing summary for a marketplace integration. | 500 | 60 | `user` |
| Get invoices by month for a marketplace integration. | 500 | 60 | `user` |
| Webhooks updates per minute. | 60 | 60 | `user` |
| Webhooks tests per minute. | 60 | 60 | `user` |
| Log Drain retrieval per minute. | 100 | 60 | `user` |
| Log Drain creation per minute. | 20 | 60 | `user` |
| Log Drain deletion per minute. | 60 | 60 | `user` |
| Log Drain test per minute. | 30 | 60 | `user` |
| Log Drain update per minute. | 30 | 60 | `user` |
| Drain create per minute. | 30 | 60 | `user` |
| Drain delete per minute. | 30 | 60 | `user` |
| Drain retrieval per minute. | 100 | 60 | `user` |
| Drain update per minute. | 30 | 60 | `user` |
| Drain test per minute. | 30 | 60 | `user` |
| Runtime Logs retrieval per minute. | 100 | 60 | `user` |
| Logs UI preset creation per minute. | 100 | 60 | `user` |
| Logs UI preset reads per minute. | 100 | 60 | `user` |
| Logs UI preset edits per minute. | 100 | 60 | `user` |
| Log Drain retrieval per minute. | 100 | 60 | `user` |
| Suggested teams retrieval per minute. | 30 | 60 | `user` |

| Integration installed retrieval per minute (All). | 20 | 60 | `user` |
| Integration otel endpoint creation/updates per minute. | 20 | 60 | `user` |
| Integration otel endpoint retrieval per minute. | 100 | 60 | `user` |
| Integration otel endpoint deletion per minute. | 60 | 60 | `user` |
| Check retrieval per minute. | 500 | 60 | `user` |
| Check retrieval per minute. | 500 | 60 | `user` |
| Checks retrieval per minute. | 300 | 60 | `owner` |
| Check retrieval per minute. | 300 | 60 | `owner` |
| Check runs retrieval per minute. | 500 | 60 | `owner` |
| Check runs for check retrieval per minute. | 500 | 60 | `owner` |
| Check runs retrieval per minute. | 500 | 60 | `owner` |
| State retrieval per minute. | 500 | 60 | `user` |
| Deployment integrations skip action. | 200 | 60 | `user` |
| Edge Config writes per day (Paid). | 480 | 86400 | `owner` |
| Edge Config writes per month (Free). | 250 | 2592000 | `owner` |
| Edge Config token changes per day. | 500 | 86400 | `owner` |
| Edge Config deletions per 5 minutes. | 60 | 300 | `owner` |
| Edge Configs reads per minute. | 500 | 60 | `owner` |
| Edge Config reads per minute. | 500 | 60 | `owner` |
| Edge Config Items reads per minute. | 20 | 60 | `owner` |
| Edge Config schema reads per minute. | 500 | 60 | `owner` |
| Edge Config schema updates per minute. | 60 | 60 | `owner` |
| Edge Config backup queries per minute. | 100 | 60 | `owner` |
| Edge Config backup retrievals per minute. | 60 | 60 | `owner` |
| Endpoint Verification retrieval per minute. | 100 | 60 | `user` |
| Secure Compute networks created per hour. | 5 | 3600 | `owner` |
| Secure Compute networks deleted per hour. | 25 | 3600 | `owner` |
| Secure Compute network lists per minute. | 250 | 60 | `owner` |
| Secure Compute network reads per minute. | 250 | 60 | `owner` |
| Secure Compute network updates per hour. | 25 | 3600 | `owner` |
| Recents create per minute. | 100 | 60 | `user` |
| Recents delete per minute. | 100 | 60 | `user` |
| Recents get retrieval per minute. | 100 | 60 | `user` |
| Update notification settings preferences. | 20 | 60 | `user` |
| Stores get retrieval per minute. | 200 | 60 | `user` |
| Accept storage terms of service. | 100 | 60 | `user` |
| Store get retrieval per minute. | 400 | 60 | `user` |
| Access credentials per minute. | 1000 | 60 | `user` |
| Blob stores create per minute. | 100 | 60 | `user` |
| Blob stores update per minute. | 100 | 60 | `user` |
| Blob stores delete per minute. | 100 | 60 | `user` |
| Postgres stores create per minute. | 100 | 60 | `user` |
| Postgres stores update per minute. | 100 | 60 | `user` |
| Postgres stores delete per minute. | 100 | 60 | `user` |
| Postgres stores warm-up per minute. | 100 | 60 | `user` |
| Stores connect per minute. | 100 | 60 | `user` |
| Stores disconnect per minute. | 100 | 60 | `user` |
| Integration stores create per minute. | 100 | 60 | `user` |
| Integration stores update per minute. | 100 | 60 | `user` |
| Integration stores delete per minute. | 100 | 60 | `user` |
| Integration stores repl commandse per minute. | 100 | 60 | `user` |
| Stores rotate default store token set per minute. | 100 | 60 | `user` |
| Transfer Stores per minute. | 100 | 60 | `user` |
| Vercel Blob Simple Operations per minute for Hobby plan. | 1200 | 60 | `team` |
| Vercel Blob Simple Operations per minute for Pro plan. | 7200 | 60 | `team` |
| Vercel Blob Simple Operations per minute for Enterprise plan. | 9000 | 60 | `team` |
| Vercel Blob Advanced Operations per minute for Hobby plan. | 900 | 60 | `team` |
| Vercel Blob Advanced Operations per minute for Pro plan. | 4500 | 60 | `team` |
| Vercel Blob Advanced Operations per minute for Enterprise plan. | 7500 | 60 | `team` |
| Ip Blocking create per minute. | 60 | 60 | `user` |
| Ip Blocking list executed per minute. | 100 | 60 | `user` |
| Ip Blocking reads executed per minute. | 100 | 60 | `user` |
| Ip Blocking delete per minute. | 100 | 60 | `user` |
| IP Bypass reads per minute. | 100 | 60 | `user` |
| IP Bypass updates per minute. | 30 | 60 | `user` |
| Attack Status | 20 | 60 | `user` |
| Project Bulk Redirect reads per minute | 200 | 60 | `owner` |
| Project Bulk Redirect mutations per minute | 30 | 60 | `owner` |
| Project Bulk Redirect version reads per minute | 500 | 60 | `owner` |
| Project Bulk Redirect version updates per minute | 20 | 60 | `owner` |
| Project Bulk Redirect settings reads per minute | 300 | 60 | `owner` |
| Project Bulk Redirect settings updates per minute | 10 | 60 | `owner` |
| Vade review configuration requests per minute. | 30 | 60 | `owner` |
| Vade tasks retrieval requests per minute. | 100 | 60 | `owner` |
| Vade runtime fix trigger requests per minute. | 100 | 60 | `owner` |
| Vade apply patch requests per minute. | 30 | 60 | `owner` |
| Vade ignore patch requests per minute. | 30 | 60 | `owner` |
| Vade code generation and follow-up requests per minute. | 20 | 60 | `owner` |
| Vade code threads retrieval requests per minute. | 100 | 60 | `owner` |
| Vade code messages retrieval requests per minute. | 100 | 60 | `owner` |
| Vade audit retrieval requests per minute. | 250 | 60 | `owner` |
| Vade audit creation requests per minute. | 30 | 60 | `owner` |
| Vade apply trial credits requests per minute. | 10 | 60 | `owner` |
| Manual AI code review requests per minute. | 30 | 60 | `owner` |

---
title: "Logs"
description: "Use logs to find information on deployment builds, function executions, and more."
last_updated: "2026-01-16T02:19:32.937Z"
source: "https://vercel.com/docs/logs"
---

# Logs

## Build Logs

When you deploy your website to Vercel, the platform generates build logs that show the deployment progress. The build logs contain infor

- The version of the build tools

- Warnings or errors encountered during the build process
- Details about the files and dependencies that were installed, compiled, or built during the deployment

Learn more about [Build Logs](/docs/deployments/logs).

## Runtime Logs

Runtime logs allow you to search, inspect, and share your team's runtime logs at a project level. You can search runtime logs from the de

Learn more about [Runtime Logs](/docs/logs/runtime).

## Activity Logs

Activity Logs provide chronologically organized events on your personal or team account. You get an overview of changes to your environme

Learn more about [Activity Logs](/docs/observability/activity-log).

## Audit Logs

Audit Logs allow owners to track events performed by other team members. The feature helps you verify who accessed what, for what reason,

Learn more about [Audit Logs](/docs/observability/audit-log).

## Log Drains

Log Drains allow you to export your log data, making it easier to debug and analyze. You can configure Log Drains through the Vercel dash

Learn more about [Log Drains](/docs/drains).

---
title: "Runtime Logs"
description: "Learn how to search, inspect, and share your runtime logs with the Logs tab."
last_updated: "2026-01-16T02:19:32.970Z"
source: "https://vercel.com/docs/logs/runtime"
---

# Runtime Logs

The **Logs** tab allows you to view, search, inspect, and [share](#log-sharing) your runtime logs without any third-party integration. Yo

> **💡 Note:** You can only view runtime logs from the Logs tab. [Build
> logs](/docs/deployments/logs) can be accessed from the production deployment
> tile.

## What are runtime logs?

**Runtime logs** include all logs generated by [Vercel Functions](/docs/functions) invocations in both [preview](/docs/deployments/enviro

With runtime logs:

- Logs are shown in realtime and grouped as per request.
- Each action of writing to standard output, such as using `console.log`, results in a separate log entry.
- The maximum number of logs is 256 lines *per request*
- Each of those logs can be up to 256 KB *per line*
- The sum of all log lines can be up to 1 MB *per request*

## Available Log Types

You can view the following log types in the [Logs tab](#view-runtime-logs):

| **Log Type** | **Available in Runtime Logs** |
| --- | --- |
| Vercel Function Invocation | Yes |
| Routing Middleware Invocation | Yes |
| Static Request | Only static request that serves cache; to get all static logs check [Log Drains](/docs/drains) |

## View runtime logs

To view runtime logs:

1. From the dashboard, select the project that you wish to see the logs for
2. Select the **Logs** tab from your project overview
3. From here you can view, filter, and search through the runtime logs. Each log row shares [basic info](#log-details) about the request,

## Log filters

You can use the following filters from the left sidebar to get a refined search experience.

### Timeline

You can filter runtime logs based on a specific timeline. It can vary from the past hour, last 3 days, or a custom timespan [depending on

> **💡 Note:** All displayed dates and times are in UTC.

### Level

You can filter requests that contain **Warning**, and **Error** logs. A request can contain both types of logs at the same time. [Streami

| Source | [Streaming functions](/docs/functions/streaming-functions) | Non-streaming Functions |
| --- | --- | --- |
| `stdout` (e.g. `console.log`) | `info` | `info` |
| `stderr` (e.g. `console.error`) | `error` | `error` |
| `console.warn` | `warning` | `error` |

Additionally:

- Requests with a status code of `4xx` are marked with **Warning** amber
- Requests with a status code of `5xx` are marked with **Error** red
- All other individual log lines are considered **Info**

### Function

You can filter and analyze logs for one or more functions defined in your project. The log output is generated for the [Vercel Functions]

### Host

You can view logs for one or more domains and subdomains attached to your team’s project. Alternatively, you can use the **Search hosts..

### Deployment

Like host and functions, you can filter your logs based on deployments URLs.

### Resource

Using the resource filter, you can search for requests containing logs generated as a result of:

| **Resource** | **Description** |
| --- | --- |
| **[Vercel Functions](/docs/functions)** | Logs generated from your Vercel Functions invocations. Log details include additio |
| **[Routing Middleware](/docs/routing-middleware)** | Logs generated as a result of your Routing Middleware invocations |
| **Vercel CDN Cache** | Logs generated from proxy serving cache |

### Request Type

You can filter your logs based on framework-defined mechanism or rendering strategy used such as API routes, Incremental Static Regenerat

### Request Method

You can filter your logs based on the request method used by a function such as `GET` or `POST`.

### Request Path

You can filter your logs based on the request path used by a function such as `/api/my-function`.

### Cache

You can filter your logs based on the cache behavior such as `HIT` or `MISS`. See [`x-vercel-cache`](/docs/headers/response-headers#x-ver

### Logs from your browser

You can filter logs to only show requests made from your current browser by clicking the user button. This is helpful for debugging your

> **💡 Note:** The matching is based on your IP address and User Agent. In some cases, this
> data may not be accurate, especially if you're using a VPN or proxy, or if
> other people in your network are using the same IP address and browser.

## Search log fields

You can use the main search field to filter logs by their messages. In the current search state, filtered log results are sorted chronolo

| **Value** | **Description** |
| --- | --- |
| **[Function](#function)** | The function name |
| **[RequestPath](#request-path)** | The request path name |
| **[RequestType](#request-type)** | The request rendering type. For example API endpoints or Incremental Static Regeneration (ISR) |
| **[Level](#level)** | The level type. Can be Info, Warning, or Error |
| **[Resource](#resource)** | Can be Vercel CDN Cache, [Vercel Function](/docs/functions), [Routing Middleware](/docs/routing-midd |
| **[Host](#host)** | Name of the [domain](/docs/domains) or subdomain for which the log was generated |
| **[Deployment](#deployment)** | The name of your deployment |
| **[Method](#request-method)** | The request method used. For example `GET`, `POST` etc. |
| **[Cache](#cache)** | The Vercel CDN Cache status, see [`x-vercel-cache`](/docs/headers/response-headers#x-vercel-cache) f |
| **Status** | HTTP status code for the log message |
| **RequestID** | Unique identifier of request. This is visible on a 404 page, for example. |

> **💡 Note:** This  feature is limited to the
>  and 
> field. Other fields can be filtered using the left sidebar or the filters in
> the search bar.

## Log details

You can view details for each request to analyze and improve your debugging experience. When you click a log from the list, the following

| **Info** | **Description** |
| --- | --- |
| **Request Path** | Request path of the log |
| **Time** | Timestamp at which the log was recorded in UTC |
| **Status Code** | HTTP status code for the log message |
| **Host** | Name of the [domain](/docs/domains) or subdomain for which the log was generated |
| **Request Id** | Unique identifier of request created only for runtime logs |
| **Request User Agent** | Name of the browser from which the request originated |
| **Search Params** | Search parameters of the request path |
| **Firewall** | If request was allowed through firewall |
| **Vercel Cache** | The Vercel CDN Cache status, see [`x-vercel-cache`](/docs/headers/response-headers#x-vercel-cache) for the pos |
| **Middleware** | Metadata about middleware execution such as location and external api |
| **Function** | Metadata about function execution including function name, location, runtime, and duration |
| **Deployment** | Metadata about the deployment that produced the logs including id, environment and branch |
| **Log Message** | The bottom panel shows a list of log messages produced in chronological order |

### Show additional logs

Towards the end of the log results window is a button called **Show New Logs**. By default, it is set to display log results for the past

Click this button, and it loads new log rows. The latest entries are added based on the selected filters.

## Log sharing

You can share a log entry with other [team members](/docs/rbac/managing-team-members) to view the particular log and context you are look

## Limits

Logs are streamed. Each `log` output can be up to 256KB, and each request can log up to 1MB of data in total, with a limit of 256 individ

Runtime logs are stored with the following observability limits:

| Plan | Retention time |
| --- | --- |
| **Hobby** | 1 hour of logs |
| **Pro** | 1 day of logs |
| **Pro** with Observability Plus | 30 days of logs |
| **Enterprise** | 3 days of logs |
| **Enterprise** with Observability Plus | 30 days of logs |

Users who have purchased the [Observability Plus](/docs/observability/observability-plus) add-on can view up to 14 consecutive days of ru

> **💡 Note:** The above limits are applied immediately when [upgrading
> plans](/docs/plans/hobby#upgrading-to-pro). For example, if you upgrade from
> [Hobby](/docs/plans/hobby) to [Pro](/docs/plans/pro-plan), you will have access to
> the Pro plan limits, and access historical logs for up to 1 day.

---
title: "Manage and optimize usage for Observability"
description: "Learn how to understand the different charts in the Vercel dashboard, how usage relates to billing, and how to optimize you
last_updated: "2026-01-16T02:19:32.981Z"
source: "https://vercel.com/docs/manage-and-optimize-observability"
---

# Manage and optimize usage for Observability

The Observability section covers usage for Observability, Monitoring, Web Analytics, and Speed insights.

## Plan usage

## Managing Web Analytics events

The **Events** chart shows the number of page views and custom events that were tracked across all of your projects. You can filter the d

Every plan has an included limit of events per month. On Pro, Pro with Web Analytics Plus, and Enterprise plans, you're billed based on t

> **💡 Note:** Speed Insights and Web Analytics require scripts to do collection of [data
> points](/docs/speed-insights/metrics#understanding-data-points). These scripts
> are loaded on the client-side and therefore may incur additional usage and
> costs for [Data Transfer](/docs/manage-cdn-usage#fast-data-transfer) and [Edge
> Requests](/docs/manage-cdn-usage#edge-requests).

### Optimizing Web Analytics events

- Your usage is based on the total number of events used across all projects within your team. You can see this number by selecting **Pro
- Reduce the amount of custom events they send. Users can find the most sent events in the [events panel](/docs/analytics#panels) in Web A
- Use [beforeSend](/docs/analytics/package#beforesend) to exclude page views and events that might not be relevant

## Managing Speed Insights data points

You are initially billed a set amount for each project on which you enable Speed Insights. Each plan includes a set number of data points

Data points are a single unit of information that represent a measurement of a specific Web Vital metric during a user's visit to your we

> **💡 Note:** Speed Insights and Web Analytics require scripts to do collection of [data
> points](/docs/speed-insights/metrics#understanding-data-points). These scripts
> are loaded on the client-side and therefore may incur additional usage and
> costs for [Data Transfer](/docs/manage-cdn-usage#fast-data-transfer) and [Edge
> Requests](/docs/manage-cdn-usage#edge-requests).

### Optimizing Speed Insights data points

- To reduce cost, you can change the sample rate at a project level by using the `@vercel/speed-insights` package as explained in [Sample
- Use [beforeSend](/docs/speed-insights/package#beforesend) to exclude page views and events that might not be relevant
- You may want to [disable speed insights](/docs/speed-insights/disable) for projects that no longer need it. This will stop data points

## Managing Monitoring events

> **💡 Note:** Monitoring has become part of Observability, and is therefore included with
> Observability Plus at no additional cost. If you are currently paying for
> Monitoring, you should
> [migrate](/docs/observability#enabling-observability-plus) to Observability
> Plus to get access to additional product features with a longer retention
> period for the same [base
> fee](/docs/observability/limits-and-pricing#pricing).

Vercel creates an event each time a request is made to your website. These events include unique parameters such as execution time and ba

You pay for monitoring based on the **total** number of events used above the included limit included in your plan. You can see this numb

You can also view the number of events used by each project in your team by selecting **Projects** in the chart. This will show you the n

### Optimizing Monitoring events

Because events are based on the amount of requests to your site, there is no way to optimize the number of events used.

## Optimizing drains usage

You can optimize your log drains usage by:

- [**Filtering by environment**](/docs/drains/reference/logs#log-environments): You can filter logs by environment to reduce the number o
- [**Sampling rate**](/docs/drains/reference/logs#sampling-rate): You can reduce the number of logs sent to your log drain by using a sam

## Managing Observability events

Vercel creates one or many events each time a request is made to your website. To learn more, see [Events](/docs/observability#tracked-ev

You pay for Observability Plus based on the **total** number of events used above the included limit included in your plan.

The Observability chart allows you to view by the total **Count**, **Event Type**, or **Projects** over the selected time period.

### Optimizing Observability events

Because events are based on the amount of requests to your site, there is no way to optimize the number of events used.

---
title: "Manage and optimize CDN usage"
description: "Learn how to understand the different charts in the Vercel dashboard. Learn how usage relates to billing, and how to optimi
last_updated: "2026-01-16T02:19:33.282Z"
source: "https://vercel.com/docs/manage-cdn-usage"
---

# Manage and optimize CDN usage

The **Networking** section shows the following metrics:

## Top Paths

**Top Paths** displays the paths that consume the most resources on your team. These are resources such as bandwidth, execution, invocati

This section helps you find ways to optimize your project.

### Managing Top Paths

In the compact view, you can view the top ten resource-consuming paths in your projects.

You can filter these by:

- **Bandwidth**
- **Execution**
- **Invocations**
- or **Requests**

Select the **View** button to view a full page, allowing you to apply filters such as billing cycle, date, or project.

### Using Top Paths and Monitoring

Using **Top Paths** you can identify and optimize the most resource-intensive paths within your project. This is particularly useful for

When analyzing your bandwidth consumption you may see a path that ends with `_next/image`. The path will also detail a consumption value,

To investigate further, you can:

1. Navigate to the **Monitoring** tab and select the **Bandwidth by Optimized Image** example query from the left navigation
2. Select the **Edit Query** button and edit the **Where** clause to filter by `host = 'my-site.com'`. The full query should look like `r

This will show you the bandwidth consumption of images served through Vercel's Image Optimization for your project hosting the domain `my

Remove filters to get a better view of image optimization usage across all your projects. You can remove the `host = 'my-site.com'` filte

For a breakdown of the available clauses, fields, and variables that you can use to construct a query, see the [Monitoring Reference](/do

For more guidance on optimizing your image usage, see [managing image optimization and usage costs](/docs/image-optimization/managing-ima

## Fast Data Transfer

When a user visits your site, the data transfer between Vercel's CDN and the user's device gets measured as Fast Data Transfer. The data

Fast Data transfer usage incurs alongside [Edge Requests](#edge-requests) every time a user visits your site, and is [priced regionally](

### Optimizing Fast Data Transfer

The **Fast Data Transfer** chart on the **Usage** tab of your dashboard shows the incoming and outgoing data transfer of your projects.

- The **Direction** filter allows you to see the data transfer direction (incoming or outgoing)
- The **Projects** filter allows you to see the data transfer of a specific project
- The **Regions** filter allows you to see the data transfer of a specific region. This is can be helpful due to the nature of [regional

As with all charts on the **Usage** tab, you can select the caret icon to view the chart as a full page.

To optimize Fast Data Transfer, you must optimize the assets that are being transferred. You can do this by:

- **Using Vercel's Image Optimization**: [Image Optimization](/docs/image-optimization) on Vercel uses advanced compression and modern fi
- **Analyzing your bundles**: See your preferred frameworks documentation for guidance on how to analyze and reduce the size of your bund

Similar to **Top Paths**, you can use the **Monitoring** tab to further analyze the data transfer of your projects. See the [**Using Top

### Calculating Fast Data Transfer

Fast Data Transfer is calculated based on the full size of each HTTP request and response transmitted to or from Vercel's [CDN](/docs/cdn

## Fast Origin Transfer

Fast Origin Transfer is incurred when using any of Vercel's compute products. These include Vercel Functions, Middleware, and the Data Ca

### Calculating Fast Origin Transfer

Usage is incurred on both the input and output data transfer when using compute on Vercel. For example:

- **Incoming:** The number of bytes sent as part of the [HTTP Request (Headers & Body)](https://developer.mozilla.org/en-US/docs/Web/HTTP
  - For common `GET` requests, the incoming bytes are normally inconsequential (less than 1KB for a normal request).
  - For `POST` requests, like a file upload API, the incoming bytes would include the entire uploaded file.
- **Outgoing:** The number of bytes sent as the [HTTP Response (Headers & Body)](https://developer.mozilla.org/en-US/docs/Web/HTTP/Messag

### Optimizing Fast Origin Transfer

#### Functions

> **💡 Note:** When using Incremental Static Regeneration (ISR) on Vercel, a Vercel Function
> is used to generate the static page. This optimization section applies for
> both server-rendered function usage, as well as usage for ISR. ISR usage on
> Vercel is billed under the Vercel Data Cache.

If using Vercel Functions, you can optimize Fast Origin Transfer by reducing the size of the response. Ensure your Function is only respo

You can also add [caching headers](/docs/cdn-cache) to the function response. By caching the response, future requests serve from the CDN

Ensure your Function supports `If-Modified-Since` or `Etag` to prevent duplicate data transmission ([on by default for Next.js applicatio

#### Middleware

If using Middleware, it is possible to accrue Fast Origin Transfer twice for a single Function request. To prevent this, you want to only

#### Investigating usage

- Look at the Fast Origin Transfer section of the Usage page:
  - Observe incoming vs outgoing usage. Reference the list above for optimization tips.
  - Observe the breakdown by project.
  - Observe the breakdown by region (Fast Origin Transfer is [priced regionally](#fast-origin-transfer))
- If optimizing Outgoing Fast Origin Transfer:
  - Observe the Top Paths on the Usage page
  - Filter by invocations to see which specific compute is being accessed most

## Edge Requests

When visiting your site, requests are made to a Vercel CDN [region](/docs/pricing/regional-pricing). Traffic is routed to the nearest reg

> **💡 Note:** Requests to regions are not only for Functions using the edge runtime. Edge
> Requests are for all requests made to your site, including static assets and
> functions.

### Managing Edge Requests

You can view the **Edge Requests** chart on the **Usage** tab of your dashboard. This chart shows:

- **Count**: The total count of requests made to your deployments
- **Projects**: The projects that received the requests
- **Region**: The region where the requests are made

As with all charts on the **Usage** tab, you can select the caret icon to view the chart in full screen mode.

### Optimizing Edge Requests

Frameworks such as [Next.js](/docs/frameworks/nextjs), [SvelteKit](/docs/frameworks/sveltekit), [Nuxt](/docs/frameworks/nuxt), and others

The most significant opportunities for optimizing Edge Requests include:

- **Identifying frequent re-mounting**: If your application involves rendering a large number of images and re-mounts them, it can inadve
  - **To identify**: Use your browsers devtools and browse your site. Pay attention to responses with a [304 status code](# "What is 304
- **Excessive polling or data fetching**: Applications that poll APIs for live updates, or use tools like SWR or React Query to reload da

## Edge Request CPU duration

Edge Request CPU Duration is the measurement of CPU processing time per Edge Request. Edge Requests of 10ms or less in duration do not in

### Managing Edge Request CPU duration

View the **Edge Request CPU Duration** chart on the **Usage** tab of your dashboard. If you notice an increase in CPU Duration, investiga

- Number of routes.
- Number of redirects.
- Complex regular expressions in routing.

To investigate further:

- Identify the deployment where the metric increased.
- Compare rewrites, redirects, and pages to the previous deployment.

---
title: "Deploy MCP servers to Vercel"
description: "Learn how to deploy Model Context Protocol (MCP) servers on Vercel with OAuth authentication and efficient scaling."
last_updated: "2026-01-16T02:19:33.304Z"
source: "https://vercel.com/docs/mcp/deploy-mcp-servers-to-vercel"
---

# Deploy MCP servers to Vercel

Deploy your Model Context Protocol (MCP) servers on Vercel to [take advantage of features](/docs/mcp/deploy-mcp-servers-to-vercel#deploy-

- Get started with [deploying MCP servers on Vercel](#deploy-an-mcp-server-on-vercel)
- Learn how to [enable authorization](#enabling-authorization) to secure your MCP server

## Deploy MCP servers efficiently

Vercel provides the following features for production MCP deployments:

- **Optimized cost and performance**: [Vercel Functions](/docs/functions) with [Fluid compute](/docs/fluid-compute) handle MCP servers' i
- [**Instant Rollback**](/docs/instant-rollback): Quickly revert to previous production deployments if issues arise with your MCP server.
- [**Preview deployments with Deployment Protection**](/docs/deployment-protection): Secure your preview MCP servers and test changes saf
- [**Vercel Firewall**](/docs/vercel-firewall): Protect your MCP servers from malicious attacks and unauthorized access with multi-layere
- [**Rolling Releases**](/docs/rolling-releases): Gradually roll out new MCP server deployments to a fraction of users before promoting t

## Deploy an MCP server on Vercel

Use the `mcp-handler` package and create the following API route to host an MCP server that provides a single tool that rolls a dice.

```ts filename="app/api/mcp/route.ts"
import { z } from 'zod';
import { createMcpHandler } from 'mcp-handler';

const handler = createMcpHandler(
  (server) => {
    server.tool(
      'roll_dice',
      'Rolls an N-sided die',
      { sides: z.number().int().min(2) },
      async ({ sides }) => {
        const value = 1 + Math.floor(Math.random() * sides);
        return {
          content: [{ type: 'text', text: `🎲 You rolled a ${value}!` }],
        };
      },
    );
  },
  {},
  { basePath: '/api' },
);

export { handler as GET, handler as POST, handler as DELETE };
```

### Test the MCP server locally

This assumes that your MCP server application, with the above-mentioned API route, runs locally at `http://localhost:3000`.

1. Run the MCP inspector:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i
    ```
  </Code>
</CodeBlock>

2. Open the inspector interface:
   - Browse to `http://127.0.0.1:6274` where the inspector runs by default
3. Connect to your MCP server:
   - Select **Streamable HTTP** in the drop-down on the left
   - In the **URL** field, use `http://localhost:3000/api/mcp`
   - Expand **Configuration**
   - In the **Proxy Session Token** field, paste the token from the terminal where your MCP server is running
   - Click **Connect**
4. Test the tools:
   - Click **List Tools** under Tools
   - Click on the `roll_dice` tool
   - Test it through the available options on the right of the tools section

When you deploy your application on Vercel, you will get a URL such as `https://my-mcp-server.vercel.app`.

### Configure an MCP host

Using [Cursor](https://www.cursor.com/), add the URL of your MCP server to the [configuration file](https://docs.cursor.com/context/model

```json filename=".cursor/mcp.json"
{
  "mcpServers": {
    "server-name": {
      "url": "https://my-mcp-server.vercel.app/api/mcp"
    }
  }
}
```

You can now use your MCP roll dice tool in [Cursor's AI chat](https://docs.cursor.com/context/model-context-protocol#using-mcp-in-chat) o

## Enabling authorization

The `mcp-handler` provides built-in OAuth support to secure your MCP server.
This ensures that only authorized clients with valid tokens can access your tools.

### Secure your server with OAuth

To add OAuth authorization to [the MCP server you created in the previous section](#deploy-an-mcp-server-on-vercel):

1. Use the `withMcpAuth` function to wrap your MCP handler
2. Implement token verification logic
3. Configure required scopes and metadata path

```typescript filename="app/api/[transport]/route.ts"
import { withMcpAuth } from 'mcp-handler';
import { AuthInfo } from '@modelcontextprotocol/sdk/server/auth/types.js';

const handler = createMcpHandler(/* ... same configuration as above ... */);

const verifyToken = async (
  req: Request,
  bearerToken?: string,
): Promise<AuthInfo | undefined> => {
  if (!bearerToken) return undefined;

  const isValid = bearerToken === '123';
  if (!isValid) return undefined;

  return {
    token: bearerToken,
    scopes: ['read:stuff'],
    clientId: 'user123',
    extra: {
      userId: '123',
    },
  };
};

const authHandler = withMcpAuth(handler, verifyToken, {
  required: true,
  requiredScopes: ['read:stuff'],
  resourceMetadataPath: '/.well-known/oauth-protected-resource',
});

export { authHandler as GET, authHandler as POST };
```

### Expose OAuth metadata endpoint

To comply with the MCP specification, your server must expose a [metadata endpoint](https://modelcontextprotocol.io/specification/draft/b
Among other things, this endpoint allows MCP clients to discover, how to authorize with your server, which authorization servers can issu
and what scopes are supported.

#### How to add OAuth metadata endpoint

1. In your `app/` directory, create a `.well-known` folder.
2. Inside this directory, create a subdirectory called `oauth-protected-resource`.
3. In this subdirectory, create a `route.ts` file with the following code for that specific route.
4. Replace the `https://example-authorization-server-issuer.com` URL with your own [Authorization Server (AS) Issuer URL](https://datatra

```typescript filename="app/.well-known/oauth-protected-resource/route.ts"
import {
  protectedResourceHandler,
  metadataCorsOptionsRequestHandler,
} from 'mcp-handler';

const handler = protectedResourceHandler({
  authServerUrls: ['https://example-authorization-server-issuer.com'],
});

const corsHandler = metadataCorsOptionsRequestHandler();

export { handler as GET, corsHandler as OPTIONS };
```

To view the full list of values available to be returned in the OAuth Protected Resource Metadata JSON, see the protected resource metada

MCP clients that are compliant with the latest version of the MCP spec can now securely connect and invoke tools defined in your MCP serv

## More resources

Learn how to deploy MCP servers on Vercel, connect to them using the AI SDK, and explore curated lists of public MCP servers.

- [Deploy an MCP server with Next.js on Vercel](https://vercel.com/templates/ai/model-context-protocol-mcp-with-next-js)
- [Deploy an MCP server with Vercel Functions](https://vercel.com/templates/other/model-context-protocol-mcp-with-vercel-functions)
- [Deploy an xmcp server](https://vercel.com/templates/backend/xmcp-boilerplate)
- [Learn about MCP server support on Vercel](https://vercel.com/changelog/mcp-server-support-on-vercel)
- [Use the AI SDK to initialize an MCP client on your MCP host to connect to an MCP server](https://ai-sdk.dev/docs/ai-sdk-core/tools-and
- [Use the AI SDK to call tools that an MCP server provides](https://ai-sdk.dev/docs/ai-sdk-core/tools-and-tool-calling#using-mcp-tools)
- [Explore the list from MCP servers repository](https://github.com/modelcontextprotocol/servers)
- [Explore the list from awesome MCP servers](https://github.com/punkpeye/awesome-mcp-servers)

---
title: "Model Context Protocol"
description: "Learn more about MCP and how you can use it on Vercel."
last_updated: "2026-01-16T02:19:33.318Z"
source: "https://vercel.com/docs/mcp"
---

# Model Context Protocol

[Model Context Protocol](https://modelcontextprotocol.io/) (MCP) is a standard interface that lets large language models (LLMs) communica

- [Get started with deploying MCP servers on Vercel](/docs/mcp/deploy-mcp-servers-to-vercel)
- Try out [Vercel's MCP server](/docs/mcp/vercel-mcp)

## Connecting LLMs to external systems

LLMs don't have access to real-time or external data by default. To provide relevant context—such as current financial data, pricing, or

Each tool or service has its own API, schema, and authentication. Managing these differences becomes difficult and error-prone as the num

## Standardizing LLM interaction with MCP

MCP standardizes the way LLMs interact with tools and data sources. Developers implement a single integration with MCP, and use it to man

Tool and data providers only need to expose an MCP interface once. After that, their system can be accessed by any MCP-enabled applicatio

MCP is like the USB-C standard: instead of needing different connectors for every device, you use one port to handle many types of connec

## MCP servers, hosts and clients

MCP uses a client-server architecture for the AI model to external system communication. The user connects to the AI application, referre

## More resources

Learn more about Model Context Protocol and explore available MCP servers.

- [Deploy your own MCP servers on Vercel](/docs/mcp/deploy-mcp-servers-to-vercel)
- [Use the AI SDK to initialize an MCP client on your MCP host to connect to an MCP server](https://ai-sdk.dev/docs/ai-sdk-core/tools-and
- [Use the AI SDK to call tools that an MCP server provides](https://ai-sdk.dev/docs/ai-sdk-core/tools-and-tool-calling#using-mcp-tools)
- [Use Vercel's MCP server](/docs/mcp/vercel-mcp)
- [Explore the list from MCP servers repository](https://github.com/modelcontextprotocol/servers)

---
title: "Use Vercel"
description: "Vercel MCP has tools available for searching docs along with managing teams, projects, and deployments."
last_updated: "2026-01-16T02:19:33.365Z"
source: "https://vercel.com/docs/mcp/vercel-mcp"
---

# Use Vercel

Connect your AI tools to Vercel using the [Model Context Protocol (MCP)](https://modelcontextprotocol.io),
an open standard that lets AI assistants interact with your Vercel projects.

## What is Vercel MCP?

Vercel MCP is Vercel's official MCP server. It's a remote MCP with OAuth that gives AI tools secure access to your Vercel projects availa

`https://mcp.vercel.com`

It integrates with popular AI assistants like Claude, enabling you to:

- Search and navigate Vercel documentation
- Manage projects and deployments
- Analyze deployment logs

Vercel MCP implements the latest [MCP Authorization](https://modelcontextprotocol.io/specification/2025-06-18/basic/authorization)
and [Streamable HTTP](https://modelcontextprotocol.io/specification/2025-06-18/basic/transports#streamable-http)
specifications.

## Available tools

Vercel MCP provides a comprehensive set of tools for searching documentation and managing your Vercel projects. See the [tools reference]

## Connecting to Vercel MCP

To ensure secure access, Vercel MCP only supports AI clients that have been reviewed and approved by Vercel.

## Supported clients

The list of supported AI tools that can connect to Vercel MCP to date:

- [Claude Code](#claude-code)
- [Claude.ai and Claude for desktop](#claude.ai-and-claude-for-desktop)
- [ChatGPT](#chatgpt)
- [Codex CLI](#codex-cli)
- [Cursor](#cursor)
- [VS Code with Copilot](#vs-code-with-copilot)
- [Devin](#devin)
- [Raycast](#raycast)
- [Goose](#goose)
- [Windsurf](#windsurf)
- [Gemini Code Assist](#gemini-code-assist)
- [Gemini CLI](#gemini-cli)

Additional clients will be added over time.

## Setup

Connect your AI client to Vercel MCP and authorize access to manage your Vercel projects.

### Claude Code

```bash
# Install Claude Code
npm install -g @anthropic-ai/claude-code

# Navigate to your project
cd your-awesome-project

# Add Vercel MCP (general access)
claude mcp add --transport http vercel https://mcp.vercel.com

# Add Vercel MCP (project-specific access)
claude mcp add --transport http vercel-awesome-ai https://mcp.vercel.com/my-team/my-awesome-project

# Start coding with Claude
claude
```

```
# Authenticate the MCP tools by typing /mcp
/mcp
```

> **💡 Note:** You can add multiple Vercel MCP connections with different names for different
> projects. For example: `vercel-cool-project`, `vercel-awesome-ai`,
> `vercel-super-app`, etc.

### Claude.ai and Claude for desktop

> **💡 Note:** Custom connectors using remote MCP are available on Claude and Claude Desktop
> for users on [Pro, Max, Team, and Enterprise
> plans](https://support.anthropic.com/en/articles/11175166-getting-started-with-custom-connectors-using-remote-mcp).

1. Open **Settings** in the sidebar
2. Navigate to **Connectors** and select **Add custom connector**
3. Configure the connector:
   - Name: `Vercel`
   - URL: `https://mcp.vercel.com`

### ChatGPT

> **💡 Note:** Custom connectors using MCP are available on ChatGPT for [Pro and Plus
> accounts](https://platform.openai.com/docs/guides/developer-mode#how-to-use)
> on the web.

Follow these steps to set up Vercel as a connector within ChatGPT:

1. Enable [Developer mode](https://platform.openai.com/docs/guides/developer-mode):
   - Go to [Settings → Connectors](https://chatgpt.com/#settings/Connectors) → Advanced settings → Developer mode
2. Open [ChatGPT settings](https://chatgpt.com/#settings)
3. In the Connectors tab, `Create` a new connector:
   - Give it a name: `Vercel`
   - MCP server URL: `https://mcp.vercel.com`
   - Authentication: `OAuth`
4. Click **Create**

The Vercel connector will appear in the composer's ["Developer mode"](https://platform.openai.com/docs/guides/developer-mode) tool later

### Codex CLI

[Codex CLI](https://developers.openai.com/codex/cli/) is OpenAI’s local coding agent that can run directly from your terminal.

```bash
# Install Codex
npm i -g @openai/codex

# Add Vercel MCP
codex mcp add vercel --url https://mcp.vercel.com

# Start Codex
codex
```

When adding the MCP server, Codex will detect OAuth support and open your browser to authorize the connection.

### Cursor

Click the button above to open Cursor and automatically add Vercel MCP. You can
also add the snippet below to your project-specific or global `.cursor/mcp.json`
file manually. For more details, see the [Cursor
documentation](https://docs.cursor.com/en/context/mcp).

```json
{
  "mcpServers": {
    "vercel": {
      "url": "https://mcp.vercel.com"
    }
  }
}
```

Once the server is added, Cursor will attempt to connect and display a `Needs login` prompt. Click on this prompt to authorize Cursor to

### VS Code with Copilot

#### Installation

Use the one-click installation by clicking the button above to add Vercel MCP, or follow the steps below to do it manually:

1. Open the Command Palette ( on Windows/Linux or  on macOS)
2. Run **MCP: Add Server**
3. Select **HTTP**
4. Enter the following details:
   - **URL:** `https://mcp.vercel.com`
   - **Name:** `Vercel`
5. Select **Global** or **Workspace** depending on your needs
6. Click **Add**

#### Authorization

Now that you've added Vercel MCP, let's start the server and authorize:

1. Open the Command Palette ( on Windows/Linux or  on macOS)
2. Run **MCP: List Servers**
3. Select **Vercel**
4. Click **Start Server**
5. When the dialog appears saying `The MCP Server Definition 'Vercel' wants to authenticate to Vercel MCP`, click **Allow**
6. A popup will ask `Do you want Code to open the external website?` — click **Cancel**

7. You'll see a message: `Having trouble authenticating to 'Vercel MCP'? Would you like to try a different way? (URL Handler)`
8. Click **Yes**
9. Click **Open** and complete the Vercel sign-in flow to connect to Vercel MCP

### Devin

1. Navigate to [Settings > MCP Marketplace](https://app.devin.ai/settings/mcp-marketplace)
2. Search for "Vercel" and select the MCP
3. Click **Install**

### Raycast

1. Run the **Install Server** command
2. Enter the following details:
   - **Name:** `Vercel`
   - **Transport:** HTTP
   - **URL:** `https://mcp.vercel.com`
3. Click **Install**

### Goose

Use the one-click installation by clicking the button below to add Vercel MCP. For more details, see the [Goose documentation](https://block.github.io/goose/docs/getting-started/using-extensions/#mcp-servers).

### Windsurf

Add the snippet below to your `mcp_config.json` file. For more details, see the [Windsurf documentation](https://docs.windsurf.com/windsurf/cascade/mcp#adding-a-new-mcp-plugin).

```json
{
  "mcpServers": {
    "vercel": {
      "serverUrl": "https://mcp.vercel.com"
    }
  }
}
```

### Gemini Code Assist

Gemini Code Assist is an IDE extension that supports MCP integration. To set up Vercel MCP with Gemini Code Assist:

1. Ensure you have Gemini Code Assist installed in your IDE
2. Add the following configuration to your `~/.gemini/settings.json` file:

```json
{
  "mcpServers": {
    "vercel": {
      "command": "npx",
      "args": ["mcp-remote", "https://mcp.vercel.com"]
    }
  }
}
```

3. Restart your IDE to apply the configuration
4. When prompted, authenticate with Vercel to grant access

### Gemini CLI

Gemini CLI shares the same configuration as [Gemini Code Assist](#gemini-code-assist). To set up Vercel MCP with Gemini CLI:

1. Ensure you have the Gemini CLI installed
2. Add the following configuration to your `~/.gemini/settings.json` file:

```json
{
  "mcpServers": {
    "vercel": {
      "command": "npx",
      "args": ["mcp-remote", "https://mcp.vercel.com"]
    }
  }
}
```

3. Run the Gemini CLI and use the `/mcp list` command to see available MCP servers
4. When prompted, authenticate with Vercel to grant access

For more details on configuring MCP servers with Gemini tools, see the [Google documentation](https://developers.google.com/gemini-code-a

> **💡 Note:** Setup steps may vary based on your MCP client version. Always check your client's documentation for the latest instructions.

## Security best practices

The MCP ecosystem and technology are evolving quickly. Here are our current best practices to help you keep your workspace secure:

- **Verify the official endpoint**
  - Always confirm you're connecting to Vercel's official MCP endpoint: `https://mcp.vercel.com`

- **Trust and verification**
  - Only use MCP clients from trusted sources and review our [list of supported clients](#supported-clients)
  - Connecting to Vercel MCP grants the AI system you're using the same access as your Vercel user account
  - When you use "one-click" MCP installation from a third-party marketplace, double-check the domain name/URL to ensure it's one you and

- **Security awareness**

  - Familiarize yourself with key security concepts like [prompt injection](https://vercel.com/blog/building-secure-ai-agents) to better

- **Confused deputy protection**
  - Vercel MCP protects against [confused deputy attacks](https://modelcontextprotocol.io/specification/draft/basic/security_best_practic
  - This prevents attackers from exploiting consent cookies to gain unauthorized access to your Vercel account through malicious authoriz

- **Protect your data**
  - Bad actors could exploit untrusted tools or agents in your workflow by inserting malicious instructions like "ignore all previous ins

  - If the agent follows those instructions using the Vercel MCP, it could lead to unauthorized data sharing.

  - When setting up workflows, carefully review the permissions and data access levels of each agent and MCP tool.

  - Keep in mind that while Vercel MCP only operates within your Vercel account, any external tools you connect could potentially share d

- **Enable human confirmation**
  - Always enable human confirmation in your workflows to maintain control and prevent unauthorized changes
  - This allows you to review and approve each step before it's executed
  - Prevents accidental or harmful changes to your projects and deployments

## Advanced Usage

### Project-specific MCP access

For enhanced functionality and better tool performance, you can use project-specific MCP URLs that automatically provide the necessary pr

`https://mcp.vercel.com/<teamSlug>/<projectSlug>`

#### Benefits of project-specific URLs

- **Automatic context**: The MCP server automatically knows which project and team you're working with
- **Improved tool performance**: Tools can execute without requiring manual parameter input
- **Better error handling**: Reduces errors from missing project slug or team slug parameters
- **Streamlined workflow**: No need to manually specify project context in each tool call

#### When to use project-specific URLs

Use project-specific URLs when:

- You're working on a specific Vercel project
- You want to avoid manually providing project and team slugs
- You're experiencing errors like "Project slug and Team slug are required"

#### Finding your team slug and project slug

You can find your team slug and project slug in several ways:

1. **From the Vercel [dashboard](/dashboard)**:
   - **Project slug**: Navigate to your project → Settings → General (sidebar tab)
   - **Team slug**: Navigate to your team → Settings → General (sidebar tab)
2. **From the Vercel CLI**: Use `vercel projects ls` to list your projects

#### Example usage

Instead of using the general MCP endpoint and manually providing parameters, you can use:

```
https://mcp.vercel.com/my-team/my-awesome-project
```

This automatically provides the context for team `my-team` and project `my-awesome-project`, allowing tools to execute without additional


---
title: "Tools"
description: "Available tools in Vercel MCP for searching docs and managing teams, projects, and deployments."
last_updated: "2026-01-16T02:19:33.395Z"
source: "https://vercel.com/docs/mcp/vercel-mcp/tools"
---

# Tools

The Vercel MCP server provides the following [MCP tools](https://modelcontextprotocol.io/specification/2025-06-18/server/tools).
To enhance security, enable human confirmation for tool execution and exercise caution when using Vercel MCP alongside other servers to p

## Tools

### Documentation tools

| Name | Description | Parameters |
| --- | --- | --- |
| **search_documentation** | Search Vercel documentation for specific topics and information | `topic` (string, required): Topic to focu

### Project Management Tools

| Name | Description |
| --- | --- |
| **list_teams** | List all [teams](/docs/accounts) that include the authenticated user as a member |
| **list_projects** | List all Vercel [projects](/docs/projects) associated with a user |
| **get_project** | Retrieve detailed information about a specific [project](/docs/projects) including framework, domains, and latest

### Deployment Tools

| Name | Description |
| --- | --- |
| **list_deployments** | List [deployments](/docs/deployments) associated with a specific project with creation time, state, an |
| **get_deployment** | Retrieve detailed information for a specific [deployment](/docs/deployments) including build status, r |
| **get_deployment_build_logs** | Get the build logs of a deployment by deployment ID or URL. Can be used to investigate why a deploym

### Domain Management Tools

| Name | Description | Parameters
| --- | --- | ---
| **check_domain_availability_and_price** | Check if domain names are available for purchase and get pricing information | `names` (a
| **buy_domain** | Purchase a domain name with registrant information | `name` (strin

### Access Tools

| Name | Description
| --- | ---
| **get_access_to_vercel_url** | Creates a temporary [shareable link](/docs/deployment-protection/methods-to-bypass-deployment-protec
| **web_fetch_vercel_url** | Allows agents to directly fetch content from a Vercel deployment URL (with [authentication](/docs/dep

### CLI Tools

| Name | Description | Parameters
| --- | --- | ---
| **use_vercel_cli** | Instructs the LLM to use Vercel CLI commands with --help flag for information | `command` (string, optional):
| **deploy_to_vercel** | Deploy the current project to Vercel | None

---
title: "Microfrontends Configuration"
description: "Configure your microfrontends.json."
last_updated: "2026-01-16T02:19:33.400Z"
source: "https://vercel.com/docs/microfrontends/configuration"
---

# Microfrontends Configuration

The `microfrontends.json` file is used to configure your microfrontends. If this file is not deployed with your [default application](/do

## Schema

## Example

```json filename="microfrontends.json"
{
  "$schema": "https://openapi.vercel.sh/microfrontends.json",
  "applications": {
    "nextjs-pages-dashboard": {
      "development": {
        "fallback": "nextjs-pages-dashboard.vercel.app"
      }
    },
    "nextjs-pages-blog": {
      "routing": [
        {
          "paths": ["/blog/:path*"]
        },
        {
          "flag": "enable-flagged-blog-page",
          "paths": ["/flagged/blog"]
        }
      ]
    }
  }
}
```

## Application Naming

If the application name differs from the `name` field in `package.json` for the application, you should either rename the name field in `

```json filename="microfrontends.json"
    "docs": {
      "packageName": "name-from-package-json",
      "routing": [
        {
          "group": "docs",
          "paths": ["/docs/:path*"]
        }
      ]
    }
```

## File Naming

The microfrontends configuration file can be named either `microfrontends.json` or `microfrontends.jsonc`.

You can also define a custom configuration file by setting the `VC_MICROFRONTENDS_CONFIG_FILE_NAME` environment variable — for example, `

Be sure to add the [environment variable](/docs/environment-variables/managing-environment-variables) to all projects within the microfro

Using a custom file name allows the same repository to support multiple microfrontends groups, since each group can have its own configur

If you're using Turborepo, define the environment variable **outside** of the Turbo invocation when running `turbo dev`, so the local pro

```bash
VC_MICROFRONTENDS_CONFIG_FILE_NAME="microfrontends-dev.json" turbo dev
```

---
title: "Microfrontends local development"
description: "Learn how to run and test your microfrontends locally."
last_updated: "2026-01-16T02:19:33.460Z"
source: "https://vercel.com/docs/microfrontends/local-development"
---

# Microfrontends local development

To provide a seamless local development experience, `@vercel/microfrontends` provides a microfrontends aware local development proxy to r

## The need for a microfrontends proxy

Microfrontends allow teams to split apart an application and only run an individual microfrontend to improve developer velocity. A downsi

For example, if you have two microfrontends `web` and `docs`:

```json filename="microfrontends.json"
{
  "$schema": "https://openapi.vercel.sh/microfrontends.json",
  "applications": {
    "web": {
      "development": {
        "fallback": "vercel.com"
      }
    },
    "docs": {
      "routing": [
        {
          "paths": ["/docs/:path*"]
        }
      ]
    }
  }
}
```

A developer working on `/docs` only runs the **Docs** microfrontend, while a developer working on `/blog` only runs the **Web** microfron

Therefore, the microfrontends proxy allows developers to run only the microfrontend they are working on locally and be able to test paths

> **⚠️ Warning:** When developing locally with Next.js any traffic a child application receives
> will be redirected to the local proxy. Setting the environment variable
> `MFE_DISABLE_LOCAL_PROXY_REWRITE=1` will disable the redirect and allow you to
> visit the child application directly.

## Setting up microfrontends proxy

### Prerequisites

- Set up your [microfrontends on Vercel](/docs/microfrontends/quickstart)
- All applications that are part of the microfrontend have `@vercel/microfrontends` listed as a dependency
- Optional: [Turborepo](https://turborepo.com) in your repository

- ### Application setup
  In order for the local proxy to redirect traffic correctly, it needs to know which port each application's development server will be u
  ```json {4} filename="package.json"
  {
    "name": "web",
    "scripts": {
      "dev": "next --port $(microfrontends port)"
    },
    "dependencies": {
      "@vercel/microfrontends": "latest"
    }
  }
  ```
  If you would like to use a specific port for each application, you may configure that in `microfrontends.json`:
  ```json {11-15} filename="microfrontends.json"
  {
    "$schema": "https://openapi.vercel.sh/microfrontends.json",
    "applications": {
      "web": {},
      "docs": {
        "routing": [
          {
            "paths": ["/docs/:path*"]
          }
        ],
        "development": {
          "task": "start",
          "local": 3001
        }
      }
    }
  }
  ```
  The `local` field may also contain a host or protocol (for example, `my.special.localhost.com:3001` or `https://my.localhost.com:3030`)

  If the name of the application in `microfrontends.json` (such as `web` or `docs`) does not match the name used in `package.json`, you c
  ```json {11} filename="microfrontends.json"
  {
    "$schema": "https://openapi.vercel.sh/microfrontends.json",
    "applications": {
      "web": {},
      "docs": {
        "routing": [
          {
            "paths": ["/docs/:path*"]
          }
        ],
        "packageName": "my-docs-package"
      }
    }
  }
  ```
  ```json {2} filename="package.json"
  ```

  {
    "name": "my-docs-package",
    "scripts": {
      "dev": "next --port $(microfrontends port)"
    },
    "dependencies": {
      "@vercel/microfrontends": "latest"
    }
  }
  ```

- ### Starting local proxy
  The local proxy is started automatically when running a microfrontend development task with `turbo`. By default a microfrontend applica

  Running `turbo web#dev` will start the `web` microfrontends development server along with a local proxy that routes all requests for `d
  > **💡 Note:** This requires version `2.3.6` or `2.4.2` or newer of the `turbo` package.

- ### Setting up your monorepo
  - ### Option 1: Adding Turborepo to a monorepo
    Turborepo is the suggested way to work with microfrontends as it provides a managed way for running multiple applications and a proxy

    If you don't already use [Turborepo](https://turborepo.com) in your monorepo, `turbo` can infer a configuration from your `microfront

    To get started, follow the [Installing `turbo`](https://turborepo.com/docs/getting-started/installation#installing-turbo) guide.

    Once you have installed `turbo`, run your development tasks using `turbo` instead of your package manager. This will start the local

    You can start the development task for the **Web** microfrontend by running `turbo run dev --filter=web`. Review Turborepo's [filter

    For more information on adding Turborepo to your repository, review [adding Turborepo to an existing repository](https://turborepo.co
  - ### Option 2: Using without Turborepo
    If you do not want to use Turborepo, you can invoke the proxy directly.
    ```json {5} filename="package.json"
    {
      "name": "web",
      "scripts": {
        "dev": "next --port $(microfrontends port)",
        "proxy": "microfrontends proxy microfrontends.json --local-apps web"
      },
      "dependencies": {
        "@vercel/microfrontends": "latest"
      }
    }
    ```
    Review [Understanding the proxy command](#understanding-the-proxy-command) for more details.

- ### Accessing the microfrontends proxy
  When testing locally, you should use the port from the microfrontends proxy to test your application. For example, if `docs` runs on po

  You can change the port of the local development proxy by setting `options.localProxyPort` in `microfrontends.json`:
  ```json {6} filename="microfrontends.json"
  {
    "applications": {
      // ...
    },
    "options": {
      "localProxyPort": 4001
    }
  }
  ```

## Debug routing

To debug issues with microfrontends locally, enable microfrontends debug mode when running your application. Details about changes to you

1. Set an environment variable `MFE_DEBUG=1`
2. Or, set `debug` to `true` in when calling `withMicrofrontends`

## Polyrepo setup

If you're working with a polyrepo setup where microfrontends are distributed across separate repositories, you'll need additional configu

### Accessing the configuration file

First, ensure that each microfrontend repository has access to the shared configuration:

- **Option 1: Use the Vercel CLI** to fetch the configuration:

  ```bash
  vercel microfrontends pull
  ```

  This command will download the `microfrontends.json` file from your default application to your local repository.

  If you haven't linked your project yet, the command will prompt you to [link your project to Vercel](https://vercel.com/docs/cli/projec

  > **💡 Note:** This command requires the Vercel CLI 44.2.2 to be installed.

- **Option 2: Set the `VC_MICROFRONTENDS_CONFIG` environment variable** with a path pointing to your `microfrontends.json` file:

  ```bash
  export VC_MICROFRONTENDS_CONFIG=/path/to/microfrontends.json
  ```

  You can also add this to your `.env` file:

  ```bash filename=".env"
  VC_MICROFRONTENDS_CONFIG=/path/to/microfrontends.json
  ```
  ```

### Running the local development proxy

In a polyrepo setup, you'll need to start each microfrontend application separately since they're in different repositories. Unlike monor

- ### Start your local microfrontend application
  Start your microfrontend application with the proper port configuration. Follow the [Application setup](/docs/microfrontends/local-deve

- ### Run the microfrontends proxy
  In the same or a separate terminal, start the microfrontends proxy:
  ```bash
  microfrontends proxy --local-apps your-app-name
  ```
  Make sure to specify the correct application name that matches your `microfrontends.json` configuration.

- ### Access your application
  Visit the proxy URL shown in the terminal output (typically `http://localhost:3024`) to test the full microfrontends experience. This U

Since you're working across separate repositories, you'll need to manually start any other microfrontends you want to test locally, each

## Understanding the proxy command

When setting up your monorepo without turborepo, the `proxy` command used inside the `package.json` scripts has the following specificati

- `microfrontends` is an executable provided by the `@vercel/microfrontends` package.
  - You can also run it with a command like `npm exec microfrontends ...` (or the equivalent for your package manager), as long as it's f
- `proxy` is a sub-command to run the local proxy.
- `microfrontends.json` is the path to your microfrontends configuration file. If you have a monorepo, you may also leave this out and th
- `--local-apps` is followed by a space separated list of the applications running locally. For the applications provided in this list, t

For example, if you are running the **Web** and **Docs** microfrontends locally, this command would set up the local proxy to route reque

```json filename="package.json"
microfrontends proxy microfrontends.json --local-apps web docs
```

We recommend having a proxy command associated with each application in your microfrontends group. For example:

- If you run `npm run docs-dev` to start up your `docs` application for local development, set up `npm run docs-proxy` as well
  - This should pass `--local-apps docs` so it sends requests to the local `docs` application, and everything else to the fallback.

Therefore, you can run `npm run docs-dev` and `npm run docs-proxy` to get the full microfrontends setup running locally.

## Falling back to protected deployments

To fall back to a Vercel deployment protected with [Deployment Protection](/docs/deployment-protection), set an environment variable with

You must name the environment variable `AUTOMATION_BYPASS_<transformed app name>`. The name is transformed to be uppercase, and any non l

For example, the env var name for an app named `my-docs-app` would be:
`AUTOMATION_BYPASS_MY_DOCS_APP`.

### Set the protection bypass environment variable

- ### Enable the Protection Bypass for Automation for your project
  1. Navigate to the Vercel **project for the protected fallback deployment**
  2. Click on the **Settings** tab
  3. Click on **Deployment Protection**
  4. If not enabled, create a new [Protection Bypass for Automation](/docs/deployment-protection/methods-to-bypass-deployment-protection/
  5. Copy the value of the secret

- ### Set the environment variable in the default app project
  1. Navigate to the Vercel project for the **default application** (may or may not be the same project)
  2. Click on the **Settings** tab
  3. Click on **Environment Variables**
  4. Add a new variable with the name `AUTOMATION_BYPASS_<transformed app name>` (e.g. `AUTOMATION_BYPASS_MY_DOCS_APP`) and the value of
  5. Set the selected environments for the variable to `Development`
  6. Click on **Save**

- ### Import the secret using vc env pull
  1. Ensure you have [vc](https://vercel.com/cli) installed
  2. Navigate to the root of the default app folder
  3. Run `vc login` to authenticate with Vercel
  4. Run `vc link` to link the folder to the Vercel project
  5. Run `vc env pull` to pull the secret into your local environment

- ### Update your README.md
  Include [the previous step](#import-the-secret-using-vc-env-pull) in your repository setup instructions, so that other users will also

---
title: "Managing microfrontends"
description: "Learn how to manage your microfrontends on Vercel."
last_updated: "2026-01-16T02:19:33.438Z"
source: "https://vercel.com/docs/microfrontends/managing-microfrontends"
---

# Managing microfrontends

With a project's **Microfrontends** settings of the Vercel dashboard, you can:

- [Add](#adding-microfrontends) and [remove](#removing-microfrontends) microfrontends
- [Share settings](#sharing-settings-between-microfrontends) between microfrontends
- [Route Observability data](#observability-data-routing)
- [Manage security](/docs/microfrontends/managing-microfrontends/security) with Deployment Protection and Firewall

You can also use the [Vercel Toolbar to manage microfrontends](/docs/microfrontends/managing-microfrontends/vercel-toolbar).

## Adding microfrontends

To add projects to a microfrontends group:

1. Visit the **Settings** tab for the project that you would like to add or remove.
2. Click on the **Microfrontends** tab.
3. Find the microfrontends group that it is being added to and Click **Add to Group**.

These changes will take effect on the next deployment.

## Removing microfrontends

To remove projects from a microfrontends group:

1. Remove the microfrontend from the `microfrontends.json` in the default application.
2. Visit the **Settings** tab for the project that you would like to add or remove.
3. Click on the **Microfrontends** tab.
4. Find the microfrontends group that the project is a part of. Click **Remove from Group** to remove it from the group.

Make sure that no other microfrontend is referring to this project. These changes will take effect on the next deployment.

> **💡 Note:** Projects that are the default application for the microfrontends group can
> only be removed after all other projects in the group have been removed. A
> microfrontends group can be deleted once all projects have been removed.

## Fallback environment

> **💡 Note:** This setting only applies to
> [preview](/docs/deployments/environments#preview-environment-pre-production)
> and [custom environments](/docs/deployments/environments#custom-environments).
> Requests for the
> [production](/docs/deployments/environments#production-environment)
> environment are always routed to the production deployment for each
> microfrontend project.

When microfrontend projects are not built for a commit in [preview](/docs/deployments/environments#preview-environment-pre-production)
or [custom environments](/docs/deployments/environments#custom-environments), Vercel will route those requests to a specified fallback so

There are three options for the fallback environment setting:

- `Same Environment` - Requests to microfrontends not built for that commit will fall back to a deployment for the other microfrontend pr
  - For example, in the `Preview` environment, requests to a microfrontend that was not built for that commit would fallback to the `Prev
  - When this setting is used, Vercel will generate `Preview` deployments on the production branch for each microfrontend project automat
- `Production` - Requests to microfrontends not built for this commit will fall back to the promoted Production deployment for that other
- A specific [custom environment](/docs/deployments/environments#custom-environments) - Requests to microfrontends not built for this com

This table illustrates the different fallback scenarios that could arise:

| Current Environment | Fallback Environment | If Microfrontend Built for Commit | If Microfrontend Did Not Build for Co |
| --- | --- | --- | --- |
| `Preview` | `Same Environment` | `Preview` | `Preview` |
| `Preview` | `Production` | `Preview` | `Production` |
| `Preview` | `staging` Custom Environment | `Preview` | `staging` Custom Environment |
| `staging` Custom Environment | `Same Environment` | `staging` Custom Environment | `staging` Custom Environment |
| `staging` Custom Environment | `Production` | `staging` Custom Environment | `Production` |
| `staging` Custom Environment | `staging` Custom Environment | `staging` Custom Environment | `staging` Custom Environment |

If the current environment is `Production`, requests will always be routed to the `Production` environment of the other project.

> **💡 Note:** If using the `Same Environment` or `Custom Environment` options, you may need
> to make sure that those environments have a deployment to fall back to. For
> example, if using the `Custom Environment` option, each project in the
> microfrontends group will need to have a Custom Environment with the specified
> name. If environments are not configured correctly, you may see a
> [MICROFRONTENDS_MISSING_FALLBACK_ERROR](/docs/errors/MICROFRONTENDS_MISSING_FALLBACK_ERROR)
> on the request.

To configure this setting, visit the **Settings** tab for the microfrontends group and configure the **Fallback Environment** setting.

### Project domains for git branches

If your project has a [project domain assigned to a Git branch](/docs/domains/working-with-domains/assign-domain-to-a-git-branch), and th

To use that branch across the microfrontends group, add a project domain for the branch to every project in the group.

## Sharing settings between microfrontends

To share settings between Vercel microfrontend projects, you can use the [Vercel Terraform Provider](https://registry.terraform.io/provid

- [Microfrontend group resource](https://registry.terraform.io/providers/vercel/vercel/latest/docs/resources/microfrontend_group)
- [Microfrontend group membership resource](https://registry.terraform.io/providers/vercel/vercel/latest/docs/resources/microfrontend_gro

### Sharing environment variables

[Shared Environment Variables](/docs/environment-variables/shared-environment-variables) allow you to manage a single secret and share it

To use environment variables with the same name but different values for different project groups, you can create a shared environment va

## Optimizing navigation's between microfrontends

> **💡 Note:** This feature is currently only supported for Next.js.

Navigations between different top level microfrontends will introduce a hard navigation for users. Vercel optimizes these navigations by

> For \['nextjs-app']:

To get started, add the `PrefetchCrossZoneLinks` element to your `layout.tsx` or `layout.jsx` file in all your microfrontend applications

> For \['nextjs']:

To get started, add the `PrefetchCrossZoneLinks` element to your `_app.tsx` or `_app.jsx` file:

Then in all microfrontends, use the `Link` component from `@vercel/microfrontends/next/client` anywhere you would use a normal link to au

```tsx
import { Link } from '@vercel/microfrontends/next/client';

export function MyComponent() {
  return (
    <>
      <Link href="/docs">Docs</Link>
    </>
  );
}
```

> **💡 Note:** When using this feature, all paths from the `microfrontends.json` file will be
> visible on the client side. This information is used to know which
> microfrontend each link comes from in order to apply prefetching and
> prerendering.

## Observability data routing

By default, observability data from [Speed Insights](/docs/speed-insights) and [Analytics](/docs/analytics) is routed to the default appl

Microfrontends also provides an option to route a project's own observability data directly to that Vercel project's page.

1. Ensure your Speed Insights and Analytics package dependencies are up to date. For this feature to work:
   - `@vercel/speed-insights` (if using) must be at version `1.2.0` or newer
   - `@vercel/analytics` (if using) must be at version `1.5.0` or newer
2. Visit the **Settings** tab for the project that you would like to change data routing.
3. Click on the **Microfrontends** tab.
4. Search for the **Observability Routing** setting.
5. Enable the setting to route the project's data to the project. Disable the setting to route the project's data to the default applicat
6. The setting will go into effect for the project's next production deployment.

> **💡 Note:** Enabling or disabling this feature will **not** move existing data between the
> default application and the individual project. Historical data will remain in
> place.

If you are using Turborepo with `--env-mode=strict`, you need to either add `ROUTE_OBSERVABILITY_TO_THIS_PROJECT` and `NEXT_PUBLIC_VERCEL_

---
title: "Managing microfrontends security"
description: "Learn how to manage your Deployment Protection and Firewall for your microfrontend on Vercel."
last_updated: "2026-01-16T02:19:33.415Z"
source: "https://vercel.com/docs/microfrontends/managing-microfrontends/security"
---

# Managing microfrontends security

Understand how and where you manage [Deployment Protection](/docs/deployment-protection) and [Vercel Firewall](/docs/vercel-firewall) for

- [Deployment Protection and microfrontends](#deployment-protection-and-microfrontends)
- [Vercel Firewall and microfrontends](#vercel-firewall-and-microfrontends)

## Deployment Protection and microfrontends

For requests to a microfrontend host (a domain belonging to the microfrontend default application):

- Requests are **only** verified by the [Deployment Protection](/docs/security/deployment-protection) settings for the project of your **

For requests directly to a child application (a domain belonging to a child microfrontend):

- Requests are **only** verified by the [Deployment Protection](/docs/security/deployment-protection) settings for the project of the **c

This applies to all [protection methods](/docs/security/deployment-protection/methods-to-protect-deployments) and [bypass methods](/docs/

- [Vercel Authentication](/docs/security/deployment-protection/methods-to-protect-deployments/vercel-authentication)
- [Password Protection](/docs/security/deployment-protection/methods-to-protect-deployments/password-protection)
- [Trusted IPs](/docs/security/deployment-protection/methods-to-protect-deployments/trusted-ips)
- [Shareable Links](/docs/security/deployment-protection/methods-to-bypass-deployment-protection/sharable-links)
- [Protection Bypass for Automation](/docs/security/deployment-protection/methods-to-bypass-deployment-protection/protection-bypass-autom
- [Deployment Protection Exceptions](/docs/security/deployment-protection/methods-to-bypass-deployment-protection/deployment-protection-e
- [OPTIONS Allowlist](/docs/security/deployment-protection/methods-to-bypass-deployment-protection/options-allowlist).

### Managing Deployment Protection for your microfrontend

Use the [Deployment Protection](/docs/security/deployment-protection) settings for the project of the default application for the group.

## Vercel Firewall and microfrontends

- The [Platform-wide firewall](/docs/vercel-firewall#platform-wide-firewall) is applied to all requests.
- The customizable [Web Application Firewall (WAF)](/docs/vercel-firewall/vercel-waf) from the default application and the corresponding

### Vercel WAF and microfrontends

For requests to a microfrontend host (a domain belonging to the microfrontend default application):

- All requests are verified by the [Vercel WAF](/docs/vercel-firewall/vercel-waf) for the project of your default application
- Requests to child applications are **additionally** verified by the [Vercel WAF](/docs/vercel-firewall/vercel-waf) for their project

For requests directly to a child application (a domain belonging to a child microfrontend):

- Requests are **only** verified by the [Vercel WAF](/docs/vercel-firewall/vercel-waf) for the project of the child application.

This applies for the entire [Vercel WAF](/docs/vercel-firewall/vercel-waf), including [Custom Rules](/docs/vercel-firewall/vercel-waf/cus

### Managing the Vercel WAF for your microfrontend

- To set a WAF rule that applies to all requests to a microfrontend, use the [Vercel WAF](/docs/vercel-firewall/vercel-waf) for your defa

- To set a WAF rule that applies **only** to requests to paths of a child application, use the [Vercel WAF](/docs/vercel-firewall/vercel-

---
title: "Managing with the Vercel Toolbar"
description: "Learn how to use the Vercel Toolbar to make it easier to manage microfrontends."
last_updated: "2026-01-16T02:19:33.407Z"
source: "https://vercel.com/docs/microfrontends/managing-microfrontends/vercel-toolbar"
---

# Managing with the Vercel Toolbar

Using the [Vercel Toolbar](/docs/vercel-toolbar), you can visualize and independently test your microfrontends so you can develop microfr

You can access it in all microfrontends that you have [enabled the toolbar for](/docs/vercel-toolbar/in-production-and-localhost).

> **💡 Note:** This requires version `0.1.33` or newer of the `@vercel/toolbar` package.

## View all microfrontends

In the **Microfrontends** panel of the toolbar shows all microfrontends that are available in that microfrontends group. By clicking on e

## Microfrontends zone indicator

Since multiple microfrontends can serve content on the same domain, it's easy to lose track of which application is serving that page. Us

You find the **Zone Indicator** toggle at the bottom of the **Microfrontends** panel in the Vercel toolbar.

## Routing overrides

While developing microfrontends, you often want to build and test just your microfrontend in isolation to avoid dependencies on other pro

1. Open the **microfrontends panel** in the Vercel Toolbar.
2. Find the application that you want to modify in the list of microfrontends.
3. In the **Routing** section, choose the environment and branch (if applicable) that you want to send requests to.
4. Select **Reload Preview** to see the microfrontend with the new values.

To undo the changes back to the original values, open the microfrontends panel and click **Reset to Default**.

## Enable routing debug mode

You can enable [debug headers](/docs/microfrontends/troubleshooting#debug-headers) on microfrontends responses to help [debug issues with

---
title: "Microfrontends"
description: "Learn how to use microfrontends on Vercel to split apart large applications, improve developer experience and make incremen
last_updated: "2026-01-16T02:19:33.482Z"
source: "https://vercel.com/docs/microfrontends"
---

# Microfrontends

Microfrontends allow you to split a single application into smaller, independently deployable units that render as one cohesive applicati

## When to use microfrontends?

They are valuable for:

- **Improved developer velocity**: You can split large applications into smaller units, improving development and build times.
- **Independent teams**: Large organizations can split features across different teams, with each team choosing their technology stack, f
- **Incremental migration**: You can gradually migrate from legacy systems to modern frameworks without rewriting everything at once.

Microfrontends may add additional complexity to your development process. To improve developer velocity, consider alternatives like:

- [Monorepos](/docs/monorepos) with [Turborepo](https://turborepo.com/)
- [Feature flags](/docs/feature-flags)
- Faster compilation with [Turbopack](https://nextjs.org/docs/app/api-reference/turbopack)

## Getting started with microfrontends

- Learn how to set up and configure microfrontends using our [Quickstart](/docs/microfrontends/quickstart) guide
- [Test your microfrontends locally](/docs/microfrontends/local-development) before merging the code to preview and production

To make the most of your microfrontend experience, [install the Vercel Toolbar](/docs/vercel-toolbar/in-production-and-localhost).

## Managing microfrontends

Once you have configured the basic structure of your microfrontends,

- Learn the different ways in which you can [route paths](/docs/microfrontends/path-routing) to different microfrontends as well as avail
- Learn how to [manage your microfrontends](/docs/microfrontends/managing-microfrontends) to add and remove microfrontends, share setting
- Learn how to [optimize navigation's](/docs/microfrontends/managing-microfrontends#optimizing-navigations-between-microfrontends) betwee
- Use the [Vercel Toolbar](/docs/microfrontends/managing-microfrontends/vercel-toolbar) to manage different aspects of microfrontends suc
- Learn how to [troubleshoot](/docs/microfrontends/troubleshooting#troubleshooting) your microfrontends setup or [add unit tests](/docs/m

## Limits and pricing

Users on all plans can use microfrontends support with some limits, while [Pro](/docs/plans/pro-plan) and [Enterprise](/docs/plans/enterp

| | Hobby | Pro / Enterprise |
| --- | --- | --- |
| Included Microfrontends Routing | 50K requests / month | N/A |
| Additional Microfrontends Routing | - | $2 per 1M requests |
| Included Microfrontends Projects | 2 projects | 2 projects |
| Additional Microfrontends Projects | - | $250/project/month |

Microfrontends usage can be viewed in the **Vercel Delivery Network** section of **Usage** tab in the Vercel dashboard.

## More resources

- [Incremental migrations with microfrontends](/kb/guide/incremental-migrations-with-microfrontends)
- [How Vercel adopted microfrontends](https://vercel.com/blog/how-vercel-adopted-microfrontends)

---
title: "Microfrontends path routing"
description: "Route paths on your domain to different microfrontends."
last_updated: "2026-01-16T02:19:33.522Z"
source: "https://vercel.com/docs/microfrontends/path-routing"
---

# Microfrontends path routing

Vercel handles routing to microfrontends directly in Vercel's network infrastructure, simplifying the setup and improving latency. When V

You can also route paths to a different microfrontend based on custom application logic using middleware.

## Add a new path to a microfrontend

To route paths to a new microfrontend, modify your `microfrontends.json` file. In the `routing` section for the project, add the new path

```json {8} filename="microfrontends.json"
{
  "$schema": "https://openapi.vercel.sh/microfrontends.json",
  "applications": {
    "web": {},
    "docs": {
      "routing": [
        {
          "paths": ["/docs/:path*", "/new-path-to-route"]
        }
      ]
    }
  }
}
```

The routing for this new path will take effect when the code is merged and the deployment is live. You can test the routing changes in Pr

Additionally, if you need to revert, you can use [Instant Rollback](/docs/instant-rollback) to rollback the project to a deployment befor

> **⚠️ Warning:** Changes to separate microfrontends are not rolled out in lockstep. If you need
> to modify `microfrontends.json`, make sure that the new application can handle
> the requests before merging the change. Otherwise use
> [flags](#roll-out-routing-changes-safely-with-flags) to control whether the
> path is routed to the microfrontend.

### Supported path expressions

You can use following path expressions in `microfrontends.json`:

- `/path` - Constant path.
- `/:path` - Wildcard that matches a single path segment.
- `/:path/suffix` - Wildcard that matches a single path segment with a constant path at the end.
- `/prefix/:path*` - Path that ends with a wildcard that can match zero or more path segments.
- `/prefix/:path+` - Path that ends with a wildcard that matches one or more path segments.
- `/\\(a\\)` - Path is `/(a)`, special characters in paths are escaped with a backslash.
- `/:path(a|b)` - Path is either `/a` or `/b`.
- `/:path(a|\\(b\\))` - Path is either `/a` or `/(b)`, special characters are escaped with a backslash.
- `/:path((?!a|b).*)` - Path is any single path except `/a` or `/b`.
- `/prefix-:path-suffix` - Path that starts with `/prefix-`, ends with `-suffix`, and contains a single path segment.

The following are not supported:

- Conflicting or overlapping paths: Paths must uniquely map to one microfrontend
- Regular expressions not included above
- Wildcards that can match multiple path segments (`+`, `*`) that do not come at the end of the expression

To assert whether the path expressions will work for your path, use the [`validateRouting` test utility](/docs/microfrontends/troubleshoo

## Asset Prefix

An *asset prefix* is a unique prefix prepended to paths in URLs of static assets, like JavaScript, CSS, or images. This is needed so that

When using `withMicrofrontends`, a default auto-generated asset prefix is automatically added. The default value is an obfuscated hash of

If you would like to use a human readable asset prefix, you can also set the asset prefix that is used in `microfrontends.json`.

```json filename="microfrontends.json"
"your-application": {
  "assetPrefix": "marketing-assets",
  "routing": [...]
}
```

> **⚠️ Warning:** Changing the asset prefix is not guaranteed to be backwards compatible. Make
> sure that the asset prefix that you choose is routed to the correct project in
> production before changing the `assetPrefix` field.

### Next.js

JavaScript and CSS URLs are automatically prefixed with the asset prefix, but content in the `public/` directory needs to be manually mov

## Setting a default route

Some functionality in the Vercel Dashboard, such as screenshots and links to the deployment domain, automatically links to the `/` path. I

To update the default route, visit the **Microfrontends Settings** page.

1. Go to the **Settings** tab for your project
2. Click on the **Microfrontends** tab
3. Search for the **Default Route** setting
4. Enter a new default path (starting with `/`) such as `/docs` and click **Save**

Deployments created after this change will now use the provided path as the default route.

## Routing to externally hosted applications

If a microfrontend is not yet hosted on Vercel, you can [create a new Vercel project](/docs/projects/managing-projects#creating-a-project

## Routing changes safely with flags

> **💡 Note:** This is only compatible with Next.js.

If you want to dynamically control the routing for a path, you can use flags to make sure that the change is safe before enabling the rou

This is compatible with the [Flags SDK](https://flags-sdk.dev) or it can be used with custom feature flag implementations.

> **💡 Note:** If using this with the Flags SDK, make sure to share the same value of the
> `FLAGS_SECRET` environment between all microfrontends in the same group.

- ### Specify a flag name
  In your `microfrontends.json` file, add a name in the `flag` field for the group of paths:
  ```json {8} filename="microfrontends.json"
  {
    "$schema": "https://openapi.vercel.sh/microfrontends.json",
    "applications": {
      "web": {},
      "docs": {
        "routing": [
          {
            "flag": "name-of-feature-flag",
            "paths": ["/flagged-path"]
          }
        ]
      }
    }
  }
  ```
  Instead of being automatically routed to the `docs` microfrontend, requests to `/flagged-path` will now be routed to the default applic

- ### Add microfrontends middleware
  The `@vercel/microfrontends` package uses middleware to route requests to the correct location for flagged paths and based on what micr

  You can add it to your Next.js application with the following code:
  ```ts filename="middleware.ts"
  import type { NextRequest } from 'next/server';
  import { runMicrofrontendsMiddleware } from '@vercel/microfrontends/next/middleware';

  export async function middleware(request: NextRequest) {
    const response = await runMicrofrontendsMiddleware({
      request,
      flagValues: {
        'name-of-feature-flag': async () => { ... },
      }
    });
    if (response) {
      return response;
    }
  }

  // Define routes or paths where this middleware should apply
  export const config = {
    matcher: [
      '/.well-known/vercel/microfrontends/client-config', // For prefetch optimizations for flagged paths
      '/flagged/path',
    ],
  };
  ```
  Your middleware matcher should include `/.well-known/vercel/microfrontends/client-config`. This endpoint is used by the client to know
  > **💡 Note:** Make sure that any flagged paths are also configured in the [middleware
  > matcher](https://nextjs.org/docs/app/building-your-application/routing/middleware#matcher)
  > so that middleware runs for these paths.
  Any function that returns `Promise<boolean>` can be used as the implementation of the flag. This also works directly with [feature flag

  If the flag returns true, the microfrontends middleware will route the path to the microfrontend specified in `microfrontends.json`. If

  We recommend setting up [`validateMiddlewareConfig`](/docs/microfrontends/troubleshooting#validatemiddlewareconfig) and [`validateMiddl

## Microfrontends domain routing

Vercel automatically determines which deployment to route a request to for the microfrontends projects in the same group. This allows dev

Domains that use this microfrontends routing will have an M icon next to the name on the deployment page.

Microfrontends routing for a domain is set when a domain is created or updated, for example when a deployment is built, promoted, or roll

### Custom domain routing

Domains assigned to the [production environment](/docs/deployments/environments#production-environment) will always route to each project
This is the same deployment that would be reached by accessing the project's production domain. If a microfrontends project is [rolled ba

Domains assigned to a [custom environment](/docs/deployments/environments#custom-environments) will route requests to other microfrontend

### Branch URL routing

Automatically generated branch URLs will route to the latest built deployment for the project on the branch. If no deployment exists for

### Deployment URL routing

Automatically generated deployment URLs are fixed to the point in time they were created. Vercel will route requests to other microfronte

If there is no deployment for the commit or branch for the project at that point in time, routing will fallback to the deployment at that

## Identifying microfrontends by path

To identify which microfrontend is responsible for serving a specific path, you can use the [Deployment Summary](/docs/deployments#resour

### Using the Vercel dashboard

1. Go to the **Project** page for the default microfrontend application.
2. Click on the **Deployment** for the production deployment.
3. Open the **[Deployment Summary](/docs/deployments#resources-tab-and-deployment-summary)** for the deployment.
4. Open up the Microfrontends accordion to see all paths that are served to that microfrontend. If viewing the default application, all p

### Using the Vercel Toolbar

1. On any page in the microfrontends group, open up the **[Vercel Toolbar](/docs/vercel-toolbar)**.
2. Open up the **Microfrontends Panel**.
3. Look through the **Directory** of each microfrontend to find the application that serves the path. If no microfrontends match, the pat

---
title: "Getting started with microfrontends"
description: "Learn how to get started with microfrontends on Vercel."
last_updated: "2026-01-16T02:19:33.636Z"
source: "https://vercel.com/docs/microfrontends/quickstart"
---

# Getting started with microfrontends

This quickstart guide will help you set up microfrontends on Vercel. Microfrontends can be used with different frameworks, and separate f

## Prerequisites

- Have at least two [Vercel projects](/docs/projects/overview#creating-a-project) created on Vercel that will be part of the same microfr

## Key concepts

Before diving into implementation, it's helpful to understand these core concepts:

- **Default app**: The main application that manages the `microfrontends.json` configuration file and handles routing decisions. The defa
- **Shared domain**: All microfrontends appear under a single domain, allowing microfrontends to reference relative paths that point to t
- **Path-based routing**: Requests are automatically directed to the appropriate microfrontend based on URL paths.
- **Independent deployments**: Teams can deploy their microfrontends without affecting other parts of the application.

## Set up microfrontends on Vercel

- ### Create a microfrontends group
  1. Navigate to [your Vercel dashboard](/dashboard) and make sure that you have selected your team from the [scope selector](/docs/dashb
  2. Visit the **Settings** tab.
  3. Find the **Microfrontends** tab from the Settings navigation menu.
  4. Click **Create Group** in the upper right corner.
  5. Follow the instructions to add projects to the microfrontends group and choose one of those applications to be the *default applicat
  Creating a microfrontends group and adding projects to that group does not change any behavior for those applications until you deploy

- ### Define `microfrontends.json`
  Once the microfrontends group is created, you can define a `microfrontends.json` file at the root in the default application. This conf

  Production behavior will not be changed until the `microfrontends.json` file is merged and promoted, so you test in the [Preview](/docs

  On the Settings page for the new microfrontends group, click the **Add Config** button to copy the `microfrontends.json` to your code.

  You can also create the configuration manually in code:
  ```json filename="microfrontends.json"
  {
    "$schema": "https://openapi.vercel.sh/microfrontends.json",
    "applications": {
      "web": {
        "development": {
          "fallback": "TODO: a URL in production that should be used for requests to apps not running locally"
        }
      },
      "docs": {
        "routing": [
          {
            "group": "docs",
            "paths": ["/docs/:path*"]
          }
        ]
      }
    }
  }
  ```
  Application names in `microfrontends.json` should match the Vercel project names, see the [microfrontends configuration](/docs/microfro

  See the [path routing](/docs/microfrontends/path-routing) documentation for details on how to configure the routing for your microfront

- ### Install the `@vercel/microfrontends` package
  In the directory of the microfrontend application, install the package using the following command:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i @vercel/microfrontends
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      ```

````
      yarn i @vercel/microfrontends
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i @vercel/microfrontends
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i @vercel/microfrontends
      ```
    </Code>
  </CodeBlock>
  You need to perform this step for every microfrontend application.

- ### Set up microfrontends with your framework
  Once the `microfrontends.json` file has been added, Vercel will be able to start routing microfrontend requests to each microfrontend.
  > For \['nextjs-app', 'nextjs']:
  To handle JavaScript and CSS assets in Next.js, add the `withMicrofrontends`
  wrapper to your `next.config.js` file.
  > For \['nextjs-app', 'nextjs']:
  > For \['nextjs-app', 'nextjs']:
  The `withMicrofrontends` function will automatically add an [asset
  prefix](/docs/microfrontends/path-routing#asset-prefix) to the application so
  that you do not have to worry about that. Next.js applications that use
  [`basePath`](https://nextjs.org/docs/app/api-reference/config/next-config-js/basePath)
  are not supported right now.
  > For \['sveltekit']:
  To handle static assets for [SvelteKit](/docs/frameworks/sveltekit), add the `withMicrofrontends` wrapper around your SvelteKit configu
  > For \['sveltekit']:
  Then, add the microfrontends plugin to your Vite configuration:
  ```ts filename="vite.config.ts" framework=sveltekit
  import { microfrontends } from '@vercel/microfrontends/experimental/vite';

  export default defineConfig({
    plugins: [microfrontends()],
  });
  ```
  ```js filename="vite.config.js" framework=sveltekit
  import { microfrontends } from '@vercel/microfrontends/experimental/vite';

  export default defineConfig({
    plugins: [microfrontends()],
  });
  ```
  > For \['sveltekit']:
  This requires version `1.0.1` of the `@vercel/microfrontends` package or higher.
  > For \['vite']:
  To handle static assets for [Vite](/docs/frameworks/vite), add the following
  plugin to your Vite configuration:
  > For \['vite']:
  The Vite plugin by default will prefix static assets with a unique path prefix. Using a [base path](https://vite.dev/guide/build#public

  The specified `basePath` must then also be listed in the `microfrontends.json` file:
  ```json filename="microfrontends.json" framework=vite
  "applications": {
    "docs": {
      "routing": [
        {
          "paths": ["/my-base-path/:path*"]
        }
      ],
    }
  }
  ```
  Vite support requires version `1.0.1` of the `@vercel/microfrontends` package or higher.
  > For \['other']:
  For other frameworks not listed here, you will need to manually ensure that assets for child applications have a unique path prefix to

  For example, if you choose `/docs-assets` to be the unique asset prefix for the Docs application, you will need to move all JS and CSS
  ```json filename="microfrontends.json" framework=other
  "applications": {
    "docs": {
      "routing": [
        {
          "paths": ["/docs-assets/:path*"]
        }
      ],
    }
  }
  ```
  Any static asset not covered by the framework instructions above, such as images or any file in the `public/` directory, will also need

- ### Run through steps 3 and 4 for all microfrontend applications in the group
  Set up the other microfrontends in the group by running through steps [3](#install-the-@vercel/microfrontends-package) and [4](#set-up-m

- ### Set up the local development proxy
  To provide a seamless local development experience, `@vercel/microfrontends` provides a microfrontends aware local development proxy to

  If you are using [Turborepo](https://turborepo.com), the proxy will automatically run when you [run the development task](/docs/microfr

  If you don't use `turbo`, you can set this up by adding a script to your `package.json` like this:
  ```json {2} filename="package.json"
  "scripts": {
    "proxy": "microfrontends proxy --local-apps my-local-app-name"
  }
  ```
  Next, use the auto-generated port in your `dev` command so that the proxy knows where to route the requests to:
  ```json filename="package.json"
````

```
  "scripts": {
    "dev": "next dev --port $(microfrontends port)"
  }
```
  Once you have your application and the local development proxy running (either via `turbo` or manually), visit the "Microfrontends Prox

- ### Deploy your microfrontends to Vercel
  You can now deploy your code to Vercel. Once live, you can then visit the domain for that deployment and visit any of the paths configu

  In the example above, visiting the `/` page will see the content from the `web` microfrontend while visiting `/docs` will see the conte
  > **💡 Note:** Microfrontends functionality can be tested in
  > [Preview](/docs/deployments/environments#preview-environment-pre-production)
  > before deploying the code to production.

## Next steps

- Learn how to use the `@vercel/microfrontends` package to manage [local development](/docs/microfrontends/local-development).
- For polyrepo setups (separate repositories), see the [polyrepo configuration guide](/docs/microfrontends/local-development#polyrepo-set
- [Route more paths](/docs/microfrontends/path-routing) to your microfrontends.
- To learn about other microfrontends features, visit the [Managing Microfrontends](/docs/microfrontends/managing-microfrontends) documen
- [Set up the Vercel Toolbar](/docs/microfrontends/managing-microfrontends/vercel-toolbar) for access to developer tools to debug and man

Microfrontends changes how paths are routed to your projects. If you encounter any issues, look at the [Testing & Troubleshooting](/docs/r


---
title: "Testing & troubleshooting microfrontends"
description: "Learn about testing, common issues, and how to troubleshoot microfrontends on Vercel."
last_updated: "2026-01-16T02:19:33.503Z"
source: "https://vercel.com/docs/microfrontends/troubleshooting"
---

# Testing & troubleshooting microfrontends

## Testing

The `@vercel/microfrontends` package includes test utilities to help avoid common misconfigurations.

### `validateMiddlewareConfig`

The `validateMiddlewareConfig` test ensures Middleware is configured to work correctly with microfrontends. Passing this test does *not* 

Since Middleware only runs in the default application, you should only run this test on the default application. If it finds a configurat

```ts filename="tests/middleware.test.ts"
/* @jest-environment node */

import { validateMiddlewareConfig } from '@vercel/microfrontends/next/testing';
import { config } from '../middleware';

describe('middleware', () => {
  test('matches microfrontends paths', () => {
    expect(() =>
      validateMiddlewareConfig(config, './microfrontends.json'),
    ).not.toThrow();
  });
});
```

### `validateMiddlewareOnFlaggedPaths`

The `validateMiddlewareOnFlaggedPaths` test checks that Middleware is correctly configured for flagged paths by ensuring that Middleware

```ts filename="tests/middleware.test.ts"
/* @jest-environment node */

import { validateMiddlewareOnFlaggedPaths } from '@vercel/microfrontends/next/testing';
import { middleware } from '../middleware';

// For this test to work, all flags must be enabled before calling
// validateMiddlewareOnFlaggedPaths. There are many ways to do this depending
// on your flag framework, test framework, etc. but this is one way to do it
// with https://flags-sdk.dev/
jest.mock('flags/next', () => ({
  flag: jest.fn().mockReturnValue(jest.fn().mockResolvedValue(true)),
}));

describe('middleware', () => {
  test('rewrites for flagged paths', async () => {
    await expect(
      validateMiddlewareOnFlaggedPaths('./microfrontends.json', middleware),
    ).resolves.not.toThrow();
  });
});
```

### `validateRouting`

The `validateRouting` test validates that the given paths route to the correct microfrontend. You should only add this test to the defaul

```ts filename="tests/microfrontends.test.ts"
import { validateRouting } from '@vercel/microfrontends/next/testing';

describe('microfrontends', () => {
  test('routing', () => {
    expect(() => {
      validateRouting('./microfrontends.json', {
        marketing: ['/', '/products'],
        docs: ['/docs', '/docs/api'],
        dashboard: [
```

```
          '/dashboard',
          { path: '/new-dashboard', flag: 'enable-new-dashboard' },
        ],
      });
    }).not.toThrow();
  });
});
```

The above test confirms that microfrontends routing:

- Routes `/` and `/products` to the `marketing` microfrontend.
- Routes `/docs` and `/docs/api` to the `docs` microfrontend.
- Routes `/dashboard` and `/new-dashboard` (with the `enable-new-dashboard` flag enabled) to the `dashboard` microfrontend.

## Debugging routing

### Debug logs when running locally

See [debug routing](/docs/microfrontends/local-development#debug-routing) for how to enable debug logs to see where and why the local pro

### Debug headers when deployed

Debug headers expose the internal reason for the microfrontend response. You can use these headers to debug issues with routing.

You can enable debug headers in the [Vercel Toolbar](/docs/microfrontends/managing-microfrontends/vercel-toolbar#enable-routing-debug-mod

Requests to your domain will then return additional headers on every response:

- `x-vercel-mfe-app`: The name of the microfrontend project that handled the request.
- `x-vercel-mfe-target-deployment-id`: The ID of the deployment that handled the request.
- `x-vercel-mfe-default-app-deployment-id`: The ID of the default application deployment, the source of the `microfrontends.json` configu
- `x-vercel-mfe-zone-from-middleware`: For flagged paths, the name of the microfrontend that middleware decided should handle the request
- `x-vercel-mfe-matched-path`: The path from `microfrontends.json` that was matched by the routing configuration.
- `x-vercel-mfe-response-reason`: The internal reason for the MFE response.

## Observability

Microfrontends routing information is stored in [Observability](/docs/observability) and can be viewed in the team or project scopes. Cli

## Tracing

Microfrontends routing is captured by Vercel [Session tracing](/docs/tracing/session-tracing). Once you have captured a trace, you can in

You may need to zoom in to the Microfrontends span. The span includes:

- `vercel.mfe.app`: The name of the microfrontend project that handled the request.
- `vercel.mfe.target_deployment_id`: The ID of the deployment that handled the request.
- `vercel.mfe.default_app_deployment_id`: The ID of the default application deployment, the source of the `microfrontends.json` configura
- `vercel.mfe.app_from_middleware`: For flagged paths, the name of the microfrontend that middleware decided should handle the request.
- `vercel.mfe.matched_path`: The path from `microfrontends.json` that was matched by the routing configuration.

## Troubleshooting

The following are common issues you might face with debugging tips:

### Microfrontends aren't working in local development

See [debug routing](/docs/microfrontends/local-development#debug-routing) for how to enable debug logs to see where and why the local pro

### Requests are not routed to the correct microfrontend in production

To validate where requests are being routed to in production, follow these steps:

1. [Verify](/docs/microfrontends/path-routing#identifying-microfrontends-by-path) that the path is covered by the microfrontends routing
2. Inspect the [debug headers](/docs/microfrontends/troubleshooting#debug-headers) or view a [page trace](/docs/microfrontends/troublesho

---
title: "Monorepos FAQ"
description: "Learn the answer to common questions about deploying monorepos on Vercel."
last_updated: "2026-01-16T02:19:33.527Z"
source: "https://vercel.com/docs/monorepos/monorepo-faq"
---

# Monorepos FAQ

## How can I speed up builds?

Whether or not your deployments are queued depends on the amount of
Concurrent Builds you have available. Hobby plans are limited to 1
Concurrent Build, while Pro or Enterprise plans can customize the amount
on the "Billing" page in the team settings.

Learn more about [Concurrent Builds](/docs/deployments/concurrent-builds).

## How can I make my projects available on different paths under the same domain?

After having set up your monorepo as described above, each of the
directories will be a separate Vercel project, and therefore be available
on a separate domain.

If you'd like to host multiple projects under a single domain, you can
create a new project, assign the domain in the project settings, and proxy
requests to the other upstream projects. The proxy can be implemented
using a `vercel.json` file with the [rewrites](/docs/project-configuration#rewrites) property, where each
`source` is the path under the main domain and each `destination` is the
upstream project domain.

## How are projects built after I push?

Pushing a commit to a Git repository that is connected with multiple
Vercel projects will result in multiple deployments being created and
built in parallel for each.

## Can I share source files between projects? Are shared packages supported?

To access source files outside the Root Directory, enable the **Include source files outside of the Root Directory in the Build Step** op

For information on using Yarn workspaces, see [Deploying a Monorepo Using
Yarn Workspaces to Vercel](/kb/guide/deploying-yarn-monorepos-to-vercel).

Vercel projects created after August 27th 2020 23:50 UTC have this option
enabled by default.
If you're using Vercel CLI, at least version 20.1.0 is required.

## How can I use Vercel CLI without Project Linking?

Vercel CLI will accept Environment Variables instead of Project Linking,
which can be useful for deployments from CI providers. For example:

```zsh filename="terminal"
VERCEL_ORG_ID=team_123 VERCEL_PROJECT_ID=prj_456 vercel
```

Learn more about [Vercel CLI for custom workflows](/kb/guide/using-vercel-cli-for-custom-workflows).

## Can I use Turborepo on the Hobby plan?

Yes. Turborepo is available on **all** plans.

## Can I use Nx with environment variables on Vercel?

When using [Nx](https://nx.dev/getting-started/intro) on Vercel with
[environment variables](/docs/environment-variables), you may
encounter an issue where some of your environment variables are not being
assigned the correct value in a specific deployment.

This can happen if the environment variable is not initialized or defined
in that deployment. If that's the case, the system will look for a value
in an existing cache which may or may not be the value you would like to
use. It is a recommended practice to define all environment variables in
each deployment for all monorepos.

With Nx, you also have the ability to prevent the environment variable
from using a cached value. You can do that by using [Runtime Hash
Inputs](https://nx.dev/using-nx/caching#runtime-hash-inputs). For
example, if you have an environment variable `MY_VERCEL_ENV` in your project, you will add the
following line to your `nx.json` configuration
file:

```json filename="nx.json"
"runtimeCacheInputs": ["echo $MY_VERCEL_ENV"]
```

---
title: "Deploying Nx to Vercel"
description: "Nx is an extensible build system with support for monorepos, integrations, and Remote Caching on Vercel. Learn how to deplo
last_updated: "2026-01-16T02:19:33.537Z"
source: "https://vercel.com/docs/monorepos/nx"
---

# Deploying Nx to Vercel

Nx is an extensible build system with support for monorepos, integrations, and Remote Caching on Vercel.

Read the [Intro to Nx](https://nx.dev/getting-started/intro) docs to learn about the benefits of using Nx to manage your monorepos.

## Deploy Nx to Vercel

- ### Ensure your Nx project is configured correctly
  If you haven't already connected your monorepo to Nx, you can follow the [Getting Started](https://nx.dev/recipe/adding-to-monorepo) on

  To ensure the best experience using Nx with Vercel, the following versions and settings are recommended:
  - Use `nx` version `14.6.2` or later
  - Use `nx-cloud` version `14.6.0` or later
  There are also additional settings if you are [using Remote Caching](/docs/monorepos/nx#setup-remote-caching-for-nx-on-vercel)

  > **💡 Note:** All Nx starters and examples are preconfigured with these settings.

- ### Import your project
  [Create a new Project](/docs/projects/overview#creating-a-project) on the Vercel dashboard and [import](/docs/getting-started-with-verc

  Vercel handles all aspects of configuring your monorepo, including setting [build commands](/docs/deployments/configure-a-build#build-c

- ### Next steps
  Your Nx monorepo is now configured and ready to be used with Vercel!

  You can now [setup Remote Caching for Nx on Vercel](#setup-remote-caching-for-nx-on-vercel) or configure additional deployment options,

## Using `nx-ignore`

`nx-ignore` provides a way for you to tell Vercel if a build should continue or not. For more details and information on how to use `nx-i

## Setup Remote Caching for Nx on Vercel

Before using remote caching with Nx, do one of the following:

- Ensure the `NX_CACHE_DIRECTORY=/tmp/nx-cache` is set

**or**

- Set the `cacheDirectory` option to `/tmp/nx-cache` at `tasksRunnerOptions.{runner}.options` in your `nx.json`. For example:

```json filename="nx.json"
"tasksRunnerOptions": {
  "default": {
    "runner": "nx/tasks-runners/default",
    "options": {
      "cacheDirectory": "/tmp/nx-cache"
    }
  }
}
```

To configure Remote Caching for your Nx project on Vercel, use the [`@vercel/remote-nx`](https://github.com/vercel/remote-cache/tree/main

- ### Install the `@vercel/remote-nx` plugin
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i @vercel/remote-nx
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i @vercel/remote-nx
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i @vercel/remote-nx
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i @vercel/remote-nx
      ```
    </Code>
  </CodeBlock>

- ### Configure the `@vercel/remote-nx` runner
  In your `nx.json` file you will find a `tasksRunnerOptions` field. Update this field so that it's using the installed `@vercel/remote-n
  ```json filename="nx.json"
  {
    "tasksRunnerOptions": {
      "default": {
        "runner": "@vercel/remote-nx",
        "options": {
          "cacheableOperations": ["build", "test", "lint", "e2e"],
          "token": "<token>",
          "teamId": "<teamId>"
        }
      }
    }
  }
  ```
  You can specify your `token` and `teamId` in your nx.json or set them as environment variables.

  | Parameter | Description | Environment V
  | --- | --- | ---
  | Vercel Access Token | Vercel access token with access to the provided team | `NX_VERCEL_RE
  | Vercel [Team ID](/docs/accounts#find-your-team-id) (optional) | The Vercel Team ID that should share the Remote Cache | `NX_VERCEL_RE
  > **💡 Note:** When deploying on Vercel, these variables will be automatically set for you.

- ### Clear cache and run
  Clear your local cache and rebuild your project.
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```

    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>


---
title: "Using Monorepos"
description: "Vercel provides support for monorepos. Learn how to deploy a monorepo here."
last_updated: "2026-01-16T02:19:33.561Z"
source: "https://vercel.com/docs/monorepos"
---

# Using Monorepos

Monorepos allow you to manage multiple projects in a single directory. They are a great way to organize your projects and make them easie

## Deploy a template monorepo

Get started with monorepos on Vercel in a few minutes by using one of our monorepo quickstart templates.

## Add a monorepo through the Vercel Dashboard

1. Go to the [Vercel Dashboard](https://vercel.com/dashboard) and ensure your team is selected from the [scope selector](/docs/dashboard-
2. Select the **Add New…** button, and then choose **Project** from the list. You'll create a new [project](/docs/projects/overview) for
3. From the **Import Git Repository** section, select the **Import** button next to the repository you want to import.
4. Before you deploy, you'll need to specify the directory within your monorepo that you want to deploy. Click the **Edit** button next t

5) Configure any necessary settings and click the **Deploy** button to deploy that project.
6) Repeat steps 2-5 to [import each directory](/docs/git#deploying-a-git-repository) from your monorepo that you want to deploy.

Once you've created a separate project for each of the directories within your Git repository, every commit will issue a deployment for a

The number of Vercel Projects connected with the same Git repository is [limited depending on your plan](/docs/limits#general-limits).

## Add a monorepo through Vercel CLI

> **💡 Note:** You should use [Vercel CLI 20.1.0](/docs/cli#updating-vercel-cli) or newer.

1. Ensure you're in the root directory of your monorepo. Vercel CLI should not be invoked from the subdirectory.
2. Run `vercel link` to link multiple Vercel projects at once. To learn more, see the [CLI documentation](/docs/cli/link#repo-alpha):
   ```bash filename="Terminal"
   vercel link --repo
   ```
3. Once linked, subsequent commands such as `vercel dev` will use the selected Vercel Project. To switch to a different Project in the sa

Alternatively, you can use  to create multiple copies of your monorepo in different directories and link each one to a different Vercel P

> **💡 Note:** See this [example](https://github.com/vercel-support/yarn-ws-monorepo) of a
> monorepo with Yarn Workspaces.

## When does a monorepo build occur?

By default, pushing a commit to your monorepo will create a deployment for each of the connected Vercel projects. However, you can choose

- [Skip unaffected projects](#skipping-unaffected-projects) by only building projects whose files have changed.
- [Ignore the build step](#ignoring-the-build-step) for projects whose files have not changed.

### Skipping unaffected projects

A project in a monorepo is considered to be changed if any of the following conditions are true:

1. The project source code has changed
2. Any of the project's internal dependencies have changed.
3. A change to a package manager lockfile has occurred, that *only* impacts the dependencies of the project.

Vercel automatically skips builds for projects in a monorepo that are unchanged by the commit.

This setting does **not** occupy [concurrent build slots](/docs/deployments/concurrent-builds), unlike the [Ignored Build Step](/docs/pro

#### Requirements

- This feature is only available for projects connected to GitHub repositories.
- The monorepo must be using npm, yarn, or pnpm workspaces, following JavaScript ecosystem conventions. Packages in the workspace must be
  - Changes that are not a part of the workspace definition will be considered global changes and deploy all applications in the reposito
  - We automatically detect your package manager using the lockfile at the repository root. You can also explicitly set a package manager
- All packages within the workspace must have a **unique** `name` field in their `package.json` file.
- Dependencies between packages in the monorepo must be explicitly stated in each package's `package.json` file. This is necessary to det
  - For example, an end-to-end tests package (`package-e2e`) tests must depend on the package it tests (`package-core`) in the `package.j

#### Disable the skipping unaffected projects feature

To disable this behavior, [visit the project's Root Directory settings](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Fsettings%

1. From the [Dashboard](https://vercel.com/dashboard), select the project you want to configure and navigate to the **Settings** tab.
2. Go to the Build and Deployment page of the project's Settings.
3. Scroll down to **Root Directory**
4. Toggle the **Skip deployment** switch to **Disabled**.
5. Click **Save** to apply the changes.

### Ignoring the build step

If you want to cancel the Build Step for projects if their files didn't change, you can do so with the [Ignored Build Step](/docs/project

If you have created a script to ignore the build step, you can skip the [the
script](/kb/guide/how-do-i-use-the-ignored-build-step-field-on-vercel) when
redeploying or promoting your app to production. This can be done through the
dashboard when you click on the **Redeploy** button, and unchecking the **Use
project's Ignore Build Step** checkbox.

## How to link projects together in a monorepo

When working in a monorepo with multiple applications—such as a frontend and a backend—it can be challenging to manage the connection str
Traditionally, referencing one project from another requires manually setting URLs or environment variables for each deployment, in *ever

With Related Projects, this process is streamlined, enabling teams to:

- Verify changes in pre-production environments without manually updating URLs or environment variables.
- Eliminate misconfigurations when referencing internal services across multiple deployments, and environments.

For example, if your monorepo contains:

1. A frontend project that fetches data from an API
2. A backend API project that serves the data

Related Projects can ensure that each preview deployment of the frontend automatically references the corresponding preview deployment of
changes that span both projects.

### Requirements

- A maximum of 3 projects can be linked together
- Only supports projects within the same repository
- CLI deployments are not supported

### Getting started

- ### Define Related Projects
  Specify the projects your app needs to reference in a `vercel.json` configuration file at the root of the app.
  While every app in your monorepo can list related projects in their own `vercel.json`, you can only specify up to three related project
  ```json filename="apps/frontend/vercel.json"
  {
    "relatedProjects": ["prj_123"]
  }
  ```
  This will make the preview, and production hosts of `prj_123` available as an environment variable in the deployment of the `frontend`
  > **💡 Note:** You can [find your project
  > ID](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Fsettings%23project-id\&title=Find+your+Vercel+project+ID)
  > in the project **Settings** page in the Vercel dashboard.

- ### Retrieve Related Project Information
  The next deployment will have the `VERCEL_RELATED_PROJECTS` environment variable set containing the urls of the related projects for us
  > **💡 Note:** View the data provided for each project in the
  > [`@vercel/related-projects`](https://github.com/vercel/vercel/blob/main/packages/related-projects/src/types.ts#L9-L58)
  > package.
  For easy access to this information, you can use the [`@vercel/related-projects`](https://github.com/vercel/vercel/tree/main/packages/r
  ```bash filename="Terminal" package-manager="npm"
  npm i @vercel/related-projects
  ```
  ```bash filename="Terminal" package-manager="bun"
  bun add @vercel/related-projects
  ```
  ```bash filename="Terminal" package-manager="yarn"
  yarn add @vercel/related-projects
  ```
  ```bash filename="Terminal" package-manager="pnpm"
  pnpm add @vercel/related-projects
  ```
  1. Easily reference hosts of related projects
  ```ts
  import { withRelatedProject } from '@vercel/related-projects';

  const apiHost = withRelatedProject({
    projectName: 'my-api-project',
    /**
     * Specify a default host that will be used for my-api-project if the related project
     * data cannot be parsed or is missing.
     */
    defaultHost: process.env.API_HOST,
  });
  ```
  2. Retrieve just the related project data:
  ```ts filename="index.ts"
  import {
    relatedProjects,
    type VercelRelatedProject,
  } from '@vercel/related-projects';

  // fully typed project data
  const projects: VercelRelatedProject[] = relatedProjects();
  ```

---
title: "Remote Caching"
description: "Vercel Remote Cache allows you to share build outputs and artifacts across distributed teams."
last_updated: "2026-01-16T02:19:33.580Z"
source: "https://vercel.com/docs/monorepos/remote-caching"
---

# Remote Caching

Remote Caching saves you time by ensuring you never repeat the same task twice, by automatically sharing a cache across your entire Verce

When a team is working on the same PR, Remote Caching identifies the necessary artifacts (such as build and log outputs) and recycles ther

This speeds up your workflow by avoiding the need to constantly re-compile, re-test, or re-execute your code if it is unchanged.

## Vercel Remote Cache

The first tool to leverage Vercel Remote Cache is [Turborepo](https://turborepo.com), a high-performance build system for JavaScript and

Turborepo caches the output of any previously run command such as testing and building, so it can replay the cached results instantly ins

With Remote Caching, you can share the Turborepo cache across your entire team and CI, resulting in even faster builds and days saved.

> **💡 Note:** Remote Caching is a powerful feature of Turborepo, but with great power comes
> great responsibility. Make sure you are caching correctly first and
> double-check the [handling of environment
> variables](/docs/monorepos/turborepo#step-0:-cache-environment-variables). You
> should also remember that Turborepo treats logs as artifacts, so be aware of
> what you are printing to the console.

The Vercel Remote Cache can also be used with any build tool by integrating with the [Remote Cache SDK](https://github.com/vercel/remote-
This provides plugins and examples for popular monorepo build tools like [Nx](https://github.com/vercel/remote-cache/tree/main/packages/r

## Get started

For this guide, your monorepo should be using [Turborepo](/docs/monorepos/turborepo). Alternatively, use `npx create-turbo` to set up a s

- ### Enable and disable Remote Caching for your team
  Remote Caching is **automatically enabled on Vercel** for organizations with Turborepo enabled on their monorepo.

  As an Owner, you can enable or disable Remote Caching from your team settings.
  1. From the [Vercel Dashboard](/dashboard), select your team from the [scope selector](/docs/dashboard-features#scope-selector).
  2. Select the **Settings** tab and go to the **Billing** section
  3. From the **Remote Caching** section, toggle the switch to enable or disable the feature.

- ### Authenticate with Vercel
  Once your Vercel project is using Turborepo, authenticate the Turborepo CLI with your Vercel account:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>
  If you are connecting to an SSO-enabled Vercel team, you should provide your Team slug as an argument:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>

- ### Link to the remote cache
  **To enable Remote Caching and connect to the Vercel Remote Cache**, every member of that team that wants use Remote Caching should run
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```

    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>
  You will be prompted to enable Remote Caching for the current repo. Enter `Y` for yes to enable Remote Caching.

  Next, select the team scope you'd like to connect to. Selecting the scope tells Vercel who the cache should be shared with and allows f
  > **⚠️ Warning:** If you run these commands but the owner has [disabled Remote
  > Caching](#enabling-and-disabling-remote-caching-for-your-team) for your team,
  > Turborepo will present you with an error message: "Please contact your account
  > owner to enable Remote Caching on Vercel."

- ### Unlink the remote cache
  To disable Remote Caching and unlink the current directory from the Vercel Remote Cache, run:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>
  This is run on a per-developer basis, so each developer that wants to unlink the remote cache must run this command locally.

- ### Test the cache
  Once your project has the remote cache linked, run `turbo run build` to see the caching in action. Turborepo caches the filesystem outp

  Now try making a change in any file and running `turbo run build` again.
  The builds speed will have dramatically improved, because Turborepo will only rebuild the changed packages.

## Use Remote Caching during Vercel Build

When you run `turbo` commands during a Vercel Build, Remote Caching will be automatically enabled. No additional configuration is require

## Use Remote Caching from external CI/CD

To use Vercel Remote Caching with Turborepo from an external CI/CD system, you can set the following environment variables in your CI/CD

- `TURBO_TOKEN`: A [Vercel Access Token](/docs/rest-api#authentication)
- `TURBO_TEAM`: The slug of the Vercel team to share the artifacts with

When these environment variables are set, Turborepo will use Vercel Remote Caching to store task artifacts.

## Usage

Vercel Remote Cache is free for all plans, subject to fair use guidelines.

| **Plan**   | **Fair use upload limit** | **Fair use artifacts request limit** |
| ---------- | ------------------------- | ------------------------------------ |
| Hobby      | 100GB / month             | 100 / minute                         |
| Pro        | 1TB / month               | 10000 / minute                       |
| Enterprise | 4TB / month               | 10000 / minute                       |

### Artifacts

| Metric                                                                    | Description
| ------------------------------------------------------------------------- | -----------------------------------------------------------
| [**Number of Remote Cache Artifacts**](#number-of-remote-cache-artifacts) | The number of uploaded and downloaded artifacts using the R
| **Total Size of Remote Cache Artifacts**                                  | The size of uploaded and downloaded artifacts using the Rem
| [**Time Saved**](#time-saved)                                             | The time saved by using artifacts cached on the Vercel Remo

Artifacts are blobs of data or files that are uploaded and downloaded using the [Vercel Remote Cache API](/docs/monorepos/remote-caching)

Vercel automatically expires uploaded artifacts after 7 days to avoid unbounded cache growth.

#### Time Saved

Artifacts get annotated with a task duration, which is the time required for the task to run and generate the artifact. The time saved is

- **Remote Cache**: The time saved by using artifacts cached on the Vercel Remote Cache API
- **Local Cache**: The time saved by using artifacts cached on your local filesystem cache

#### Number of Remote Cache Artifacts

When your team enables [Vercel Remote Cache](/docs/monorepos/remote-caching#enable-and-disable-remote-caching-for-your-team), Vercel will

For other monorepo implementations like [Nx](/docs/monorepos/nx), you need to manually configure your project using the [Remote Cache SDK

You are not charged based on the number of artifacts, but rather the size in GB downloaded.

#### Optimizing total size of Remote Cache artifacts

Caching only the files needed for the task will improve cache restoration performance.

For example, the `.next` folder contains your build artifacts. You can avoid caching the `.next/cache` folder since it is only used for d

## Billing information

Vercel Remote Cache is free for all plans, subject to [fair use guidelines](#usage).

### Pro and Enterprise

Remote Caching can only be enabled by [team owners](/docs/rbac/access-roles#owner-role). When Remote Caching is enabled, anyone on your t

## More resources

- [Use this SDK to manage Remote Cache Artifacts](https://github.com/vercel/remote-cache)

---
title: "Deploying Turborepo to Vercel"
description: "Learn about Turborepo, a build system for monorepos that allows you to have faster incremental builds, content-aware hashin
last_updated: "2026-01-16T02:19:33.614Z"
source: "https://vercel.com/docs/monorepos/turborepo"
---

# Deploying Turborepo to Vercel

Turborepo is a high-performance build system for JavaScript and TypeScript codebases with:

- Fast incremental builds
- Content-aware hashing, meaning only the files you changed will be rebuilt
- [Remote Caching](/docs/monorepos/remote-caching) for sharing build caches with your team and CI/CD pipelines

And more. Read the [Why Turborepo](https://turborepo.com/docs#why-turborepo) docs to learn about the benefits of using Turborepo to manag

## Deploy Turborepo to Vercel

Follow the steps below to deploy your Turborepo to Vercel:

- ### Handling environment variables
  It's important to ensure you are managing environment variables (and files outside of packages and apps) correctly.

  If your project has environment variables, you'll need to create a list of them in your `turbo.json` so Turborepo knows to use differen

  Frameworks like Next.js inline build-time environment variables (e.g. `NEXT_PUBLIC_XXX`) in bundled outputs as strings. Turborepo will

  You can control Turborepo's cache behavior (hashing) based on the values of both environment variables and the contents of files in a f
  > **💡 Note:** `env` and `globalEnv` key support is available in Turborepo version 1.5 or
  > later. You should update your Turborepo version if you're using an older
  > version.
  The following example shows a Turborepo configuration, that handles these suggestions:
  ```json filename="turbo.json"
  {
    "$schema": "https://turborepo.com/schema.json",
    "pipeline": {
      "build": {
        "dependsOn": ["^build"],
        "env": [
          // env vars will impact hashes of all "build" tasks
          "SOME_ENV_VAR"
        ],
        "outputs": ["dist/**"]
      },
      "web#build": {
        // override settings for the "build" task for the "web" app
        "dependsOn": ["^build"],
        "env": ["SOME_OTHER_ENV_VAR"],
        "outputs": [".next/**", "!.next/cache/**"]
      }
    },
    "globalEnv": [
      "GITHUB_TOKEN" // env var that will impact the hashes of all tasks,
    ],
    "globalDependencies": [
      "tsconfig.json" // file contents will impact the hashes of all tasks,
    ]
  }
  ```
  > **💡 Note:** In most monorepos, environment variables are usually used in applications
  > rather than in shared packages. To get higher cache hit rates, you should only
  > include environment variables in the app-specific tasks where they are used or
  > inlined.
  Once you've declared your environment variables, commit and push any changes you've made. When you update or add new inlined build-time

- ### Import your Turborepo to Vercel
  > **💡 Note:** If you haven't already connected your monorepo to Turborepo, you can follow
  > the [quickstart](https://turborepo.com/docs) on the Turborepo docs to do so.
  [Create a new Project](/new) on the Vercel dashboard and [import](/docs/getting-started-with-vercel/import) your Turborepo project.

  Vercel handles all aspects of configuring your monorepo, including setting [build commands](/docs/deployments/configure-a-build#build-c

  The table below reflects the values that Vercel will set if you'd like to set them manually in your Dashboard or in the `vercel.json` o

  | **Field**          | **Command**                                                                                 |
  | ------------------ | ------------------------------------------------------------------------------------------- |
  | Framework Preset   | [One of 35+ framework presets](/docs/frameworks/more-frameworks)                            |
  | Build Command      | `turbo run build` (requires version >=1.8) or `cd ../.. && turbo run build --filter=web` |
  | Output Directory   | Framework default                                                                           |
  | Install Command    | Automatically detected by Vercel                                                            |
  | Root Directory     | App location in repository (e.g. `apps/web`)                                                |
  | Ignored Build Step | `npx turbo-ignore --fallback=HEAD^1`                                                        |

## Using global `turbo`

Turborepo is also available globally when you deploy on Vercel, which means that you do **not** have to add `turbo` as a dependency in yo

Thanks to [automatic workspace scoping](https://turborepo.com/blog/turbo-1-8-0#automatic-workspace-scoping) and [globally installed turbo

```bash
turbo build
```

The appropriate [filter](https://turborepo.com/docs/core-concepts/monorepos/filtering) will be automatically inferred based on the config

> **💡 Note:** To override this behavior and use a specific version of Turborepo, install the
> desired version of `turbo` in your project. [Learn
> more](https://turborepo.com/blog/turbo-1-7-0#global-turbo)

## Ignoring unchanged builds

You likely don't need to build a preview for every application in your monorepo on every commit. To ensure that only applications that ha

## Setup Remote Caching for Turborepo on Vercel

You can optionally choose to connect your Turborepo to the [Vercel Remote Cache](/docs/monorepos/remote-caching) from your local machine,

You do not need to host your project on Vercel to use Vercel Remote Caching. For more information, see the [Remote Caching](/docs/monorepo

- ### Link your project to the Vercel Remote Cache
  First, authenticate with the Turborepo CLI **from the root of your monorepo**:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>
  Then, use [`turbo link`](https://turborepo.com/docs/reference/command-line-reference#turbo-link) to link your Turborepo to your [remote
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>
  Next, `cd` into each project in your Turborepo and run `vercel link` to link each directory within the monorepo to your Vercel Project.

  As a Team owner, you can also [enable caching within the Vercel Dashboard](/docs/monorepos/remote-caching#enable-and-disable-remote-cac

- ### Test the caching
  Your project now has the Remote Cache linked. Run `turbo run build` to see the caching in action. Turborepo caches the filesystem outpu

  Now try making a change in a file and running `turbo run build` again.
  The builds speed will have dramatically improved. This is because Turborepo will only rebuild the changed files.

  To see information about the [Remote Cache usage](/docs/limits/usage#artifacts), go to the **Artifacts** section of the **Usage** tab.

## Troubleshooting

### Build outputs cannot be found on cache hit

For Vercel to deploy your application, the outputs need to be present for your [Framework Preset](/docs/deployments/configure-a-build#fra

- Confirm that your outputs match [the expected Output Directory for your Framework Preset](/docs/monorepos/turborepo#import-your-turbore
- Make sure the application outputs defined in the `outputs` key of your `turbo.json` for your build task are aligned with your Framework

```json filename="turbo.json"
{
  "$schema": "https://turborepo.com/schema.json",
  "pipeline": {
    "build": {
      "dependsOn": ["^build"],
```

```
      "outputs": [
        // Next.js
        ".next/**", "!.next/cache/**"
        // SvelteKit
        ".svelte-kit/**", ".vercel/**",
        // Build Output API
        ".vercel/output/**"
        // Other frameworks
        ".nuxt/**", "dist/**" "other-output-directory/**"
      ]
    }
  }
}
```

Visit [the Turborepo documentation](https://turborepo.com/docs/reference/configuration#outputs) to learn more about the `outputs` key.

### Unexpected cache misses

When using Turborepo on Vercel, all information used by `turbo` during the build process is automatically collected to help debug cache m

> **💡 Note:** Turborepo Run Summary is only available in Turborepo version `1.9` or later.
> To upgrade, use `npx @turbo/codemod upgrade`.

To view the Turborepo Run Summary for a deployment, use the following steps:

1. From your [dashboard](/dashboard), select your project and go to the **Deployments** tab.
2. Select a **Deployment** from the list to view the deployment details
3. Select the **Run Summary** button to the right of the **Building** section, under the **Deployment Status** heading:

This opens a view containing a review of the build, including:

- All [tasks](https://turborepo.com/docs/core-concepts/caching) that were executed as part of the build
- The execution time and cache status for each task
- All data that `turbo` used to construct the cache key (the [task hash](https://turborepo.com/docs/core-concepts/caching#hashing))

> **💡 Note:** If a previous deployment from the same branch is available, the difference
> between the cache inputs for the current and previous build will be
> automatically displayed, highlighting the specific changes that caused the
> cache miss.

This information can be helpful in identifiying exactly why a cache miss occurred, and can be used to determine if a cache miss is due to
project, or a change in the environment.

To change the comparison, select a different deployment from the dropdown, or search for a deployment ID. The summary data can also be do

> **💡 Note:** Environment variable values are encrypted when displayed in Turborepo Run
> Summary, and can only be compared with summary files generated locally when
> viewed by a team member with access to the projects environment variables.
> [Learn more](/docs/rbac/access-roles/team-level-roles)

## Limitations

Building a Next.js application that is using [Skew Protection](/docs/skew-protection) always results in a Turborepo cache miss. This occu

If you are using a version of Turborepo below 2.4.1, you may encounter issues with Skew Protection related to missing assets in productio

---
title: "Domain management for multi-tenant"
description: "Manage custom domains, wildcard subdomains, and SSL certificates programmatically for multi-tenant applications using Verce
last_updated: "2026-01-16T02:19:33.589Z"
source: "https://vercel.com/docs/multi-tenant/domain-management"
---

# Domain management for multi-tenant

Learn how to programmatically manage domains for your multi-tenant application using Vercel for Platforms.

## Using wildcard domains

If you plan on offering subdomains like `*.acme.com`, add a **wildcard domain** to your Vercel project. This requires using [Vercel's nam

1. Point your domain to Vercel's nameservers (`ns1.vercel-dns.com` and `ns2.vercel-dns.com`).
2. In your Vercel project settings, add the apex domain (e.g., `acme.com`).
3. Add a wildcard domain: `.acme.com`.

Now, any `tenant.acme.com` you create—whether it's `tenant1.acme.com` or `docs.tenant1.acme.com`—automatically resolves to your Vercel de

## Offering custom domains

You can also give tenants the option to bring their own domain. In that case, you'll want your code to:

1. Provision and assign the tenant's domain to your Vercel project.
2. Verify the domain (to ensure the tenant truly owns it).
3. Automatically generate an SSL certificate.

## Adding a domain programmatically

You can add a new domain through the [Vercel SDK](https://vercel.com/docs/sdk). For example:

```ts
import { VercelCore as Vercel } from '@vercel/sdk/core.js';
import { projectsAddProjectDomain } from '@vercel/sdk/funcs/projectsAddProjectDomain.js';

const vercel = new Vercel({
  bearerToken: process.env.VERCEL_TOKEN,
});

// The 'idOrName' is your project name in Vercel, for example: 'multi-tenant-app'
```

```
await projectsAddProjectDomain(vercel, {
  idOrName: 'my-multi-tenant-app',
  teamId: 'team_1234',
  requestBody: {
    // The tenant's custom domain
    name: 'customacmesite.com',
  },
});
```

Once the domain is added, Vercel attempts to issue an SSL certificate automatically.

## Verifying domain ownership

If the domain is already in use on Vercel, the user needs to set a TXT record to prove ownership of it.

You can check the verification status and trigger manual verification:

```ts
import { VercelCore as Vercel } from '@vercel/sdk/core.js';
import { projectsGetProjectDomain } from '@vercel/sdk/funcs/projectsGetProjectDomain.js';
import { projectsVerifyProjectDomain } from '@vercel/sdk/funcs/projectsVerifyProjectDomain.js';

const vercel = new Vercel({
  bearerToken: process.env.VERCEL_TOKEN,
});

const domain = 'customacmesite.com';

const [domainResponse, verifyResponse] = await Promise.all([
  projectsGetProjectDomain(vercel, {
    idOrName: 'my-multi-tenant-app',
    teamId: 'team_1234',
    domain,
  }),
  projectsVerifyProjectDomain(vercel, {
    idOrName: 'my-multi-tenant-app',
    teamId: 'team_1234',
    domain,
  }),
]);

const { value: result } = verifyResponse;

if (!result?.verified) {
  console.log(`Domain verification required for ${domain}.`);
  // You can prompt the tenant to add a TXT record or switch nameservers.
}
```

## Handling redirects and apex domains

### Redirecting between apex and "www"

Some tenants might want `www.customacmesite.com` to redirect automatically to their apex domain `customacmesite.com`, or the other way ar

1. Add both `customacmesite.com` and `www.customacmesite.com` to your Vercel project.
2. Configure a redirect for `www.customacmesite.com` to the apex domain by setting `redirect: customacmesite.com` through the API or your

This ensures a consistent user experience and prevents issues with duplicate content.

### Avoiding duplicate content across subdomains

If you offer both `tenant.acme.com` and `customacmesite.com` for the same tenant, you may want to redirect the subdomain to the custom do

## Deleting or removing domains

If a tenant cancels or no longer needs their custom domain, you can remove it from your Vercel account using the SDK:

```ts
import { VercelCore as Vercel } from '@vercel/sdk/core.js';
import { projectsRemoveProjectDomain } from '@vercel/sdk/funcs/projectsRemoveProjectDomain.js';
import { domainsDeleteDomain } from '@vercel/sdk/funcs/domainsDeleteDomain.js';

const vercel = new Vercel({
  bearerToken: process.env.VERCEL_TOKEN,
});

await Promise.all([
  projectsRemoveProjectDomain(vercel, {
    idOrName: 'my-multi-tenant-app',
    teamId: 'team_1234',
    domain: 'customacmesite.com',
  }),
  domainsDeleteDomain(vercel, {
    domain: 'customacmesite.com',
  }),
]);
```

The first call disassociates the domain from your project, and the second removes it from your account entirely.

## Troubleshooting common issues

Here are a few common issues you might run into and how to solve them:

**DNS propagation delays**

After pointing your nameservers to Vercel or adding CNAME records, changes can take 24–48 hours to propagate. Use [WhatsMyDNS](https://ww

**Forgetting to verify domain ownership**

If you add a tenant's domain but never verify it (e.g., by adding a `TXT` record or using Vercel nameservers), SSL certificates won't be

**Wildcard domain requires Vercel nameservers**

If you try to add `.acme.com` without pointing to `ns1.vercel-dns.com` and `ns2.vercel-dns.com`, wildcard SSL won't work. Make sure the a

**Exceeding subdomain length for preview URLs**

Each DNS label has a [63-character limit](/kb/guide/why-is-my-vercel-deployment-url-being-shortened#rfc-1035). If you have a very long br

**Duplicate content SEO issues**

If the same site is served from both subdomain and custom domain, consider using [canonical](https://nextjs.org/docs/app/api-reference/fu

**Misspelled domain**

A small typo can block domain verification or routing, so double-check your domain spelling.

---
title: "Multi-tenant Limits"
description: "Understand the limits and features available for Vercel for Platforms."
last_updated: "2026-01-16T02:19:33.657Z"
source: "https://vercel.com/docs/multi-tenant/limits"
---

# Multi-tenant Limits

This page provides an overview of the limits and feature availability for Vercel for Platforms across different plan types.

## Feature availability

| Feature | Hobby
| --- | --- | --- | --- | ---
| Compute | Included | Included | Included |
| Firewall | Included | Included | Included |
| WAF (Web Application Firewall) | Included | Included | Included |
| Custom Domains | 50 | Unlimited* | Unlimited* |
| Multi-tenant preview URLs | Enterprise only | Enterprise only | Enterprise only |
| Custom SSL certificates | Enterprise only | Enterprise only | Enterprise only |

- To prevent abuse, Vercel implements soft limits of 100,000 domains per project for the Pro plan and 1,000,000 domains for the Enterpris

### Wildcard domains

- **All plans**: Support for wildcard domains (e.g., `*.acme.com`)
- **Requirement**: Must use [Vercel's nameservers](https://vercel.com/docs/projects/domains/working-with-nameservers) for wildcard SSL ce

### Custom domains

- **All plans**: Unlimited custom domains per project
- **SSL certificates**: Automatically issued for all verified domains
- **Verification**: Required for domains already in use on Vercel

## Multi-tenant preview URLs

Multi-tenant preview URLs are available exclusively for **Enterprise** customers. This feature allows you to:

- Generate unique preview URLs for each tenant during development
- Test changes for specific tenants before deploying to production
- Use dynamic subdomains like `tenant1---project-name-git-branch.yourdomain.dev`

To enable this feature, Enterprise customers should contact their Customer Success Manager (CSM) or Account Executive (AE).

## Custom SSL certificates

Custom SSL certificates are available exclusively for **Enterprise** customers. This feature allows you to:

- Upload your own SSL certificates for tenant domains
- Maintain complete control over certificate management
- Meet specific compliance or security requirements

Learn more about [custom SSL certificates](https://vercel.com/docs/domains/custom-SSL-certificate).

## Rate limits

Domain management operations through the Vercel API are subject to standard [API rate limits](https://vercel.com/docs/rest-api#rate-limit

- **Domain addition**: 100 requests per hour per team
- **Domain verification**: 50 requests per hour per team
- **Domain removal**: 100 requests per hour per team

## DNS propagation

After configuring domains or nameservers, DNS typically takes 24-48 hours to propagate globally. Use tools like [WhatsMyDNS](https://www.

## Subdomain length limits

Each DNS label has a [63-character limit](/kb/guide/why-is-my-vercel-deployment-url-being-shortened#rfc-1035). For preview URLs with long

---
title: "Vercel for Platforms"
description: "Build multi-tenant applications that serve multiple customers from a single codebase with custom domains and subdomains."
last_updated: "2026-01-16T02:19:33.665Z"
source: "https://vercel.com/docs/multi-tenant"
---

# Vercel for Platforms

A **multi-tenant application** serves multiple customers (tenants) from a single codebase.

Each tenant gets its own domain or subdomain, but you only have one Next.js (or similar) deployment running on Vercel. This approach simp

Get started with our [detailed docs](/platforms/docs), [multi-tenant Next.js example](https://vercel.com/templates/next.js/platforms-star

## Why build multi-tenant apps?

Some popular multi-tenant apps on Vercel include:

- **Content platforms**: [Hashnode](https://townhall.hashnode.com/powerful-and-superfast-hashnode-blogs-now-powered-by-nextjs-11-and-verc
- **Documentation platforms:** [Mintlify](https://mintlify.com/), [Fern](https://buildwithfern.com/), [Plain](https://www.plain.com/chann
- **Website and ecommerce store builders**: [Super](https://vercel.com/blog/super-serves-thousands-of-domains-on-one-project-with-next-js
- **B2B SaaS platforms**: [Zapier](https://zapier.com/interfaces), [Instatus](https://instatus.com/), [Cal](http://cal.com/)

For example, you might have:

- A root domain for your platform: `acme.com`
- Subdomains for tenants: `tenant1.acme.com`, `tenant2.acme.com`
- Fully custom domains for certain customers: `tenantcustomdomain.com`

Vercel's platform automatically issues [SSL certificates](https://vercel.com/docs/domains/working-with-ssl), handles DNS routing via its 

## Getting started

The fastest way to get started is with our [multi-tenant Next.js starter kit](https://vercel.com/templates/next.js/platforms-starter-kit)

- Custom subdomain routing with Next.js middleware
- Tenant-specific content and pages
- Redis for tenant data storage
- Admin interface for managing tenants
- Compatible with Vercel preview deployments

## Multi-tenant features on Vercel

- Unlimited custom domains
- Unlimited `*.yourdomain.com` subdomains
- Automatic SSL certificate issuance and renewal
- Domain management through REST API or SDK
- Low-latency responses globally with the Vercel CDN
- Preview environment support to test changes
- Support for 35+ frontend and backend frameworks

## Next steps

- [Full Vercel for Platforms docs](/platforms/docs)
- [Learn about limits and features](/docs/multi-tenant/limits)
- [Set up domain management](/docs/multi-tenant/domain-management)
- [Deploy the starter template](https://vercel.com/templates/next.js/platforms-starter-kit)

---
title: "On-Demand Usage Pricing"

last_updated: "2026-01-16T02:19:33.677Z"
source: "https://vercel.com/docs/no-index-on-demand-ent-usage-pricing-01"
---

# On-Demand Usage Pricing

Vercel prices its [CDN](/docs/cdn) resources by region to help optimize costs and performance for your projects. This is to ensure you ar

### Managed Infrastructure Units

Managed Infrastructure Units (MIUs) serve as both a financial commitment and a measurement of the infrastructure consumption of an Enterp

Each MIU is valued at $1.00 USD and is used to pay for the resources consumed by your project. MIUs are billed monthly and do not roll ov

### Regional pricing

The following table lists the pricing for each resource in Managed Infrastructure. Resources that depend on the region of your Vercel pro

Use the dropdown to select the region you are interested in.

### Additional usage based products

The following table lists the pricing for additional usage based products in Managed Infrastructure.

### Secure Compute

Secure Compute is a feature that allows you to run Vercel Functions in a secure environment. **Purchasing Secure Compute will result in a

---
title: "Notebooks"
description: "Learn more about Notebooks and how they allow you to organize and save your queries."
last_updated: "2026-01-16T02:19:33.705Z"
source: "https://vercel.com/docs/notebooks"
---

# Notebooks

**Notebooks** allow you to collect and manage multiple queries related to your application's metrics and performance data.

Within a single notebook, you can store multiple queries that examine different aspects of your system - each with its own specific filte
This facilitates the building of comprehensive dashboards or analysis workflows by grouping related queries together.

> **💡 Note:** You need to enable [Observability

> Plus](/docs/observability/observability-plus) to use Notebooks since you need
> run queries.

## Using and managing notebooks

You can use notebooks to organize and save your queries. Each notebook is a collection of queries that you can keep personal or share wit

### Create a notebook

1. From the **Observability** tab of your dashboard, click **Notebooks** from the left navigation of the Observability Overview page
2. Edit the notebook name by clicking the pencil icon on the top left of the default title which uses your username and created date and

### Add a query to a notebook

1. From the **Notebooks** page, click the **Create Notebook** button or select an existing **Notebook**
2. Click the + icon to open the query builder and build your query
3. Edit the query name by clicking the pencil icon on the top left of the default query title
4. Select the most appropriate view for your query: line chart, volume chart, table or big number
5. Once you're happy with your query results, save it by clicking **Save Query**
6. Your query is now available in your notebook

### Delete a query

1. From the **Notebooks** page, select an existing **Notebook**
2. Click the three-dot menu on the top-right corner of a query, and select **Delete**. This action is permanent and cannot be undone.

### Delete a notebook

1. From the **Notebooks** page, select the **Notebook** you'd like to delete from the list
2. Click the three-dot menu on the top-right corner of the notebook, and select **Delete notebook**. This action is permanent and cannot

## Notebook types and access

You can create 2 types of notebooks.

- Personal Notebooks: Only the creator and owner can view them.
- Team Notebooks: All team members can view them and they share ownership.

When created, notebooks are personal by default. You can use the **Share** button to turn them to Team Notebooks for collaboration. When

As a Notebook owner, you have complete control over your notebook. You can add new queries, edit existing ones, remove individual queries


---
title: "Notifications"
description: "Learn how to use Notifications to view and manage important alerts about your deployments, domains, integrations, account,
last_updated: "2026-01-16T02:19:33.722Z"
source: "https://vercel.com/docs/notifications"
---

# Notifications

Vercel sends configurable notifications to you through the [dashboard](/dashboard) and email. These notifications enable you to view and

## Receiving notifications

There are a number of places where you can receive notifications:

- **Web**: The Vercel dashboard displays a popover, which contains all relevant notifications
- **Email**: You'll receive an email when any of the alerts that you set on your team have been triggered
- **Push**: You'll receive a push notification when any of the alerts that you set on your team have been triggered
- **SMS**: SMS notifications can only be configured on a per-user basis for [Spend Management](/docs/spend-management#managing-alert-thre

By default, you will receive both web and email notifications for all [types of alerts](#types-of-notifications). Push notifications are

## Basic capabilities

There are two main ways to interact with web notifications:

- **Read**: Unread notifications are displayed with a counter on the bell icon. When you view a notification on the web, it will be marke
- **Archive**: You can manage the list of notifications by archiving them. You can view these archived notifications in the archive tab,

## Managing notifications

You can manage **your own** notifications by using the following steps:

1. Select your team from the [scope selector](/docs/dashboard-features#scope-selector).
2. Go to the **Settings** tab of your account or team's dashboard, and under **Account**, select **My Notifications**.
3. From here, you can toggle [where](#receiving-notifications) *you* would like to receive notifications for each different [type of noti

Any changes you make will only be reflected for your notifications and not for any other members of the team. You cannot configure notifi

### Notifications for Comments

You can receive feedback on your deployments with the Comments feature. When someone leaves a comment, you'll receive a notification on V

[Learn more in the Comments docs](/docs/comments/managing-comments#notifications).

### On-demand usage notifications

You'll receive notifications as you accrue usage past the [included amounts](/docs/limits#included-usage) for products like Vercel Functi

**Team owners** on the **Pro** plan can customize which usage categories they want to receive notifications for based on percentage thres

Emails are sent out at specific usage thresholds which vary based on the feature and plan you are on.

> **💡 Note:** If you choose to disable notifications, you won't receive alerts for any
> excessive charges within that category. This may result in unexpected
> additional costs on your bill. It is recommended that you carefully consider
> the implications of turning off notifications for any usage thresholds before

> making changes to your notification settings.

## Types of notifications

The types of notifications available for you to manage depend on the [role](/docs/rbac/access-roles/team-level-roles) you are assigned wi

### Critical notifications

It is *not* possible to disable all notifications for alerts that are critical to your Vercel workflow. You **can** opt-out of [one speci

### Notification details

| Notification group | Type of notification | Explanation |
| --- | --- | --- |
| **Account** | | |
| | Team join requests | Team owners will be notified when someone req |
| **Alerts** | | |
| | Usage Anomalies | Triggered when the usage of your project exce |
| | Error Anomalies | Triggered when a high rate of failed function |
| **Deployment** | | |
| | Deployment Failures | Deployment owners will be notified about any |
| | Deployment Promotions | Deployment owners will be notified about any |
| **Domain** | | |
| | Configuration - Certificate renewal failed | Team owners will be notified if the SSL Certi |
| | Configuration - Domain Configured | Team owners will be notified of any domains t |
| | Configuration - Domain Misconfigured | Team owners will be notified of any domains t |
| | Configuration - Domain no payment source or payment failure | Team owners will be notified if there were an |
| | Renewals - Domain renewals | Team owners will be notified 17 days and 7 da |
| | Renewals - Domain expiration | Team owners will be notified 24 and 14 days b |
| | Transfers - Domain moves requested or completed | Team owners will be notified when a domain ha |
| | Transfers - Domain transfers initiated, cancelled, and completed | Team owners will be notified about any inform |
| | Transfers - Domain transfers pending approval | Team owners will be notified when a domain is |
| **Integrations** | | |
| | Integration configuration disabled | Everyone will be notified about integration u |
| | Integration scope changed | Team owners will be notified if any of the In |
| **Usage** | | |
| | Usage increased | Team owners will be notified about all [usage |
| | Usage limit reached | Users will be notified when they reach the li |
| **Non-configurable** | | |
| | Email changed confirmation | You will be notified when you have successful |
| | Email changed verification | You will be notified when you have updated th |
| | User invited | You will be sent this when you have been invi |
| | Invoice payment failed | Users who can manage billing settings will be |
| | Project role changed | You will be sent this when your [role](/docs/ |
| | User deleted | You will be sent this when you have chosen to |
| **Edge Config** | Size Limit Alerts | Members will be notified when Edge Config siz |
| | Schema Validation Errors | Members will be notified (at most once per ho |

---
title: "Observability Insights"
description: "List of available data sources that you can view and monitor with Observability on Vercel."
last_updated: "2026-01-16T02:19:33.734Z"
source: "https://vercel.com/docs/observability/insights"
---

# Observability Insights

Vercel organizes Observability through sections that correspond to different features and traffic sources that you can view, monitor and

## Vercel Functions

The **Vercel Functions** tab provides a detailed view of the performance of your Vercel Functions. You can see the number of invocations

For more information, see [Vercel Functions](/docs/functions). See [understand the cost impact of function invocations](/kb/guide/underst

### CPU Throttling

When your function uses too much CPU time, Vercel pauses its execution periodically to stay within limits. This means your function may t

CPU throttling itself isn't necessarily a problem as it's designed to keep functions within their resource limits. Some throttling is nor

To reduce throttling, optimize heavy computations, add caching, or increase the memory size of the affected functions.

## External APIs

You can use the **External APIs** tab to understand more information about requests from your functions to external APIs. You can organiz

### External APIs Recipes

- [Investigate Latency Issues and Slowness on Vercel](/kb/guide/investigate-latency-issues-and-slowness)

## Middleware

The **Middleware** observability tab shows invocation counts and performance metrics of your application's middleware.

Observability Plus users receive additional insights and tooling:

- Analyze invocations by request path, matched against your middleware config
- Break down middleware actions by type (e.g., redirect, rewrite)
- View rewrite targets and frequency
- Query middleware invocations using the query builder

## Edge Requests

You can use the **Edge Requests** tab to understand the requests to each of static and dynamic routes through the global network. This in

It also provides detailed breakdowns for individual bots and bot categories, including AI crawlers and search engines.

Additionally, Observability Plus users can:

- Filter traffic by bot category, such as AI
- View metrics for individual bots
- Break down traffic by bot or category in the query builder
- Filter traffic by redirect location
- Break down traffic by redirect location in the query builder

## Fast Data Transfer

You can use the **Fast Data Transfer** tab to understand how data is being transferred within the global network for your project.

For more information, see [Fast Data Transfer](/docs/manage-cdn-usage#fast-data-transfer).

## Image Optimization

The **Image Optimization** tab provides deeper insights into image transformations and efficiency.

It contains:

- Transformation insights: View formats, quality settings, and width adjustments
- Optimization analysis: Identify high-frequency transformations to help inform caching strategies
- Bandwidth savings: Compare transformed images against their original sources to measure bandwidth reduction and efficiency
- Image-specific views: See all referrers and unique variants of an optimized image in one place

For more information, see [Image Optimization](/docs/image-optimization).

## ISR (Incremental Static Regeneration)

You can use the **ISR** tab to understand your revalidations and cache hit ratio to help you optimize towards cached requests by default.

For more information on ISR, see [Incremental Static Regeneration](/docs/incremental-static-regeneration).

## Blob

Use the **Vercel Blob** tab to gain visibility into how Blob stores are used across your applications.
It allows you to understand usage patterns, identify inefficiencies, and optimize how your application stores and serves assets.

At the team level, you will access:

- Total data transfer
- Download volume
- Cache activity
- API operations

You can also drill into activity by user agent, edge region, and client IP.

Learn more about [Vercel Blob](/docs/storage/vercel-blob).

## Build Diagnostics

You can use the **Build Diagnostics** tab to view the performance of your builds. You can see the build time and resource usage for each

To learn more, see [Builds](/docs/deployments/builds).

## AI Gateway

With the AI Gateway you can switch between ~100 AI models without needing to manage API keys, rate limits, or provider accounts.

The **AI Gateway** tab surfaces metrics related to the AI Gateway, and provides visibility into:

- Requests by model
- Time to first token (TTFT)
- Request duration
- Input/output token count
- Cost per request (free while in alpha)

You can view these metrics across all projects or drill into per-project and per-model usage to understand which models are performing we

For more information, see [the AI Gateway announcement](/blog/ai-gateway).

## Sandbox

With [Vercel Sandbox](/docs/vercel-sandbox), you can safely run untrusted or user-generated code on Vercel in an ephemeral compute primit

You can view a list of sandboxes that were started for this project. For each sandbox, you can see:

- Time started
- Status such as pending or stopped
- Runtime such as `node24`
- Resources such as `4x CPU 8.19 KB`
- Duration it ran for

Clicking on a sandbox item from the list takes you to the detail page that provides detailed information, including the URL and port of t

## External Rewrites

The **External Rewrites** tab gives you visibility into how your external rewrites are performing at both the team and project levels. Fo

- Total external rewrites
- External rewrites by hostnames

Additionally, Observability Plus users can view:

- External rewrite connection latency
- External rewrites by source/destination paths

To learn more, see [External Rewrites](/docs/rewrites#external-rewrites).

## Microfrontends

Vercel's microfrontends support allows you to split large applications into smaller ones to move faster and develop with independent tech

The **Microfrontends** tab provides visibility into microfrontends routing on Vercel:

- The response reason from the microfrontends routing logic
- The path expression used to route the request to that microfrontend

For more information, see [Microfrontends](/docs/microfrontends).

---
title: "Observability Plus"
description: "Learn about using Observability Plus and its limits."
last_updated: "2026-01-16T02:19:33.687Z"
source: "https://vercel.com/docs/observability/observability-plus"
---

# Observability Plus

**Observability Plus** is an optional upgrade that enables Pro and Enterprise teams to explore data at a more granular level, helping you

To learn more about Observability Plus, see [Limitations](#limitations) or [pricing](#pricing).

## Using Observability Plus

### Enabling Observability Plus

By default, all users on all plans have access to Observability at both a team and project level.

To upgrade to Observability Plus:

1. From your [dashboard](/dashboard), navigate to [the **Observability** tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fobservability
2. Next to the time range selector, click the  button and select **Upgrade to Observability Plus**.
3. From the **Upgrade to Observability Plus** modal, click **Continue**.
   - If you're an existing Monitoring user, the modal will be **Migrate from Monitoring to Observability Plus** and will display the redu
4. Then, view the charges and click **Confirm and Pay**.

You'll be charged and upgraded immediately. You will immediately have access to the Observability Plus features and can view [events](/do

### Disabling Observability Plus

1. From your [dashboard](/dashboard), navigate to [the **Observability** tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fobservability
2. Next to the time range selector, click the  button and select **Observability Settings**.
3. This takes you to the [**Observability Plus** section of your project's **Billing** settings](https://vercel.com/d?to=%2F%5Bteam%5D%2F
   - Click the toggle button to disable it
   - Click the **Confirm** button in the **Turn off Observability Plus** dialog

## Pricing

Users on all plans can use Observability at no additional cost, with some [limitations](#limitations). Observability is available for all

Owners on Pro and Enterprise teams can upgrade to **Observability Plus** to get access to additional features, higher limits, and increas

| Resource | Base Fee | Usage-based pri |
| --- | --- | --- |
| Observability Plus | Pro: $10/month  Enterprise: none | $1.20 per 1 million [events](/docs/observability#tracked-events) |

## Limitations

| Feature | Observability | Obse |
| --- | --- | --- |
| Data Retention | Hobby: 12 hours  Pro: 1 day  Enterprise: 3 days | 30 days |
| Monitoring access | Not Included | Incl |
| Vercel Functions | No Latency (p75) data, no breakdown by path | Late |
| External APIs | No ability to sort by error rate or p75 duration, only request totals for each hostname | Sort |
| Edge Requests | No breakdown by path | Full |
| Fast Data Transfer | No breakdown by path | Full |
| ISR (Incremental Static Regeneration) | No access to average duration or revalidation data. Limited function data for each route | Acce |
| Build Diagnostics | Full access | Full |
| In-function Concurrency | Full access when enabled | Full |
| Runtime logs | Hobby: 1 hour  Pro: 1 day  Enterprise: 3 days | 30 days, max s |

## Prorating

Pro teams are charged a base fee when enabling Observability Plus. However, you will only be charged for the remaining time in your billi

- If ten days remain in your current billing cycle, you will only pay around $3. For every new billing cycle after that, you'll be charge
- Events are prorated. This means that if your team incurs 100K events over the included allotment, you would will only pay $0.12 over th
- Suppose you disable Observability Plus before the billing cycle ends. In that case, Observability Plus will automatically turn off, we w
- Once the billing cycle is over, you will be charged for the events collected prior to disabling. You won't be refunded any amounts alre
- Re-enabling Observability Plus before the end of the billing cycle won't cost you another base fee. Instead, the usual base fee of $10 w

---
title: "Observability"
description: "Observability on Vercel provides framework-aware insights enabling you to optimize infrastructure and application performan
last_updated: "2026-01-16T02:19:33.699Z"
source: "https://vercel.com/docs/observability"
---

# Observability

Observability provides a way for you to monitor and analyze the performance and traffic of your projects on Vercel through a variety of [

- Learn how to [use Observability](#using-observability) and the available [insight sections](/docs/observability#available-insights)
- Learn how you can save and organize your Observability queries with [Notebooks](/docs/notebooks)

### Observability feature access

You can use Observability on all plans to monitor your projects. If you are on the Pro or Enterprise plan, you can [upgrade](/docs/observ

[Try Observability](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fobservability\&title=Try+Observability) to get started.

## Using Observability

How you use Observability depends on the needs of your project, for example, perhaps builds are taking longer than expected, or your Verc

1. Decide what feature you want to investigate. For example, **Vercel Functions**.
2. Use the date picker or the time range selector to choose the time period you want to investigate. Users on [Observability Plus](/docs/
3. Let's investigate our graphs in more detail, for example, **Error Rate**. Click and drag to select a period of time and press the **Zo
4. Then, from the list of routes below, choose to reorder either based on the error rate or the duration to get an idea of which routes a
5. To learn more about specific routes, click on the route.
6. The functions view will show you the performance of each route or function, including details about the function, latency, paths, and
7. The function view also provides a direct link to the logs for that function, enabling you to pinpoint the cause of the issue.

### Available insights

Observability provides different sections of features and traffic sources that help you monitor, analyze, and manage your applications ei

| Data source | Team Level | Project Level |
| --- | --- | --- |
| [Vercel Functions](/docs/observability/insights#vercel-functions) | ✓ | ✓ |
| [External APIs](/docs/observability/insights#external-apis) | ✓ | ✓ |
| [Edge Requests](/docs/observability/insights#edge-requests) | ✓ | ✓ |
| [Middleware](/docs/observability/insights#middleware) | ✓ | ✓ |
| [Fast Data Transfer](/docs/observability/insights#fast-data-transfer) | ✓ | ✓ |
| [Image Optimization](/docs/observability/insights#image-optimization) | ✓ | ✓ |
| [ISR (Incremental Static Regeneration)](/docs/observability/insights#isr-incremental-static-regeneration) | ✓ | ✓ |
| [Blob](/docs/observability/insights#blob) | ✓ | |
| [Build Diagnostics](/docs/observability/insights#build-diagnostics) | | ✓ |
| [AI Gateway](/docs/observability/insights#ai-gateway) | ✓ | ✓ |
| [External Rewrites](/docs/observability/insights#external-rewrites) | ✓ | ✓ |
| [Microfrontends](/docs/observability/insights#microfrontends) | ✓ | ✓ |

## Tracked events

Vercel tracks the following event types for Observability:

- Edge Requests
- Vercel Function Invocations
- External API Requests
- Routing Middleware Invocations
- AI Gateway Requests

Vercel creates one or more of these events each time a request is made to your site. Depending on your application and configuration a si

- 1 edge request event if it's cached.
- 1 Edge Request, 1 Middleware, 1 Function Invocation, 2 External API calls, and 1 AI Gateway request, for a total of 6 events.
- 1 edge request event if it's a static asset.

Events are tracked on a team level, and so the events are counted across all projects in the team.

## Pricing and limitations

Users on all plans can use Observability at no additional cost, with some [limitations](/docs/observability/observability-plus#limitation

[Owners](/docs/rbac/access-roles#owner-role) on Pro and Enterprise teams can [upgrade](/docs/observability/observability-plus#enabling-ob

For more information on pricing, see [Pricing](/docs/observability/observability-plus#pricing).

## Existing Monitoring users

Monitoring is now automatically included with [Observability Plus](/docs/observability/observability-plus) and cannot be purchased separa

Teams that are currently paying for Monitoring, will not automatically see the [Observability Plus](/docs/observability/observability-plu

In addition, teams that subscribe to [Observability Plus](/docs/observability/observability-plus) will have access to the **Monitoring**

---
title: "OG Image Generation Examples"
description: "Learn how to use the @vercel/og library with examples."
last_updated: "2026-01-16T02:19:33.811Z"
source: "https://vercel.com/docs/og-image-generation/examples"
---

# OG Image Generation Examples

## Dynamic title

## Dynamic external image

## Emoji

## SVG

## Custom font

## Tailwind CSS

## Internationalization

## Secure URL

---
title: "@vercel/og Reference"
description: "This reference provides information on how the @vercel/og package works on Vercel."
last_updated: "2026-01-16T02:19:33.747Z"
source: "https://vercel.com/docs/og-image-generation/og-image-api"

---

# @vercel/og Reference

The package exposes an `ImageResponse` constructor, with the following parameters:

```ts v0="build" filename="ImageResponse Interface" framework=all
import { ImageResponse } from '@vercel/og'

new ImageResponse(
  element: ReactElement,
  options: {
    width?: number = 1200
    height?: number = 630
    emoji?: 'twemoji' | 'blobmoji' | 'noto' | 'openmoji' = 'twemoji',
    fonts?: {
      name: string,
      data: ArrayBuffer,
      weight: number,
      style: 'normal' | 'italic'
    }[]
    debug?: boolean = false

    // Options that will be passed to the HTTP response
    status?: number = 200
    statusText?: string
    headers?: Record<string, string>
  },
)
```

### Main parameters

| Parameter | Type | Default | Description |
| --- | --- | --- | --- |
| `element` | `ReactElement` | – | The React element to generate the image from. |
| `options` | `object` | – | Options to customize the image and HTTP response. |

### Options parameters

| Parameter | Type | Default | Description |
| --- | --- | --- | --- |
| `width` | `number` | `1200` | The width of the image. |
| `height` | `number` | `630` | The height of the image. |
| `emoji` | `twemoji` `blobmoji` `noto` `openmoji` `twemoji` | The emoji set to use. | |
| `debug` | `boolean` | `false` | Debug mode flag. |
| `status` | `number` | `200` | The HTTP status code for the response. |
| `statusText` | `string` | – | The HTTP status text for the response. |
| `headers` | `Record<string, string>` | – | The HTTP headers for the response. |

### Fonts parameters (within options)

| Parameter | Type | Default | Description |
| --- | --- | --- | --- |
| `name` | `string` | – | The name of the font. |
| `data` | `ArrayBuffer` | – | The font data. |
| `weight` | `number` | – | The weight of the font. |
| `style` | `normal` `italic` | – | The style of the font. |

By default, the following headers will be included by `@vercel/og`:

```javascript filename="included-headers"

'content-type': 'image/png',
'cache-control': 'public, immutable, no-transform, max-age=31536000',

```

## Supported HTML and CSS features

Refer to [Satori's documentation](https://github.com/vercel/satori#documentation) for a list of supported HTML and CSS features.

By default, `@vercel/og` only has the Noto Sans font included. If you need to use other fonts, you can pass them in the `fonts` option. V

## Acknowledgements

- [Twemoji](https://github.com/twitter/twemoji)
- [Google Fonts](https://fonts.google.com) and [Noto Sans](https://www.google.com/get/noto/)
- [Resvg](https://github.com/RazrFalcon/resvg) and [Resvg.js](https://github.com/yisibl/resvg-js)


---
title: "Open Graph (OG) Image Generation"
description: "Learn how to optimize social media image generation through the Open Graph Protocol and @vercel/og library."
last_updated: "2026-01-16T02:19:33.933Z"
source: "https://vercel.com/docs/og-image-generation"
---

# Open Graph (OG) Image Generation

To assist with generating dynamic [Open Graph (OG)](https://ogp.me/ "Open Graph (OG)") images, you can use the Vercel `@vercel/og` librar

## Benefits

- **Performance:** With a small amount of code needed to generate images, [functions](/docs/functions) can be started almost instantly. T
- **Ease of use:** You can define your images using HTML and CSS and the library will dynamically generate images from the markup
- **Cost-effectiveness:** `@vercel/og` automatically adds the correct headers to cache computed images on the CDN, helping reduce cost an

## Supported features

- Basic CSS layouts including flexbox and absolute positioning
- Custom fonts, text wrapping, centering, and nested images
- Ability to download the subset characters of the font from Google Fonts
- Compatible with any framework and application deployed on Vercel
- View your OG image and other metadata before your deployment goes to production through the [Open Graph](/docs/deployments/og-preview)

## Runtime support

Vercel OG image generation is supported on the [Node.js runtime](/docs/functions/runtimes/node-js).

Local resources can be loaded directly using `fs.readFile`. Alternatively, `fetch` can be used to load remote resources.

```js filename="og.js"
const fs = require('fs').promises;

const loadLocalImage = async () => {
  const imageData = await fs.readFile('/path/to/image.png');
  // Process image data
};
```

### Runtime caveats

There are limitations when using `vercel/og` with the **Next.js Pages Router** and the Node.js runtime. Specifically, this combination do

| Configuration | Supported Syntax | Notes |
| --- | --- | --- |
| `pages/` + Edge runtime | `return new Response(…)` | Fully supported. |
| `app/` + Node.js runtime | `return new Response(…)` | Fully supported. |
| `app/` + Edge runtime | `return new Response(…)` | Fully supported. |
| `pages/` + Node.js runtime | Not supported | Does not support `return new Response(…)` syntax with `vercel/og`. |

## Usage

### Requirements

- Install  or newer by visiting [nodejs.org](https://nodejs.org)
- Install `@vercel/og` by running the following command inside your project directory. **This isn't required for Next.js App Router proje

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i @vercel/og
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @vercel/og
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @vercel/og
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i @vercel/og
    ```
  </Code>
</CodeBlock>

- For Next.js implementations, make sure you are using Next.js v12.2.3 or newer
- Create API endpoints that you can call from your front-end to generate the images. Since the HTML code for generating the image is incl
- To avoid the possibility of social media providers not being able to fetch your image, it is recommended to add your OG image API route
  ```txt filename="robots.txt"
  Allow: /api/og/*
  ```
  If you are using Next.js, review [robots.txt](https://nextjs.org/docs/app/api-reference/file-conventions/metadata/robots#static-robotst

### Getting started

Get started with an example that generates an image from static text using Next.js by setting up a new app with the following command:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i
    ```
  </Code>
</CodeBlock>

> For \["nextjs"]:

Create an API endpoint by adding  under the `/pages/api` directory in the root of your project.

> For \["nextjs-app"]:

Create an API endpoint by adding  under the `app/api/og` directory in the root of your project.

> For \["other"]:

Create an API endpoint by adding  under the `api` directory in the root of your project.

Then paste the following code:

```ts v0="build" filename="app/api/og/route.tsx" framework=nextjs-app
import { ImageResponse } from 'next/og';
// App router includes @vercel/og.
// No need to install it.

export async function GET() {
  return new ImageResponse(
    (
      <div
        style={{
          fontSize: 40,
          color: 'black',
          background: 'white',
          width: '100%',
          height: '100%',
          padding: '50px 200px',
          textAlign: 'center',
          justifyContent: 'center',
          alignItems: 'center',
        }}
      >
        👋 Hello
      </div>
    ),
    {
      width: 1200,
      height: 630,
    },
  );
}
```

```js v0="build" filename="app/api/og/route.jsx" framework=nextjs-app
import { ImageResponse } from 'next/og';
// App router includes @vercel/og.
// No need to install it.

export async function GET() {
  return new ImageResponse(
    (
      <div
        style={{
          fontSize: 40,
          color: 'black',
          background: 'white',
          width: '100%',
          height: '100%',
          padding: '50px 200px',
          textAlign: 'center',
          justifyContent: 'center',
          alignItems: 'center',
        }}
      >
        👋 Hello
      </div>
    ),
    {
      width: 1200,
      height: 630,
    },
  );
}
```

```ts v0="build" filename="pages/api/og.tsx" framework=nextjs
import { ImageResponse } from '@vercel/og';

export default async function handler() {
  return new ImageResponse(
    (
      <div
        style={{
          fontSize: 40,
          color: 'black',
          background: 'white',
          width: '100%',
          height: '100%',
          padding: '50px 200px',
          textAlign: 'center',
          justifyContent: 'center',
          alignItems: 'center',
        }}
      >
        👋 Hello 你好 नमस्ते こんにちは สวัสดีค่ะ 안녕 добрий день Hallá
      </div>
    ),
    {
      width: 1200,
      height: 630,
```

```
    },
  );
}
```

```js v0="build" filename="pages/api/og.jsx" framework=nextjs
import { ImageResponse } from '@vercel/og';

export default async function handler() {
  return new ImageResponse(
    (
      <div
        style={{
          fontSize: 40,
          color: 'black',
          background: 'white',
          width: '100%',
          height: '100%',
          padding: '50px 200px',
          textAlign: 'center',
          justifyContent: 'center',
          alignItems: 'center',
        }}
      >
        👋 Hello 你好 नमस्ते こんにちは สวัสดีค่ะ 안녕 добрий день Hallá
      </div>
    ),
    {
      width: 1200,
      height: 630,
    },
  );
}
```

```ts filename="api/og.tsx" framework=other
import { ImageResponse } from '@vercel/og';

export default async function handler() {
  return new ImageResponse(
    (
      <div
        style={{
          fontSize: 40,
          color: 'black',
          background: 'white',
          width: '100%',
          height: '100%',
          padding: '50px 200px',
          textAlign: 'center',
          justifyContent: 'center',
          alignItems: 'center',
        }}
      >
        👋 Hello
      </div>
    ),
    {
      width: 1200,
      height: 630,
    },
  );
}
```

```js filename="api/og.jsx" framework=other
import { ImageResponse } from '@vercel/og';

export default async function handler() {
  return new ImageResponse(
    (
      <div
        style={{
          fontSize: 40,
          color: 'black',
          background: 'white',
          width: '100%',
          height: '100%',
          padding: '50px 200px',
          textAlign: 'center',
          justifyContent: 'center',
          alignItems: 'center',
        }}
      >
        👋 Hello
      </div>
    ),
    {
      width: 1200,
      height: 630,
    },
  );
}
```

> **💡 Note:** If you're not using a framework, you must either add
>  to your
>  or change your JavaScript Functions'
> file extensions from  to
>

Run the following command:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i
    ```
  </Code>
</CodeBlock>

Then, browse to `http://localhost:3000/api/og`. You will see the following image:

### Consume the OG route

Deploy your project to obtain a publicly accessible path to the OG image API endpoint. You can find an example deployment at <https://og-

Then, based on the [Open Graph Protocol](https://ogp.me/#metadata), create the web content for your social media post as follows:

- Create a `<meta>` tag inside the `<head>` of the webpage
- Add the `property` attribute with value `og:image` to the `<meta>` tag
- Add the `content` attribute with value as the absolute path of the `/api/og` endpoint to the `<meta>` tag

With the example deployment at <https://og-examples.vercel.sh/api/static>, use the following code:

```html filename="index.js"
<head>
  <title>Hello world</title>
  <meta
    property="og:image"
    content="https://og-examples.vercel.sh/api/static"
  />
</head>
```

Every time you create a new social media post, you need to update the API endpoint with the new content. However, if you identify which p

In the examples below, we explore using parameters and including other types of content with `ImageResponse`.

## Examples

- [Dynamic title](/docs/og-image-generation/examples#dynamic-title): Passing the image title as a URL parameter
- [Dynamic external image](/docs/og-image-generation/examples#dynamic-external-image): Passing the username as a URL parameter to pull an
- [Emoji](/docs/og-image-generation/examples#emoji): Using emojis to generate the image
- [SVG](/docs/og-image-generation/examples#svg): Using SVG embedded content to generate the image
- [Custom font](/docs/og-image-generation/examples#custom-font): Using a custom font available in the file system to style your image tit
- [Tailwind CSS](/docs/og-image-generation/examples#tailwind-css): Using Tailwind CSS (Experimental) to style your image content
- [Internationalization](/docs/og-image-generation/examples#internationalization): Using other languages in the text for generating your
- [Secure URL](/docs/og-image-generation/examples#secure-url): Encrypting parameters so that only certain values can be passed to generat

## Technical details

- Recommended OG image size: 1200x630 pixels
- `@vercel/og` uses [Satori](https://github.com/vercel/satori) and Resvg to convert HTML and CSS into PNG
- `@vercel/og` [API reference](/docs/og-image-generation/og-image-api)

## Limitations

- Only `ttf`, `otf`, and `woff` font formats are supported. To maximize the font parsing speed, `ttf` or `otf` are preferred over `woff`
- Only flexbox (`display: flex`) and a subset of CSS properties are supported. Advanced layouts (`display: grid`) will not work. See [Sat
- Maximum bundle size of 500KB. The bundle size includes your JSX, CSS, fonts, images, and any other assets. If you exceed the limit, con

---
title: "Connect to your own API"
description: "Learn how to configure your own API to trust Vercel"
last_updated: "2026-01-16T02:19:33.827Z"
source: "https://vercel.com/docs/oidc/api"
---

# Connect to your own API

## Validate the tokens

To configure your own API to accept Vercel's OIDC tokens, you need to validate the tokens using Vercel's JSON Web Keys (JWTs), available

### Use the `jose.jwtVerify` function

Install the following package:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i jose
    ```

  </Code>
  <Code tab="yarn">
    ```bash
    yarn i jose
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i jose
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i jose
    ```
  </Code>
</CodeBlock>

In the code example below, you use the `jose.jwtVerify` function to verify the token. The `issuer`, `audience`, and `subject` are validat

```ts filename="server.ts"
import http from 'node:http';
import * as jose from 'jose';

const ISSUER_URL = `https://oidc.vercel.com/[TEAM_SLUG]`;
// or use `https://oidc.vercel.com` if your issuer mode is set to Global.

const JWKS = jose.createRemoteJWKSet(new URL(ISSUER_URL, '/.well-known/jwks'));

const server = http.createServer((req, res) => {
  const token = req.headers['authorization']?.split('Bearer ')[1];

  if (!token) {
    res.statusCode = 401;
    res.end('Unauthorized');
    return;
  }

  try {
    const { payload } = jose.jwtVerify(token, JWKS, {
      issuer: ISSUER_URL,
      audience: 'https://vercel.com/[TEAM_SLUG]',
      subject:
        'owner:[TEAM_SLUG]:project:[PROJECT_NAME]:environment:[ENVIRONMENT]',
    });

    res.statusCode = 200;
    res.end('OK');
  } catch (error) {
    res.statusCode = 401;
    res.end('Unauthorized');
  }
});

server.listen(3000);
```

Make sure that you:

- Replace `[TEAM_SLUG]` with your team identifier from the Vercel's team URL
- Replace `[PROJECT_NAME]` with your [project's name](https://vercel.com/docs/projects/overview#project-name) in your [project's settings](https://vercel.com/docs/projects/overview#project-settings)
- Replace `[ENVIRONMENT]` with one of Vercel's [environments](https://vercel.com/docs/deployments/environments#deployment-environments), `development`, `preview` or `production`

### Use the `getVercelOidcToken` function

Install the following package:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i @vercel/functions
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @vercel/functions
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @vercel/functions
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i @vercel/functions
    ```
  </Code>
</CodeBlock>

In the code example below, the `getVercelOidcToken` function is used to retrieve the OIDC token from your Vercel environment.
You can then use this token to authenticate the request to the external API.

```ts filename="/api/custom-api/route.ts"
import { getVercelOidcToken } from '@vercel/oidc';

export const GET = async () => {
  const result = await fetch('https://api.example.com', {
```

```
    headers: {
      Authorization: `Bearer ${await getVercelOidcToken()}`,
    },
  });

  return Response.json(await result.json());
};
```

---
title: "Connect to Amazon Web Services (AWS)"
description: "Learn how to configure your AWS account to trust Vercel"
last_updated: "2026-01-16T02:19:33.837Z"
source: "https://vercel.com/docs/oidc/aws"
---

# Connect to Amazon Web Services (AWS)

To understand how AWS supports OIDC, and for a detailed user guide on creating an OIDC identity provider with AWS, consult the [AWS OIDC

## Configure your AWS account

- ### Create an OIDC identity provider
  1. Navigate to the [AWS Console](https://console.aws.amazon.com/)
  2. Navigate to **IAM** then **Identity Providers**
  3. Select **Add Provider**
  4. Select **OpenID Connect** from the provider type
  5. Enter the **Provider URL**, the URL will depend on the issuer mode setting:
     - **Team**: `https://oidc.vercel.com/[TEAM_SLUG]`, replacing `[TEAM_SLUG]` with the path from your Vercel team URL
     - **Global**: `https://oidc.vercel.com`
  6. Enter `https://vercel.com/[TEAM_SLUG]` in the **Audience** field, replacing `[TEAM_SLUG]` with the path from your Vercel team URL
  7. Select **Add Provider**

- ### Create an IAM role
  To use AWS OIDC Federation you must have an [IAM role](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html). [IAM roles](htt

  Here is an example of a trust policy using the **Team** issuer mode:
  ```json filename="trust-policy.json"
  {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Federated": "arn:aws:iam::[YOUR AWS ACCOUNT ID]:oidc-provider/oidc.vercel.com/[TEAM_SLUG]"
        },
        "Action": "sts:AssumeRoleWithWebIdentity",
        "Condition": {
          "StringEquals": {
            "oidc.vercel.com/[TEAM_SLUG]:sub": "owner:[TEAM SLUG]:project:[PROJECT NAME]:environment:production",
            "oidc.vercel.com/[TEAM_SLUG]:aud": "https://vercel.com/[TEAM SLUG]"
          }
        }
      }
    ]
  }
  ```
  The above policy's conditions are quite strict. It requires the `aud` sub `sub` claims to match exactly,
  but it's possible to configure less strict trust policies conditions:
  ```json filename="trust-policy.json"
  {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Federated": "arn:aws:iam::[YOUR AWS ACCOUNT ID]:oidc-provider/oidc.vercel.com/[TEAM_SLUG]"
        },
        "Action": "sts:AssumeRoleWithWebIdentity",
        "Condition": {
          "StringEquals": {
            "oidc.vercel.com/[TEAM_SLUG]:aud": "https://vercel.com/[TEAM SLUG]"
          },
          "StringLike": {
            "oidc.vercel.com/[TEAM_SLUG]:sub": [
              "owner:[TEAM SLUG]:project:*:environment:preview",
              "owner:[TEAM SLUG]:project:*:environment:production"
            ]
          }
        }
      }
    ]
  }
  ```
  This policy allows any project matched by the `*` that are targeted to `preview` and `production` but not `development`.

- ### Define the role ARN as environment variable
  Once you have created the role, copy the [role's ARN](https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_identifiers.html#ident
  ```env filename=".env.local"
  AWS_ROLE_ARN=arn:aws:iam::accountid:user/username
  ```
  You are now ready to connect to your AWS resource in your project's code. Review the examples below.

## Examples

In the following examples, you create a [Vercel function](/docs/functions/quickstart#create-a-vercel-function) in the Vercel project wher

### List objects in an AWS S3 bucket

Install the following packages:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i @aws-sdk/client-s3 @vercel/functions
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @aws-sdk/client-s3 @vercel/functions
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @aws-sdk/client-s3 @vercel/functions
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i @aws-sdk/client-s3 @vercel/functions
    ```
  </Code>
</CodeBlock>

In the API route for the function, use the AWS SDK for JavaScript to list objects in an S3 bucket with the following code:

```ts filename="/api/aws-s3/route.ts"
import * as S3 from '@aws-sdk/client-s3';
import { awsCredentialsProvider } from '@vercel/oidc-aws-credentials-provider';

const AWS_REGION = process.env.AWS_REGION!;
const AWS_ROLE_ARN = process.env.AWS_ROLE_ARN!;
const S3_BUCKET_NAME = process.env.S3_BUCKET_NAME!;

// Initialize the S3 Client
const s3client = new S3.S3Client({
  region: AWS_REGION,
  // Use the Vercel AWS SDK credentials provider
  credentials: awsCredentialsProvider({
    roleArn: AWS_ROLE_ARN,
  }),
});

export async function GET() {
  const result = await s3client.send(
    new S3.ListObjectsV2Command({
      Bucket: S3_BUCKET_NAME,
    }),
  );
  return result?.Contents?.map((object) => object.Key) ?? [];
}
```

Vercel sends the OIDC token to the SDK using the `awsCredentialsProvider` function from `@vercel/functions`.

### Query an AWS RDS instance

Install the following packages:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i @aws-sdk/rds-signer @vercel/functions pg
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @aws-sdk/rds-signer @vercel/functions pg
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @aws-sdk/rds-signer @vercel/functions pg
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i @aws-sdk/rds-signer @vercel/functions pg
    ```
  </Code>
</CodeBlock>

In the API route for the function, use the AWS SDK for JavaScript to perform a database `SELECT` query from an AWS RDS instance with the

```ts filename="/api/aws-rds/route.ts"
import { awsCredentialsProvider } from '@vercel/oidc-aws-credentials-provider';
import { Signer } from '@aws-sdk/rds-signer';
import { Pool } from 'pg';

const RDS_PORT = parseInt(process.env.RDS_PORT!);
const RDS_HOSTNAME = process.env.RDS_HOSTNAME!;
const RDS_DATABASE = process.env.RDS_DATABASE!;
const RDS_USERNAME = process.env.RDS_USERNAME!;
const AWS_REGION = process.env.AWS_REGION!;
const AWS_ROLE_ARN = process.env.AWS_ROLE_ARN!;

// Initialize the RDS Signer
const signer = new Signer({
  // Use the Vercel AWS SDK credentials provider
```

```
  credentials: awsCredentialsProvider({
    roleArn: AWS_ROLE_ARN,
  }),
  region: AWS_REGION,
  port: RDS_PORT,
  hostname: RDS_HOSTNAME,
  username: RDS_USERNAME,
});

// Initialize the Postgres Pool
const pool = new Pool({
  password: signer.getAuthToken,
  user: RDS_USERNAME,
  host: RDS_HOSTNAME,
  database: RDS_DATABASE,
  port: RDS_PORT,
});

// Export the route handler
export async function GET() {
  try {
    const client = await pool.connect();
    const { rows } = await client.query('SELECT * FROM my_table');
    return Response.json(rows);
  } finally {
    client.release();
  }
}
```

---
title: "Connect to Microsoft Azure"
description: "Learn how to configure your Microsoft Azure account to trust Vercel"
last_updated: "2026-01-16T02:19:33.846Z"
source: "https://vercel.com/docs/oidc/azure"
---

# Connect to Microsoft Azure

To understand how Azure supports OIDC through Workload Identity Federation, consult the [Azure documentation](https://learn.microsoft.com

## Configure your Azure account

- ### Create a Managed Identity
  - Navigate to **All services**
  - Select **Identity**
  - Select **Manage Identities** and select **Create**
  - Choose your Azure Subscription, Resource Group, Region and Name

- ### Create a Federated Credential
  - Go to **Federated credentials** and select **Add Credential**
  - In the **Federated credential scenario** field select **Other**
  - Enter the **Issuer URL**, the URL will depend on the issuer mode setting:
    - **Team**: `https://oidc.vercel.com/[TEAM_SLUG]`, replacing `[TEAM_SLUG]` with the path from your Vercel team URL
    - **Global**: `https://oidc.vercel.com`
  - In the **Subject identifier** field use: `owner:[TEAM_SLUG]:project[PROJECT_NAME]:environment:[preview | production | development]`
    - Replace `[TEAM_SLUG]` with your team identifier from the Vercel's team URL
    - Replace `[PROJECT_NAME]` with your [project's name](https://vercel.com/docs/projects/overview#project-name) in your
      [project's settings](https://vercel.com/docs/projects/overview#project-settings)
  - In the **Name** field, use a name for your own reference such as: `[Project name] - [Environment]`
  - In the **Audience** field use: `https://vercel.com/[TEAM_SLUG]`
    - Replace `[TEAM_SLUG]` with your team identifier from the Vercel's team URL
  > **💡 Note:** Azure does not allow for partial claim conditions so you must specify the
  > `Subject` and `Audience` fields exactly. However, it is possible to create
  > mutliple federated credentials on the same managed identity to allow for the
  > various `sub` claims.

- ### Grant access to the Azure service
  In order to connect to the Azure service that you would like to use, you need to allow your Managed Identity to access it.

  For example, to use Azure CosmosDB, associate a role definition to the Managed Identity using the Azure CLI, as explained in the [Azure

  You are now ready to connect to your Azure service from your project's code. Review the example below.

## Example

In the following example, you create a [Vercel function](/docs/functions/quickstart#create-a-vercel-function) in a Vercel project where y

### Query an Azure CosmosDB instance

Install the following packages:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i @azure/identity @azure/cosmos @vercel/functions
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @azure/identity @azure/cosmos @vercel/functions
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @azure/identity @azure/cosmos @vercel/functions
    ```
  </Code>
  <Code tab="bun">

    ```bash
    bun i @azure/identity @azure/cosmos @vercel/functions
    ```
  </Code>
</CodeBlock>

In the API route for this function, use the following code to perform a database `SELECT` query from an Azure CosmosDB instance:

```ts filename="/api/azure-cosmosdb/route.ts"
import {
  ClientAssertionCredential,
  AuthenticationRequiredError,
} from '@azure/identity';
import * as cosmos from '@azure/cosmos';
import { getVercelOidcToken } from '@vercel/oidc';

/**
 * The Azure Active Directory tenant (directory) ID.
 * Added to environment variables
 */
const AZURE_TENANT_ID = process.env.AZURE_TENANT_ID!;

/**
 * The client (application) ID of an App Registration in the tenant.
 * Added to environment variables
 */
const AZURE_CLIENT_ID = process.env.AZURE_CLIENT_ID!;
const COSMOS_DB_ENDPOINT = process.env.COSMOS_DB_ENDPOINT!;
const COSMOS_DB_ID = process.env.COSMOS_DB_ID!;
const COSMOS_DB_CONTAINER_ID = process.env.COSMOS_DB_CONTAINER_ID!;

const tokenCredentials = new ClientAssertionCredential(
  AZURE_TENANT_ID,
  AZURE_CLIENT_ID,
  getVercelOidcToken,
);

const cosmosClient = new cosmos.CosmosClient({
  endpoint: COSMOS_DB_ENDPOINT,
  aadCredentials: tokenCredentials,
});

const container = cosmosClient
  .database(COSMOS_DB_ID)
  .container(COSMOS_DB_CONTAINER_ID);

export async function GET() {
  const { resources } = await container.items
    .query('SELECT * FROM my_table')
    .fetchAll();

  return Response.json(resources);
}
```

---
title: "Connect to Google Cloud Platform (GCP)"
description: "Learn how to configure your GCP project to trust Vercel"
last_updated: "2026-01-16T02:19:33.869Z"
source: "https://vercel.com/docs/oidc/gcp"
---

# Connect to Google Cloud Platform (GCP)

To understand how GCP supports OIDC through Workload Identity Federation, consult the [GCP documentation](https://cloud.google.com/iam/do

## Configure your GCP project

- ### Configure a Workload Identity Federation
  1. Navigate to the [Google Cloud Console](https://console.cloud.google.com/)
  2. Navigate to **IAM & Admin** then **Workload Identity Federation**
  3. Click on **Create Pool**

- ### Create an identity pool
  1. Enter a name for the pool, e.g. `Vercel`
  2. Enter an ID for the pool, e.g. `vercel` and click **Continue**

- ### Add a provider to the identity pool
  1. Select `OpenID Connect (OIDC)` from the provider types
  2. Enter a name for the provider, e.g. `Vercel`
  3. Enter an ID for the provider, e.g. `vercel`
  4. Enter the **Issuer URL**, the URL will depend on the issuer mode setting:
     - **Team**: `https://oidc.vercel.com/[TEAM_SLUG]`, replacing `[TEAM_SLUG]` with the path from your Vercel team URL
     - **Global**: `https://oidc.vercel.com`
  5. Leave JWK file (JSON) empty
  6. Select `Allowed audiences` from "Audience"
  7. Enter `https://vercel.com/[TEAM_SLUG]` in the "Audience 1" field and click "Continue"

- ### Configure the provider attributes
  1. Assign the `google.subject` mapping to `assertion.sub`
  2. Click **Save**

- ### Create a service account
  1. Copy the **IAM Principal** from the pool details page from the previous step. It should look like `principal://iam.googleapis.com/pr
  2. Navigate to **IAM & Admin** then **Service Accounts**
  3. Click on **Create Service Account**

- ### Enter the service account details
  1. Enter a name for the service account, e.g. `Vercel`.

  2. Enter an ID for the service account, e.g. `vercel` and click **Create and continue**.

- ### Grant the service account access to the project
  1. Select a role or roles for the service account, e.g. `Storage Object Admin`.
  2. Click **Continue**.

- ### Grant users access to the service account
  1. Paste in the **IAM Principal** copied from the pool details page in the **Service account users role** field.
     - Replace `SUBJECT_ATTRIBUTE_VALUE` with `owner:[VERCEL_TEAM]:project:[PROJECT_NAME]:environment:[ENVIRONMENT]`. e.g. `principal://i
     - You can add multiple principals to this field, add a principal for each project and environment you want to grant access to.
  2. Click **Done**.

- ### Define GCP account values as environment variables
  Once you have configured your GCP project with OIDC access, gather the following values from the Google Cloud Console:
  | Value | Location | Environment Variable | Example |
  | ---------------------------------- | ---------------------------------------------------------------------- | ---------------------------
  | Project ID | IAM & Admin -> Settings | `GCP_PROJECT_ID` | `my-project-123456` |
  | Project Number | IAM & Admin -> Settings | `GCP_PROJECT_NUMBER` | `1234567890` |
  | Service Account Email | IAM & Admin -> Service Accounts | `GCP_SERVICE_ACCOUNT_EMAIL` | `vercel@my-project-123456.iam.gserviceaccount
  | Workload Identity Pool ID | IAM & Admin -> Workload Identity Federation -> Pools | `GCP_WORKLOAD_IDENTITY_POOL_ID` | `vercel` |
  | Workload Identity Pool Provider ID | IAM & Admin -> Workload Identity Federation -> Pools -> Providers | `GCP_WORKLOAD_IDENTITY_POOL_

  Then, [declare them as environment variables](/docs/environment-variables#creating-environment-variables) in your Vercel project.

  You are now ready to connect to your GCP resource from your project's code. Review the example below.

## Example

In the following example, you create a [Vercel function](/docs/functions/quickstart#create-a-vercel-function) in the Vercel project where

### Return GCP Vertex AI generated text

Install the following packages:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i google-auth-library @ai-sdk/google-vertex ai @vercel/functions
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i google-auth-library @ai-sdk/google-vertex ai @vercel/functions
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i google-auth-library @ai-sdk/google-vertex ai @vercel/functions
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i google-auth-library @ai-sdk/google-vertex ai @vercel/functions
    ```
  </Code>
</CodeBlock>

In the API route for this function, use the following code to perform the following tasks:

- Use `google-auth-library` to create an External Account Client
- Use it to authenticate with Google Cloud Services
- Use Vertex AI with [Google Vertex Provider](https://sdk.vercel.ai/providers/ai-sdk-providers/google-vertex) to generate text from a pro

```ts filename="/api/gcp-vertex-ai/route.ts"
import { getVercelOidcToken } from '@vercel/oidc';
import { ExternalAccountClient } from 'google-auth-library';
import { createVertex } from '@ai-sdk/google-vertex';
import { generateText } from 'ai';

const GCP_PROJECT_ID = process.env.GCP_PROJECT_ID;
const GCP_PROJECT_NUMBER = process.env.GCP_PROJECT_NUMBER;
const GCP_SERVICE_ACCOUNT_EMAIL = process.env.GCP_SERVICE_ACCOUNT_EMAIL;
const GCP_WORKLOAD_IDENTITY_POOL_ID = process.env.GCP_WORKLOAD_IDENTITY_POOL_ID;
const GCP_WORKLOAD_IDENTITY_POOL_PROVIDER_ID =
  process.env.GCP_WORKLOAD_IDENTITY_POOL_PROVIDER_ID;

// Initialize the External Account Client
const authClient = ExternalAccountClient.fromJSON({
  type: 'external_account',
  audience: `//iam.googleapis.com/projects/${GCP_PROJECT_NUMBER}/locations/global/workloadIdentityPools/${GCP_WORKLOAD_IDENTITY_POOL_ID}/
  subject_token_type: 'urn:ietf:params:oauth:token-type:jwt',
  token_url: 'https://sts.googleapis.com/v1/token',
  service_account_impersonation_url: `https://iamcredentials.googleapis.com/v1/projects/-/serviceAccounts/${GCP_SERVICE_ACCOUNT_EMAIL}:ge
  subject_token_supplier: {
    // Use the Vercel OIDC token as the subject token
    getSubjectToken: getVercelOidcToken,
  },
});

const vertex = createVertex({
  project: GCP_PROJECT_ID,
  location: 'us-central1',
  googleAuthOptions: {
    authClient,
    projectId: GCP_PROJECT_ID,
  },
});

// Export the route handler
export const GET = async (req: Request) => {
```

```
  const result = generateText({
    model: vertex('gemini-1.5-flash'),
    prompt: 'Write a vegetarian lasagna recipe for 4 people.',
  });
  return Response.json(result);
};
```

---
title: "OpenID Connect (OIDC) Federation"
description: "Secure the access to your backend using OIDC Federation to enable auto-generated, short-lived, and non-persistent credentia
last_updated: "2026-01-16T02:19:33.876Z"
source: "https://vercel.com/docs/oidc"
---

# OpenID Connect (OIDC) Federation

When you create long-lived, persistent credentials in your backend to allow access from your web applications, you increase the security

Cloud providers such as Amazon Web Services, Google Cloud Platform, and Microsoft Azure can trust these tokens and exchange them for shor

### Benefits

- **No persisted credentials**: There is no need to copy and paste long-lived access tokens
  from your cloud provider into your Vercel environment variables. Instead, you can exchange the OIDC token for short-lived
  access tokens with your trusted cloud provider
- **Granular access control**: You can configure your cloud providers to grant different permissions depending
  on project or environment. For instance, you can separate your development, preview and production environments on your cloud provider
  only grant Vercel issued OIDC tokens access to the necessary environment(s)
- **Local development access**: You can configure your cloud provider to trust local development environments so that long-lived credenti

## Getting started

To securely connect your deployment with your backend, configure your backend to trust Vercel's OIDC Identity Provider and connect to it

- [Connect to Amazon Web Services (AWS)](/docs/oidc/aws)
- [Connect to Google Cloud Platform (GCP)](/docs/oidc/gcp)
- [Connect to Microsoft Azure](/docs/oidc/azure)
- [Connect to your own API](/docs/oidc/api)

## Issuer mode

There are two options available configure the token's issuer URL (`iss`):

1. **Team** *(Recommended)*: The issuer URL is bespoke to your team e.g. `https://oidc.vercel.com/acme`.
2. **Global**: The issuer URL is generic e.g. `https://oidc.vercel.com`

To change the issuer mode:

- Open your project from the Vercel dashboard
- Select the Settings tab
- Navigate to Security
- From **Secure backend access with OIDC federation** section, toggle between **Team** or **Global** and click "Save".

## How OIDC token federation works

### In Builds

When you run a build, Vercel automatically generates a new token and assigns it to the `VERCEL_OIDC_TOKEN`
environment variable. You can then exchange the token for short-lived access tokens with your cloud provider.

### In Vercel Functions

When your application invokes a function, the OIDC token is set to the `x-vercel-oidc-token` header
on the function's `Request` object.

Vercel does not generate a fresh OIDC token for each execution but caches the token for a maximum of 45 minutes. While the token has a Ti

### In Local Development

You can download the `VERCEL_OIDC_TOKEN` straight to your local development environment using the CLI command
`vercel env pull`.

```bash filename="terminal"
vercel env pull
```

This writes the `VERCEL_OIDC_TOKEN` environment variable and other environment variables targeted
to `development` to the `.env.local` file of your project folder. See the [CLI docs](/docs/cli/env) for more information.

## Related

---
title: "OIDC Federation Reference"
description: "Review helper libraries to help you connect with your backend and understand the structure of an OIDC token."
last_updated: "2026-01-16T02:19:33.890Z"
source: "https://vercel.com/docs/oidc/reference"
---

# OIDC Federation Reference

## Helper libraries

Vercel provides helper libraries to make it easier to exchange the OIDC token for short-lived credentials with your cloud provider.
They are available from the [@vercel/oidc](https://www.npmjs.com/package/@vercel/oidc) and [@vercel/oidc-aws-credentials-provider](https:

### AWS SDK credentials provider

`awsCredentialsProvider()` is a helper function that returns a function that can be used as the `credentials` property of the AWS SDK client. It exchanges the OIDC token for short-lived credentials with AWS by calling the `AssumeRoleWithWebIdentity` operation.

#### AWS S3 usage example

```ts
import { awsCredentialsProvider } from '@vercel/oidc-aws-credentials-provider';
import * as s3 from '@aws-sdk/client-s3';

const s3client = new s3.S3Client({
  region: process.env.AWS_REGION!,
  credentials: awsCredentialsProvider({
    roleArn: process.env.AWS_ROLE_ARN!,
  }),
});
```

### Other cloud providers

`getVercelOidcToken()` returns the OIDC token from the `VERCEL_OIDC_TOKEN` environment variable in builds and local development environments or the `x-vercel-oidc-token` in Vercel functions.

#### Azure / CosmosDB example

```ts
import { getVercelOidcToken } from '@vercel/oidc';
import { ClientAssertionCredential } from '@azure/identity';
import { CosmosClient } from '@azure/cosmos';

const credentialsProvider = new ClientAssertionCredential(
  process.env.AZURE_TENANT_ID,
  process.env.AZURE_CLIENT_ID,
  getVercelOidcToken,
);

const cosmosClient = new CosmosClient({
  endpoint: process.env.COSMOS_DB_ENDPOINT,
  aadCredentials: credentialsProvider,
});
```

> **💡 Note:** In the Vercel function environments, you cannot execute the
> `getVercelOidcToken()` function directly at the module level because the token
> is only available in the `Request` object as the `x-vercel-oidc-token` header.

## Team and project name changes

If you change the name of your team or project, the claims within the OIDC token will reflect the new names. This can affect your trust and access control policies. You should consider this when you plan to rename your team or project and update your policies accordingly.

AWS roles can support multiple conditions so you can allow access to both the old and new team and project names. The following example s

```json filename="aws-trust-policy.json"
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::[YOUR AWS ACCOUNT ID]:oidc-provider/oidc.vercel.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "oidc.vercel.com:aud": [
            "https://vercel.com/[OLD_TEAM_SLUG]",
            "https://vercel.com/[NEW_TEAM_SLUG]"
          ],
          "oidc.vercel.com:sub": [
            "owner:[OLD_TEAM_SLUG]:project:[OLD_PROJECT_NAME]:environment:production",
            "owner:[NEW_TEAM_SLUG]:project:[NEW_PROJECT_NAME]:environment:production"
          ]
        }
      }
    }
  ]
}
```

If your project is using the `team` issuer mode, you will need to create a new OIDC provider and add another statement to the trust polic

```json filename="aws-trust-policy.json"
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "OldTeamName",
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::[YOUR AWS ACCOUNT ID]:oidc-provider/oidc.vercel.com/[OLD_TEAM_SLUG]"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "oidc.vercel.com/[OLD_TEAM_SLUG]:aud": [
            "https://vercel.com/[OLD_TEAM_SLUG]"
          ],
          "oidc.vercel.com/[OLD_TEAM_SLUG]:sub": [
```

```
            "owner:[OLD_TEAM_SLUG]:project:[OLD_PROJECT_NAME]:environment:production"
          ]
        }
      }
    },
    {
      "Sid": "NewTeamName",
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::[YOUR AWS ACCOUNT ID]:oidc-provider/oidc.vercel.com/[NEW_TEAM_SLUG]"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "oidc.vercel.com/[NEW_TEAM_SLUG]:aud": [
            "https://vercel.com/[NEW_TEAM_SLUG]"
          ],
          "oidc.vercel.com/[NEW_TEAM_SLUG]:sub": [
            "owner:[NEW_TEAM_SLUG]:project:[NEW_PROJECT_NAME]:environment:production"
          ]
        }
      }
    }
  ]
}
```

## OIDC token anatomy

You can validate OpenID Connect tokens by using the issuer's OpenID Connect Discovery Well Known location, which is either `https://oidc.v
provides a URI to Vercel's public JSON Web Keys (JWKs). You can use the corresponding JWK identified by `kid` to verify tokens
that are signed with the same `kid` in the token's header.

### Example token

```json
// Header:
{
  "typ": "JWT",
  "alg": "RS256",
  "kid": "example-key-id"
}
// Claims:
{
  "iss": "https://oidc.vercel.com/acme",
  "aud": "https://vercel.com/acme",
  "sub": "owner:acme:project:acme_website:environment:production",
  "iat": 1718885593,
  "nfb": 1718885593,
  "exp": 1718889193,
  "owner": "acme",
  "owner_id": "team_7Gw5ZMzpQA8h9OF832KGp7nwbuh3",
  "project": "acme_website",
  "project_id": "prj_7Gw5ZMBpQA8h9GF832KGp7nwbuh3",
  "environment": "production"
}
```

### Standard OpenID Connect claims

This is a list of standard tokens that you can expect from an OpenID Connect JWT:

| Claim | Kind | Description |
| --- | --- | --- |
| `iss` | Issuer | When using the **team** issuer mode, the issuer is set to `https://oidc.vercel.com/[TEAM_SLUG]`When using the **gl |
| `aud` | Audience | The audience is set to `https://vercel.com/[TEAM_SLUG]` |
| `sub` | Subject | The subject is set to `owner:[TEAM_SLUG]:project:[PROJECT_NAME]:environment:[ENVIRONMENT]` |
| `iat` | Issued at | The time the token was created |
| `nbf` | Not before | The token is not valid before this time |
| `exp` | Expires at | The time the token has or will expire. `preview` and `production` tokens expire one hour after creation, `developm |

### Additional claims

These claims provide more granular access control:

| Claim | Description |
| --- | --- |
| `owner` | The team slug, e.g. `acme` |
| `owner_id` | The team ID, e.g. `team_7Gw5ZMzpQA8h9OF832KGp7nwbuh3` |
| `project` | The project name, e.g. `acme_website` |
| `project_id` | The project ID, e.g. `prj_7Gw5ZMBpQA8h9GF832KGp7nwbuh3` |
| `environment` | The environment: `development` or `preview` or `production` |
| `user_id` | When environment is `development`, this is the ID of the user who was issued the token |

### JWT headers

These headers are standard to the JWT tokens:

| Header | Kind | Description |
| --- | --- | --- |
| `alg` | Algorithm | The algorithm used by the issuer |
| `kid` | Key ID | The identifier of the key used to sign the token |
| `typ` | Type | The type of token, this is set to `jwt`. |

---
title: "Open Source Program"
description: "Vercel provides platform credits, exclusive community support, and extra benefits for your open source project."
last_updated: "2026-01-16T02:19:33.896Z"
source: "https://vercel.com/docs/open-source-program"

---

# Open Source Program

Applications are now closed for the Spring 2025 cohort. **Summer cohort applications will open in July.**

The program opens applications on a seasonal basis. Each cohort is curated to include a small group of impactful, open source projects. I

> **💡 Note:** Applications are currently closed. They will reopen in July.

## Program Benefits

If selected, your open source project will receive:

- **Vercel credits**: $3,600 Vercel platform credits over 12 months
- **OSS starter pack**: Additional credits from third-party services to boost your project
- **Community support**: Get prioritized support and guidance from the Vercel team

## Who Should Apply?

To be considered for the Vercel OSS Program, projects must:

- Be an open source project that is actively being developed and maintained
- Be hosted on or intended to host on Vercel
- Show measurable impact or growth potential
- Follow a Code of Conduct ([example](https://github.com/vercel/vercel/blob/main/.github/CODE_OF_CONDUCT.md))
- Use credits exclusively for open source work and the project itself

## Frequently Asked Questions

**Does the program support nonprofits?**

Yes! If your nonprofit is fully open source, you're welcome to apply.

**What if I'm a startup?**

Startups with open source projects are eligible. You might also want to check out our Startups Program for additional benefits. [Learn mo

**Do you allow funded open-source companies to enter?**

We recommend applying for our [Startups Program](https://vercel.com/startups/credits) instead.

**How are applications evaluated?**

Applications are reviewed based on their impact, community engagement, and adherence to the criteria above. We look for projects that dem

**Can I apply if my project is just starting?**

Absolutely! We encourage applications from projects at all stages of development.

**Are Vercel Marketplace providers covered in credits?**

No. Vercel Marketplace providers can offer credits directly, separately from Vercel's open source program.

**What happens after 12 months?**

The program is designed to support projects as they grow. After 12 months, you graduate out of the program and we open it up to new appli

**Have any questions outside of these?**
Let us know in the [Vercel Community](https://community.vercel.com/c/open-source/45). We're happy to help!

> **💡 Note:** Applications are currently closed. They will reopen in July.

---
title: "Package Managers"
description: "Discover the package managers supported by Vercel for dependency management. Learn how Vercel detects and uses npm, Yarn, p
last_updated: "2026-01-16T02:19:33.906Z"
source: "https://vercel.com/docs/package-managers"
---

# Package Managers

Vercel will automatically detect the package manager used in your project and install the dependencies when you [create a deployment](/do

If you are using [Corepack](/docs/deployments/configure-a-build#corepack), Vercel will use the package manager specified in the `package.

## Supported package managers

The following table lists the package managers supported by Vercel, with their install commands and versions:

| Package Manager | Loc
| --- | ---
| Yarn | [`y
| npm | [`p
| pnpm | [`p
| Bun 1 | [`b
| Vlt | `vlt-lock.json` |

While Vercel automatically selects the package manager based on the lock file present in your project, the specific version of that packa

The npm and pnpm package managers create a `lockfileVersion` property when they generate a lock file. This property specifies the lock fi

| Package Manager | Condition | Install Command | Version Used |
| --- | --- | --- | --- |
| pnpm | `pnpm-lock.yaml`: present | `pnpm install` | Varies |
| | `lockfileVersion`: 9.0 | - | pnpm 9 or 10* |
| | `lockfileVersion`: 7.0 | - | pnpm 9 |
| | `lockfileVersion`: 6.0/6.1 | - | pnpm 8 |
| | `lockfileVersion`: 5.3/5.4 | - | pnpm 7 |

| | Otherwise | - | pnpm 6 |
| --- | --- | --- | --- |
| npm | `package-lock.json`: present | `npm install` | Varies |
| | `lockfileVersion`: 2 | - | npm 8 |
| | Node 20 | - | npm 10 |
| | Node 22 | - | npm 10 |
| Bun | `bun.lockb`: present | `bun install` | Bun <1.2 |
| | `bun.lock`: present | `bun install --save-text-lockfile` | Bun 1 |
| | `bun.lock`: present | `bun install` | Bun >=1.2 |
| Yarn | `yarn.lock`: present | `yarn install` | Yarn 1 |
| Vlt | `vlt-lock.json`: present | `vlt install` | Vlt 0.x |

> **💡 Note:** `pnpm-lock.yaml` version 9.0 can be generated by pnpm 9 or 10. Newer projects
> will prefer 10, while older prefer 9. Check [build
> logs](/docs/deployments/logs) to see which version is used for your project.

When no lock file exists, Vercel uses npm by default. Npm's default version aligns with the Node.js version as described in the table abo

## Manually specifying a package manager

You can manually specify a package manager to use on a per-project, or per-deployment basis.

### Project override

To specify a package manager for all deployments in your project, use the **Override** setting in your project's [**Build & Development S

1. Navigate to your [dashboard](/dashboard) and select your project
2. Select the **Settings** tab
3. From the left navigation, select **General**
4. Enable the **Override** toggle in the [**Build & Development Settings**](/docs/deployments/configure-a-build#build-and-development-set

> **💡 Note:** When using an override install command like
> , Vercel will use the oldest version of
> the specified package manager available in the build container. For example,
> if you specify  as your override install
> command, Vercel will use pnpm 6.

### Deployment override

To specify a package manager for a deployment, use the [`installCommand`](/docs/project-configuration#installcommand) property in your pr

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "installCommand": "pnpm install"
}
```

---
title: "Vercel Documentation"
description: "Vercel is the AI Cloud - a unified platform for building, deploying, and scaling AI-powered applications and agentic worklo
last_updated: "2026-01-16T02:19:33.919Z"
source: "https://vercel.com/docs"
---

# Vercel Documentation

Vercel is the AI Cloud for building and deploying modern web applications, from static sites to AI-powered agents.

## Get started with Vercel

You can build and host many different types of applications on Vercel, static sites with your favorite [framework](/docs/frameworks), [mu

You can also use the [Vercel Marketplace](/docs/integrations) to find and install integrations such as AI providers, databases, CMSs, ana

When you are ready to build, connect your [Git repository](/docs/git) to deploy on every push, with [automatic preview environments](/doc

See the [getting started guide](/docs/getting-started-with-vercel) for more information, or the [incremental migration guide](/docs/increm

## Quick references

## Build your applications

Use one or more of the following tools to build your application depending on your needs:

- **[Next.js](/docs/frameworks/nextjs)**: Build full-stack applications with Next.js, or any of our [supported frameworks](/docs/framewor
- **[Functions](/docs/functions)**: API routes with [Fluid compute](/docs/fluid-compute), [active CPU, and provisioned memory](/docs/func
- **[Routing Middleware](/docs/routing-middleware)**: Customize your application's behavior with code that runs before a request is proce
- **[Incremental Static Regeneration](/docs/incremental-static-regeneration)**: Automatically regenerate your pages on a schedule or when
- **[Image Optimization](/docs/image-optimization)**: Optimize your images for the web
- **[Manage environments](/docs/deployments/environments)**: Local, preview, production, and custom environments
- **[Feature flags](/docs/feature-flags)**: Control the visibility of features in your application

## Use Vercel's AI infrastructure

Add intelligence to your applications with Vercel's AI-first infrastructure:

- **[v0](https://v0.app/docs/introduction)**: Iterate on ideas with Vercel's AI-powered development assistant
- **[AI SDK](/docs/ai-sdk)**: Integrate language models with streaming and tool calling
- **[AI Gateway](/docs/ai-gateway)**: Route to any AI provider with automatic failover
- **[Agents](/kb/guide/how-to-build-ai-agents-with-vercel-and-the-ai-sdk)**: Build autonomous workflows and conversational interfaces
- **[MCP Servers](/docs/mcp)**: Create tools for AI agents to interact with your systems
- **[Sandbox](/docs/vercel-sandbox)**: Secure execution environments for untrusted code
- **[Claim deployments](/docs/deployments/claim-deployments)**: Allow AI agents to deploy a project and let a human take over

## Collaborate with your team

Collaborate with your team using the following tools:

- **[Toolbar](/docs/vercel-toolbar)**: An in-browser toolbar that lets you leave feedback, manage feature flags, preview drafts, edit con

- **[Comments](/docs/comments)**: Let teams and invited collaborators comment on your preview deployments and production environments
- **[Draft mode](/docs/draft-mode)**: View your unpublished headless CMS content on your site

## Secure your applications

Secure your applications with the following tools:

- **[Deployment Protection](/docs/deployment-protection)**: Protect your applications from unauthorized access
- **[RBAC](/docs/rbac)**: Role-based access control for your applications
- **[Configurable WAF](/docs/vercel-firewall/vercel-waf)**: Customizable rules to protect against attacks, scrapers, and unwanted traffic
- **[Bot Management](/docs/bot-management)**: Protect your applications from bots and automated traffic
- **[BotID](/docs/botid)**: An invisible CAPTCHA that protects against sophisticated bots without showing visible challenges or requiring
- **[AI bot filtering](/docs/bot-management#ai-bots-managed-ruleset)**: Control traffic from AI bots
- **[Platform DDoS Mitigation](/docs/security/ddos-mitigation)**: Protect your applications from DDoS attacks

## Deploy and scale

Vercel handles infrastructure automatically based on your framework and code, and provides the following tools to help you deploy and sca

- **[Vercel Delivery Network](/docs/cdn)**: Fast, globally distributed execution
- **[Rolling Releases](/docs/rolling-releases)**: Roll out new deployments in increments
- **[Rollback deployments](/docs/instant-rollback)**: Roll back to a previous deployment, for swift recovery from production incidents, l
- **[Observability suite](/docs/observability)**: Monitor performance and debug your AI workflows and apps


---
title: "Billing FAQ for Enterprise Plan"
description: "This page covers frequently asked questions around payments, invoices, and billing on the Enterprise plan."
last_updated: "2026-01-16T02:19:34.042Z"
source: "https://vercel.com/docs/plans/enterprise/billing"
---

# Billing FAQ for Enterprise Plan

The Vercel Enterprise plan is perfect for [teams](/docs/accounts/create-a-team) with increased performance, collaboration, and security n

## Payments

### When are payments taken?

- Pay by credit card: When the invoice is finalized in Stripe
- Pay by ACH/Wire: Due by due date on the invoice

### What payment methods are available?

- Credit card
- ACH/Wire

### What currency can I pay in?

You can pay in any currency so long as the credit card provider allows charging in USD *after* conversion.

### Can I delay my payment?

Contact your Customer Success Manager (CSM) or Account Executive (AE) if you feel payment might be delayed.

### Can I pay annually?

Yes.

### What card types can I pay with?

- American Express
- China UnionPay (CUP)
- Discover & Diners
- Japan Credit Bureau (JCB)
- Mastercard
- Visa

#### If paying by ACH, do I need to cover the payment fee cost on top of the payment?

Yes, when paying with ACH, the payment fee is often deducted by the sender. You need to add this fee to the amount you send, otherwise th

### Can I change my payment method?

Yes. You are free to remove your current payment method, so long as you have ACH payments set up. Once you have ACH payments set up, noti

## Invoices

### Can I pay by invoice?

- Yes. After checking the invoice, you can make a payment. You will receive a receipt after your credit card gets charged
- If you are paying with ACH, you will receive an email containing the bank account details you can wire the payment to
- If you are paying with ACH, you should provide the invoice number as a reference on the payment

### Why am I overdue?

Payment was not received from you by the invoice due date. This could be due to an issue with your credit card, like reaching your paymen

### Can I change an existing invoice detail?

No. Unless you provide specific justification to your Customer Success Manager (CSM) or Account Executive (AE). This addition will get ad

## Billing

### Is there a Billing role available?

Yes. Learn more about [Roles and Permissions](/docs/accounts/team-members-and-roles).

### How do I update my billing information?

- ### Go to the  page
  - Navigate to the [Dashboard](/dashboard)
  - Select your team from the scope selector on the top left as explained [here](/docs/teams-and-accounts/create-or-join-a-team#creating-
  - Select the **Settings** tab

- ### Go to the **Billing** section to update the appropriate fields
  Select **Billing** from the sidebar. Scroll down to find the following editable fields. You can update these if you are a [team owner](
  - **Invoice Email Recipient**: A custom destination email for your invoices. By default, they get sent to the first owner of the team
  - **Company Name**: The company name that shows up on your invoices. By default, it is set to your team name
  - **Billing Address**: A postal address added to every invoice. By default, it is blank
  - **Invoice Language**: The language of your invoices which is set to **English** by default
  - **Invoice Purchase Order**: A line that includes a purchase order on your invoices. By default, it is blank
  - **Tax ID**: A line for rendering a specific tax ID on your invoices. By default, it is blank
  > **💡 Note:** Your changes only affect future invoices, not existing ones.

### What do I do if I think my bill is wrong?

Please [open a support ticket](/help#issues) to log your request, which will allow our support team to look into the case for you.

When you contact support the following information will be needed:

- Invoice ID
- The account email
- The Team name
- If the query is related to the monthly plan, or usage billing

### Do I get billed for DDoS?

[Vercel automatically mitigates against L3, L4, and L7 DDoS attacks](/docs/security/ddos-mitigation) at the platform level for all plans.

Usage will be incurred for requests that are successfully served prior to us automatically mitigating the event. Usage will also be incur

For an additional layer of security, we recommend that you enable [Attack Challenge Mode](/docs/attack-challenge-mode) when you are under

You can monitor usage in the [Vercel Dashboard](/dashboard) under the **Usage** tab, although you will [receive notifications](/docs/noti

### What is a billing cycle?

The billing cycle refers to the period of time between invoices. The start date depends on when you created the account. You will be bill

---
title: "Using MIUs for AI Gateway and Vercel Agent"
description: "Learn how to use your MIU commitment to pay for AI Gateway and Vercel Agent."
last_updated: "2026-01-16T02:19:33.945Z"
source: "https://vercel.com/docs/plans/enterprise/buy-with-miu"
---

# Using MIUs for AI Gateway and Vercel Agent

For projects under the Enterprise plan, you can now use your existing [MIUs](/docs/pricing/understanding-my-invoice#managed-infrastructur

### Enabling Buy with MIUs

To enable buying AI with MIUs, go to your [team's Billing page](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fsettings%2Fbilling\&title=G

1. Go to your team dashboard and click **Settings**
2. Navigate to **Billing**
3. Under **Enterprise Plan**, find **AI Gateway** or **Vercel Agent** under **MIU Commitment**
4. Toggle **Buy with MIU** on for the product you want to enable it for
5. Review the dialog to confirm the conversion rate and click **Enable**. You can optionally set a maximum monthly MIU spend

> **💡 Note:** If your MIU credits include a discounted rate, the discount will not be
> applied when calculating the rate for this product.

When you toggle **Buy with MIU** on for a product, all future usage for that product category draws from your MIU balance. Whenever your

---
title: "Vercel Enterprise Plan"
description: "Learn about the Enterprise plan for Vercel, including features, pricing, and more."
last_updated: "2026-01-16T02:19:33.956Z"
source: "https://vercel.com/docs/plans/enterprise"
---

# Vercel Enterprise Plan

Vercel offers an Enterprise plan for organizations and enterprises that need high [performance](#performance-and-reliability), advanced [

## Performance and reliability

The Enterprise plan uses isolated build infrastructure on high-grade hardware with no queues to ensure exceptional performance and a seam

- Greater function limits for [Vercel Functions](/docs/functions/runtimes) including bundle size, duration, memory, and concurrency
- Automatic failover regions for [Vercel Functions](/docs/functions/configuring-functions/region#automatic-failover)
- Greater multi-region limits for [Vercel Functions](/docs/functions/configuring-functions/region#project-configuration)
- Vercel functions memory [configurable](/docs/functions/runtimes#size-limits) to 3009 MB
- Configurable [Vercel Function](/docs/functions) up to a [maximum duration](/docs/functions/runtimes#max-duration) of 900-seconds
- Unlimited [domains](/docs/domains) per project
- [Custom SSL Certificates](/docs/domains/custom-SSL-certificate)
- Automatic concurrency scaling up to 100,000 for [Vercel Functions](/docs/functions/concurrency-scaling#automatic-concurrency-scaling)
- [Isolated
  build infrastructure](/docs/security#do-enterprise-accounts-run-on-a-different-infrastructure),
  with the ability to have [larger memory and storage](/docs/deployments/troubleshoot-a-build#build-container-resources)
- [Trusted Proxy](/docs/headers/request-headers#x-forwarded-for)

## Security and compliance

Data and infrastructure security is paramount in the Enterprise plan with advanced features including:

- [SSO/SAML Login](/docs/saml)
- [Compliance measures](/docs/security)
- Access management for your deployments such as [Password Protection](/docs/security/deployment-protection/methods-to-protect-deployment
  [Private Production Deployments](/docs/security/deployment-protection#configuring-deployment-protection),
  and [Trusted IPs](/docs/security/deployment-protection/methods-to-protect-deployments/trusted-ips)
- [Secure Compute](/docs/secure-compute) (Paid add-on for Enterprise)
- [Directory Sync](/docs/security/directory-sync)
- [SIEM Integration](/docs/observability/audit-log#custom-siem-log-streaming) (Paid add-on for Enterprise)
- [Vercel Firewall](/docs/vercel-firewall), including [dedicated DDoS support](/docs/vercel-firewall/ddos-mitigation#dedicated-ddos-suppo

## Conformance and Code Owners

[Conformance](/docs/conformance) is a suite of tools designed for static code analysis. Conformance ensures high standards in performance

- [Allowlists](/docs/conformance/allowlist)
- [Curated rules](/docs/conformance/rules)
- [Custom rules](/docs/conformance/custom-rules)
- [Code Owners](/docs/code-owners) for GitHub

## Observability and Reporting

Gain actionable insights with enhanced observability & logging.

- Enhanced [Observability and Logging](/docs/observability)
- [Audit Logs](/docs/observability/audit-log)
- Increased retention with [Speed Insights](/docs/speed-insights/limits-and-pricing)
- [Custom Events](/docs/analytics/custom-events) tracking and more filters, such as UTM Parameters
- 3 days of [Runtime Logs](/docs/runtime-logs) and increased row data
- Increased retention with [Vercel Monitoring](/docs/observability/monitoring)
- [Tracing](/docs/tracing) support
- Configurable [drains](/docs/drains/using-drains)
- Integrations, like [Datadog](/integrations/datadog), [New Relic](/integrations/newrelic), and [Middleware](/integrations/middleware)

## Administration and Support

The Enterprise plan allows for streamlined team collaboration and offers robust support with:

- [Role-Based Access Control (RBAC)](/docs/rbac/access-roles)
- [Access Groups](/docs/rbac/access-groups)
- [Vercel Support Center](/docs/dashboard-features/support-center)
- A dedicated Success Manager
- [SLAs](https://vercel.com/legal/sla), including [response time](https://vercel.com/legal/support-terms)
- Audits for Next.js
- Professional services


---
title: "Vercel Hobby Plan"
description: "Learn about the Hobby plan and how it compares to the Pro plan."
last_updated: "2026-01-16T02:19:34.059Z"
source: "https://vercel.com/docs/plans/hobby"
---

# Vercel Hobby Plan

The Hobby plan is **free** and aimed at developers with personal projects, and small-scale applications. It offers a generous set of feat

| Resource | Hobby Included Usage |
| --- | --- |
| [Edge Config Reads](/docs/edge-config/using-edge-config#reading-data-from-edge-configs) | First 100,000 |
| [Edge Config Writes](/docs/edge-config/using-edge-config#writing-data-to-edge-configs) | First 100 |
| [Active CPU](/docs/functions/usage-and-pricing) | 4 CPU-hrs |
| [Provisioned Memory](/docs/functions/usage-and-pricing) | 360 GB-hrs |
| [Function Invocations](/docs/functions/usage-and-pricing) | First 1,000,000 |
| [Function Duration](/docs/functions/configuring-functions/duration) | First 100 GB-Hours |
| [Image Optimization Source Images](/docs/image-optimization/legacy-pricing#source-images) | First 1,000 |
| [Speed Insights Data Points](/docs/speed-insights/metrics#understanding-data-points) | First 10,000 |
| [Speed Insights Projects](/docs/speed-insights) | 1 Project |
| [Web Analytics Events](/docs/analytics/limits-and-pricing#what-is-an-event-in-vercel-web-analytics) | First 50,000 Events |

## Hobby billing cycle

As the Hobby plan is a free tier there are no billing cycles. In most cases, if you exceed your usage limits on the Hobby plan, you will

Some features have shorter or longer time periods:

- [Web Analytics](/docs/analytics/limits-and-pricing#hobby)

As stated in the [fair use guidelines](/docs/limits/fair-use-guidelines#commercial-usage), the Hobby plan restricts users to non-commerci

When your personal account gets converted to a Hobby team, your usage and activity log will be reset. To learn more about this change, re

## Comparing Hobby and Pro plans

The Pro plan offers more resources and advanced features compared to the Hobby plan. The following table provides a side-by-side comparis

| Feature | Hobby | Pro |
| --- | --- | --- |
| Active CPU | 4 CPU-hrs | 16 CPU-hrs |
| Provisioned Memory | 360 GB-hrs | 1440 GB-hrs |
| ISR Reads | Up to 1,000,000 Reads | 10,000,000 included |
| ISR Writes | Up to 200,000 | 2,000,000 included |
| Edge Requests | Up to 1,000,000 requests | 10,000,000 requests included |
| Projects | 200 | Unlimited |
| Vercel Function maximum duration | 10s (default) - [configurable up to 60s (1 minute)](/docs/functions/limitations#max-duration) | 15s |
| Build execution minutes | 6,000 | 24,000 |
| Team collaboration features | - | Yes |
| Domains per project | 50 | Unlimited |
| Deployments per day | 100 | 6,000 |

| Analytics | 50,000 included Events 1 month of data | 100,000 included Events 12 months of data Custom events |
| Email support | - | Yes |
| [Vercel AI Playground models](https://sdk.vercel.ai/) | Llama, GPT 3.5, Mixtral | GPT-4, Claude, Mistral Large, Code Llama |
| [RBAC](/docs/rbac/access-roles) available | N/A | [Owner](/docs/rbac/access-roles#owner-role), [Member](/docs/rbac/access-roles#member-
| [Comments](/docs/comments) | Available | Available for team collaboration |
| Log Drains | - | [Configurable](/docs/drains/using-drains) (not on a trial) |
| Spend Management | N/A | [Configurable](/docs/spend-management) |
| [Vercel Toolbar](/docs/vercel-toolbar) | Available for certain features | Available |
| [Storage](/docs/storage) | Blob (Beta) | Blob (Beta) |
| [Activity Logs](/docs/observability/activity-log) | Available | Available |
| [Runtime Logs](/docs/runtime-logs) | 1 hour of logs and up to 4000 rows of log data | 1 day of logs and up to 100,000 rows of log data
| [DDoS Mitigation](/docs/security/ddos-mitigation) | On by default. Optional [Attack Challenge Mode](/docs/attack-challenge-mode). | On
| [Vercel WAF IP Blocking](/docs/security/vercel-waf/ip-blocking) | Up to 10 | Up to 100 |
| [Vercel WAF Custom Rules](/docs/security/vercel-waf/custom-rules) | Up to 3 | Up to 40 |
| Deployment Protection | [Vercel Authentication](/docs/security/deployment-protection/methods-to-protect-deployments/vercel-authenticati
| [Deployment Retention](/docs/security/deployment-retention) | Unlimited by default. | Unlimited by default. |

## Upgrading to Pro

You can take advantage of Vercel's Pro trial to explore [Pro features](/docs/plans/pro-plan) for free during the trial period, with some

To upgrade from a Hobby plan:

1. Go to your [dashboard](/dashboard). If you're upgrading a team, make sure to select the team you want to upgrade
2. Go to the **Settings** tab and select **Billing**
3. Under **Plan**, if your team is eligible for an upgrade, you can click the **Upgrade** button. Or, you may need to create or select a
4. Optionally, add team members. Each member incurs a **$20 per user / month charge**
5. Enter your card details
6. Click **Confirm and Upgrade**

If you would like to end your paid plan, you can [downgrade to Hobby](/docs/plans/pro-plan#downgrading-to-hobby).

---
title: "Account Plans on Vercel"
description: "Learn about the different plans available on Vercel."
last_updated: "2026-01-16T02:19:34.020Z"
source: "https://vercel.com/docs/plans"
---

# Account Plans on Vercel

Vercel offers multiple account plans: Hobby, Pro, and Enterprise.

Each plan is designed to meet the needs of different types of users, from personal projects to large enterprises. The Hobby plan is free

## Hobby

The Hobby plan is designed for personal projects and developers. It includes CLI and personal [Git integrations](/docs/git), built-in CI/

It also provides base resources for [Vercel Functions](/docs/functions), [Middleware](/docs/routing-middleware), and [Image Optimization]

See the [Hobby plan](/docs/plans/hobby) page for more details.

## Pro

The Pro plan is designed for professional developers, freelancers, and businesses who need enhanced features and team collaboration. It i

Pro introduces a flexible credit-based system that provides transparent, usage-based billing. You get enhanced team collaboration with vi

Key features include team roles and permissions, credit-based resource management, enhanced monitoring, and email support with optional p

See the [Pro plan](/docs/plans/pro-plan) page for more details.

## Enterprise

The Enterprise plan caters to large organizations and enterprises requiring custom options, advanced security, and dedicated support. It

Enterprise customers benefit from [Single Sign-On (SSO)](/docs/saml), enhanced [observability and logging](/docs/observability), isolated

See the [Enterprise plan](/docs/plans/enterprise) page for more details.

## General billing information

### Where do I understand my usage?

On the [usage page of your dashboard](/dashboard). To learn how your usage relates to your bill and how to optimize your usage, see [Manag

You can also learn more about how [usage incurs on your site](/docs/pricing/how-does-vercel-calculate-usage-of-resources) or how to [unde

### What happens when I reach 100% usage?

All plans [receive notifications](/docs/notifications#on-demand-usage-notifications) by email and on the dashboard when they are approach

- Hobby plans will be paused when they exceed the included free tier usage
- Pro plans users can configure [Spend Management](/docs/spend-management) to automatically pause deployments, trigger a webhook, or send

For Pro and Enterprise teams, when you reach 100% usage your deployments are **not** automatically stopped. Rather, Vercel enables you to

One of the benefits to always being on, is that you don't have to worry about downtime in the event of a huge traffic spike caused by ann

See [Manage & optimize usage](/docs/pricing/manage-and-optimize-usage) for more information on how to optimize your usage.

---
title: "Billing FAQ for Pro Plan"
description: "This page covers frequently asked questions around payments, invoices, and billing on the Pro plan."
last_updated: "2026-01-16T02:19:34.050Z"
source: "https://vercel.com/docs/plans/pro-plan/billing"
---

# Billing FAQ for Pro Plan

The Vercel Pro plan is designed for professional developers, freelancers, and businesses who need enhanced features and team collaboratio

## Payments

### What is the price of the Pro plan?

See the [pricing page](/docs/pricing).

### When are payments taken?

At the beginning of each [billing cycle](#what-is-a-billing-cycle). Each invoice charges for the upcoming billing cycle. It includes any

### What payment methods are available?

Credit/Debit card only. Examples of invalid payment methods are gift cards, prepaid cards, EBT cards, and some virtual cards.

### What card types can I pay with?

- American Express
- China UnionPay (CUP)
- Discover & Diners
- Japan Credit Bureau (JCB)
- Mastercard
- Visa

### What currency can I pay in?

You can pay in any currency so long as the credit card provider allows charging in USD *after* conversion.

### What happens when I cannot pay?

When an account goes overdue, some account features are restricted until you make a payment. This means:

- You can't create new Projects
- You can't add new team members
- You can't redeploy existing projects

> **⚠️ Warning:** For subscription renewals, payment must be successfully made within 14 days,
> else all deployments on your account will be paused. For new subscriptions,
> the initial payment must be successfully made within 24 hours.

You can be overdue when:

- The card attached to the team expires
- The bank declined the payment
  - Possible incorrect card details
  - The card is reported lost or stolen
- There was no card on record or a payment method was removed

To fix, you can add a new payment method to bring your account back online.

### Can I delay my payment?

No, you cannot delay your payment.

### Can I pay annually?

No. Only monthly payments are supported. You can pay annually if you upgrade to an [Enterprise](/pricing) plan. The Enterprise plan offer

### Can I change my payment method?

Yes. You will have to add a new payment method before you can remove the old one. To do this:

1. From your dashboard, select your team in the **Scope selector**
2. Go to the **Settings** tab and select **Billing** from the left nav
3. Scroll to **Payment Method** and select the **Add new card** button

## Invoices

### Can I pay by invoice?

Yes. If you have a card on file, Vercel will charge it automatically. A receipt is then sent to you after your credit card gets charged.

- From your [dashboard](/docs/dashboard-features), go to the Team's page from the scope selector
- Select the **Settings** tab followed by the **Invoices** link on the left

If you do not have a card on file, then you will have to add a payment method, and you will receive a receipt of payment.

### Why am I overdue?

We were unable to charge your payment method for your latest invoice. This likely means that the payment was not successfully processed w

Some senders deduct a payment fee for transaction costs. This could mean that the amount charged on the invoice, does not reflect the amo

See [What happens when I cannot pay](#what-happens-when-i-cannot-pay) for more information.

### Can I change an existing invoice detail?

Invoice details must be accurate before adding a credit card at the end of a trial, **or prior to the upcoming invoice being finalized**.

Changes are reflected on future invoices **only**. Details on previous invoices will remain as they were issued and cannot be changed.

### Does Vercel possess and display their VAT ID on invoices?

No. Vercel is a US-based entity and does not have a VAT ID. If applicable, customers are encouraged to add their own VAT ID to their bill

### Can invoices be sent to my email?

Yes. By default, invoices are sent to the email address of the first [owner](/docs/accounts/team-members-and-roles/access-roles#owner-rol

1. From your [dashboard](/dashboard), navigate to the **Settings** tab
2. Select **Billing** from the sidebar
3. Scroll down to find the editable **Invoice Email Recipient** field

If you are having trouble receiving these emails, please review the spam settings of your email workspace as these emails may be getting

### Can I repay an invoice if I've used the wrong payment method?

No. Once an invoice is paid, it cannot be recharged with a different payment method, and refunds are not provided in these cases.

## Billing

### How are add-ons billed?

Pro add-ons are billed in the subsequent billing cycle as a line item on your invoice.

### What happens if I purchase an add-on by mistake?

[Open a support ticket](/help#issues) for your request and our team will assist you.

### What do I do if I think my bill is wrong?

Please [open a support ticket](/help#issues) and provide the following information:

- Invoice ID
- The account email
- The Team name
- If your query relates to the monthly plan, or usage billing

### Do I get billed for DDoS?

[Vercel automatically mitigates against L3, L4, and L7 DDoS attacks](/docs/security/ddos-mitigation) at the platform level for all plans.

Usage will be incurred for requests that are successfully served prior to us automatically mitigating the event. Usage will also be incur

For an additional layer of security, we recommend that you enable [Attack Challenge Mode](/docs/attack-challenge-mode) when you are under

You can monitor usage in the [Vercel Dashboard](/dashboard) under the **Usage** tab, although you will [receive notifications](/docs/noti

### What is a billing cycle?

The billing cycle refers to the period of time between invoices. The start date depends on when you created the account, or the account's

The second tab indicates the range of the billing cycle. During this period, you would get billed for:

- The amount of Team seats you have, and any addons you have purchased - Billed for the next 30 days of usage
- The usage consumed during the last billing cycle - Billed for the last 30 days of additional usage

You can't change a billing cycle or the dates on which you get billed. You can view the current billing cycle by going to the **Settings*

### What if my usage goes over the included credit?

You will be charged for on-demand usage, which is billed at the end of the month.

### What's the benefit of the credit-based model?

The monthly credit gives teams flexibility to allocate usage based on their actual workload, rather than being locked into rigid usage bu

## Access

### What can the Viewer seat do?

[Viewer seats](/docs/plans/pro-plan#viewer-team-seat) can:

- View and comment on deployments
- Access analytics and project insights

---
title: "Vercel Pro Plan"
description: "Learn about the Vercel Pro plan with credit-based billing, free viewer seats, and self-serve enterprise features for profes
last_updated: "2026-01-16T02:19:33.968Z"
source: "https://vercel.com/docs/plans/pro-plan"
---

# Vercel Pro Plan

The Vercel Pro plan is designed for professional developers, freelancers, and businesses who need enhanced features and team collaboratio

## Pro plan features

- **[Credit-based billing](#monthly-credit)**: Pro includes monthly credit that can be used flexibly across [usage dimensions](/docs/pric
- **[Free viewer seats](#viewer-team-seat)**: Unlimited read-only access to the Vercel dashboard so that project collaborators can view d
- **[Paid add-ons](#paid-add-ons)**: Additional enterprise-grade features are available as add-ons

For a full breakdown of the features included in the Pro plan, see the [pricing page](https://vercel.com/pricing).

## Monthly credit

You can use your monthly credit across all infrastructure resources. Once you have used your monthly credit, Vercel bills additional usag

The monthly credit applies to all [managed infrastructure billable resources](/docs/pricing#managed-infrastructure-billable-resources) af

### Credit and usage allocation

- **Monthly credit**: Every Pro plan has $20 in monthly credit.
- **Included infrastructure usage**: Each month, you have 1 TB [Fast Data Transfer](/docs/manage-cdn-usage#fast-data-transfer) and 10,000

### Credit expiration

The credit and allocations expire at the end of the month if they are not used, and are reset at the beginning of the following month.

### Managing your spend amount

You will receive automatic notifications when your usage has reached 75% of your monthly credit. Once you exceed the monthly credit, Verc

You can also set up alerts and automatic actions when your account hits a certain spend threshold as described in the [spend management d

> **💡 Note:** By default, Vercel enables spend management notifications for new customers at
> a spend amount of $200 per billing cycle.

## Pro plan pricing

The Pro plan is billed monthly based on the number of deploying team seats, paid add-ons, and any on-demand usage during the billing peri

### Platform fee

- $20/month Pro platform fee
  - 1 deploying team seat included
  - $20/month in usage credit

See the [pricing](/docs/pricing) page for more information about the pricing for resource usage.

## Team seats

On the Pro plan, your team starts with 1 included paid seat that can deploy projects, manage the team, and access all member-level permis

You can add (See the [Managing Team Members documentation](/docs/rbac/managing-team-members#adding-team-members-and-assigning-roles) for

- Additional paid seats ([Owner](/docs/rbac/access-roles#owner-role) or [Member](/docs/rbac/access-roles#member-role) roles) for $20/mont
- Unlimited free [Viewer seats](#viewer-team-seat) with read-only access

See the [Team Level Roles Reference](/docs/rbac/access-roles/team-level-roles) for a complete list of roles and their permissions.

### Viewer team seat

Each viewer team seat has the [Viewer Pro](/docs/rbac/access-roles#viewer-pro-role) role with the following access:

- Read-only access to Vercel to view analytics, speed insights, or access project deployments
- Ability to comment and collaborate on deployed previews

Viewers cannot configure or deploy projects.

### Additional team seats

- Seats with [Owner](/docs/rbac/access-roles#owner-role) or [Member](/docs/rbac/access-roles#member-role) roles: $20/month each
  - These team seats have the ability to configure & deploy projects
- [Viewer Pro](/docs/rbac/access-roles#viewer-pro-role) (read-only) seats: Free

## Paid add-ons

The following features are available as add-ons:

- **[SAML Single Sign-On](/docs/saml)**: $300/month
- **[HIPAA BAA](/docs/security/compliance#hipaa)**: Healthcare compliance agreements for $350/month
- **[Flags Explorer](/docs/feature-flags/flags-explorer)**: $250/month
- **[Observability Plus](/docs/observability/observability-plus)**: $10/month
- **[Web Analytics Plus](/docs/analytics/limits-and-pricing#pro-with-web-analytics-plus)**: $10/month
- **[Speed Insights](/docs/speed-insights)**: $10/month per project

## Downgrading to Hobby

Each account is limited to one team on the Hobby plan. If you attempt to downgrade a Pro team while already having a Hobby team, the plat

To downgrade from a Pro to Hobby plan without losing access to the team's projects:

1. Navigate to your [dashboard](/dashboard) and select your team from the [scope selector](/docs/dashboard-features#scope-selector)
2. Select the **Settings** tab
3. Select **Billing** in the Settings navigation
4. Click **Downgrade Plan** in the **Plan** sub-section

When you downgrade a Pro team, all active members except for the original owner are removed.

Due to restrictions in the downgrade flow, Pro teams will need to [manually transfer any connected Stores](/docs/storage#transferring-you

---
title: "Understanding Vercel"
description: "Learn all about Vercel"
last_updated: "2026-01-16T02:19:33.978Z"
source: "https://vercel.com/docs/plans/pro-plan/trials"
---

# Understanding Vercel

Vercel offers three plan tiers: **Hobby**, **Pro**, and **Enterprise**.

The Pro trial offers an opportunity to explore [Pro features](/docs/plans/pro-plan) for free during the trial period. There are some [lim

## Starting a trial

> **💡 Note:** There is a limit of one Pro plan trial per user account.

1. Select the [scope selector](/docs/dashboard-features#scope-selector) from the dashboard. From the bottom of the list select **Create T
2. Name your team

3. Select the **Pro Trial** option from the dialog. If this option does not appear, it means you have already reached your limit of one t

## Trial Limitations

The trial plan includes a $20 credit and follows the same [general limits](/docs/limits#general-limits) as a regular plan but with specif

| | Pro Trial Limits |
| --- | --- |
| Owner Members | 1 |
| Team Members (total, including Owners) | 10 |
| Projects | 200 |
| [Active CPU](/docs/functions/usage-and-pricing) | 8 CPU-hrs |
| [Provisioned Memory](/docs/functions/usage-and-pricing) | 720 GB-hrs |
| [Function Invocations](/docs/functions/usage-and-pricing) | 1,000,000/month |
| Build Execution | Max. 200 Hrs |
| [Image transformations](/docs/image-optimization/limits-and-pricing#image-transformations) | Max. 5K/month |
| [Image cache reads](/docs/image-optimization/limits-and-pricing#image-cache-reads) | Max. 300K/month |
| [Image cache writes](/docs/image-optimization/limits-and-pricing#image-cache-writes) | Max. 100K/month |
| [Monitoring](/docs/observability/monitoring) | Max. 125,000 metrics |
| Domains per Project | 50 |

To monitor the current usage of your Team's projects, see the [Usage](/docs/limits/usage) guide.

The following Pro features are **not available** on the trial:

- [Log drains](/docs/log-drains)
- [Account webhooks](/docs/webhooks#account-webhooks)
- Certain models (GPT-5 and Claude) on [Vercel AI Playground](https://sdk.vercel.ai/)

Once your usage of [Active CPU](/docs/functions/usage-and-pricing), [Provisioned Memory](/docs/functions/usage-and-pricing), or [Function

It is not possible to change Owners during the Pro trial period. Owners can be changed once the Pro trial has upgraded to a paid Pro plan

## Post-Trial Decision

Your trial finishes after 14 days or once your team exceeds the usage limits, whichever happens first. After which, you can opt for one o

- [Upgrade to a paid Pro plan](#upgrade-to-a-paid-pro-plan)
- [Revert to a Hobby plan](#revert-to-a-hobby-plan)

### Upgrade to a paid Pro plan

If you wish to continue on the Pro plan, you must add a payment method to ensure a seamless transition from the trial to the paid plan wh

To add a payment method, navigate to the Billings page through **Settings > Billing**. From this point, you will get billed according to

#### When will I get billed?

Billing begins immediately after your trial ends if you have added a payment method.

### Revert to a Hobby plan

Without a payment method, your account reverts to a Hobby plan when the trial ends. Alternatively, you can use the **Downgrade** button l

> **💡 Note:** Charges apply only if you have a payment method. If a trial finishes and you
> haven't set payment method, you will  get charged.

You can upgrade to a Pro plan anytime later by visiting **Settings > Billing** and adding a payment method.

### Downgraded to Hobby

If your Pro trial account gets downgraded to a Hobby team, you can revert this by **upgrading to Pro**. If you've transferred out the pro

When you upgrade to Pro, the pause status on your account will get lifted. This reinstates:

- **Full access** to all previous projects and deployments
- Access to the increased limits and features of a Pro account

#### What if I resume using Vercel months after my trial ends?

No charges apply for the months of inactivity. Billing will only cover the current billing cycle.

---
title: "Postgres on Vercel"
description: "Learn how to use Postgres databases through the Vercel Marketplace."
last_updated: "2026-01-16T02:19:33.990Z"
source: "https://vercel.com/docs/postgres"
---

# Postgres on Vercel

Vercel lets you connect external Postgres databases through the [Marketplace](/marketplace), allowing you to connect external Postgres da

> **💡 Note:** Vercel Postgres is no longer available. If you had an existing Vercel Postgres database, we automatically moved it to [Neo

- Explore [Marketplace storage postgres integrations](/marketplace?category=storage\&search=postgres).
- Learn how to [add a Marketplace native integration](/docs/integrations/install-an-integration/product-integration).

## Connecting to the Marketplace

Vercel enables you to use Postgres by integrating with external database providers. By using the Marketplace, you can:

- Select from a [range of Postgres providers](/marketplace?category=storage\&search=postgres)
- Provision and configure a Postgres database with minimal setup.
- Have credentials and [environment variables](/docs/environment-variables) injected into your Vercel project.

---
title: "Calculating usage of resources"

description: "Understand how Vercel measures and calculates your resource usage based on a typical user journey."
last_updated: "2026-01-16T02:19:34.009Z"
source: "https://vercel.com/docs/pricing/how-does-vercel-calculate-usage-of-resources"
---

# Calculating usage of resources

It's important to understand how usage and accrual happen on Vercel, in order to make the best choices for your project. This guide helps

You'll learn how resources are used at each stage of the journey, from entering the site, to browsing products, interacting with dynamic

## Understanding Vercel resources

> **💡 Note:** The scenarios and resource usage described in this guide are for illustrative
> purposes only.

Usage is accrued as users visit your site. Vercel's framework-defined infrastructure determines how your site renders and how your costs

A typical user journey through an ecommerce store touches on multiple resources used in Vercel's [managed infrastructure](/docs/pricing#m

The ecommerce store employs a combination of caching strategies to optimize both static and dynamic content delivery. For static pages, i

For dynamic content like product price discounts, the site uses [Vercel Data Cache](/docs/infrastructure/data-cache) to store and retriev

For dynamic, user-specific content like shopping cart states, a Redis database from the [Vercel Marketplace](/marketplace?category=storag

The site also uses [Middleware](/docs/routing-middleware) to A/B test a product carousel, showing different variants to different users b

The following sections outline the resources used at each stage of the user journey.

### 1. User enters the site

The browser requests the page from Vercel. Since it's static and cached on our global [CDN](/docs/cdn), this only involves [Edge Requests

**Priced resources**

- :
  Charged per network request to the CDN
- : Charged based on data moved to the user from the CDN

### 2. Product browsing

During the user's visit to the site, they browse the **All Products** page, which is populated with a list of cached product images and p

**Priced resources**

- :
  Charged for network requests to fetch product images/details
- : Data movement charges from CDN to the user

### 3. Viewing updated product details

The user decides to view the details of a product. This products price was recently updated and the first view of the page shows the stal

Behind the scenes the site uses [Incremental Static Regeneration (ISR)](/docs/incremental-static-regeneration) to update the products des

For products with real-time discounts, these discounts are calculated using a [Vercel Function](/docs/functions) that fetches the latest

Upon viewing a product, if the discount data is already in the Data Cache and still fresh, it will be served from there. If the data is s

**Priced resources**

- :
  Network request charges for fetching updated product information
- : Charges for activating a function to update content
- : CPU runtime charges for the function processing the update

### 4. Dynamic interactions (Cart)

The user decides to add a product to their cart. The cart is a dynamic feature that requires real-time updates. When the user adds an ite

**Priced resources**

- :
  Network request charges for cart updates
- : Function activation charges for managing cart logic
- : CPU runtime charges for the function processing the cart logic
- : Data movement charges for fetching cart state from the cache
- Redis Requests: Charges for reading and writing cart state to the Redis store
- Redis Storage: Charges for storing cart state in the Redis store
- Redis Data Transfer: Data movement charges for fetching cart state from the
  Redis store

### 5. Engaging with A/B testing for personalized content

Having added an item to the cart, the user decides to continue browsing the site. They scroll to the bottom of the page and are shown a p

**Priced resources**

- :
  Network request charges for delivering test variants

## Summary and next steps

Throughout the user journey through the site, a variety of resources are used from Vercel's [managed infrastructure](/docs/pricing#manage

To learn more about each of the resources used in this guide, see the [managed infrastructure billable resources](/docs/pricing#managed-i

## More resources

For more information on Vercel's pricing, guidance on optimizing consumption, and invoices, see the following resources:

- [Learn about Vercel's pricing model and how it works](/docs/pricing)
- [Learn how Vercel usage is calculated and how it accrues](/docs/pricing/manage-and-optimize-usage)
- [Learn how to understand your Vercel invoice](/docs/pricing/understanding-my-invoice)

---
title: "Legacy Metrics"
description: "Learn about Bandwidth, Requests, Vercel Function Invocations, and Vercel Function Execution metrics."
last_updated: "2026-01-16T02:19:34.067Z"
source: "https://vercel.com/docs/pricing/legacy"
---

# Legacy Metrics

## Bandwidth

Bandwidth is the amount of data your deployments have sent or received.
This chart includes traffic for both [preview](/docs/deployments/environments#preview-environment-pre-production) and
[production](/docs/deployments/environments#production-environment) deployments.

> **💡 Note:** You are not billed for bandwidth usage on [blocked or
> paused](/kb/guide/why-is-my-account-deployment-blocked#pausing-process)
> deployments.

The total traffic of your projects is the sum of the outgoing and incoming bandwidth.

- **Outgoing**: Outgoing bandwidth measures the amount of data that your deployments have **sent** to your users.
  Data used by [ISR](/docs/incremental-static-regeneration) and the responses from the [CDN](/docs/cdn) and [Vercel functions](/docs/func
- **Incoming**: Incoming bandwidth measures the amount of data that your deployments have **received** from your users

An example of incoming bandwidth would be page views requested by the browser. All requests sent to the [CDN](/docs/cdn) and [Vercel func

Incoming bandwidth is usually much smaller than outgoing bandwidth for website projects.

## Requests

Requests are the number of requests made to your deployments. This chart includes traffic for both [preview](/docs/deployments/environmen

Requests can be filtered by:

- **Ratio**: The ratio of requests that are cached and uncached by the [CDN](/docs/cdn)
- **Projects**: The projects that the requests are made to

## Vercel Function Invocations

Vercel Function Invocations are the number of times your [Vercel functions](/docs/functions) have receive a request, excluding cache hits

Vercel Function Invocations can be filtered by:

- **Ratio**: The ratio of invocations that are **Successful**, **Errored**, or **Timed out**
- **Projects**: The projects that the invocations are made to

## Vercel Function Execution

Vercel Function Execution is the amount of time your [Vercel functions](/docs/functions) have spent computing resources.

Vercel Function Execution can be filtered by:

- **Ratio**: The ratio of execution time that is **Completed**, **Errored**, or **Timed out**
- **Projects**: The projects that the execution time is spent on

---
title: "Manage and optimize usage"
description: "Understand how to manage and optimize your usage on Vercel, learn how to track your usage, set up alerts, and optimize your
last_updated: "2026-01-16T02:19:34.082Z"
source: "https://vercel.com/docs/pricing/manage-and-optimize-usage"
---

# Manage and optimize usage

## What pricing plan am I on?

There are three plans on Vercel: Hobby, Pro, and Enterprise. To see which plan you are on, select your team from the [scope selector](/do

## Viewing usage

The Usage page shows the usage of all projects in your Vercel account by default. To access it, select the **Usage** tab from your Vercel

To use the usage page:

1. To investigate the usage of a specific team, use the scope selector to select your team
2. From your dashboard, select the **Usage** tab
3. We recommend you look at usage over the last 30 days to determine patterns. Change the billing cycle dropdown under Usage to **Last 30
4. You can choose to view the usage of a particular project by selecting it from the dropdown
5. In the overview, you'll see an allotment indicator. It shows how much of your usage you've consumed in the current cycle and the proje
6. Use the [**Top Paths**](/docs/manage-cdn-usage#top-paths) chart to understand the metrics causing the high usage

## Usage alerts, notification, and spend management

The usage dashboard helps you understand and project your usage. You can also set up alerts to notify you when you're approaching usage l

- **Spend Management**: Spend management is an opt-in feature. Pro teams can set up a spend amount for your team to trigger notifications
- **Usage Notifications**: Usage notifications are set up automatically. Pro teams can also [configure the threshold](/docs/notifications

## Networking

The table below shows the metrics for the [Networking](/docs/pricing/networking) section of the **Usage** dashboard.

To view information on managing each resource, select the resource link in the **Metric** column. To jump straight to guidance on optimiz

## Serverless Functions

The table below shows the metrics for the [**Serverless Functions**](/docs/pricing/serverless-functions) section of the **Usage** dashboa

To view information on managing each resource, select the resource link in the **Metric** column. To jump straight to guidance on optimiz

## Builds

The table below shows the metrics for the [**Builds**](/docs/builds/managing-builds) section of the **Usage** dashboard.

To view information on managing each resource, select the resource link in the **Metric** column. To jump straight to guidance on optimiz

## Artifacts

The table below shows the metrics for the [**Remote Cache Artifacts**](/docs/monorepos/remote-caching#artifacts) section of the **Usage**

To view information on managing each resource, select the resource link in the **Metric** column. To jump straight to guidance on optimiz

## Edge Config

The table below shows the metrics for the [**Edge Config**](/docs/pricing/edge-config) section of the **Usage** dashboard.

To view information on managing each resource, select the resource link in the **Metric** column. To jump straight to guidance on optimiz

## Data Cache

The table below shows the metrics for the [**Data Cache**](/docs/data-cache) section of the **Usage** dashboard.

To view information on managing each resource, select the resource link in the **Metric** column. To jump straight to guidance on optimiz

## Incremental Static Regeneration (ISR)

The table below shows the metrics for the [**Incremental Static Regeneration**](/docs/pricing/incremental-static-regeneration) section of

To view information on managing each resource, select the resource link in the **Metric** column. To jump straight to guidance on optimiz

## Observability

The table below shows the metrics for the [Web Analytics](/docs/pricing/observability#managing-web-analytics-events), [Speed Insights](/d

To view information on managing each resource, select the resource link in the **Metric** column. To jump straight to guidance on optimiz

## Image Optimization

The table below shows the metrics for the [**Image Optimization**](/docs/image-optimization/managing-image-optimization-costs) section of

To view information on managing each resource, select the resource link in the **Metric** column. To jump straight to guidance on optimiz

## Viewing Options

### Count

Count shows the **total** number of a certain metric, across all projects in your account. This is useful to understand past trends about

### Project

Project shows the total usage of a certain metric, per project. This is useful to understand how different projects are using resources a

### Region

For region-based pricing, you can view the usage of a certain metric, per region. This is useful to understand the requests your site is

### Ratio

- **Requests**: The ratio of cached vs uncached requests
- **Fast Data Transfer**: The ratio of incoming vs outgoing data transfer
- **Fast Origin Transfer**: The ratio of incoming vs outgoing data transfer
- **Serverless Functions invocations**: Successful vs errored vs timed out invocations
- **Serverless Functions execution**: Successful vs errored vs timed out invocations
- **Builds**: Completed vs errored builds
- **Remote Cache Artifacts**: Uploaded vs downloaded artifacts
- **Remote Cache total size**: Uploaded vs downloaded artifacts

### Average

This shows the average usage of a certain metric over a 24 hour period.

## More resources

For more information on Vercel's pricing, guidance on optimizing consumption, and invoices, see the following
resources:

- [How are resources used on Vercel?](/docs/pricing/how-does-vercel-calculate-usage-of-resources)
- [Understanding my invoice](/docs/pricing/understanding-my-invoice)

---
title: "Pricing on Vercel"
description: "Learn about Vercel"
last_updated: "2026-01-16T02:19:34.090Z"
source: "https://vercel.com/docs/pricing"
---

# Pricing on Vercel

This page provides an overview of Vercel's pricing model and outlines all billable metrics and their pricing models.

For a full breakdown of Vercel's pricing by plan, see the .

To learn how resources are triggered through a real-world app scenario, see the [calculating resource usage](/docs/pricing/how-does-verce

## Managed Infrastructure

Vercel provides  to deploy, scale, and secure your applications.

These resources are usage based, and billed based on the amount of data transferred, the number of requests made, and the duration of com

Each product's usage breaks down into resources, with each one billed based on the usage of a specific metric. For example, [Function Dur

### Managed Infrastructure billable resources

Most resources include an amount of usage your projects can use within your billing cycle. If you exceed the included amount, you are cha

See the following pages for more information on the pricing of each managed infrastructure resource:

- [Vercel Functions](/docs/functions/usage-and-pricing)
- [Image Optimization](/docs/image-optimization/limits-and-pricing)
- [Edge Config](/docs/edge-config/edge-config-limits)
- [Web Analytics](/docs/analytics/limits-and-pricing)
- [Speed Insights](/docs/speed-insights/limits-and-pricing)
- [Drains](/docs/drains#usage-and-pricing)
- [Monitoring](/docs/monitoring/limits-and-pricing)
- [Observability](/docs/observability/limits-and-pricing)
- [Blob](/docs/vercel-blob/usage-and-pricing)
- [Microfrontends](/docs/microfrontends#limits-and-pricing)
- [Bulk redirects](/docs/redirects/bulk-redirects#limits-and-pricing)

For [Enterprise](/docs/plans/enterprise) pricing, contact our [sales team](/contact/sales).

#### Pro plan add-ons

To enable any of the Pro plan add-ons:

1. Visit the Vercel [dashboard](/dashboard) and select your team from the [scope selector](/docs/dashboard-features#scope-selector).
2. Select the **Settings** tab and go to Billing.
3. In the **Add-Ons** section, find the add-on you'd like to add. Switch the toggle to **Enabled** and configure the add-on as necessary.

#### Regional pricing

See the [regional pricing](/docs/pricing/regional-pricing) page for more information on Managed Infrastructure pricing in different regio

## Developer Experience Platform

Vercel's Developer Experience Platform  offers a monthly billed suite of tools and services focused on building, deploying, and optimizin

### DX Platform billable resources

The below table lists the billable DX Platform resources for the Pro plan. These resources are not usage based, and are billed at a fixed

## More resources

For more information on Vercel's pricing, guidance on optimizing consumption, and invoices, see the following resources:

- [How are resources used on Vercel?](/docs/pricing/how-does-vercel-calculate-usage-of-resources)
- [Manage and optimize usage](/docs/pricing/manage-and-optimize-usage)
- [Understanding my invoice](/docs/pricing/understanding-my-invoice)
- [Improved infrastructure pricing](/blog/improved-infrastructure-pricing)
- [Regional pricing](/docs/pricing/regional-pricing)

---
title: "Stockholm, Sweden (arn1) pricing"
description: "Vercel pricing for the Stockholm, Sweden (arn1) region."
last_updated: "2026-01-16T02:19:34.121Z"
source: "https://vercel.com/docs/pricing/regional-pricing/arn1"
---

# Stockholm, Sweden (arn1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "Mumbai, India (bom1) pricing"
description: "Vercel pricing for the Mumbai, India (bom1) region."
last_updated: "2026-01-16T02:19:34.118Z"
source: "https://vercel.com/docs/pricing/regional-pricing/bom1"
---

# Mumbai, India (bom1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid

> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "Paris, France (cdg1) pricing"
description: "Vercel pricing for the Paris, France (cdg1) region."
last_updated: "2026-01-16T02:19:34.124Z"
source: "https://vercel.com/docs/pricing/regional-pricing/cdg1"
---

# Paris, France (cdg1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "Cleveland, USA (cle1) pricing"
description: "Vercel pricing for the Cleveland, USA (cle1) region."
last_updated: "2026-01-16T02:19:34.128Z"
source: "https://vercel.com/docs/pricing/regional-pricing/cle1"
---

# Cleveland, USA (cle1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "Cape Town, South Africa (cpt1) pricing"
description: "Vercel pricing for the Cape Town, South Africa (cpt1) region."
last_updated: "2026-01-16T02:19:34.131Z"
source: "https://vercel.com/docs/pricing/regional-pricing/cpt1"
---

# Cape Town, South Africa (cpt1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "Dublin, Ireland (dub1) pricing"
description: "Vercel pricing for the Dublin, Ireland (dub1) region."
last_updated: "2026-01-16T02:19:34.134Z"
source: "https://vercel.com/docs/pricing/regional-pricing/dub1"
---

# Dublin, Ireland (dub1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "Dubai, United Arab Emirates (dxb1) pricing"
description: "Vercel pricing for the Dubai, UAE (dxb1) region."
last_updated: "2026-01-16T02:19:34.112Z"
source: "https://vercel.com/docs/pricing/regional-pricing/dxb1"
---

# Dubai, United Arab Emirates (dxb1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "Frankfurt, Germany (fra1) pricing"
description: "Vercel pricing for the Frankfurt, Germany (fra1) region."
last_updated: "2026-01-16T02:19:34.139Z"
source: "https://vercel.com/docs/pricing/regional-pricing/fra1"
---

# Frankfurt, Germany (fra1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "São Paulo, Brazil (gru1) pricing"
description: "Vercel pricing for the São Paulo, Brazil (gru1) region."
last_updated: "2026-01-16T02:19:34.143Z"
source: "https://vercel.com/docs/pricing/regional-pricing/gru1"
---

# São Paulo, Brazil (gru1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "Hong Kong (hkg1) pricing"
description: "Vercel pricing for the Hong Kong (hkg1) region."
last_updated: "2026-01-16T02:19:34.147Z"
source: "https://vercel.com/docs/pricing/regional-pricing/hkg1"
---

# Hong Kong (hkg1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "Tokyo, Japan (hnd1) pricing"
description: "Vercel pricing for the Tokyo, Japan (hnd1) region."
last_updated: "2026-01-16T02:19:34.151Z"
source: "https://vercel.com/docs/pricing/regional-pricing/hnd1"
---

# Tokyo, Japan (hnd1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "Washington, D.C., USA (iad1) pricing"
description: "Vercel pricing for the Washington, D.C., USA (iad1) region."
last_updated: "2026-01-16T02:19:34.155Z"
source: "https://vercel.com/docs/pricing/regional-pricing/iad1"
---

# Washington, D.C., USA (iad1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "Seoul, South Korea (icn1) pricing"
description: "Vercel pricing for the Seoul, South Korea (icn1) region."
last_updated: "2026-01-16T02:19:34.159Z"
source: "https://vercel.com/docs/pricing/regional-pricing/icn1"
---

# Seoul, South Korea (icn1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "Osaka, Japan (kix1) pricing"
description: "Vercel pricing for the Osaka, Japan (kix1) region."
last_updated: "2026-01-16T02:19:34.163Z"
source: "https://vercel.com/docs/pricing/regional-pricing/kix1"
---

# Osaka, Japan (kix1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "London, UK (lhr1) pricing"
description: "Vercel pricing for the London, UK (lhr1) region."
last_updated: "2026-01-16T02:19:34.167Z"
source: "https://vercel.com/docs/pricing/regional-pricing/lhr1"
---

# London, UK (lhr1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "Regional pricing"
description: "Vercel pricing for Managed Infrastructure resources in different regions."
last_updated: "2026-01-16T02:19:34.188Z"
source: "https://vercel.com/docs/pricing/regional-pricing"
---

# Regional pricing

When using Managed Infrastructure resources on Vercel, some, but not all, are priced based on region. The following table shows the price

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

| Resource | Included (Billing Cycle) | On-demand (Billing Cycle) |
| --- | --- | --- |
| [Fast Data Transfer](/docs/manage-cdn-usage#fast-data-transfer) | First 1 TB | 1 GB for  -  |
| [Edge Requests](/docs/manage-cdn-usage#edge-requests) | First 10,000,000 | 1,000,000 Requests for  -  |

| Resource | On-demand (Billing Cycle) |
| --- | --- |
| [ISR Writes](/docs/incremental-static-regeneration/limits-and-pricing#isr-writes-chart) | 1,000,000 Write Units for  - |
| [ISR Reads](/docs/incremental-static-regeneration/limits-and-pricing#isr-reads-chart) | 1,000,000 Read Units for  - |

| [Fast Origin Transfer](/docs/manage-cdn-usage#fast-origin-transfer) | 1 GB for  - |
| [Edge Request Additional CPU Duration](/docs/manage-cdn-usage#edge-request-cpu-duration) | 1 Hour for  - |
| [Image Optimization Transformations](/docs/image-optimization/limits-and-pricing#image-transformations) |  -  per 1K |
| [Image Optimization Cache Reads](/docs/image-optimization/limits-and-pricing#image-cache-reads) |  -  per 1M |
| [Image Optimization Cache Writes](/docs/image-optimization/limits-and-pricing#image-cache-writes) |  -  per 1M |
| [Runtime Cache Writes](/docs/functions/functions-api-reference/vercel-functions-package#getcache) | 1,000,000 Write Units for  - |
| [Runtime Cache Reads](/docs/functions/functions-api-reference/vercel-functions-package#getcache) | 1,000,000 Read Units for  - |
| [WAF Rate Limiting](/docs/security/vercel-waf/usage-and-pricing#rate-limiting-pricing) | 1,000,000 Allowed Requests fo |
| [OWASP CRS per request number](/docs/security/vercel-waf/usage-and-pricing#managed-ruleset-pricing) | 1,000,000 Inspected Requests |
| [OWASP CRS per request size](/docs/security/vercel-waf/usage-and-pricing#managed-ruleset-pricing) | 1 GB of inspected request pay |
| [Blob Storage Size](/docs/vercel-blob/usage-and-pricing#pricing) | 1 GB for  - |
| [Blob Simple Operations](/docs/vercel-blob/usage-and-pricing#pricing) | 1,000,000 for  - |
| [Blob Advanced Operations](/docs/vercel-blob/usage-and-pricing#pricing) | 1,000,000 for  - |
| [Blob Data Transfer](/docs/vercel-blob/usage-and-pricing#pricing) | 1 GB for  - |
| [Private Data Transfer](/docs/connectivity/static-ips) | 1 GB for  - |

## Specific region pricing

For specific, region based pricing, see the following pages:

- [Cape Town, South Africa (cpt1)](/docs/pricing/regional-pricing/cpt1)
- [Cleveland, USA (cle1)](/docs/pricing/regional-pricing/cle1)
- [Dubai, UAE (dxb1)](/docs/pricing/regional-pricing/dxb1)
- [Dublin, Ireland (dub1)](/docs/pricing/regional-pricing/dub1)
- [Frankfurt, Germany (fra1)](/docs/pricing/regional-pricing/fra1)
- [Hong Kong (hkg1)](/docs/pricing/regional-pricing/hkg1)
- [London, UK (lhr1)](/docs/pricing/regional-pricing/lhr1)
- [Mumbai, India (bom1)](/docs/pricing/regional-pricing/bom1)
- [Osaka, Japan (kix1)](/docs/pricing/regional-pricing/kix1)
- [Paris, France (cdg1)](/docs/pricing/regional-pricing/cdg1)
- [Portland, USA (pdx1)](/docs/pricing/regional-pricing/pdx1)
- [San Francisco, USA (sfo1)](/docs/pricing/regional-pricing/sfo1)
- [Seoul, South Korea (icn1)](/docs/pricing/regional-pricing/icn1)
- [Singapore (sin1)](/docs/pricing/regional-pricing/sin1)
- [Stockholm, Sweden (arn1)](/docs/pricing/regional-pricing/arn1)
- [Sydney, Australia (syd1)](/docs/pricing/regional-pricing/syd1)
- [São Paulo, Brazil (gru1)](/docs/pricing/regional-pricing/gru1)
- [Tokyo, Japan (hnd1)](/docs/pricing/regional-pricing/hnd1)
- [Washington, D.C. USA (iad1)](/docs/pricing/regional-pricing/iad1)
- [Montréal, Canada (yul1)](/docs/pricing/regional-pricing/yul1)

For more information on Managed Infrastructure pricing, see the [pricing documentation](/docs/pricing#managed-infrastructure).

---
title: "Portland, USA (pdx1) pricing"
description: "Vercel pricing for the Portland, USA (pdx1) region."
last_updated: "2026-01-16T02:19:34.175Z"
source: "https://vercel.com/docs/pricing/regional-pricing/pdx1"
---

# Portland, USA (pdx1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "San Francisco, USA (sfo1) pricing"
description: "Vercel pricing for the San Francisco, USA (sfo1) region."
last_updated: "2026-01-16T02:19:34.194Z"
source: "https://vercel.com/docs/pricing/regional-pricing/sfo1"
---

# San Francisco, USA (sfo1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "Singapore (sin1) pricing"
description: "Vercel pricing for the Singapore (sin1) region."
last_updated: "2026-01-16T02:19:34.198Z"
source: "https://vercel.com/docs/pricing/regional-pricing/sin1"
---

# Singapore (sin1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on

> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "Sydney, Australia (syd1) pricing"
description: "Vercel pricing for the Sydney, Australia (syd1) region."
last_updated: "2026-01-16T02:19:34.202Z"
source: "https://vercel.com/docs/pricing/regional-pricing/syd1"
---

# Sydney, Australia (syd1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "Montréal, Canada (yul1) pricing"
description: "Vercel pricing for the Montréal, Canada (yul1) region."
last_updated: "2026-01-16T02:19:34.206Z"
source: "https://vercel.com/docs/pricing/regional-pricing/yul1"
---

# Montréal, Canada (yul1) pricing

The table below shows Managed Infrastructure products with pricing specific to the  region. This pricing is available only to [Pro plan](

The **Included** column shows the amount of usage covered in your [billing cycle](/docs/pricing/understanding-my-invoice#understanding-yo

> **💡 Note:** Active CPU and Provisioned Memory are billed at different rates depending on
> the region your [fluid compute](/docs/fluid-compute) is deployed. The rates
> for each region can be found in the [fluid
> pricing](/docs/functions/usage-and-pricing) documentation.

Learn more about the different regions available on Vercel in the [regions](/docs/regions) documentation. See the [pricing](/docs/pricing

---
title: "Sales Tax"
description: "This page covers frequently asked questions around sales tax."
last_updated: "2026-01-16T02:19:34.213Z"
source: "https://vercel.com/docs/pricing/sales-tax"
---

# Sales Tax

### Do you charge sales tax on your services?

Yes. Beginning November 1, 2025, we will start collecting sales tax for US-based customers on all Vercel products and services where requ

### Why are you starting to collect sales tax now?

State regulations now require cloud service providers to collect sales tax in many jurisdictions. We're updating our billing practices ef

### Will all customers be charged sales tax?

Not necessarily. Sales tax is only charged in states where Vercel is registered to collect tax. If your billing address is in one of thos

### How will sales tax appear on my invoice?

Invoices will now show a separate line item for sales tax, clearly indicating the amount charged in addition to the products and services

### Do I need to take any action regarding sales tax?

For most customers, no action is required. Sales tax will automatically be calculated and added to your invoice based on your billing inf

### What if my organization is tax-exempt?

If you qualify for tax exemption, please send your exemption certificate to <tax@vercel.com>. Once verified by our team, your account wil

### Are international customers charged any additional fees or taxes?

For international customers, we will begin collecting VAT, GST, or similar taxes where required by law in the near future. We will commun

### When will US customers start being charged for sales tax?

Sales tax collection for US-based customers will begin on November 1, 2025. All invoices issued on or after that date will include applic

### Where can I find more information about Vercel’s terms of service about tax?

You can refer to our [terms of service](/legal/terms#payments) on collecting sales tax.

### Who can I contact with tax-related questions?

If you have specific questions about tax collection or exemptions, please contact our team at <tax@vercel.com>.

---

title: "Billing & Invoices"
description: "Learn how Vercel invoices get structured for Pro and Enterprise plans. Learn how usage allotments and on-demand charges get
last_updated: "2026-01-16T02:19:34.222Z"
source: "https://vercel.com/docs/pricing/understanding-my-invoice"
---

# Billing & Invoices

You can view your current invoice from the **Settings** tab of your [dashboard](/dashboard) in two ways:

- By navigating to the **Billing** tab of the dashboard
- Or selecting the latest entry in the list of invoices on the **Invoices** tab.

## Understanding your invoice

Your invoice is a breakdown of the charges you have incurred for the current billing cycle. It includes the total amount due, the billing

When you access your invoice through the **Invoice** tab:

- You can choose to download the invoice as a PDF through selecting the  icon on the invoice row
- You can select an invoice to view the detailed breakdown of the charges. Each invoice includes an invoice number, the date issued, and

### Pro plan invoices

Pro plan users receive invoices based on on-demand usage. Each feature under [Managed Infrastructure](/docs/pricing#managed-infrastructur

- A specific usage allotment. Charges incur on-demand when you exceed the usage allotment
- [Managed Infrastructure](/docs/pricing#managed-infrastructure-billable-resources) charges get metered and billed on a monthly basis
- [Developer Experience Platform](/docs/pricing#dx-platform-billable-resources) features get billed at fixed prices when purchased, and c

When viewing an invoice, Pro plan users will see a section called **[On-demand Charges](#pro-plan-on-demand-charges)**. This section has

#### Pro plan on-demand charges

For Pro plan users, on-demand charges incur in two ways. Either when you exceed the usage allotment for a specific feature under [Managed

### Enterprise plan invoices

Enterprise customer's invoicing gets tailored around a flexible usage model. It's based on a periodic commitment to [Managed Infrastructu

The top of the invoice shows a summary of the commitment period, the total MIUs committed, and the current usage towards that commitment.

#### Managed Infrastructure Units (MIU)

MIUs are a measure of the infrastructure consumption of an Enterprise project. These consist of a variety of resources like [Fast Data Tr

#### Enterprise on-demand charges

When Enterprise customers exceed their commitment for a period, they will see individual line items for the on-demand amount under the **

## More resources

For more information on Vercel's pricing, and guidance on optimizing consumption, see the following resources:

- [Vercel Pricing](/docs/pricing)
- [Manage and optimize usage](/docs/pricing/manage-and-optimize-usage)


---
title: "Working with Vercel"
description: "Learn how to set up Vercel"
last_updated: "2026-01-16T02:19:34.241Z"
source: "https://vercel.com/docs/private-registry"
---

# Working with Vercel

Vercel distributes packages with the `@vercel-private` scope through our
private npm registry, requiring authentication through a Vercel account for
each user.

This guide covers Vercel's private registry packages. For information on using your own private npm packages with Vercel, see our guide o

> **💡 Note:** Access to `@vercel-private` packages is linked to access to products. If you
> have trouble accessing a package, please check that you have access to the
> corresponding Vercel product.

## Setting up your local environment

- ### Set up your workspace
  If you're the first person on your team to use Vercel's private registry,
  you'll need to set up your workspace to fetch packages from the private
  registry.

  Execute the following command to configure your package manager to fetch
  packages with the `@vercel-private` scope from the private registry. If you're using modern Yarn (v2 or newer) see the [Using modern ve
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```

````
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>
  This command creates an `.npmrc` file (or updates one if it exists) at the root
  of your workspace. We recommend committing this file to your repository, as it
  will help other engineers get on board faster.

- ### Setting registry server using modern versions of Yarn
  Yarn version 2 or newer ignores the `.npmrc` config file so you will need to use this command instead to add the
  registry to your project's `.yarnrc.yml` file:
  ```sh copy
  yarn config set npmScopes.vercel-private.npmRegistryServer "https://vercel-private-registry.vercel.sh/registry"
  ```

- ### Log in to the private registry
  Each team member will need to complete this step. It may be helpful to
  summarize this step in your team's onboarding documentation.

  To log in, use the following command and follow the prompts:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>
  > **⚠️ Warning:** The minimum required version of npm to log into the registry is 8.14.0. For
  > pnpm, version 7.0.0 or higher is required.
  During this process, you will be asked to log in to your Vercel account. Ensure
  that the account that you log in to has access to the Vercel product(s) that
  you're trying to install.

  You should now have a `.npmrc` file in your home directory that contains the
  authentication token for the private registry.

- #### Setting token using modern versions of Yarn
  Yarn version 2 or newer requires the authentication token to be saved in a
  `.yarnrc.yml` file. After running the above command, you can copy the token
  from the `.npmrc` file with:
  ```sh copy
  auth_token=$(awk -F'=' '/vercel-private-registry.vercel.sh\/:_authToken/ {print $2}' $(npm config get userconfig)) \
  && yarn config set --home 'npmRegistries["https://vercel-private-registry.vercel.sh/registry"].npmAuthToken' $auth_token
  ```
  Note the `--home` flag, which ensures the token is saved in the global `.yarnrc.yml`
  rather then in your project so that it isn't committed.

- ### Verify your setup
  Verify your login status by executing:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>
  > **⚠️ Warning:** The Yarn command only works with Yarn version 2 or newer, use the npm command
  > if using Yarn v1.
  You should see your Vercel username returned if everything is set up correctly.

- ### Optionally set up a pre-install message for missing credentials
  When a user tries to install a package from the private registry without first
  logging in, the error message might be unclear. To help, we suggest adding a
  pre-install message that provides instructions to those unauthenticated users.

  Create a `preinstall.mjs` file with your error message:
````
````

  ```javascript copy filename="preinstall.mjs"
  import { exec } from 'node:child_process';
  import { promisify } from 'node:util';

  const execPromise = promisify(exec);

  // Detect which package manager is being used
  const userAgent = process.env.npm_config_user_agent || '';
  const isYarn = userAgent.includes('yarn');
  const isPnpm = userAgent.includes('pnpm');
  const isBun = userAgent.includes('bun');

  let checkCommand;
  let loginCommand;

  if (isPnpm) {
    checkCommand =
      'pnpm whoami --registry=https://vercel-private-registry.vercel.sh/registry';
    loginCommand = 'pnpm login --scope=@vercel-private';
  } else if (isYarn) {
    checkCommand = 'yarn npm whoami --scope=vercel-private';
    loginCommand = 'npm login --scope=@vercel-private';
  } else {
    // npm or bun
    checkCommand =
      'npm whoami --registry=https://vercel-private-registry.vercel.sh/registry';
    loginCommand = 'npm login --scope=@vercel-private';
  }

  try {
    await execPromise(checkCommand);
  } catch (error) {
    throw new Error(
      `Please log in to the Vercel private registry to install \`@vercel-private\`-scoped packages:\n\`${loginCommand}\``,
    );
  }
  ```
  Then add the following script to the `scripts` field in your `package.json`:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i
      ```
    </Code>
  </CodeBlock>

## Setting up Vercel

Now that your local environment is set up, you can configure Vercel to use the
private registry.

1. Create a [Vercel authentication token](/docs/rest-api#creating-an-access-token) on the [Tokens](https://vercel.com/account/tokens) pag
2. To set the newly created token in Vercel, navigate to the [Environment Variables](https://vercel.com/docs/environment-variables)
   settings for your Project
3. Add a new environment variable with the name `VERCEL_TOKEN`, and set the
   value to the token you created above. We recommend using a [Sensitive Environmental Variable](/docs/environment-variables/sensitive-en
4. Add a new environment variable with the name `NPM_RC`, and set the value to
   the following:

```sh copy
@vercel-private:registry=https://vercel-private-registry.vercel.sh/registry
//vercel-private-registry.vercel.sh/:_authToken=${VERCEL_TOKEN}
```

> **💡 Note:** If you already have an `NPM_RC` environment variable, you can append the above
> to that existing value.

Vercel should now be able to install packages from the private registry when
building your Project.

## Setting up your CI provider

The instructions below are for [GitHub Actions](https://github.com/features/actions),
but configuring other CI providers should be similar:

1. Create a [Vercel authentication token](/docs/rest-api#creating-an-access-token) on the [Tokens](https://vercel.com/account/tokens) pag
2. Once you have a new token, add it as a secret named `VERCEL_TOKEN` to your
   GitHub repository or organization. To learn more about how to add secrets, [Using secrets in GitHub Actions](https://docs.github.com/e
3. Finally, create a [workflow](https://docs.github.com/en/actions/using-workflows) for the product you're setting up. The example workfl
   and assumes that you're using [pnpm](https://pnpm.io/) as your package manager. In this example we also pass the token to the Conforma

```yaml filename=".github/workflows/conformance.yml"
name: Conformance

on:
```

```
  pull_request:
    branches:
      - main

jobs:
  conformance:
    name: 'Run Conformance'
    runs-on: ubuntu-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v3

      - name: Set up Node.js
        uses: actions/setup-node@v4
        with:
          node-version-file: '.node-version'

      - name: Set up pnpm
        uses: pnpm/action-setup@v3

      - name: Set up Vercel private registry
        run: npm config set //vercel-private-registry.vercel.sh/:_authToken $VERCEL_TOKEN
        env:
          VERCEL_TOKEN: ${{ secrets.VERCEL_TOKEN }}

      - name: Install dependencies
        run: pnpm install

      - name: Run Conformance
        run: pnpm conformance
        env:
          VERCEL_TOKEN: ${{ secrets.VERCEL_TOKEN }}
```

By default, GitHub workflows are not required. To require the workflow in your repository, [create a branch protection rule on GitHub](ht

---
title: "Production checklist for launch"
description: "Ensure your application is ready for launch with this comprehensive production checklist by the Vercel engineering team. Co
last_updated: "2026-01-16T02:19:34.254Z"
source: "https://vercel.com/docs/production-checklist"
---

# Production checklist for launch

When launching your application on Vercel, it is important to ensure that it's ready for production. This checklist is prepared by the Ve

- [Operational excellence](#operational-excellence)
- [Security](#security)
- [Reliability](#reliability)
- [Performance efficiency](#performance)
- [Cost optimization](#cost-optimization).

## Operational excellence

## Security

## Reliability

## Performance

## Cost optimization

## Enterprise support

Need help with your production rollout?

---
title: "General settings"
description: "Configure basic settings for your Vercel project, including the project name, build and development settings, root director
last_updated: "2026-01-16T02:19:34.260Z"
source: "https://vercel.com/docs/project-configuration/general-settings"
---

# General settings

## Project name

Project names can be up to 100 characters long and must be lowercase. They can include letters, digits, and the following characters: `.`

## Build and development settings

You can edit settings regarding the build and development settings, root directory, and the [install command](/docs/deployments/configure

The changes you make to these settings will only be applied starting from your **next deployment**.

## Nodejs version

Learn more about how to customize the Node.js version of your project in the [Node.js runtime](/docs/functions/runtimes/node-js/node-js-v

You can also learn more about [all supported versions](/docs/functions/runtimes/node-js/node-js-versions#default-and-available-versions)

## Project ID

Your project ID can be used by the REST API to carry out tasks relating to your project. To locate your Project ID:

1. Ensure you have selected your Team from the [scope selector](/docs/dashboard-features#scope-selector).
2. Choose your project from the [dashboard](/dashboard).

3. Select the **Settings** tab.
4. Under **General**, scroll down until you find **Project ID**. The ID should start `prj_`.
5. Copy the Project ID to use as needed.

## Vercel Toolbar settings

The Vercel Toolbar is a tool that assists you in iterating and developing your project and is enabled by default on preview deployments. `

- Leave feedback on deployments with [Comments](/docs/comments)
- Navigate [through dashboard pages](/docs/vercel-toolbar#using-the-toolbar-menu), and [share deployments](/docs/vercel-toolbar#sharing-d
- Read and set [Feature Flags](/docs/feature-flags)
- Use [Draft Mode](/docs/draft-mode) for previewing unpublished content
- Edit content in real-time using [Edit Mode](/docs/edit-mode)
- Inspect for [Layout Shifts](/docs/vercel-toolbar/layout-shift-tool) and [Interaction Timing](/docs/vercel-toolbar/interaction-timing-to
- Check for accessibility issues with the [Accessibility Audit Tool](/docs/vercel-toolbar/accessibility-audit-tool)

---
title: "Git Configuration"
description: "Learn how to configure Git for your project through vercel.json or vercel.ts."
last_updated: "2026-01-16T02:19:34.283Z"
source: "https://vercel.com/docs/project-configuration/git-configuration"
---

# Git Configuration

The following configuration options can be used through a `vercel.json` file via [Static Configuration](/docs/project-configuration/verce

## git.deploymentEnabled

**Type**: `Object` of key branch identifier `String` and value `Boolean`, or `Boolean`.

**Default**: `true`

Specify branches that should not trigger a deployment upon commits. By default, any unspecified branch is set to `true`.

#### \['vercel.json'

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "git": {
    "deploymentEnabled": {
      "dev": false
    }
  }
}
```

#### 'vercel.ts']

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  git: {
    deploymentEnabled: {
      dev: false,
    },
  },
};
```

### Matching multiple branches

Use [minimatch syntax](https://github.com/isaacs/minimatch) to define behavior for multiple branches.

The example below prevents automated deployments for any branch that starts with `internal-`.

#### \['vercel.json'

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "git": {
    "deploymentEnabled": {
      "internal-*": false
    }
  }
}
```

#### 'vercel.ts']

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  git: {
    deploymentEnabled: {
      'internal-*': false,
    },
  },
};
```

### Branches matching multiple rules

If a branch matches multiple rules and at least one rule is `true`, a deployment will occur.

#### \['vercel.json'

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "git": {
    "deploymentEnabled": {
      "experiment-*": false,
      "*-dev": true
    }
  }
}
```

#### 'vercel.ts']

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  git: {
    deploymentEnabled: {
      'experiment-*': false,
      '*-dev': true,
    },
  },
};
```

A branch named `experiment-my-branch-dev` will create a deployment.

### Turning off all automatic deployments

To turn off automatic deployments for all branches, set the property value to `false`.

#### \['vercel.json'

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "git": {
    "deploymentEnabled": false
  }
}
```

#### 'vercel.ts']

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  git: {
    deploymentEnabled: false,
  },
};
```

## github.autoAlias

**Type**: `Boolean`.

When set to `false`, [Vercel for GitHub](/docs/git/vercel-for-github) will create preview deployments upon merge.

> **⚠️ Warning:** Follow the [deploying a staged production
> build](/docs/deployments/promoting-a-deployment#staging-and-promoting-a-production-deployment)
> workflow instead of this setting.

#### \['vercel.json'

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "github": {
    "autoAlias": false
  }
}
```

#### 'vercel.ts']

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  github: {
    autoAlias: false,
  },
};
```

## github.autoJobCancelation

**Type**: `Boolean`.

When set to false, [Vercel for GitHub](/docs/git/vercel-for-github) will always build pushes in sequence without cancelling a build for t

#### \['vercel.json'

```json filename="vercel.json"
```

```
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "github": {
    "autoJobCancelation": false
  }
}
```

#### 'vercel.ts']

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  github: {
    autoJobCancelation: false,
  },
};
```

## Legacy

### github.silent

The `github.silent` property has been deprecated in favor of the new settings in the dashboard, which allow for more fine-grained control

**Type**: `Boolean`.

When set to `true`, [Vercel for GitHub](/docs/git/vercel-for-github) will stop commenting on pull requests and commits.

#### \['vercel.json'

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "github": {
    "silent": true
  }
}
```

#### 'vercel.ts']

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  github: {
    silent: true,
  },
};
```

### github.enabled

The `github.enabled` property has been deprecated in favor of [git.deploymentEnabled](/docs/project-configuration/git-configuration#git.d

**Type**: `Boolean`.

When set to `false`, [Vercel for GitHub](/docs/git/vercel-for-github) will not deploy the given project regardless of the GitHub app bein

#### \['vercel.json'

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "github": {
    "enabled": false
  }
}
```

#### 'vercel.ts']

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  github: {
    enabled: false,
  },
};
```

---
title: "Git settings"
description: "Use the project settings to manage the Git connection, enable Git LFS, create deploy hooks, and configure the build step."
last_updated: "2026-01-16T02:19:34.296Z"
source: "https://vercel.com/docs/project-configuration/git-settings"
---

# Git settings

Once you have [connected a Git repository](/docs/git#deploying-a-git-repository), select the **Git** menu item from your project settings

- Managing Git Large File Storage (LFS)
- Creating Deploy Hooks
- Ignoring the build step when a commit is pushed to the Git repository

## Disconnect your Git repository

To disconnect your Git repository from your Vercel project:

1. Choose a project from the [dashboard](/dashboard)
2. Select the **Settings** tab and then select the **Git** menu item
3. Under **Connected Git Repository**, select the **Disconnect** button.

## Git Large File Storage (LFS)

If you have [LFS objects](https://git-lfs.com/) in your repository, you can enable or disable support for them from the [project settings
When support is enabled, Vercel will pull the LFS objects that are used in your repository.

> **💡 Note:** You must [redeploy your
> project](/docs/deployments/managing-deployments#redeploy-a-project) after
> turning Git LFS on.

## Deploy Hooks

Vercel supports **deploy hooks**, which are unique URLs that accept HTTP POST requests and trigger deployments. Check out [our Deploy Hoo

## Ignored Build Step

By default, Vercel creates a new [deployment](/docs/deployments) and build (unless the Build Step is [skipped](/docs/deployments/configur

Each commit in Git is assigned a unique hash value commonly referred to as SHA. If the SHA of the commit was already deployed in the past

To ignore the build step:

1. Choose a project from the [dashboard](/dashboard)
2. Select the **Settings** tab and then select the **Git** menu item
3. In the **Ignored Build Step** section, select the behavior you would like. This behavior provides a command that outputs a code, which
   - **Automatic**: Each commit will issue a new build
   - **Only build production**: When the `VERCEL_ENV` is production, a new build will be issued
   - **Only build preview**: When the `VERCEL_ENV` is preview, a new build will be issued
   - **Only build if there are changes**: A new build will be issued only when the Git diff contains changes
   - **Only build if there are changes in a folder**: A new build will be issued only when the Git diff contains changes in a folder that
   - **Don't build anything**: A new build will never be issued
   - **Run my Bash script**: [Run a Bash script](/kb/guide/how-do-i-use-the-ignored-build-step-field-on-vercel) from a location that you
   - **Run my Node script**: [Run a Node script](/kb/guide/how-do-i-use-the-ignored-build-step-field-on-vercel) from a location that you
   - **Custom**: You can enter any other command here, for example, only building an Nx app ([`npx nx-ignore <project-name>`](https://git
4. When your deployment enters the `BUILDING` state, the command you've entered in the **Ignored Build Step** section will be run. The co
   - If the command exits with code `1`, the build continues as normal
   - If the command exits with code `0`, the build is immediately aborted, and the deployment state is set to `CANCELED`

> **⚠️ Warning:** Canceled builds are counted as full deployments as they execute a build
> command in the build step. This means that any canceled builds initiated using
> the ignore build step will still count towards your [deployment quotas](/docs/limits#deployments-per-day-hobby) and [concurrent build s

To learn about more advanced usage see the ["How do I use the Ignored Build Step field on Vercel?"](/kb/guide/how-do-i-use-the-ignored-bu

### Ignore Build Step on redeploy

If you have set an ignore build step command or [script](/kb/guide/how-do-i-use-the-ignored-build-step-field-on-vercel), you can also ski

1. From the Vercel dashboard, select your project
2. Select the **Deployments** tab and find your deployment
3. Click the ellipses (...) and from the context menu, select **Redeploy**
4. Uncheck the **Use project's Ignore Build Step** checkbox

---
title: "Global Vercel CLI Configuration"
description: "Learn how to configure Vercel CLI under your system user."
last_updated: "2026-01-16T02:19:34.425Z"
source: "https://vercel.com/docs/project-configuration/global-configuration"
---

# Global Vercel CLI Configuration

Using the following files and configuration options, you can configure [Vercel CLI](/cli) under your system user.

The two global configuration files are: `config.json` and `auth.json`. These files are stored in the `com.vercel.cli` directory inside [`

- Linux: `~/.local/share/com.vercel.cli`
- macOS: `~/Library/Application Support/com.vercel.cli`
- Windows: `%APPDATA%\Roaming\xdg.data\com.vercel.cli`

> **💡 Note:** These files are automatically generated by Vercel CLI, and shouldn't need to
> be altered.

## config.json

This file is used for global configuration of Vercel deployments. Vercel CLI uses this file as a way to co-ordinate how deployments shoul

The first option is a single `_` that gives a description to the file, if a user should find themselves looking through it without contex

You can use the following options to configure all Vercel deployments on your system's user profile:

### currentTeam

**Type**: `String`.

**Valid values**: A [team ID](/docs/accounts#find-your-team-id).

This option tells [Vercel CLI](/cli) which context is currently active. If this property exists and contains a team ID, that team is used

```json filename="config.json"
{
  "currentTeam": "team_ofwUZockJlL53hINUGCc1ONW"
```

}
```

### collectMetrics

**Type**: `Boolean`.

**Valid values**: `true` (default), `false`.

This option defines whether [Vercel CLI](/cli) should collect anonymous metrics about which commands are invoked the most, how long they

```json filename="config.json"
{
  "collectMetrics": true
}
```

## auth.json

This file should not be edited manually. It exists to contain the authentication information for the Vercel clients.

In the case that you are uploading your global configuration setup to a potentially insecure destination, we highly recommend ensuring th

---
title: "Project Configuration"
description: "Learn how to configure your Vercel projects using vercel.json, vercel.ts, or the dashboard to control builds, routing, func
last_updated: "2026-01-16T02:19:34.307Z"
source: "https://vercel.com/docs/project-configuration"
---

# Project Configuration

Vercel automatically detects your framework and sets sensible defaults for builds, deployments, and routing. Project configuration lets y

In addition to configuring your project through the [dashboard](/docs/projects/project-dashboard), you have the following options:

- [Static file-based configuration](/docs/project-configuration/vercel-json) - Static JSON configuration in your repository
- [Programmatic file-based configuration](/docs/project-configuration/vercel-ts) - Dynamic TypeScript configuration that runs at build ti
- [Global CLI configuration](/docs/project-configuration/global-configuration) - System-wide Vercel CLI settings

Each method lets you control different aspects of your project.

## File-based configuration

File-based configuration lives in your repository and gets version-controlled with your code. You can use either [`vercel.json`](/docs/pr

The table below shows all available configuration properties:

| Property | vercel.json | v |
| --- | :---: | :---: |
| **$schema** | [View](/docs/project-configuration/vercel-json#schema-autocomplete) | [View](/docs/project-configur |
| **buildCommand** | [View](/docs/project-configuration/vercel-json#buildcommand) | [View](/docs/project-confi |
| **bunVersion** | [View](/docs/project-configuration/vercel-json#bunversion) | [View](/docs/project-conf |
| **cleanUrls** | [View](/docs/project-configuration/vercel-json#cleanurls) | [View](/docs/project-con |
| **crons** | [View](/docs/project-configuration/vercel-json#crons) | [View](/docs/project-c |
| **devCommand** | [View](/docs/project-configuration/vercel-json#devcommand) | [View](/docs/project-conf |
| **fluid** | [View](/docs/project-configuration/vercel-json#fluid) | [View](/docs/project-c |
| **framework** | [View](/docs/project-configuration/vercel-json#framework) | [View](/docs/project-con |
| **functions** | [View](/docs/project-configuration/vercel-json#functions) | [View](/docs/project-con |
| **headers** | [View](/docs/project-configuration/vercel-json#headers) | [View](/docs/project-co |
| **ignoreCommand** | [View](/docs/project-configuration/vercel-json#ignorecommand) | [View](/docs/project-confi |
| **images** | [View](/docs/project-configuration/vercel-json#images) | [View](/docs/project-co |
| **installCommand** | [View](/docs/project-configuration/vercel-json#installcommand) | [View](/docs/project-config |
| **outputDirectory** | [View](/docs/project-configuration/vercel-json#outputdirectory) | [View](/docs/project-config |
| **public** | [View](/docs/project-configuration/vercel-json#public) | [View](/docs/project-co |
| **redirects** | [View](/docs/project-configuration/vercel-json#redirects) | [View](/docs/project-con |
| **bulkRedirectsPath** | [View](/docs/project-configuration/vercel-json#bulkredirectspath) | [View](/docs/project-configu |
| **regions** | [View](/docs/project-configuration/vercel-json#regions) | [View](/docs/project-co |
| **functionFailoverRegions** | [View](/docs/project-configuration/vercel-json#functionfailoverregions) | [View](/docs/project-configurat |
| **rewrites** | [View](/docs/project-configuration/vercel-json#rewrites) | [View](/docs/project-con |
| **trailingSlash** | [View](/docs/project-configuration/vercel-json#trailingslash) | [View](/docs/project-confi |

## Global CLI configuration

[Global Configuration](/docs/project-configuration/global-configuration) affects how Vercel CLI behaves on your machine. These settings a

## Configuration areas

For detailed information about specific configuration areas, see:

- [General Settings](/docs/project-configuration/general-settings) - Project name, Node.js version, build settings, and Vercel Toolbar
- [Project Settings](/docs/project-configuration/project-settings) - Overview of all project settings in the dashboard
- [Git Configuration](/docs/project-configuration/git-configuration) - Configure Git through vercel.json and vercel.ts
- [Git Settings](/docs/project-configuration/git-settings) - Manage Git connection, LFS, and build step settings
- [Security settings](/docs/project-configuration/security-settings) - Attack Challenge Mode, logs protection, fork protection, OIDC, and

---
title: "Project settings"
description: "Use the project settings, to configure custom domains, environment variables, Git, integrations, deployment protection, fun
last_updated: "2026-01-16T02:19:34.317Z"
source: "https://vercel.com/docs/project-configuration/project-settings"
---

# Project settings

From the Vercel [dashboard](/dashboard), there are two areas where you can configure settings:

- **Team Settings**: Any settings configured here, are applied at the team-level, although you can select which project's the settings sh
```

- **Project Settings**: These are specific settings, accessed through the [project dashboard](/docs/projects/project-dashboard), that are

This guide focuses on the project settings. To edit project settings:

1. Ensure you have selected your Team from the [scope selector](/docs/dashboard-features#scope-selector).
2. Choose a project from the [dashboard](/dashboard).
3. Select the **Settings** tab.
4. Find the settings you need and make changes.

## Configuring your project with a vercel.json file

While many settings can be set from the dashboard, you can also define a `vercel.json` file at the project root that allows you to set an

To learn more, see [Configuring projects with vercel.json](/docs/project-configuration).

## General settings

This provides all the foundational information and settings for your Vercel project, including the name, build and deployment settings, t

To learn more, see [General Settings](/docs/project-configuration/general-settings)

## Build and deployment settings

In your build and deployment settings, adjust configurations such as framework settings, code directory, and Node.js version.

In this section, you can adjust build-related configurations, such as framework settings, code directory, Node.js version, and more.

- [Node.js version](/docs/functions/runtimes/node-js/node-js-versions#setting-the-node.js-version-in-project-settings)
- [Prioritize production builds](/docs/deployments/concurrent-builds#prioritize-production-builds)
- [On-demand concurrent builds](/docs/deployments/managing-builds#on-demand-concurrent-builds)

## Custom domains

You can [add **custom domains**](/docs/domains/add-a-domain) for each project.

To learn more, [see the Domains documentation](/docs/domains)

## Environment Variables

You can configure Environment Variables for each environment directly from your project's settings. This includes [linking Shared Environ

To learn more, [see the Environment Variables documentation](/docs/environment-variables).

## Git

In your project settings, you can manage the Git connection, enable Git LFS, and manage your build step settings.

To learn more about the settings, see [Git Settings](/docs/project-configuration/git-settings). To learn more about working with your Git

## Integrations

To manage third-party integrations for your project, you can use the Integrations settings.

To learn more, see [Integrations](/docs/integrations).

## Deployment Protection

Protect your project deployments with [Vercel Authentication](/docs/security/deployment-protection/methods-to-protect-deployments/vercel-

To learn more, see [Deployment Protection](/docs/security/deployment-protection).

## Functions

You can configure the default settings for your Vercel Functions, including the Node.js version, memory, timeout, region, and more.

To learn more, see [Configuring Functions](/docs/functions/configuring-functions).

## Cron Jobs

You can enable and disable Cron Jobs for your project from the Project Settings. Configuring cron jobs is done in your codebase.

To learn more, see [Cron Jobs](/docs/cron-jobs).

## Project members

Team owners can manage who has access to the project by adding or removing members to that specific project from the project settings.

To learn more, see [project-level roles](/docs/rbac/access-roles/project-level-roles).

## Webhooks

Webhooks allow your external services to respond to events in your project. You can enable them on a per-project level from the project s

To learn more, see the [Webhooks documentation](/docs/webhooks).

## Drains

Drains are a Pro and Enterprise feature that allow you to send observability data (logs, traces, speed insights, and analytics) to extern

To learn more, see the [Drains documentation](/docs/drains/using-drains).

## Security settings

From your project's security settings you can enable or disable [Attack Challenge Mode](/docs/attack-challenge-mode), [Logs and Source Pr

To learn more, see [Security Settings](/docs/project-configuration/security-settings).

## Advanced

Vercel provides some additional features in order to configure your project in a more advanced way. This includes:

- Displaying [directory listing](/docs/directory-listing)
- Enabling [Skew protection](/docs/skew-protection)

---
title: "Security settings"
description: "Configure security settings for your Vercel project, including Logs and Source Protection, Customer Success Code Visibility
last_updated: "2026-01-16T02:19:34.327Z"
source: "https://vercel.com/docs/project-configuration/security-settings"
---

# Security settings

To adjust your project's security settings:

1. Select your project from your [dashboard](/dashboard)
2. Select the **Settings** tab
3. Choose the **Security** menu item

From here you can enable or disable [Attack Challenge Mode](/docs/attack-challenge-mode), [Logs and Source Protection](#build-logs-and-so

## Build logs and source protection

By default, the following paths mentioned below can only be accessed by you and authenticated members of your Vercel team:

- `/_src`: Displays the source code and build output.
- `/_logs`: Displays the build logs.

> **⚠️ Warning:** Disabling **Build Logs and Source Protection** will make your source code and
> logs publicly accessible. **Do not** edit this setting if you don't want them
> to be publicly accessible. 

None of your existing deployments will be affected when you toggle this
setting. If you'd like to make the source code or logs private on your
existing deployments, the only option is to delete these deployments.

This setting is overwritten when a deployment is created using Vercel CLI with the [`--public` option](/docs/cli/deploy#public) or the [`

> **💡 Note:** For deployments created before July 9th, 2020 at 7:05 AM (UTC), only the
> Project Settings is considered for determining whether the deployment's Logs
> and Source are publicly accessible or not. It doesn't matter if the `--public`
> flag was passed when creating those Deployments.

## Customer Success Code Visibility

Vercel provides a setting that controls the visibility of your source code to our Customer Success team. By default, this setting is disa
The Customer Success team might request for this setting to be enabled to troubleshoot specific issues related to your code.

## Git fork protection

If you receive a pull request from a fork of your repository, Vercel will require authorization from you or a [Team Member](/docs/rbac/ma

This behavior protects you from leaking sensitive project information such as environment variables and the [OIDC Token](/docs/oidc).

You can disable this protection in the Security section of your Project Settings.

> **💡 Note:** Do not disable this setting until you review Environment Variables in your
> project as well as  in your source code.

## Secure Backend Access with OIDC Federation

This feature allows you to secure access to your backend services by using short-lived, non-persistent tokens that are signed by Vercel's

To learn more, see [Secure Backend Access with OIDC Federation](/docs/oidc).

## Deployment Retention Policy

Deployment Retention Policy allows you to set a limit on how long older deployments are kept for your project. To learn more, see [Deploy

This section also provides information on the recently deleted deployments

---
title: "Static Configuration with vercel.json"
description: "Learn how to use vercel.json to configure and override the default behavior of Vercel from within your project. "
last_updated: "2026-01-16T02:19:34.683Z"
source: "https://vercel.com/docs/project-configuration/vercel-json"
---

# Static Configuration with vercel.json

The `vercel.json` file lets you configure, and override the default behavior of Vercel from within your project.

This file should be created in your project's root directory and allows you to set:

- [schema autocomplete](#schema-autocomplete)
- [buildCommand](#buildcommand)
- [bunVersion](#bunversion)
- [cleanUrls](#cleanurls)
- [crons](#crons)
- [devCommand](#devcommand)
- [fluid](#fluid)
- [framework](#framework)
- [functions](#functions)
- [headers](#headers)
- [ignoreCommand](#ignorecommand)
- [images](#images)
- [installCommand](#installcommand)
- [outputDirectory](#outputdirectory)

- [public](#public)
- [redirects](#redirects)
- [bulkRedirectsPath](#bulkredirectspath)
- [regions](#regions)
- [functionFailoverRegions](#functionfailoverregions)
- [rewrites](#rewrites)
- [trailingSlash](#trailingslash)

## schema autocomplete

To add autocompletion, type checking, and schema validation to your `vercel.json` file, add the following to the top of your file:

```json
{
  "$schema": "https://openapi.vercel.sh/vercel.json"
}
```

## buildCommand

**Type:** `string | null`

The `buildCommand` property can be used to override the Build Command in the Project Settings dashboard, and the `build` script from the

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "buildCommand": "next build"
}
```

This value overrides the [Build Command](/docs/deployments/configure-a-build#build-command) in Project Settings.

## bunVersion

**Type:** `string`

**Value:** `"1.x"`

The `bunVersion` property configures your project to use the Bun runtime instead of Node.js. When set, all [Vercel Functions](/docs/funct

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "bunVersion": "1.x"
}
```

> **💡 Note:** Vercel manages the Bun minor and patch versions automatically. `1.x` is the
> only valid value currently.

When using Next.js with [ISR](/docs/incremental-static-regeneration) (Incremental Static Regeneration), you must also update your `build`

```json filename="package.json"
{
  "scripts": {
    "dev": "bun run --bun next dev",
    "build": "bun run --bun next build"
  }
}
```

To learn more about using Bun with Vercel Functions, see the [Bun runtime documentation](/docs/functions/runtimes/bun).

## cleanUrls

**Type**: `Boolean`.

**Default Value**: `false`.

When set to `true`, all HTML files and Vercel functions will have their extension removed. When visiting a path that ends with the extens

For example, a static file named `about.html` will be served when visiting the `/about` path. Visiting `/about.html` will redirect to `/a

Similarly, a Vercel Function named `api/user.go` will be served when visiting `/api/user`. Visiting `/api/user.go` will redirect to `/api

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "cleanUrls": true
}
```

If you are using Next.js and running `vercel dev`, you will get a 404 error when visiting a route configured with `cleanUrls` locally. It

## crons

Used to configure [cron jobs](/docs/cron-jobs) for the production deployment of a project.

**Type**: `Array` of cron `Object`.

**Limits**:

- A maximum of string length of 512 for the `path` value.
- A maximum of string length of 256 for the `schedule` value.

### Cron object definition

- `path` - **Required** - The path to invoke when the cron job is triggered. Must start with `/`.
- `schedule` - **Required** - The [cron schedule expression](/docs/cron-jobs#cron-expressions) to use for the cron job.

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "crons": [
    {
      "path": "/api/every-minute",
      "schedule": "* * * * *"
    },
    {
      "path": "/api/every-hour",
      "schedule": "0 * * * *"
    },
    {
      "path": "/api/every-day",
      "schedule": "0 0 * * *"
    }
  ]
}
```

## devCommand

This value overrides the [Development Command](/docs/deployments/configure-a-build#development-command) in Project Settings.

**Type:** `string | null`

The `devCommand` property can be used to override the Development Command in the Project Settings dashboard. For more information on the

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "devCommand": "next dev"
}
```

## fluid

This value allows you to enable [Fluid compute](/docs/fluid-compute) programmatically.

**Type:** `boolean | null`

The `fluid` property allows you to test Fluid compute on a per-deployment or per [custom environment](/docs/deployments/environments#custo

> **💡 Note:** As of April 23, 2025, Fluid compute is enabled by default for new projects.

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "fluid": true
}
```

## framework

This value overrides the [Framework](/docs/deployments/configure-a-build#framework-preset) in Project Settings.

**Type:** `string | null`

Available framework slugs:

The `framework` property can be used to override the Framework Preset in the Project Settings dashboard. The value must be a valid framew

> **💡 Note:** To select "Other" as the Framework Preset, use .

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "framework": "nextjs"
}
```

## functions

**Type:** `Object` of key `String` and value `Object`.

### Key definition

A [glob](https://github.com/isaacs/node-glob#glob-primer) pattern that matches the paths of the Vercel functions you would like to custom

- `api/*.js` (matches one level e.g. `api/hello.js` but not `api/hello/world.js`)
- `api/**/*.ts` (matches all levels `api/hello.ts` and `api/hello/world.ts`)
- `src/pages/**/*` (matches all functions from `src/pages`)
- `api/test.js`

### Value definition

- `runtime` (optional): The npm package name of a [Runtime](/docs/functions/runtimes), including its version.
- `memory`: Memory cannot be set in `vercel.json` with [Fluid compute](/docs/fluid-compute) enabled. Instead set it in the **Functions**
- `maxDuration` (optional): An integer defining how long your Vercel Function should be allowed to run on every request in seconds (betwe
- `supportsCancellation` (optional): A boolean defining whether your Vercel Function should [support request cancellation](/docs/function
- `includeFiles` (optional): A [glob](https://github.com/isaacs/node-glob#glob-primer) pattern to match files that should be included in
- `excludeFiles` (optional): A [glob](https://github.com/isaacs/node-glob#glob-primer) pattern to match files that should be excluded fro

### Description

By default, no configuration is needed to deploy Vercel functions to Vercel.

For all [officially supported runtimes](/docs/functions/runtimes), the only requirement is to create an `api` directory at the root of yo

The `functions` property cannot be used in combination with `builds`. Since the latter is a legacy configuration property, we recommend d

Because [Incremental Static Regeneration (ISR)](/docs/incremental-static-regeneration) uses Vercel functions, the same configurations app

When deployed, each Vercel Function receives the following properties:

- **Memory:** 1024 MB (1 GB) - **(Optional)**
- **Maximum Duration:** 10s default - 60s / 1 minute (Hobby), 15s default - 300s / 5 minutes (Pro), or 15s default - 900s / 15 minutes (E

To configure them, you can add the `functions` property.

#### `functions` property with Vercel functions

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "api/test.js": {
      "memory": 3009,
      "maxDuration": 30
    },
    "api/*.js": {
      "memory": 3009,
      "maxDuration": 30
    }
  }
}
```

#### `functions` property with ISR

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "pages/blog/[hello].tsx": {
      "memory": 1024
    },
    "src/pages/isr/**/*": {
      "maxDuration": 10
    }
  }
}
```

### Using unsupported runtimes

In order to use a runtime that is not [officially supported](/docs/functions/runtimes), you can add a `runtime` property to the definitio

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functions": {
    "api/test.php": {
      "runtime": "vercel-php@0.5.2"
    }
  }
}
```

In the example above, the `api/test.php` Vercel Function does not use one of the [officially supported runtimes](/docs/functions/runtimes

For more information on Runtimes, see the [Runtimes documentation](/docs/functions/runtimes):

## headers

**Type:** `Array` of header `Object`.

**Valid values:** a list of header definitions.

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "headers": [
    {
      "source": "/service-worker.js",
      "headers": [
        {
          "key": "Cache-Control",
          "value": "public, max-age=0, must-revalidate"
        }
      ]
    },
    {
      "source": "/(.*)",
      "headers": [
        {
          "key": "X-Content-Type-Options",
          "value": "nosniff"
        },
        {
          "key": "X-Frame-Options",
          "value": "DENY"
        },
        {
          "key": "X-XSS-Protection",
          "value": "1; mode=block"
        }
      ]
    },
    {
```

```
      "source": "/:path*",
      "has": [
        {
          "type": "query",
          "key": "authorized"
        }
      ],
      "headers": [
        {
          "key": "x-authorized",
          "value": "true"
        }
      ]
    }
  ]
}
```

This example configures custom response headers for static files, [Vercel functions](/docs/functions), and a wildcard that matches all ro

### Header object definition

| Property  | Description
| --------- | ----------------------------------------------------------------------------------------------------------------------------------
| `source`  | A pattern that matches each incoming pathname (excluding querystring).
| `headers` | A non-empty array of key/value pairs representing each response header.
| `has`     | An optional array of `has` objects with the `type`, `key` and `value` properties. Used for conditional path matching based
| `missing` | An optional array of `missing` objects with the `type`, `key` and `value` properties. Used for conditional path matching ba

### Header `has` or `missing` object definition

If `value` is an object, it has one or more of the following fields:

This example demonstrates using the expressive `value` object to append the header `x-authorized: true` if the `X-Custom-Header` request

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "headers": [
    {
      "source": "/:path*",
      "has": [
        {
          "type": "header",
          "key": "X-Custom-Header",
          "value": {
            "pre": "valid",
            "suf": "value"
          }
        }
      ],
      "headers": [
        {
          "key": "x-authorized",
          "value": "true"
        }
      ]
    }
  ]
}
```

Learn more about [rewrites](/docs/headers) on Vercel and see [limitations](/docs/cdn-cache#limits).

## ignoreCommand

This value overrides the [Ignored Build Step](/docs/project-configuration/git-settings#ignored-build-step) in Project Settings.

**Type:** `string | null`

This `ignoreCommand` property will override the Command for Ignoring the Build Step for a given deployment. When the command exits with c

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "ignoreCommand": "git diff --quiet HEAD^ HEAD ./"
}
```

## installCommand

This value overrides the [Install Command](/docs/deployments/configure-a-build#install-command) in Project Settings.

**Type:** `string | null`

The `installCommand` property can be used to override the Install Command in the Project Settings dashboard for a given deployment. This
section.

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "installCommand": "npm install"
}
```

## images

The `images` property defines the behavior of [Vercel's native Image Optimization API](/docs/image-optimization), which allows on-demand

**Type**: `Object`

### Value definition

- `sizes` - **Required** - Array of allowed image widths. The Image Optimization API will return an error if the `w` parameter is not def
- `localPatterns` - Allow-list of local image paths which can be used with the Image Optimization API.
- `remotePatterns` - Allow-list of external domains which can be used with the Image Optimization API.
- `minimumCacheTTL` - Cache duration (in seconds) for the optimized images.
- `qualities` - Array of allowed image qualities. The Image Optimization API will return an error if the `q` parameter is not defined in
- `formats` - Supported output image formats. Allowed values are either `"image/avif"` and/or `"image/webp"`.
- `dangerouslyAllowSVG` - Allow SVG input image URLs. This is disabled by default for security purposes.
- `contentSecurityPolicy` - Specifies the [Content Security Policy](https://developer.mozilla.org/docs/Web/HTTP/CSP) of the optimized ima
- `contentDispositionType` - Specifies the value of the `"Content-Disposition"` response header. Allowed values are `"inline"` or `"attac

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "images": {
    "sizes": [256, 640, 1080, 2048, 3840],
    "localPatterns": [{
      "pathname": "^/assets/.*$",
      "search": ""
    }],
    "remotePatterns": [
      {
        "protocol": "https",
        "hostname": "example.com",
        "port": "",
        "pathname": "^/account123/.*$",
        "search": "?v=1"
      }
    ],
    "minimumCacheTTL": 60,
    "qualities": [25, 50, 75],
    "formats": ["image/webp"],
    "dangerouslyAllowSVG": false,
    "contentSecurityPolicy": "script-src 'none'; frame-src 'none'; sandbox;",
    "contentDispositionType": "inline"
  }
}
```

## outputDirectory

This value overrides the [Output Directory](/docs/deployments/configure-a-build#output-directory) in Project Settings.

**Type:** `string | null`

The `outputDirectory` property can be used to override the Output Directory in the Project Settings dashboard for a given deployment.

In the following example, the deployment will look for the `build` directory rather than the default `public` or `.` root directory. For

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "outputDirectory": "build"
}
```

## public

**Type**: `Boolean`.

**Default Value**: `false`.

When set to `true`, both the [source view](/docs/deployments/build-features#source-view) and [logs view](/docs/deployments/build-features

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "public": true
}
```

## redirects

**Type:** `Array` of redirect `Object`.

**Valid values:** a list of redirect definitions.

### Redirects examples

This example redirects requests to the path `/me` from your site's root to the `profile.html` file relative to your site's root with a [3

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "redirects": [
    { "source": "/me", "destination": "/profile.html", "permanent": false }
  ]
}
```

This example redirects requests to the path `/me` from your site's root to the `profile.html` file relative to your site's root with a [3

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "redirects": [
    { "source": "/me", "destination": "/profile.html", "permanent": true }
  ]
```

}
```

This example redirects requests to the path `/user` from your site's root to the api route `/api/user` relative to your site's root with

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "redirects": [
    { "source": "/user", "destination": "/api/user", "statusCode": 301 }
  ]
}
```

This example redirects requests to the path `/view-source` from your site's root to the absolute path `https://github.com/vercel/vercel`

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "redirects": [
    {
      "source": "/view-source",
      "destination": "https://github.com/vercel/vercel"
    }
  ]
}
```

This example redirects requests to all the paths (including all sub-directories and pages) from your site's root to the absolute path `ht

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "redirects": [
    {
      "source": "/(.*)",
      "destination": "https://vercel.com/docs"
    }
  ]
}
```

This example uses wildcard path matching to redirect requests to any path (including subdirectories) under `/blog/` from your site's root

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "redirects": [
    {
      "source": "/blog/:path*",
      "destination": "/news/:path*"
    }
  ]
}
```

This example uses regex path matching to redirect requests to any path under `/posts/` that only contain numerical digits from your site'

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "redirects": [
    {
      "source": "/post/:path(\\d{1,})",
      "destination": "/news/:path*"
    }
  ]
}
```

This example redirects requests to any path from your site's root that does not start with `/uk/` and has `x-vercel-ip-country` header va

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "redirects": [
    {
      "source": "/:path((?!uk/).*)",
      "has": [
        {
          "type": "header",
          "key": "x-vercel-ip-country",
          "value": "GB"
        }
      ],
      "destination": "/uk/:path*",
      "permanent": false
    }
  ]
}
```

> **💡 Note:** Using  does not yet work locally while using
> , but does work when deployed.

### Redirect object definition

| Property | Description |
| ---------- | ---------- |
| `source` | A pattern that matches each incoming pathname (excluding querystring). |
```

| `destination` | A location destination defined as an absolute pathname or external URL.
| `permanent`   | An optional boolean to toggle between permanent and temporary redirect (default `true`). When `true`, the status code i
| `statusCode`  | An optional integer to define the status code of the redirect. Used when you need a value other than 307/308 from `perm
| `has`         | An optional array of `has` objects with the `type`, `key` and `value` properties. Used for conditional redirects based
| `missing`     | An optional array of `missing` objects with the `type`, `key` and `value` properties. Used for conditional redirects ba

### Redirect `has` or `missing` object definition

If `value` is an object, it has one or more of the following fields:

This example uses the expressive `value` object to define a route that redirects users with a redirect status of 308 to `/end` only if th

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "redirects": [
    {
      "source": "/start",
      "destination": "/end",
      "has": [
        {
          "type": "header",
          "key": "X-Custom-Header",
          "value": {
            "pre": "valid",
            "suf": "value"
          }
        }
      ]
    }
  ]
}
```

Learn more about [redirects on Vercel](/docs/redirects) and see [limitations](/docs/redirects#limits).

## bulkRedirectsPath

Learn more about [bulk redirects on Vercel](/docs/redirects/bulk-redirects) and see [limits and pricing](/docs/redirects/bulk-redirects#l

**Type:** `string` path to a file or folder.

The `bulkRedirectsPath` property can be used to import many thousands of redirects per project. These redirects do not support wildcard o

CSV, JSON, and JSONL file formats are supported, and the redirect files can be generated at build time as long as they end up in the loca

### CSV

> **💡 Note:** CSV headers must match the field names below, can be specific in any order, and optional fields can be ommitted.

```csv filename="redirects.csv"
source,destination,permanent
/source/path,/destination/path,true
/source/path-2,https://destination-site.com/destination/path,true
```

### JSON

```json filename="redirects.json"
[
    {
        "source": "/source/path",
        "destination": "/destination/path",
        "permanent": true
    },
    {
        "source": "/source/path-2",
        "destination": "https://destination-site.com/destination/path",
        "permanent": true
    }
]
```

### JSONL

```jsonl filename="redirects.jsonl"
{"source": "/source/path", "destination": "/destination/path", "permanent": true}
{"source": "/source/path-2", "destination": "https://destination-site.com/destination/path", "permanent": true}
```

> **💡 Note:** Bulk redirects do not work locally while using 

### Bulk redirect field definition

| Field           | Type      | Required | Description
| --------------- | --------- | -------- | ----------------------------------------------------------------------------------------------------
| `source`        | `string`  | Yes      | An absolute path that matches each incoming pathname (excluding querystring). Max 2048 charact
| `destination`   | `string`  | Yes      | A location destination defined as an absolute pathname or external URL. Max 2048 characters.
| `permanent`     | `boolean` | No       | Toggle between permanent ([308](https://developer.mozilla.org/docs/Web/HTTP/Status/308)) and t
| `statusCode`    | `integer` | No       | Specify the exact status code. Can be [301](https://developer.mozilla.org/docs/Web/HTTP/Status
| `caseSensitive` | `boolean` | No       | Toggle whether source path matching is case sensitive. Default: `false`.
| `query`         | `boolean` | No       | Toggle whether to preserve the query string on the redirect. Default: `false`.

In order to improve space efficiency, all boolean values can be the single characters `t` (true) or `f` (false) while using the CSV forma

## regions

This value overrides the [Vercel Function Region](/docs/functions/regions) in Project Settings.

**Type:** `Array` of region identifier `String`.

**Valid values:** List of [regions](/docs/regions), defaults to `iad1`.

You can define the **regions** where your [Vercel functions](/docs/functions) are executed. Users on Pro and Enterprise can deploy to mul

Function responses [can be cached](/docs/cdn-cache) in the requested regions. Selecting a Vercel Function region does not impact static f

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "regions": ["sfo1"]
}
```

## functionFailoverRegions

Set this property to specify the [region](/docs/functions/regions) to which a Vercel Function should fallback when the default region(s)

**Type:** `Array` of region identifier `String`.

**Valid values:** List of [regions](/docs/regions).

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "functionFailoverRegions": ["iad1", "sfo1"]
}
```

These regions serve as a fallback to any regions specified in the [`regions` configuration](/docs/project-configuration#regions). The reg

- Vercel always attempts to invoke the function in the primary region. If you specify more than one primary region in the `regions` prope
- If all primary regions are unavailable, Vercel automatically fails over to the regions specified in `functionFailoverRegions`, selectin
- The order of the regions in `functionFailoverRegions` does not matter as Vercel automatically selects the region geographically closest

To learn more about automatic failover for Vercel Functions, see [Automatic failover](/docs/functions/configuring-functions/region#automa

Region failover is supported with Secure Compute, see [Region Failover](/docs/secure-compute#region-failover) to learn more.

## rewrites

**Type:** `Array` of rewrite `Object`.

**Valid values:** a list of rewrite definitions.

If [`cleanUrls`](/docs/project-configuration/vercel-json#cleanurls) is set to `true` in
your project's `vercel.json`, do not include the file extension in the source
or destination path. For example, `/about-our-company.html` would be
`/about-our-company`

### Rewrites examples

- This example rewrites requests to the path `/about` from your site's root to the `/about-our-company.html` file relative to your site's

  ```json filename="vercel.json"
  {
    "$schema": "https://openapi.vercel.sh/vercel.json",
    "rewrites": [
      { "source": "/about", "destination": "/about-our-company.html" }
    ]
  }
  ```

- This example rewrites all requests to the root path which is often used for a Single Page Application (SPA).

  ```json filename="vercel.json"
  {
    "$schema": "https://openapi.vercel.sh/vercel.json",
    "rewrites": [{ "source": "/(.*)", "destination": "/index.html" }]
  }
  ```

- This example rewrites requests to the paths under `/resize` that with 2 paths levels (defined as variables `width` and `height` that ca

  ```json filename="vercel.json"
  {
    "$schema": "https://openapi.vercel.sh/vercel.json",
    "rewrites": [
      { "source": "/resize/:width/:height", "destination": "/api/sharp" }
    ]
  }
  ```

- This example uses wildcard path matching to rewrite requests to any path (including subdirectories) under `/proxy/` from your site's ro

  ```json filename="vercel.json"
  {
    "$schema": "https://openapi.vercel.sh/vercel.json",
    "rewrites": [
      {
        "source": "/proxy/:match*",
        "destination": "https://example.com/:match*"
      }
    ]
  }
  ```

- This example rewrites requests to any path from your site's root that does not start with /uk/ and has x-vercel-ip-country header value

  ```json filename="vercel.json"
  ```

```
  {
    "$schema": "https://openapi.vercel.sh/vercel.json",
    "rewrites": [
      {
        "source": "/:path((?!uk/).*)",
        "has": [
          {
            "type": "header",
            "key": "x-vercel-ip-country",
            "value": "GB"
          }
        ],
        "destination": "/uk/:path*"
      }
    ]
  }
```

- This example rewrites requests to the path `/dashboard` from your site's root that **does not** have a cookie with key `auth_token` to

  ```json filename="vercel.json"
  {
    "$schema": "https://openapi.vercel.sh/vercel.json",
    "rewrites": [
      {
        "source": "/dashboard",
        "missing": [
          {
            "type": "cookie",
            "key": "auth_token"
          }
        ],
        "destination": "/login"
      }
    ]
  }
  ```

### Rewrite object definition

| Property | Description |
| --- | --- |
| `source` | A pattern that matches each incoming pathname (excluding querystring). |
| `destination` | A location destination defined as an absolute pathname or external URL. |
| `permanent` | A boolean to toggle between permanent and temporary redirect (default true). When `true`, the status code is [308](http |
| `has` | An optional array of `has` objects with the `type`, `key` and `value` properties. Used for conditional rewrites based o |
| `missing` | An optional array of `missing` objects with the `type`, `key` and `value` properties. Used for conditional rewrites bas |

### Rewrite `has` or `missing` object definition

If `value` is an object, it has one or more of the following fields:

This example demonstrates using the expressive `value` object to define a route that rewrites users to `/end` only if the `X-Custom-Heade

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "rewrites": [
    {
      "source": "/start",
      "destination": "/end",
      "has": [
        {
          "type": "header",
          "key": "X-Custom-Header",
          "value": {
            "pre": "valid",
            "suf": "value"
          }
        }
      ]
    }
  ]
}
```

The `source` property should **NOT** be a file because precedence is given to the filesystem prior to rewrites being applied. Instead, yo

> **💡 Note:** Using  does not yet work locally while using
> , but does work when deployed.

Learn more about [rewrites](/docs/rewrites) on Vercel.

## trailingSlash

**Type**: `Boolean`.

**Default Value**: `undefined`.

### false

When `trailingSlash: false`, visiting a path that ends with a forward slash will respond with a 308 status code and redirect to the path w

For example, the `/about/` path will redirect to `/about`.

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "trailingSlash": false
}
```

```

### true

When `trailingSlash: true`, visiting a path that does not end with a forward slash will respond with a 308 status code and redirect to th

For example, the `/about` path will redirect to `/about/`.

However, paths with a file extension will not redirect to a trailing slash.

For example, the `/about/styles.css` path will not redirect, but the `/about/styles` path will redirect to `/about/styles/`.

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "trailingSlash": true
}
```

### undefined

When `trailingSlash: undefined`, visiting a path with or without a trailing slash will not redirect.

For example, both `/about` and `/about/` will serve the same content without redirecting.

This is not recommended because it could lead to search engines indexing two different pages with duplicate content.

## Legacy

Legacy properties are still supported for backwards compatibility, but are deprecated.

### name

The `name` property has been deprecated in favor of [Project Linking](/docs/cli/project-linking), which allows you to link a Vercel proje

**Type**: `String`.

**Valid values**: string name for the deployment.

**Limits**:

- A maximum length of 52 characters
- Only lower case alphanumeric characters or hyphens are allowed
- Cannot begin or end with a hyphen, or contain multiple consecutive hyphens

The prefix for all new deployment instances. Vercel CLI usually generates this field automatically based on the name of the directory. Bu

The defined name is also used to organize the deployment into [a project](/docs/projects/overview).

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "name": "example-app"
}
```

### version

The `version` property should not be used anymore.

**Type**: `Number`.

**Valid values**: `1`, `2`.

Specifies the Vercel Platform version the deployment should use.

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "version": 2
}
```

### alias

The `alias` property should not be used anymore. To assign a custom Domain to your project, please [define it in the Project Settings](/d

**Type**: `Array` or `String`.

**Valid values**: [domain names](/docs/domains/add-a-domain) (optionally including subdomains) added to the account, or a string for a su

**Limit**: A maximum of 64 aliases in the array.

The alias or aliases are applied automatically using [Vercel for GitHub](/docs/git/vercel-for-github), [Vercel for GitLab](/docs/git/verc

You can deploy to the defined aliases using [Vercel CLI](/docs/cli) by setting the [production deployment environment target](/docs/domai

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "alias": ["my-domain.com", "my-alias"]
}
```

### scope

The `scope` property has been deprecated in favor of [Project Linking](/docs/cli/project-linking), which allows you to link a Vercel proj

**Type**: `String`.
```

**Valid values**: For teams, either an ID or slug. For users, either a email address, username, or ID.

This property determines the scope ([Hobby team](/docs/accounts/create-an-account#creating-a-hobby-account) or [team](/docs/accounts/crea

It also affects any other actions that the user takes within the directory that contains this configuration (e.g. listing [environment va

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "scope": "my-team"
}
```

Deployments made through [Git](/docs/git) will **ignore** the `scope` property because the repository is already connected to [project](/

### env

We recommend against using this property. To add custom environment variables to your project [define them in the Project Settings](/docs

**Type:** `Object` of `String` keys and values.

**Valid values:** environment keys and values.

Environment variables passed to the invoked [Vercel functions](/docs/functions).

This example will pass the `MY_KEY` static env to all [Vercel functions](/docs/functions) and the `SECRET` resolved from the `my-secret-n

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "env": {
    "MY_KEY": "this is the value",
    "SECRET": "@my-secret-name"
  }
}
```

### build.env

We recommend against using this property. To add custom environment variables to your project [define them in the Project Settings](/docs

**Type:** `Object` of `String` keys and values inside the `build` `Object`.

**Valid values:** environment keys and values.

[Environment variables](/docs/environment-variables) passed to the [Build](/docs/deployments/configure-a-build) processes.

The following example will pass the `MY_KEY` environment variable to all [Builds](/docs/deployments/configure-a-build) and the `SECRET` r

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "env": {
    "MY_KEY": "this is the value",
    "SECRET": "@my-secret-name"
  }
}
```

### builds

We recommend against using this property. To customize Vercel functions, please use the [functions](#functions) property instead. If you'

**Type:** `Array` of build `Object`.

**Valid values:** a list of build descriptions whose `src` references valid source files.

#### Build object definition

- `src` (`String`): A glob expression or pathname. If more than one file is resolved, one build will be created per matched file. It can
- `use` (`String`): An npm module to be installed by the build process. It can include a semver compatible version (e.g.: `@org/proj@1`).
- `config` (`Object`): Optionally, an object including arbitrary metadata to be passed to the Builder.

The following will include all HTML files as-is (to be served statically), and build all Python files and JS files into [Vercel functions

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "builds": [
    { "src": "*.html", "use": "@vercel/static" },
    { "src": "*.py", "use": "@vercel/python" },
    { "src": "*.js", "use": "@vercel/node" }
  ]
}
```

When at least one `builds` item is specified, only the outputs of the build processes will be included in the resulting deployment as a s

### routes

We recommend using [cleanUrls](#cleanurls), [trailingSlash](#trailingslash), [redirects](#redirects), [rewrites](#rewrites), and/or [head

The `routes` property is only meant to be used for advanced integration purposes, such as the [Build Output API](/docs/build-output-api/v

See the [upgrading routes section](#upgrading-legacy-routes) to learn how to migrate away from this property.

**Type:** `Array` of route `Object`.

**Valid values:** a list of route definitions.

#### Route object definition

- `src`: A [PCRE-compatible regular expression](https://www.pcre.org/original/doc/html/pcrepattern.html) that matches each incoming pathn
- `methods`: A set of HTTP method types. If no method is provided, requests with any HTTP method will be a candidate for the route.
- `dest`: A destination pathname or full URL, including querystring, with the ability to embed capture groups as $1, $2…
- `headers`: A set of headers to apply for responses.
- `status`: A status code to respond with. Can be used in tandem with `Location:` header to implement redirects.
- `continue`: A boolean to change matching behavior. If `true`, routing will continue even when the `src` is matched.
- `has`: An optional array of `has` objects with the `type`, `key` and `value` properties. Used for conditional path matching based on th
- `missing`: An optional array of `missing` objects with the `type`, `key` and `value` properties. Used for conditional path matching bas
- `mitigate`: An optional object with the property `action`, which can either be "challenge" or "deny". These perform [mitigation actions
- `transforms`: An optional array of `transform` objects to apply. Transform rules let you append, set, or remove request/response header

Routes are processed in the order they are defined in the array, so wildcard/catch-all patterns should usually be last.

#### Route has and missing object definition

If `value` is an object, it has one or more of the following fields:

This example uses the expressive `value` object to define a route that will only rewrite users to `/end` if the `X-Custom-Header` header'

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "routes": [
    {
      "src": "/start",
      "dest": "/end",
      "has": [
        {
          "type": "header",
          "key": "X-Custom-Header",
          "value": {
            "pre": "valid",
            "suf": "value"
          }
        }
      ]
    }
  ]
}
```

This example configures custom routes that map to static files and [Vercel functions](/docs/functions):

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "routes": [
    {
      "src": "/redirect",
      "status": 308,
      "headers": { "Location": "https://example.com/" }
    },
    {
      "src": "/custom-page",
      "headers": { "cache-control": "s-maxage=1000" },
      "dest": "/index.html"
    },
    { "src": "/api", "dest": "/my-api.js" },
    { "src": "/users", "methods": ["POST"], "dest": "/users-api.js" },
    { "src": "/users/(?<id>[^/]*)", "dest": "/users-api.js?id=$id" },
    { "src": "/legacy", "status": 404 },
    { "src": "/.*", "dest": "https://my-old-site.com" }
  ]
}
```

### Transform object definition

| Property | Type | Description |
| -------- | ------------------------------------ | ---------------------------------------------------------------------------------------- |
| `type` | `String` | Must be either `request.query`, `request.headers`, or `response.headers`. This speci |
| `op` | `String` | These specify the possible operations:- `append` appends `args` to the value of the |
| `target` | `Object` | An object with key `key`, which is either a `String` or an `Object`. If it is a stri |
| `args` | `String` or `String[]` or `undefined` | If `args` is a string or string array, it will be used as the value for the target a |

#### Transform target object definition

Target is an object with a `key` property. For the `set` operation, the `key` property is used as the header or query key. For other oper

| Property | Type | Description |
| -------- | ------------------- | ---------------------------------------------------------------------------------------------------------- |
| `key` | `String` or `Object` | It may be a string or an object. If it is an object, it must have one or more of the properties defin |

#### Transform key object definition

When the `key` property is an object, it can contain one or more of the following conditional matching properties:

| Property | Type | Description |
| -------- | ------------------- | ------------------------------------------ |
| `eq` | `String` or `Number` | Check equality on a value |
| `neq` | `String` | Check inequality on a value |
| `inc` | `String[]` | Check inclusion in an array of values |
| `ninc` | `String[]` | Check non-inclusion in an array of values |
| `pre` | `String` | Check if value starts with a prefix |
| `suf` | `String` | Check if value ends with a suffix |
| `gt` | `Number` | Check if value is greater than |
| `gte` | `Number` | Check if value is greater than or equal to |
| `lt` | `Number` | Check if value is less than |

| `lte`    | `Number`             | Check if value is less than or equal to    |

#### Transform examples

These examples demonstrate practical use-cases for route transforms.

In this example, you remove the incoming request header `x-custom-header` from all requests and responses to the `/home` route:

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "routes": [
    {
      "src": "/home",
      "transforms": [
        {
          "type": "request.headers",
          "op": "delete",
          "target": {
            "key": "x-custom-header"
          }
        },
        {
          "type": "response.headers",
          "op": "delete",
          "target": {
            "key": "x-custom-header"
          }
        }
      ]
    }
  ]
}
```

In this example, you override the incoming query parameter `theme` to `dark` for all requests to the `/home` route, and set if it doesn't

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "routes": [
    {
      "src": "/home",
      "transforms": [
        {
          "type": "request.query",
          "op": "set",
          "target": {
            "key": "theme"
          },
          "args": "dark"
        }
      ]
    }
  ]
}
```

In this example, you append multiple values to the incoming request header `x-content-type-options` for all requests to the `/home` route

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "routes": [
    {
      "src": "/home",
      "transforms": [
        {
          "type": "request.headers",
          "op": "append",
          "target": {
            "key": "x-content-type-options"
          },
          "args": ["nosniff", "no-sniff"]
        }
      ]
    }
  ]
}
```

In this example, you delete any header that begins with `x-react-router-` for all requests to the `/home` route:

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "routes": [
    {
      "src": "/home",
      "transforms": [
        {
          "type": "request.headers",
          "op": "delete",
          "target": {
            "key": {
              "pre": "x-react-router-"
            }
          }
        }
      ]
```

```
    }
  ]
}
```

You can integrate transforms with existing matching capabilities through the [`has` and `missing` properties for routes](/docs/project-co

### Upgrading legacy routes

In most cases, you can upgrade legacy `routes` usage to the newer [`rewrites`](/docs/project-configuration#rewrites), [`redirects`](/docs

Here are some examples that show how to upgrade legacy `routes` to the equivalent new property.

#### Route Parameters

With `routes`, you use a [PCRE Regex](https://en.wikipedia.org/wiki/Perl_Compatible_Regular_Expressions) named group to match the ID and

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "routes": [{ "src": "/product/(?<id>[^/]+)", "dest": "/api/product?id=$id" }]
}
```

With [`rewrites`](/docs/project-configuration#rewrites), named parameters are automatically passed in the query string. The following exa

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "rewrites": [{ "source": "/product/:id", "destination": "/api/product" }]
}
```

#### Legacy redirects

With `routes`, you specify the status code to use a 307 Temporary Redirect. Also, this redirect needs to be defined before other routes.

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "routes": [
    {
      "src": "/posts/(.*)",
      "headers": { "Location": "/blog/$1" },
      "status": 307
    }
  ]
}
```

With [`redirects`](/docs/project-configuration#redirects), you disable the `permanent` property to use a 307 Temporary Redirect. Also, `r

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "redirects": [
    {
      "source": "/posts/:id",
      "destination": "/blog/:id",
      "permanent": false
    }
  ]
}
```

#### Legacy SPA Fallback

With `routes`, you use `"handle": "filesystem"` to give precedence to the filesystem and exit early if the requested path matched a file.

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "routes": [
    { "handle": "filesystem" },
    { "src": "/(.*)", "dest": "/index.html" }
  ]
}
```

With [`rewrites`](/docs/project-configuration#rewrites), the filesystem check is the default behavior. If you want to change the name of

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "rewrites": [{ "source": "/(.*)", "destination": "/index.html" }]
}
```

#### Legacy Headers

With `routes`, you use `"continue": true` to prevent stopping at the first match. The following example adds `Cache-Control` headers to t

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "routes": [
    {
      "src": "/favicon.ico",
      "headers": { "Cache-Control": "public, max-age=3600" },
      "continue": true
```

```
    },
    {
      "src": "/assets/(.*)",
      "headers": { "Cache-Control": "public, max-age=31556952, immutable" },
      "continue": true
    }
  ]
}
```

With [`headers`](/docs/project-configuration#headers), this is no longer necessary since that is the default behavior. The following exam

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "headers": [
    {
      "source": "/favicon.ico",
      "headers": [
        {
          "key": "Cache-Control",
          "value": "public, max-age=3600"
        }
      ]
    },
    {
      "source": "/assets/(.*)",
      "headers": [
        {
          "key": "Cache-Control",
          "value": "public, max-age=31556952, immutable"
        }
      ]
    }
  ]
}
```

#### Legacy Pattern Matching

With `routes`, you need to escape a dot with two backslashes, otherwise it would match any character [PCRE Regex](https://en.wikipedia.or

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "routes": [{ "src": "/atom\\.xml", "dest": "/api/rss" }]
}
```

With [`rewrites`](/docs/project-configuration#rewrites), the `.` is not a special character so it does not need to be escaped. The follow

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "rewrites": [{ "source": "/atom.xml", "destination": "/api/rss" }]
}
```

#### Legacy Negative Lookahead

With `routes`, you use [PCRE Regex](https://en.wikipedia.org/wiki/Perl_Compatible_Regular_Expressions) negative lookahead. The following

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "routes": [{ "src": "/(?!maintenance)", "dest": "/maintenance" }]
}
```

With [`rewrites`](/docs/project-configuration#rewrites), the Regex needs to be wrapped. The following example is equivalent to the legacy

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "rewrites": [
    { "source": "/((?!maintenance).*)", "destination": "/maintenance" }
  ]
}
```

#### Legacy Case Sensitivity

With `routes`, the `src` property is case-insensitive leading to duplicate content, where multiple request paths with difference cases se

With [`rewrites`](/docs/project-configuration#rewrites) / [`redirects`](/docs/project-configuration#redirects) / [`headers`](/docs/projec

---
title: "Programmatic Configuration with vercel.ts"
description: "Define your Vercel configuration in vercel.ts with @vercel/config for type-safe routing and build settings."
last_updated: "2026-01-16T02:19:34.781Z"
source: "https://vercel.com/docs/project-configuration/vercel-ts"
---

# Programmatic Configuration with vercel.ts

The `vercel.ts` file lets you configure and override the default behavior of Vercel from within your project. Unlike `vercel.json`, which

## Getting Started

### Requirements

Use only one configuration file: `vercel.ts` or `vercel.json`.

You can have any sort of code in your `vercel.ts` file, but the final set of rules and configuration properties must be in a `config` str

```typescript
export const config: VercelConfig = {
  buildCommand: 'npm run build',
  cleanUrls: true,
  trailingSlash: false,
  // See the sections below for all available options
};
```

To migrate from `vercel.json`, copy its contents into your `config` export, then add new capabilities as needed.

### Install @vercel/config

Install the NPM package to get access to types and helpers.

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i @vercel/config
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @vercel/config
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @vercel/config
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i @vercel/config
    ```
  </Code>
</CodeBlock>

Create a `vercel.ts` in your project root and export a typed config. The example below shows how to configure build commands, framework s

> **💡 Note:** You can also use `vercel.js`, `vercel.mjs`, `vercel.cjs`, or `vercel.mts` to create this configuration file.

```typescript filename="vercel.ts"
import { routes, deploymentEnv, type VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  buildCommand: 'npm run build',
  framework: 'nextjs',

  rewrites: [
    routes.rewrite('/api/(.*)', 'https://backend.api.example.com/$1'),
    routes.rewrite('/(.*)', 'https://api.example.com/$1', {
      requestHeaders: {
        authorization: `Bearer ${deploymentEnv('API_TOKEN')}`,
      },
    }),
    routes.rewrite(
      '/users/:userId/posts/:postId',
      'https://api.example.com/users/$1/posts/$2',
      ({ userId, postId }) => ({
        requestHeaders: {
          'x-user-id': userId,
          'x-post-id': postId,
          authorization: `Bearer ${deploymentEnv('API_KEY')}`,
        },
      }),
    ),
  ],

  redirects: [routes.redirect('/old-docs', '/docs', { permanent: true })],

  headers: [
    routes.cacheControl('/static/(.*)', {
      public: true,
      maxAge: '1 week',
      immutable: true,
    }),
  ],

  crons: [{ path: '/api/cleanup', schedule: '0 0 * * *' }],
};
```

### Migrating from vercel.json

To migrate from an existing `vercel.json`, paste its contents into a `config` export in a new vercel.ts file:

```typescript filename="vercel.ts"
export const config = {
  buildCommand: 'pnpm run generate-config',
  outputDirectory: ".next",
  headers: [
    {
      source: "/(.*)\\\\.(js|css|jpg|jpeg|gif|png|svg|txt|ttf|woff2|webmanifest)",
```

```
      headers: [
        {
          key: "Cache-Control",
          value: "public, max-age=2592000, s-maxage=2592000"
        }
      ]
    }
  ]
}
```

Then install the `@vercel/config` package and enhance your configuration:

```typescript filename="vercel.ts"
import { routes, type VercelConfig } from '@vercel/config/v1'

export const config: VercelConfig = {
  buildCommand: 'pnpm run generate-config',
  outputDirectory: `.${process.env.framework}`,
  headers: [
    routes.cacheControl(
      '/(.*)\\\\.(js|css|jpg|jpeg|gif|png|svg|txt|ttf|woff2|webmanifest)',
      {
        public: true,
        maxAge: '30days',
        sMaxAge: '30days'
      }
    )
  ]
}
```

## Config export options



## schema autocomplete

Via the types imported from the `@vercel/config` package, autocomplete for all config properties and helpers are automatically available

```typescript filename="vercel.ts"
import { routes, type VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  rewrites: [routes.rewrite('/about', '/about-our-company.html')],
  // add more properties here
};
```

## buildCommand

**Type:** `string | null`

The `buildCommand` property can be used to override the Build Command in the Project Settings dashboard, and the `build` script from the

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  buildCommand: 'next build',
};
```

This value overrides the [Build Command](/docs/deployments/configure-a-build#build-command) in Project Settings.

## bunVersion

**Type:** `string`

**Value:** `"1.x"`

The `bunVersion` property configures your project to use the Bun runtime instead of Node.js. When set, all [Vercel Functions](/docs/funct

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  bunVersion: '1.x',
};
```

```

> **💡 Note:** Vercel manages the Bun minor and patch versions automatically. `1.x` is the
> only valid value currently.

When using Next.js with [ISR](/docs/incremental-static-regeneration) (Incremental Static Regeneration), you must also update your `build`

```json filename="package.json"
{
  "scripts": {
    "dev": "bun run --bun next dev",
    "build": "bun run --bun next build"
  }
}
```

To learn more about using Bun with Vercel Functions, see the [Bun runtime documentation](/docs/functions/runtimes/bun).

## cleanUrls

**Type**: `Boolean`.

**Default Value**: `false`.

When set to `true`, all HTML files and Vercel functions will have their extension removed. When visiting a path that ends with the extens

For example, a static file named `about.html` will be served when visiting the `/about` path. Visiting `/about.html` will redirect to `/a

Similarly, a Vercel Function named `api/user.go` will be served when visiting `/api/user`. Visiting `/api/user.go` will redirect to `/api

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  cleanUrls: true,
};
```

If you are using Next.js and running `vercel dev`, you will get a 404 error when visiting a route configured with `cleanUrls` locally. It

## crons

Used to configure [cron jobs](/docs/cron-jobs) for the production deployment of a project.

**Type**: `Array` of cron `Object`.

**Limits**:

- A maximum of string length of 512 for the `path` value.
- A maximum of string length of 256 for the `schedule` value.

### Cron object definition

- `path` - **Required** - The path to invoke when the cron job is triggered. Must start with `/`.
- `schedule` - **Required** - The [cron schedule expression](/docs/cron-jobs#cron-expressions) to use for the cron job.

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  crons: [
    {
      path: '/api/every-minute',
      schedule: '* * * * *',
    },
    {
      path: '/api/every-hour',
      schedule: '0 * * * *',
    },
    {
      path: '/api/every-day',
      schedule: '0 0 * * *',
    },
  ],
};
```

## devCommand

This value overrides the [Development Command](/docs/deployments/configure-a-build#development-command) in Project Settings.

**Type:** `string | null`

The `devCommand` property can be used to override the Development Command in the Project Settings dashboard. For more information on the

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  devCommand: 'next dev',
};
```

## fluid

This value allows you to enable [Fluid compute](/docs/fluid-compute) programmatically.

**Type:** `boolean | null`

The `fluid` property allows you to test Fluid compute on a per-deployment or per [custom environment](/docs/deployments/environments#cust
```

> **💡 Note:** As of April 23, 2025, Fluid compute is enabled by default for new projects.

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  fluid: true,
};
```

## framework

This value overrides the [Framework](/docs/deployments/configure-a-build#framework-preset) in Project Settings.

**Type:** `string | null`

Available framework slugs:

The `framework` property can be used to override the Framework Preset in the Project Settings dashboard. The value must be a valid framew

> **💡 Note:** To select "Other" as the Framework Preset, use .

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  framework: 'nextjs',
};
```

## functions

**Type:** `Object` of key `String` and value `Object`.

### Key definition

A [glob](https://github.com/isaacs/node-glob#glob-primer) pattern that matches the paths of the Vercel functions you would like to custom

- `api/*.js` (matches one level e.g. `api/hello.js` but not `api/hello/world.js`)
- `api/**/*.ts` (matches all levels `api/hello.ts` and `api/hello/world.ts`)
- `src/pages/**/*` (matches all functions from `src/pages`)
- `api/test.js`

### Value definition

- `runtime` (optional): The npm package name of a [Runtime](/docs/functions/runtimes), including its version.
- `memory`: Memory cannot be set in `vercel.ts` with [Fluid compute](/docs/fluid-compute) enabled. Instead set it in the **Functions** ta
- `maxDuration` (optional): An integer defining how long your Vercel Function should be allowed to run on every request in seconds (betwe
- `supportsCancellation` (optional): A boolean defining whether your Vercel Function should [support request cancellation](/docs/function
- `includeFiles` (optional): A [glob](https://github.com/isaacs/node-glob#glob-primer) pattern to match files that should be included in
- `excludeFiles` (optional): A [glob](https://github.com/isaacs/node-glob#glob-primer) pattern to match files that should be excluded fro

### Description

By default, no configuration is needed to deploy Vercel functions to Vercel.

For all [officially supported runtimes](/docs/functions/runtimes), the only requirement is to create an `api` directory at the root of yo

The `functions` property cannot be used in combination with `builds`. Since the latter is a legacy configuration property, we recommend d

Because [Incremental Static Regeneration (ISR)](/docs/incremental-static-regeneration) uses Vercel functions, the same configurations app

When deployed, each Vercel Function receives the following properties:

- **Memory:** 1024 MB (1 GB) - **(Optional)**
- **Maximum Duration:** 10s default - 60s / 1 minute (Hobby), 15s default - 300s / 5 minutes (Pro), or 15s default - 900s / 15 minutes (E

To configure them, you can add the `functions` property.

#### `functions` property with Vercel functions

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  functions: {
    'api/test.js': {
      memory: 3009,
      maxDuration: 30,
    },
    'api/*.js': {
      memory: 3009,
      maxDuration: 30,
    },
  },
};
```

#### `functions` property with ISR

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  functions: {
    'pages/blog/[hello].tsx': {
      memory: 1024,
    },
    'src/pages/isr/**/*': {
```

```
      maxDuration: 10,
    },
  },
};
```

### Using unsupported runtimes

In order to use a runtime that is not [officially supported](/docs/functions/runtimes), you can add a `runtime` property to the definitio

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  functions: {
    'api/test.php': {
      runtime: 'vercel-php@0.5.2',
    },
  },
};
```

In the example above, the `api/test.php` Vercel Function does not use one of the [officially supported runtimes](/docs/functions/runtimes

For more information on Runtimes, see the [Runtimes documentation](/docs/functions/runtimes):

## headers

**Type:** `Array` of header `Object`.

**Valid values:** a list of header definitions.

```typescript filename="vercel.ts"
import { routes, type VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  headers: [
    routes.header('/service-worker.js', [
      { key: 'Cache-Control', value: 'public, max-age=0, must-revalidate' },
    ]),
    routes.header('/(.*)', [
      { key: 'X-Content-Type-Options', value: 'nosniff' },
      { key: 'X-Frame-Options', value: 'DENY' },
      { key: 'X-XSS-Protection', value: '1; mode=block' },
    ]),
    routes.header('/:path*', [{ key: 'x-authorized', value: 'true' }], {
      has: [{ type: 'query', key: 'authorized' }],
    }),
  ],
};
```

This example configures custom response headers for static files, [Vercel functions](/docs/functions), and a wildcard that matches all ro

### Header object definition

| Property  | Description
| --------- | ----------------------------------------------------------------------------------------------------------------------------------------
| `source`  | A pattern that matches each incoming pathname (excluding querystring).
| `headers` | A non-empty array of key/value pairs representing each response header.
| `has`     | An optional array of `has` objects with the `type`, `key` and `value` properties. Used for conditional path matching based 
| `missing` | An optional array of `missing` objects with the `type`, `key` and `value` properties. Used for conditional path matching ba

### Header `has` or `missing` object definition

If `value` is an object, it has one or more of the following fields:

This example demonstrates using the expressive `value` object to append the header `x-authorized: true` if the `X-Custom-Header` request

```typescript filename="vercel.ts"
import { routes, type VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  headers: [
    routes.header('/:path*', [{ key: 'x-authorized', value: 'true' }], {
      has: [
        {
          type: 'header',
          key: 'X-Custom-Header',
          value: { pre: 'valid', suf: 'value' },
        },
      ],
    }),
  ],
};
```

Learn more about [rewrites](/docs/headers) on Vercel and see [limitations](/docs/cdn-cache#limits).

## ignoreCommand

This value overrides the [Ignored Build Step](/docs/project-configuration/git-settings#ignored-build-step) in Project Settings.

**Type:** `string | null`

This `ignoreCommand` property will override the Command for Ignoring the Build Step for a given deployment. When the command exits with c

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';
```

```
export const config: VercelConfig = {
  ignoreCommand: 'git diff --quiet HEAD^ HEAD ./',
};
```

## installCommand

This value overrides the [Install Command](/docs/deployments/configure-a-build#install-command) in Project Settings.

**Type:** `string | null`

The `installCommand` property can be used to override the Install Command in the Project Settings dashboard for a given deployment. This
section.

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  installCommand: 'npm install',
};
```

## images

The `images` property defines the behavior of [Vercel's native Image Optimization API](/docs/image-optimization), which allows on-demand

**Type**: `Object`

### Value definition

- `sizes` - **Required** - Array of allowed image widths. The Image Optimization API will return an error if the `w` parameter is not def
- `localPatterns` - Allow-list of local image paths which can be used with the Image Optimization API.
- `remotePatterns` - Allow-list of external domains which can be used with the Image Optimization API.
- `minimumCacheTTL` - Cache duration (in seconds) for the optimized images.
- `qualities` - Array of allowed image qualities. The Image Optimization API will return an error if the `q` parameter is not defined in
- `formats` - Supported output image formats. Allowed values are either `"image/avif"` and/or `"image/webp"`.
- `dangerouslyAllowSVG` - Allow SVG input image URLs. This is disabled by default for security purposes.
- `contentSecurityPolicy` - Specifies the [Content Security Policy](https://developer.mozilla.org/docs/Web/HTTP/CSP) of the optimized ima
- `contentDispositionType` - Specifies the value of the `"Content-Disposition"` response header. Allowed values are `"inline"` or `"attac

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  images: {
    sizes: [256, 640, 1080, 2048, 3840],
    localPatterns: [
      {
        pathname: '^/assets/.*$',
        search: '',
      },
    ],
    remotePatterns: [
      {
        protocol: 'https',
        hostname: 'example.com',
        port: '',
        pathname: '^/account123/.*$',
        search: '?v=1',
      },
    ],
    minimumCacheTTL: 60,
    qualities: [25, 50, 75],
    formats: ['image/webp'],
    dangerouslyAllowSVG: false,
    contentSecurityPolicy: "script-src 'none'; frame-src 'none'; sandbox;",
    contentDispositionType: 'inline',
  },
};
```

## outputDirectory

This value overrides the [Output Directory](/docs/deployments/configure-a-build#output-directory) in Project Settings.

**Type:** `string | null`

The `outputDirectory` property can be used to override the Output Directory in the Project Settings dashboard for a given deployment.

In the following example, the deployment will look for the `build` directory rather than the default `public` or `.` root directory. For

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  outputDirectory: 'build',
};
```

## public

**Type**: `Boolean`.

**Default Value**: `false`.

When set to `true`, both the [source view](/docs/deployments/build-features#source-view) and [logs view](/docs/deployments/build-features

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';
```

```
export const config: VercelConfig = {
  public: true,
};
```

## redirects

**Type:** `Array` of redirect `Object`.

**Valid values:** a list of redirect definitions.

### Redirects examples

This example redirects requests to the path `/me` from your site's root to the `profile.html` file relative to your site's root with a [3

```typescript filename="vercel.ts"
import { routes, type VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  redirects: [
    routes.redirect('/me', '/profile.html', { permanent: false }),
  ],
};
```

This example redirects requests to the path `/me` from your site's root to the `profile.html` file relative to your site's root with a [3

```typescript filename="vercel.ts"
import { routes, type VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  redirects: [
    routes.redirect('/me', '/profile.html', { permanent: true }),
  ],
};
```

This example redirects requests to the path `/user` from your site's root to the api route `/api/user` relative to your site's root with

```typescript filename="vercel.ts"
import { routes, type VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  redirects: [
    routes.redirect('/user', '/api/user', { statusCode: 301 }),
  ],
};
```

This example redirects requests to the path `/view-source` from your site's root to the absolute path `https://github.com/vercel/vercel` (

```typescript filename="vercel.ts"
import { routes, type VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  redirects: [
    routes.redirect('/view-source', 'https://github.com/vercel/vercel'),
  ],
};
```

This example redirects requests to all the paths (including all sub-directories and pages) from your site's root to the absolute path `ht

```typescript filename="vercel.ts"
import { routes, type VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  redirects: [
    routes.redirect('/(.*)', 'https://vercel.com/docs'),
  ],
};
```

This example uses wildcard path matching to redirect requests to any path (including subdirectories) under `/blog/` from your site's root

```typescript filename="vercel.ts"
import { routes, type VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  redirects: [
    routes.redirect('/blog/:path*', '/news/:path*'),
  ],
};
```

This example uses regex path matching to redirect requests to any path under `/posts/` that only contain numerical digits from your site'

```typescript filename="vercel.ts"
import { routes, type VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  redirects: [
    routes.redirect('/post/:path(\\d{1,})', '/news/:path*'),
  ],
};
```

This example redirects requests to any path from your site's root that does not start with `/uk/` and has `x-vercel-ip-country` header va

```typescript filename="vercel.ts"
```

```
import { routes, type VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  redirects: [
    routes.redirect('/:path((?!uk/).*)', '/uk/:path*', {
      has: [
        {
          type: 'header',
          key: 'x-vercel-ip-country',
          value: 'GB',
        },
      ],
      permanent: false,
    }),
  ],
};
```

> **💡 Note:** Using  does not yet work locally while using
> , but does work when deployed.

### Redirect object definition

| Property | Description |
| --- | --- |
| `source` | A pattern that matches each incoming pathname (excluding querystring). |
| `destination` | A location destination defined as an absolute pathname or external URL. |
| `permanent` | An optional boolean to toggle between permanent and temporary redirect (default `true`). When `true`, the status code i |
| `statusCode` | An optional integer to define the status code of the redirect. Used when you need a value other than 307/308 from `perm |
| `has` | An optional array of `has` objects with the `type`, `key` and `value` properties. Used for conditional redirects based |
| `missing` | An optional array of `missing` objects with the `type`, `key` and `value` properties. Used for conditional redirects ba |

### Redirect `has` or `missing` object definition

If `value` is an object, it has one or more of the following fields:

This example uses the expressive `value` object to define a route that redirects users with a redirect status of 308 to `/end` only if th

```typescript filename="vercel.ts"
import { routes, type VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  redirects: [
    routes.redirect('/start', '/end', {
      has: [
        {
          type: 'header',
          key: 'X-Custom-Header',
          value: { pre: 'valid', suf: 'value' },
        },
      ],
    }),
  ],
};
```

Learn more about [redirects on Vercel](/docs/redirects) and see [limitations](/docs/redirects#limits).

## bulkRedirectsPath

Learn more about [bulk redirects on Vercel](/docs/redirects/bulk-redirects) and see [limits and pricing](/docs/redirects/bulk-redirects#l

**Type:** `string` path to a file or folder.

The `bulkRedirectsPath` property can be used to import many thousands of redirects per project. These redirects do not support wildcard o

CSV, JSON, and JSONL file formats are supported, and the redirect files can be generated at build time as long as they end up in the loca

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  bulkRedirectsPath: 'redirects.csv',
};
```

### CSV

> **💡 Note:** CSV headers must match the field names below, can be specific in any order, and optional fields can be ommitted.

```csv filename="redirects.csv"
source,destination,permanent
/source/path,/destination/path,true
/source/path-2,https://destination-site.com/destination/path,true
```

### JSON

```json filename="redirects.json"
[
    {
        "source": "/source/path",
        "destination": "/destination/path",
        "permanent": true
    },
    {
        "source": "/source/path-2",
        "destination": "https://destination-site.com/destination/path",
        "permanent": true
    }
```

]
```

### JSONL

```jsonl filename="redirects.jsonl"
{"source": "/source/path", "destination": "/destination/path", "permanent": true}
{"source": "/source/path-2", "destination": "https://destination-site.com/destination/path", "permanent": true}
```

> **💡 Note:** Bulk redirects do not work locally while using&#x20;

### Bulk redirect field definition

| Field | Type | Required | Description |
| --- | --- | --- | --- |
| `source` | `string` | Yes | An absolute path that matches each incoming pathname (excluding querystring). Max 2048 charact |
| `destination` | `string` | Yes | A location destination defined as an absolute pathname or external URL. Max 2048 characters. |
| `permanent` | `boolean` | No | Toggle between permanent ([308](https://developer.mozilla.org/docs/Web/HTTP/Status/308)) and t |
| `statusCode` | `integer` | No | Specify the exact status code. Can be [301](https://developer.mozilla.org/docs/Web/HTTP/Status |
| `caseSensitive` | `boolean` | No | Toggle whether source path matching is case sensitive. Default: `false`. |
| `query` | `boolean` | No | Toggle whether to preserve the query string on the redirect. Default: `false`. |

In order to improve space efficiency, all boolean values can be the single characters `t` (true) or `f` (false) while using the CSV forma

## regions

This value overrides the [Vercel Function Region](/docs/functions/regions) in Project Settings.

**Type:** `Array` of region identifier `String`.

**Valid values:** List of [regions](/docs/regions), defaults to `iad1`.

You can define the **regions** where your [Vercel functions](/docs/functions) are executed. Users on Pro and Enterprise can deploy to mul

Function responses [can be cached](/docs/cdn-cache) in the requested regions. Selecting a Vercel Function region does not impact static f

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  regions: ['sfo1'],
};
```

## functionFailoverRegions

Set this property to specify the [region](/docs/functions/regions) to which a Vercel Function should fallback when the default region(s)

**Type:** `Array` of region identifier `String`.

**Valid values:** List of [regions](/docs/regions).

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  functionFailoverRegions: ['iad1', 'sfo1'],
};
```

These regions serve as a fallback to any regions specified in the [`regions` configuration](/docs/project-configuration#regions). The reg

- Vercel always attempts to invoke the function in the primary region. If you specify more than one primary region in the `regions` prope
- If all primary regions are unavailable, Vercel automatically fails over to the regions specified in `functionFailoverRegions`, selectin
- The order of the regions in `functionFailoverRegions` does not matter as Vercel automatically selects the region geographically closest

To learn more about automatic failover for Vercel Functions, see [Automatic failover](/docs/functions/configuring-functions/region#automa

Region failover is supported with Secure Compute, see [Region Failover](/docs/secure-compute#region-failover) to learn more.

## rewrites

**Type:** `Array` of rewrite `Object`.

**Valid values:** a list of rewrite definitions.

If [`cleanUrls`](/docs/project-configuration/vercel-ts#cleanurls) is set to `true` in
your project's `vercel.ts`, do not include the file extension in the source
or destination path. For example, `/about-our-company.html` would be
`/about-our-company`

### Rewrites examples

- This example rewrites requests to the path `/about` from your site's root to the `/about-our-company.html` file relative to your site's

  ```typescript filename="vercel.ts"
  import { routes, type VercelConfig } from '@vercel/config/v1';

  export const config: VercelConfig = {
    rewrites: [routes.rewrite('/about', '/about-our-company.html')],
  };
  ```

- This example rewrites all requests to the root path which is often used for a Single Page Application (SPA).

  ```typescript filename="vercel.ts"
  import { routes, type VercelConfig } from '@vercel/config/v1';

  export const config: VercelConfig = {

```
    rewrites: [routes.rewrite('/(.*)', '/index.html')],
  };
```

- This example rewrites requests to the paths under `/resize` that with 2 paths levels (defined as variables `width` and `height` that ca

  ```typescript filename="vercel.ts"
  import { routes, type VercelConfig } from '@vercel/config/v1';

  export const config: VercelConfig = {
    rewrites: [routes.rewrite('/resize/:width/:height', '/api/sharp')],
  };
  ```

- This example uses wildcard path matching to rewrite requests to any path (including subdirectories) under `/proxy/` from your site's ro

  ```typescript filename="vercel.ts"
  import { routes, type VercelConfig } from '@vercel/config/v1';

  export const config: VercelConfig = {
    rewrites: [
      routes.rewrite('/proxy/:match*', 'https://example.com/:match*'),
    ],
  };
  ```

- This example rewrites requests to any path from your site's root that does not start with /uk/ and has x-vercel-ip-country header value

  ```typescript filename="vercel.ts"
  import { routes, type VercelConfig } from '@vercel/config/v1';

  export const config: VercelConfig = {
    rewrites: [
      routes.rewrite('/:path((?!uk/).*)', '/uk/:path*', {
        has: [
          {
            type: 'header',
            key: 'x-vercel-ip-country',
            value: 'GB',
          },
        ],
      }),
    ],
  };
  ```

- This example rewrites requests to the path `/dashboard` from your site's root that **does not** have a cookie with key `auth_token` to

  ```typescript filename="vercel.ts"
  import { routes, type VercelConfig } from '@vercel/config/v1';

  export const config: VercelConfig = {
    rewrites: [
      routes.rewrite('/dashboard', '/login', {
        missing: [
          {
            type: 'cookie',
            key: 'auth_token',
          },
        ],
      }),
    ],
  };
  ```

### Rewrite object definition

| Property | Description |
| --- | --- |
| `source` | A pattern that matches each incoming pathname (excluding querystring). |
| `destination` | A location destination defined as an absolute pathname or external URL. |
| `permanent` | A boolean to toggle between permanent and temporary redirect (default true). When `true`, the status code is [308](http |
| `has` | An optional array of `has` objects with the `type`, `key` and `value` properties. Used for conditional rewrites based o |
| `missing` | An optional array of `missing` objects with the `type`, `key` and `value` properties. Used for conditional rewrites bas |

### Rewrite `has` or `missing` object definition

If `value` is an object, it has one or more of the following fields:

This example demonstrates using the expressive `value` object to define a route that rewrites users to `/end` only if the `X-Custom-Heade

```typescript filename="vercel.ts"
import { routes, type VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  rewrites: [
    routes.rewrite('/start', '/end', {
      has: [
        {
          type: 'header',
          key: 'X-Custom-Header',
          value: { pre: 'valid', suf: 'value' },
        },
      ],
    }),
  ],
};
```

The `source` property should **NOT** be a file because precedence is given to the filesystem prior to rewrites being applied. Instead, yo

> **💡 Note:** Using  does not yet work locally while using
> , but does work when deployed.

Learn more about [rewrites](/docs/rewrites) on Vercel.

## trailingSlash

**Type**: `Boolean`.

**Default Value**: `undefined`.

### false

When `trailingSlash: false`, visiting a path that ends with a forward slash will respond with a 308 status code and redirect to the path w

For example, the `/about/` path will redirect to `/about`.

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  trailingSlash: false,
};
```

### true

When `trailingSlash: true`, visiting a path that does not end with a forward slash will respond with a 308 status code and redirect to th

For example, the `/about` path will redirect to `/about/`.

However, paths with a file extension will not redirect to a trailing slash.

For example, the `/about/styles.css` path will not redirect, but the `/about/styles` path will redirect to `/about/styles/`.

```typescript filename="vercel.ts"
import type { VercelConfig } from '@vercel/config/v1';

export const config: VercelConfig = {
  trailingSlash: true,
};
```

### undefined

When `trailingSlash: undefined`, visiting a path with or without a trailing slash will not redirect.

For example, both `/about` and `/about/` will serve the same content without redirecting.

This is not recommended because it could lead to search engines indexing two different pages with duplicate content.

## Legacy properties

Legacy properties like `routes` and `builds` are still supported in `vercel.ts` for backwards compatibility, but are deprecated. We recom

For details on legacy properties, see the [legacy section of the static configuration reference](/docs/project-configuration/vercel-json#


---
title: "Managing projects"
description: "Learn how to manage your projects through the Vercel Dashboard."
last_updated: "2026-01-16T02:19:34.395Z"
source: "https://vercel.com/docs/projects/managing-projects"
---

# Managing projects

You can manage your project on Vercel in your project's dashboard. See [our project dashboard docs](/docs/projects/project-dashboard) to

## Creating a project

#### \['Dashboard'

To create a [new](/new) project:

1. On the Vercel [dashboard](/dashboard), ensure you have selected the correct team from the [scope selector](/docs/dashboard-features#sc
2. Click the **Add New…** drop-down button and select **Project**:

3) You can either [import from an existing Git repository](/docs/git) or use one of our [templates](/templates). For more information, se
4) If you choose to import from a Git repository, you'll be prompted to select the repository you want to deploy.
5) Configure your project settings, such as the name, [framework](/docs/frameworks), [environment variables](/docs/environment-variables)
6) If you're importing from a monorepo, select the **Edit** button to select the project from the repository you want to deploy. For more

#### 'cURL'

To create an Authorization Bearer token, see the [access token](/docs/rest-api/reference/welcome#creating-an-access-token) section of the

```bash filename="cURL"
curl --request POST \
  --url https://api.vercel.com/v11/projects \
  --header "Authorization: Bearer $VERCEL_TOKEN" \
  --header "Content-Type: application/json" \
  --data '{
    "environmentVariables": [
      {
        "key": "<env-key>",
        "target": "production",
        "gitBranch": "<git-branch>",
        "type": "system",
```

```
        "value": "<env-value>"
      }
    ],
    "framework": "<framework>",
    "gitRepository": {
      "repo": "<repo-url>",
      "type": "github"
    },
    "installCommand": "<install-command>",
    "name": "<project-name>",
    "rootDirectory": "<root-directory>"
  }'
```

#### 'SDK']

To create an Authorization Bearer token, see the [access token](/docs/rest-api/reference/welcome#creating-an-access-token) section of the

```ts filename="createProject"
import { Vercel } from '@vercel/sdk';

const vercel = new Vercel({
  bearerToken: '<YOUR_BEARER_TOKEN_HERE>',
});

async function run() {
  const result = await vercel.projects.createProject({
    requestBody: {
      name: '<project-name>',
      environmentVariables: [
        {
          key: '<env-key>',
          target: 'production',
          gitBranch: '<git-branch>',
          type: 'system',
          value: '<env-value>',
        },
      ],
      framework: '<framework>',
      gitRepository: {
        repo: '<repo-url>',
        type: 'github',
      },
      installCommand: '<install-command>',
      name: '<project-name>',
      rootDirectory: '<root-directory>',
    },
  });

  // Handle the result
  console.log(result);
}

run();
```

## Pausing a project

You can choose to temporarily pause a project to ensure that you do not incur usage from [metered resources](/docs/limits#additional-reso

### Pausing a project when you reach your spend amount

To automatically pause your projects when you reach your spend amount:

1. On the Vercel [dashboard](/dashboard), ensure you have selected the correct team from the [scope selector](/docs/dashboard-features#sc
2. Select the **Settings** tab.
3. In the **Spend Management** section, select the **Pause all production deployments** option. Then follow the steps to confirm the acti

To learn more, see the [Spend Management documentation](/docs/spend-management#pausing-projects).

### Pause a project using the REST API

To pause a project manually or with a webhook you can use the [REST API](/docs/rest-api/reference/endpoints/projects/pause-a-project):

1. Ensure you have [access token](/docs/rest-api#creating-an-access-token) scoped to your team to authenticate the API.
2. Create a webhook that calls the pause project [endpoint](/docs/rest-api/reference/endpoints/projects/pause-a-project):
   - You'll need to pass a path parameter of the [Project ID](/docs/projects/overview#project-id) and query string of [Team ID](/docs/acc
     ```bash filename="request"
     https://api.vercel.com/v1/projects/<prj_ID>/pause?teamId=<team_ID>
     ```
   - Use your access token as the bearer token, to enable you to carry out actions through the API on behalf of your team.
   - Ensure that your `Content-Type` header is set to `application/json`.

When you pause your project, any users accessing your production deployment will see a [503 DEPLOYMENT_PAUSED error](/docs/errors/DEPLOY

```bash filename="cURL"
curl --request POST \
  --url "https://api.vercel.com/v1/projects/<project-id>/pause?teamId=<team-id>&slug=<team-slug>" \
  --header "Authorization: Bearer $VERCEL_TOKEN"
```

> **💡 Note:** You can also manually make a POST request to the [pause project
> endpoint](/docs/rest-api/reference/endpoints/projects/pause-a-project) without
> using webhook.

### Resuming a project

Resuming a project can either be done through the [REST API](/docs/rest-api/reference/endpoints/projects/unpause-a-project) or your proje

1. Go to your team's [dashboard](/dashboard) and select your project. When you select it, you should notice it has a **paused** icon in t

2. Select the **Settings** tab.
3. You'll be presented with a banner notifying you that your project is paused and your production deployment is unavailable.
4. Select the **Resume Service** button.
5. In the dialog that appears, confirm that you want to resume service of your project's production deployment by selecting the **Resume*

Your production deployment will resume service within a few minutes. You do not need to redeploy it.

## Deleting a project

Deleting your project will also delete the deployments, domains, environment variables, and settings within it. If you have any deploymen

To delete a project:

1. On the Vercel [dashboard](/dashboard), ensure you have selected the correct team from the [scope selector](/docs/dashboard-features#sc
2. Select the **Settings** tab.
3. At the bottom of the **General** page, you'll see the **Delete Project** section. Click the **Delete** button.
4. In the **Delete Project** dialog, confirm that you'd like to delete the project by entering the project name and prompt. Then, click t

---
title: "Projects overview"
description: "A project is the application that you have deployed to Vercel."
last_updated: "2026-01-16T02:19:34.401Z"
source: "https://vercel.com/docs/projects"
---

# Projects overview

Projects on Vercel represent applications that you have deployed to the platform from a [single Git repository](/docs/git). Each project
and [custom domains](/docs/domains/add-a-domain "Custom Domains").

While each project is only connected to a single, imported Git repository, you can have multiple projects connected to a single Git repos

You can view all projects in your team's [Vercel dashboard](/dashboard) and selecting a project will bring you to the [project dashboard]

- View an overview of the [production deployment](/docs/deployments) and any active pre-production deployments.
- Configure [project settings](/docs/project-configuration/project-settings) such as setting [custom domains](/docs/domains), [environmen
- View details about each [deployment](/docs/deployments) for that project, such as the status, the commit that triggered the deployment,
- Manage [observability](/docs/observability) for that project, including [Web Analytics](/docs/analytics), [Speed Insights](/docs/speed-
- Managing the project's [firewall](/docs/vercel-firewall).

## Project limits

To learn more about limits on the number of projects you can have, see [Limits](/docs/limits#general-limits).

---
title: "Project Dashboard"
description: "Learn about the features available for managing projects with the project Dashboard on Vercel."
last_updated: "2026-01-16T02:19:34.357Z"
source: "https://vercel.com/docs/projects/project-dashboard"
---

# Project Dashboard

Each Vercel project has a separate dashboard to configure settings, view deployments, and more.

To get started with a project on Vercel, see [Creating a Project](/docs/projects/managing-projects#creating-a-project) or [create a new p

## Project overview

The Project Overview tab provides an overview of your production deployment, including its [active Git branches](#active-branches), [buil

### Active branches

The Project Overview's Active Branches gives you a quick view of your project's branches that are being actively committed to. The metada

> **💡 Note:** If your project isn't connected to [a Git provider](/docs/git), you'll see a
> **Preview Deployments** section where **Active Branches** should be.

You can filter the list of active branches by a search term, and see the status of each branch's deployment at a glance with the colored

From the Active Branches section, you can:

- View the status of a branch's deployment
- Redeploy a branch, if you have [the appropriate permissions](/docs/rbac/access-roles/team-level-roles)
- View build and runtime logs for a branch's deployment
- View a branch's source in your chosen Git provider
- Copy a branch's deployment URL for sharing and viewing amongst members of your team. To share the preview with members outside of your

## Deployments

The project dashboard lets you manage all your current and previous deployments associated with your project. To manage a deployment, sel

You can sort your deployments by branch, or by status. You can also interact with your deployment by redeploying it, inspecting it, assig

See [our docs on managing deployments](/docs/deployments/managing-deployments) to learn more.

## Web Analytics and Speed Insights

You can learn about your site's performance metrics with [**Speed Insights**](/docs/speed-insights). When enabled, this dashboard display

Through [**Web Analytics**](/docs/analytics), Vercel exposes data about your audience, such as the top pages, top referrers, and visitor

## Runtime logs

The Logs tab inside your project dashboard allows you to view, search, inspect, and share your runtime logs without any third-party integ

Learn more in the [runtime logs docs](/docs/runtime-logs).

## Storage

The Storage tab lets you manage storage products connected to your project, including:

- [Vercel Blob stores](/docs/storage/vercel-blob)
- [Edge Config stores](/docs/edge-config)

Learn more in [our storage docs](/docs/storage).

## Settings

The Settings tab lets you configure your project. You can change the project's name, specify its root directory, configure environment va

Learn more in [our project settings docs](/docs/project-configuration/project-settings).

---
title: "Transferring a project"
description: "Learn how to transfer a project between Vercel teams."
last_updated: "2026-01-16T02:19:34.840Z"
source: "https://vercel.com/docs/projects/transferring-projects"
---

# Transferring a project

You can transfer projects between your Vercel teams with **zero downtime** and **no workflow interruptions**.

You must be an [owner](/docs/rbac/access-roles#owner-role) of the team you're transferring from, and a member of the team you're transfer

During the transfer, all of the project's dependencies will be moved or copied over to the new Vercel team namespace. To learn more about

## Starting a transfer

1. To begin transferring a project, choose a project from the Vercel [dashboard](/dashboard).

2. Then, select the **Settings** tab from the top menu to go to the project settings.

3. From the left sidebar, click **General** and scroll down to the bottom of the page, where you'll see the **Transfer Project** section.

4. Select the Vercel team you wish to transfer the project to. You can also choose to create a new team:

   If the target Vercel team does not have a valid payment method, you must add
   one before transferring your project to avoid any interruption in service.

5. You'll see a list of any domains, aliases, and environment variables that will be transferred. You can also choose a new name for your

   > **💡 Note:** The original project  when initiating the transfer,
   > but you will not experience any downtime.

6. After reviewing the information, click **Transfer** to initiate the project transfer.

7. While the transfer is in progress, Vercel will redirect you to the newly created project on the target Vercel team with in-progress in
   edit project settings or delete that project.

Transferring a project may take between 10 seconds and 10 minutes, depending on the amount of associated data. When the transfer complete

## What is transferred?

- [Deployments](/docs/deployments)
- [Environment variables](/docs/environment-variables) are copied to the target team, except for those defined in the [`env`](/docs/proje
- The project's configuration details
- [Domains and Aliases](#transferring-domains)
- Administrators
- Project name
- Builds
- Git repository link
- Security settings
- [Cron Jobs](/docs/cron-jobs)
- [Preview Comments](/docs/comments)
- [Web Analytics](/docs/analytics)
- [Speed Insights](/docs/speed-insights)
- [Function Region](/docs/regions#compute-defaults)
- [Directory listing setting](/docs/directory-listing)

Once you transfer a project from a Hobby team to a Pro or Enterprise team, you may choose to enable additional paid features on the targe

- [Concurrent Builds](/docs/deployments/concurrent-builds)
- [Preview Deployment Suffix](/docs/deployments/generated-urls#preview-deployment-suffix)
- [Password Protection](/docs/deployments/deployment-protection#password-protection)

## What is not transferred?

- [Integrations](/docs/integrations): Those associated with your project must be added again after the transfer is complete
- [Edge Configs](/docs/edge-config) have [a separate transfer mechanism](/docs/storage#transferring-your-store)
- Usage is reset on transfer
- The Active Branches section under **Project** will be empty
- Environment variables defined in the [`env`](/docs/project-configuration#env) and [`build.env`](/docs/configuration#project/build-env)
- [Monitoring](/docs/observability/monitoring) data is not transferred
- Log data ([Runtime](/docs/runtime-logs) + [build](/docs/deployments/logs) time)
- [Custom Log Drains](/docs/drains) are not transferred

- [Vercel Blob](/docs/storage/vercel-blob) has [a separate transfer mechanism](/docs/storage#transferring-your-store)

## Transferring domains

Project [domains](/docs/domains) will automatically be transferred to the target team by delegating access to domains.

For example, if your project uses the domain `example.com`, the domain will be [moved](/docs/projects/custom-domains#moving-domains) to t

If your project uses the domain `blog.example.com`, the domain `blog.example.com` will be **delegated** to the target team, but the root

If your project uses a [Wildcard domain](/docs/domains/working-with-domains#wildcard-domain) like `*.example.com`, the Wildcard domain wi

## Additional features

> **💡 Note:** This only applies when transferring away from a team.

When transferring between teams, you may be asked whether you want to add additional features to the target team to match the origin team
Adding these features is optional.

---
title: "Restricting Git Connections to a single Vercel team"
description: "Information to stop developers from deploying their repositories to a personal Vercel account by using Protected Git Scopes
last_updated: "2026-01-16T02:19:34.958Z"
source: "https://vercel.com/docs/protected-git-scopes"
---

# Restricting Git Connections to a single Vercel team

Teams often need control over who can deploy their repositories to which teams or accounts. For example, a user on your team may accident

Protected Git Scopes restrict Vercel account and team access to Organization-level Git repositories. This ensures that only authorized Ve

## Managing Protected Git Scopes

You can [add](#adding-a-protected-git-scope) up to five Protected Git Scopes to your Vercel Team. Protected Git Scopes are configured at

In order to add a Protected Git Scope to your Vercel Team, you must be an [Owner](/docs/rbac/access-roles#owner-role) of the Vercel Team,

For Github you must be an `admin`, for Gitlab you must be an `owner`, and for Bitbucket you must be a `owner`.

## Adding a Protected Git Scope

To add a Protected Git Scopes:

1. Go to your Team's dashboard and select the **Settings** tab
2. In the **Security & Privacy** section, go to **Protected Git Scopes**

3) Select **Add** to add a new Protected Git Scope
4) In the modal, select the Git provider you wish to add:
5) In the modal, select the Git namespace you wish to add:
6) Click **Save**

## Removing a Protected Git Scope

To remove a Protected Git Scopes:

1. Go to your Team's dashboard and select the **Settings** tab.
2. In the **Security & Privacy** section, go to **Protected Git Scopes**

3) Select **Remove** to remove the Protected Git Scope

---
title: "Limits and Pricing for Monitoring"
description: "Learn about our limits and pricing when using Monitoring. Different limitations are applied depending on your plan."
last_updated: "2026-01-16T02:19:34.961Z"
source: "https://vercel.com/docs/query/monitoring/limits-and-pricing"
---

# Limits and Pricing for Monitoring

## Pricing

Monitoring has become part of Observability, and is therefore included with Observability Plus at no additional cost. If you are currentl

Even if you choose not to migrate to Observability Plus, Vercel will automatically move you to the new pricing modal of $1.20 per 1 milli

To learn more, see [Limits and Pricing for Observability](/docs/observability/limits-and-pricing).

## Limitations

| Limit          | Pro           | Enterprise              |
| -------------- | ------------- | ----------------------- |
| Data retention | 30 days       | 90 days                 |
| Granularity    | 1 day, 1 hour | 1 day, 1 hour, 5 minute |

## How are events counted?

Vercel creates an event each time a request is made to your website. These events include unique parameters such as execution time. For a

---
title: "Monitoring Reference"
description: "This reference covers the clauses, fields, and variables used to create a Monitoring query."
last_updated: "2026-01-16T02:19:34.984Z"
source: "https://vercel.com/docs/query/monitoring/monitoring-reference"
---

# Monitoring Reference

## Visualize

The `Visualize` clause selects what query data is displayed. You can select one of the following fields at a time, [aggregating](#aggrega

| **Field Name** | **Description** |
| --- | --- |
| **Edge Requests** | The number of [Edge Requests](/docs/manage-cdn-usage#edge-requests) |
| **Duration** | The time spent serving a request, as measured by Vercel's CDN |
| **Incoming Fast Data Transfer** | The amount of [Fast Data Transfer](/docs/manage-cdn-usage#fast-data-transf |
| **Outgoing Fast Data Transfer** | The amount of [Fast Data Transfer](/docs/manage-cdn-usage#fast-data-transf |
| **Function Duration** | The amount of \[Vercel Function duration]\(/docs/ |
| fluid-compute#pricing-and-usage), as measured in GB-hours. | Sum, Sum per Second, Min/Max, Percentages, Percentiles |
| **Function Invocations** | The number of [Vercel Function invocations](/docs/functions/usage-and-pric |
| **Function Duration** | The amount of [Vercel Function duration](/docs/functions/usage-and-pricing |
| **Function CPU Time** | The amount of CPU time a Vercel Function has spent responding to requests, |
| **Incoming Fast Origin Transfer** | The amount of [Fast Origin Transfer](/docs/manage-cdn-usage#fast-origin-tr |
| **Outgoing Fast Origin Transfer** | The amount of [Fast Origin Transfer](/docs/manage-cdn-usage#fast-origin-tr |
| **Provisioned Memory** | The amount of memory provisioned to a Vercel Function. |
| **Peak Memory** | The maximum amount of memory used by Vercel Function at any point in time. |
| **Requests Blocked** | All requests blocked by either the system or user. |
| **Incoming Legacy Bandwidth** | Legacy Bandwidth sent from the client to Vercel |
| **Outgoing Legacy Bandwidth** | Legacy Bandwidth sent from Vercel to the client |
| **Total Legacy Bandwidth** | Sum of Incoming and Outgoing Legacy Bandwidth |

### Aggregations

The visualize field can be aggregated in the following ways:

| **Aggregation** | **Description** |
| --- | --- |
| **Count** | The number of requests that occurred |
| **Count per Second** | The average rate of requests that occurred |
| **Sum** | The sum of the field value across all requests |
| **Sum per Second** | The sum of the field value as a rate per second |
| **Minimum** | The smallest observed field value |
| **Maximum** | The largest observed field value |
| **Percentiles (75th, 90th, 95th, 99th)** | Percentiles for the field values. For example, 90% of requests will have a duration that is |
| **Percentages** | Each group is reported as a percentage of the ungrouped whole. For example, if a query for r |

Aggregations are calculated within each point on the chart (hourly, daily, etc depending on the selected granularity) and also across the

## Where

The `Where` clause defines the conditions to filter your query data. It only fetches data that meets a specified condition based on sever

| **Operator** | **Description** |
| --- | --- |
| `=` | The operator that allows you to specify a single value |
| `in` | The operator that allows you to specify multiple values. For example, `host in ('vercel.com', 'nextjs.com')` |
| `and` | The operator that displays a query result if all the filter conditions are |
| `or` | The operator that displays a query result if at least one of the filter conditions are |
| `not` | The operator that displays a query result if the filter condition(s) is |
| `like` | The operator used to search a specified pattern. This is case-sensitive. For example, `host like 'acme.com'`. You can al |
| `startsWith` | Filter data values that begin with some specific characters |
| `match` | The operator used to search for patterns based on a regular expression ([`Re2`](https://github.com/google/re2/wiki/Synta |

> **⚠️ Warning:** String literals must be surrounded by single quotes. For example, `host =
>  'vercel.com'`.

## Group by

The `Group By` clause calculates statistics for each combination of [field](#group-by-and-where-fields) values. Each group is displayed a

For example, grouping by `host` and `status` will display data broken down by each combination of `host` and `status`.

## Limit

The `Limit` clause defines the maximum number of results displayed. If the number of query results is greater than the `Limit` value, the

## Group by and where fields

There are several fields available for use within the [where](#where) and [group by](#group-by) clauses:

| **Field Name** | **Description** |
| --- | --- |
| `host` | Group by the request's domains and subdomains |
| `path_type` | Group by the request's [resource type](#path-types) |
| `project_id` | Group by the request's project ID |
| `status` | Group by the request's HTTP response code |
| `source_path` | The mapped path used by the request. For example, if you have a dynamic route like `/blog/[slug]` and a blog post i |
| `request_path` | The path used by the request. For example, if you have a dynamic route like `/blog/[slug]` and a blog post is `/blo |
| `cache` | The [cache](/docs/cdn-cache#x-vercel-cache) status for the request |
| `error_details` | Group by the [errors](/docs/errors) that were thrown on Vercel |
| `deployment_id` | Group by the request's deployment ID |
| `environment` | Group by the environment (`production` or [`preview`](/docs/deployments/environments#preview-environment-pre-produc |
| `request_method` | Group by the HTTP request method (`GET`, `POST`, `PUT`, etc.) |
| `http_referer` | Group by the HTTP referer |
| `public_ip` | Group by the request's IP address |
| `user_agent` | Group by the request's user agent |
| `asn` | The [autonomous system number (ASN)](# "ASN") for the request. This is related to what network the request came fro |
| `bot_name` | Group by the request's bot crawler name. This field will contain the name of a known crawler (e.g. Google, Bing) |
| `region` | Group by the [region](/docs/regions) the request was routed to |
| `waf_action` | Group by the WAF action taken by the [Vercel Firewall](/docs/security/vercel-waf) (`deny`, `challenge`, `rate_limit |
| `action` | Group by the action taken by [Vercel DDoS Mitigations](/docs/security/ddos-mitigation) (`deny` or `challenge`) |
| `skew_protection` | When `active`, the request would have been subject to [version skew](/docs/deployments/skew-protection) but was pro |

### Path types

All your project's resources like pages, functions, and images have a path type:

| **Path Type** | **Description**
| ---------- | ---------- |
| `static` | A static asset (`.js`, `.css`, `.png`, etc.)
| `func` | A [Vercel Function](/docs/functions)
| `external` | A resource that is outside of Vercel. This is usually caused when you have [rewrite rules](/docs/project-configurat
| `edge` | A [Vercel Function](/docs/functions) using [Edge runtime](/docs/functions/runtimes/edge)
| `prerender` | A pre-rendered page built using [Incremental Static Regeneration](/docs/incremental-static-regeneration)
| `streaming_func` | A [streaming Vercel Function](/docs/functions/streaming-functions)
| `background_func` | The [Incremental Static Regeneration Render Function](/docs/incremental-static-regeneration) used to create or upda

## Chart view

In the chart view (vertical bar or line), `Limit` is applied at the level of each day or hour (based the value of the **Data Granularity*

## Table view

In the table view (below the chart), `Limit` is applied to the sum of requests for the selected query window so that the number of rows i

## Example queries

On the left navigation bar, you will find a list of example queries to get started:

| **Query Name** | **Description**
| ---------- | ---------- |
| Requests by Hostname | The total number of requests for each `host`
| Requests Per Second by Hostname | The total number of requests per second for each `host`
| Requests by Project | The total number of requests for each `project_id`
| Requests by IP Address | The total number of requests for each `public_ip`
| Requests by Bot/Crawler | The total number of requests for each `bot_name`
| Requests by User Agent | The total number of requests for each `user_agent`
| Requests by Region | The total number of requests for each `region`
| Bandwidth by Project, Hostname | The outgoing bandwidth for each `host` and `project_id` combination
| Bandwidth Per Second by Project, Hostname | The outgoing bandwidth per second for each `host` and `project_id`
| Bandwidth by Path, Hostname | The outgoing bandwidth for each `host` and `source_path`
| Request Cache Hits | The total number of request cache hits for each `host`
| Request Cache Misses | The total number of request cache misses for each`host`
| Cache Hit Rates | The percentage of cache hits and misses over time
| 429 Status Codes by Host, Path | The total 429 (Too Many Requests) status code requests for each `host` and `source_path`
| 5XX Status Codes by Host, Path | The total 5XX (server-related HTTPS error) status code requests for each `host` and `source
| Execution by Host, Path | The total billed Vercel Function usage for each `host` and `source_path`
| Average Duration by Host, Path | The average duration for each `host` and `source_path`
| 95th Percentile Duration by Host, Path | The p95 duration for each `host` and `source_path`


---
title: "Monitoring"
description: "Query and visualize your Vercel usage, traffic, and more with Monitoring."
last_updated: "2026-01-16T02:19:34.900Z"
source: "https://vercel.com/docs/query/monitoring"
---

# Monitoring

**Monitoring** allows you to visualize and quantify the performance and traffic of your projects on Vercel. You can use [example queries]

## Monitoring chart

Charts allow you to explore your query results in detail. Use filters to adjust the date, data granularity, and chart type (line or bar).

Hover and move your mouse across the chart to view your data at a specific point in time. For example, if the data granularity is set to

## Example queries

To get started with the most common scenarios, use our **Example Queries**. You cannot edit or add new example queries. For a list of the

## Save new queries

You can no longer save new Monitoring queries as the feature has now been sunset.

Instead, use observability queries, which can be saved into [Notebooks](/docs/notebooks).

### Manage saved queries

You can manage your saved personal and team queries from the query console. Select a query from the left navigation bar and click on the '

Duplicating a query creates a copy of the query in the same folder. You cannot copy queries to another folder. To rename a saved query, u

Deleting a saved personal or team query is permanent and irreversible. To delete a saved query, click the **Delete** button in the confir

## Error messages

You may encounter errors such as **invalid queries** when using Monitoring. For example, defining an incorrect location parameter generat

## Enable Monitoring

You can no longer enable **Monitoring** on [Pro](/docs/plans/pro-plan) plans as the feature has now been sunset.

Get the most comprehensive suite of tools, including queries, by enabling [Observability Plus](/docs/observability/observability-plus).

## Disable Monitoring

1. Go to your team **Settings** > **Billing**
2. Scroll to the **Observability Plus** section
3. Set the toggle to the disabled state

## Manage IP Address visibility for Monitoring

Vercel creates events each time a request is made to your website. These events include unique parameters such as execution time and band

Certain events such as `public_ip` may be considered personal information under certain data protection laws. To hide IP addresses from y

1. Go to the Vercel [dashboard](/dashboard) and ensure your team is selected in the scope selector.
2. Go to the **Settings** tab and navigate to **Security & Privacy**.
3. Under **IP Address Visibility**, toggle the switch next to off so the text reads **IP addresses are hidden in your Monitoring queries.

> **💡 Note:** For business purposes, such as DDoS mitigation, Vercel will still collect IP
> addresses.

For a complete list of fields, see the [visualize clause](/docs/observability/monitoring/monitoring-reference#visualize) docs.

## Monitoring sunset

From the end of billing cycle in Nov 2025, Vercel will sunset Monitoring for pro plans. Pro users will no longer see the Monitoring tab.
If you want to continue using the full Monitoring capabilities or purchase a product similar to Monitoring, consider moving to [Query](/d

- Enable [Observability Plus](/docs/observability/observability-plus) to continue using query features.
- Save queries in **Observability** [Notebooks](/docs/observability/query#save-query).

## More resources

For more information on what to do next, we recommend the following articles:

- [Quickstart](/docs/observability/monitoring/quickstart): Learn how to create and run a query to understand the top bandwidth images on
  your website
- [Reference](/docs/observability/monitoring/monitoring-reference): Learn about the clauses, fields, and variables used to create a Monit
- [Limits and Pricing](/docs/observability/monitoring/limits-and-pricing): Learn about our limits and pricing when using Monitoring. Diff

---
title: "Monitoring Quickstart"
description: "In this quickstart guide, you"
last_updated: "2026-01-16T02:19:34.920Z"
source: "https://vercel.com/docs/query/monitoring/quickstart"
---

# Monitoring Quickstart

## Prerequisites

- Make sure you upgrade to [Pro](/docs/plans/pro-plan) or [Enterprise](/docs/plans/enterprise) plan.
- Pro and Enterprise teams should [Upgrade to Observability Plus](/docs/observability#enabling-observability-plus) to access Monitoring.

## Create a new query

In the following guide you will learn how to view the most requested posts on your website.

- ### Go to the dashboard
  1. Navigate to the **Monitoring** tab from your Vercel dashboard
  2. Click the **Create New Query** button to open the query builder
  3. Click the **Edit Query** button to configure your query with clauses

- ### Add Visualize clause
  The [Visualize](/docs/observability/monitoring/monitoring-reference#visualize") clause specifies which field in your query will be calc

  Click the **Run Query** button, and the [Monitoring chart](/docs/observability/monitoring#monitoring-chart) will display the total numb

- ### Add Where clause
  To filter the query data, use the [Where](/docs/observability/monitoring/monitoring-reference#where) clause and specify the conditions
  ```sql filename=Where
  host = 'my-site.com' and like(request_path, '/posts%')
  ```
  This query retrieves data with a host field of `my-site.com` and a `request_path` field that starts with /posts.

  The `%` character can be used as a wildcard to match any sequence of characters after `/posts`, allowing you to capture all `request_pa

- ### Add Group By clause
  Define a criteria that groups the data based on the selected attributes. The grouping mechanism is supported through the [Group By](/do

  Set the Group By clause to `request_path`.

  With **Visualize**, **Where**, and **Group By** fields set, the [Monitoring chart](/docs/observability/monitoring#monitoring-chart) now

- ### Add Limit clause
  To control the number of results returned by the query, use the [**Limit**](/docs/observability/monitoring/monitoring-reference#limit)

- ### Save and Run Query
  Save your query and click the  button to generate the final results. The Monitoring chart will display a comprehensive view of the top

---
title: "Query"
description: "Query and visualize your Vercel usage, traffic, and more in observability."
last_updated: "2026-01-16T02:19:34.936Z"
source: "https://vercel.com/docs/query"
---

# Query

You can use Query to get deeper visibility into your application when debugging issues, monitoring usage, or optimizing for speed and rel

- Investigate errors, slow routes, and high-latency functions
- Analyze traffic patterns and request volumes by path, region, or device
- Monitor usage and performance of AI models or API endpoints
- Track build and deployment behavior across your projects
- Save queries to notebooks for reuse and team collaboration
- Customize dashboards and automate reporting or alerts

## Getting started

To start using Query, you first need to [enable Observability Plus](#enable-observability-plus). Then, you can [create a new query](#crea

### Enable Observability Plus

- Pro and Enterprise teams should [Upgrade to Observability Plus](/docs/observability#enabling-observability-plus) to edit queries in mod
- Free observability users can still open a query, but they cannot modify any filters or create new queries.

> **💡 Note:** [Enterprise](/docs/plans/enterprise) teams can [contact sales](/contact/sales)
> to get a customized plan based on their requirements.

### Create a new query

- ### Access the Observability dashboard
  - **At the Team level**: Go to the [Vercel dashboard](/dashboard) and click the **Observability** tab
  - **At the Project level**: Go to the [Vercel dashboard](/dashboard), select the project you would like to monitor from the scope selec

- ### Initiate a new query
  - **Start a new query**: In the Observability section, click the  button (New Query) to open the query creation interface.
  - **Select a data source**: Under "Visualize", select the [metric](/docs/observability/query/query-reference#metric) you want to analyz

- ### Define query parameters
  - **Select the data aggregation**: Select how you would like the values of your selected metric to be compiled such as sum, percentage,
  - **Set Time Range**: Select the time frame for the data you want to query. This can be a predefined range like "Last 24 hours" or a cu
  - **Filter Data**: Apply filters to narrow down the data. You can filter by a list of [fields](/docs/query/reference#group-by-and-where

- ### Visualize Query
  - **View the results**: The graph below the filter updates automatically as you change the filters.
  - **Adjust as Needed**: Refine your query parameters if needed to get precise insights.

- ### Save and Share Query
  - **Save the query**: Once you are satisfied with your query, you can save it by clicking **Add to Notebook**.
  - **Select a notebook**: Select an existing [notebook](/docs/notebooks) from the dropdown.
  - **Share Query**: You can share the saved query from the notebook with team members by clicking on the **Share with team** button.

## Using Query

- When building queries, you can select the most appropriate view, and visualize results with:
  - a line or a volume chart
  - a table, if your query has a group by clause
  - a big number (with a time series), if your query has no group by clause
- You can [save your queries](#save-and-share-query) in [notebooks](/docs/notebooks) either for personal use or to share with your team.
- In the dashboard, you can [create a new query](#create-a-new-query) using the query [form fields](/docs/query/reference#group-by-and-wh
- You can export query results as CSV or JSON by clicking the download icon.

## Manage IP Address visibility for Query

Vercel creates events each time a request is made to your website. These events include unique parameters such as execution time and band

Certain events such as `public_ip` may be considered personal information under certain data protection laws. To hide IP addresses from y

1. Go to the Vercel [dashboard](/dashboard) and ensure your team is selected in the scope selector.
2. Go to the **Settings** tab and navigate to **Security & Privacy**.
3. Under **IP Address Visibility**, toggle the switch next to "Off" so the text reads **IP addresses are currently hidden in the Vercel D

> **💡 Note:** For business purposes, such as DDoS mitigation, Vercel will still collect IP
> addresses.

## More resources

- Learn about available metrics and aggregations and how you can group and filter the data in [Query Reference](/docs/observability/query

---
title: "Query Reference"
description: "This reference covers the dimensions and operators used to create a query."
last_updated: "2026-01-16T02:19:34.951Z"
source: "https://vercel.com/docs/query/reference"
---

# Query Reference

## Metric

The metric selects what query data is displayed. You can choose one field at a time, and the same metric can be applied to different even

| **Field Name** | **Description** |
| --- | --- |
| **Edge Requests** | The number of [Edge Requests](/docs/pricing/networking#edge-requests) |
| **Duration** | The time spent serving a request, as measured by Vercel's CDN |
| **Incoming Fast Data Transfer** | The incoming amount of [Fast Data Transfer](/docs/pricing/networking#fast-data-transfer) used by th |
| **Outgoing Fast Data Transfer** | The outgoing amount of [Fast Data Transfer](/docs/pricing/networking#fast-data-transfer) used by th |
| **Total Fast Data Transfer** | The total amount of [Fast Data Transfer](/docs/pricing/networking#fast-data-transfer) used by the r |
| **Function Invocations** | The number of [Function invocations](/docs/functions/usage-and-pricing#managing-function-invocation |
| **Function Duration** | The amount of [Function duration](/docs/functions/usage-and-pricing#managing-function-duration), as |
| **Function CPU Time** | The amount of CPU time a Vercel Function has spent responding to requests, as measured in milliseco |
| **Incoming Fast Origin Transfer** | The amount of [Fast Origin Transfer](/docs/pricing/networking#fast-origin-transfer) used by the req |
| **Outgoing Fast Origin Transfer** | The amount of [Fast Origin Transfer](/docs/pricing/networking#fast-origin-transfer) used by the res |
| **Provisioned Memory** | The amount of memory provisioned to a Vercel Function. |
| **Peak Memory** | The maximum amount of memory used by Vercel Function at any point in time. |
| **Requests Blocked** | All requests blocked by either the system or user. |
| **ISR Read Units** | The amount of [Read Units](/docs/pricing/incremental-static-regeneration) used to access ISR data |
| **ISR Write Units** | The amount of [Write Units](/docs/pricing/incremental-static-regeneration) used to store new ISR da |
| **ISR Read/Write** | The amount of ISR operations |
| **Time to First Byte** | The time between the request for a resource and when the first byte of a response begins to arrive. |
| **Function Wall Time** | The duration that a Vercel Function has run |
| **Firewall Actions** | The incoming web traffic observed by firewall rules. |
| **Optimizations** | The number of image transformations |

| **Source Size**                  | The source size of image optimizations
| **Optimized Size**               | The optimized size of image optimizations
| **Compression Ratio**            | The compression ratio of image optimizations
| **Size Change**                  | The size change of image optimizations

### Aggregations

Metrics can be aggregated in the following ways:

| **Aggregation**                          | **Description**
| ---------------------------------------- | ----------------------------------------------------------------------------------------------------
| **Count**                                | The number of requests that occurred
| **Count per Second**                     | The average rate of requests that occurred
| **Sum**                                  | The sum of the field value across all requests
| **Sum per Second**                       | The sum of the field value as a rate per second
| **Minimum**                              | The smallest observed field value
| **Maximum**                              | The largest observed field value
| **Percentiles (75th, 90th, 95th, 99th)** | Percentiles for the field values. For example, 90% of requests will have a duration that is
| **Percentages**                          | Each group is reported as a percentage of the ungrouped whole. For example, if a query for r

Aggregations are calculated within each point on the chart (hourly, daily, etc) and also across the entire query window.

## Filter

The filter bar defines the conditions to filter your query data. It only fetches data that meets a specified condition based on several [

| **Operator**                 | **Description**
| ---------------------------- | ----------------------------------------------------------------------------------------------------
| `is`, `is not`               | The operator that allows you to specify a single value
| `is any of `, `is not any of` | The operator that allows you to specify multiple values. For example, `host in ('vercel.com', 'nextjs.c
| `startsWith`                 | Filter data values that begin with some specific characters
| `endsWith`                   | Filter data values that end with specific characters
| `>,>=,<,<=`                  | Numerical operators that allow numerical comparisons

## Group by

The `Group By` clause calculates statistics for each combination of [field](#group-by-and-where-fields) values. Each group is displayed a

For example, grouping by `Request HostName` and `HTTP Status` will display data broken down by each combination of `Request Hostname` and

## Group by and where fields

There are several fields available for use within the [Filter](#filter) and [group by](#group-by):

| **Field Name**      | **Description**
| ------------------- | ----------------------------------------------------------------------------------------------------
| `Request Hostname`  | Group by the request's domains and subdomains
| `project`           | Group by the request's project
| `Deployment ID`     | Group by the request's deployment ID
| `HTTP Status`       | Group by the request's HTTP response code
| `route`             | The mapped path used by the request. For example, if you have a dynamic route like `/blog/[slug]` and a blog post
| `Request Path`      | The path used by the request. For example, if you have a dynamic route like `/blog/[slug]` and a blog post is `/b
| `Cache Result`      | The [cache](/docs/cdn-cache#x-vercel-cache) status for the request
| `environment`       | Group by the environment (`production` or [`preview`](/docs/deployments/environments#preview-environment-pre-prod
| `Request Method`    | Group by the HTTP request method (`GET`, `POST`, `PUT`, etc.)
| `Referrer URL`      | Group by the HTTP referrer URL
| `Referrer Hostname` | Group by the HTTP referrer domain
| `Client IP`         | Group by the request's IP address
| `Client IP Country` | Group by the request's IP country
| `Client User Agent` | Group by the request's user agent
| `AS Number`         | The [autonomous system number (ASN)](# "ASN") for the request. This is related to what network the request came f
| `CDN Region`        | Group by the [region](/docs/regions) the request was routed to
| `ISR Cache Region`  | Group by the ISR cache region
| `Cache Result`      | Group by cache result
| `WAF Action`        | Group by the WAF action taken by the [Vercel Firewall](/docs/security/vercel-waf) (`deny`, `challenge`, `rate_lim
| `WAF Rule ID`       | Group by the firewall rule ID
| `Skew Protection`   | When `active`, the request would have been subject to [version skew](/docs/skew-protection) but was protected, ot


---
title: "Access Groups"
description: "Learn how to configure access groups for team members on a Vercel account."
last_updated: "2026-01-16T02:19:34.997Z"
source: "https://vercel.com/docs/rbac/access-groups"
---

# Access Groups

Access Groups provide a way to manage groups of Vercel users across projects on your team. They are a set of project role assignations, a

An Access Group consists of one or many projects in a team and assigns project roles to team members. Any team member included in an Acce

Team administrators can apply automatic role assignments for default roles. And for more restricted projects, you can ensure only a subse

## Create an Access Group

1. Navigate to your team's **Settings** tab and then **Access Groups** (`<team-name>/~/settings/access-groups`)
2. Select **Create Access Group**
3. Create a name for your Access Group
4. Select the projects and [project roles](/docs/rbac/access-roles/project-level-roles) to assign
5. Select the **Members** tab
6. Add members with the **Developer** and **Contributor** role to the Access Group
7. Create your Access Group by pressing **Create**

## Edit projects of an Access Group

1. Navigate to your team's **Settings** tab and then **Access Groups** (`<team-name>/~/settings/access-groups`)
2. Press the **Edit Access Group** button for the Access Group you wish to edit from your list of Access Groups
3. Either:
   - Remove a project using the remove button to the right of a project

   - Add more projects using the **Add more** button below the project list and using the selection controls

## Add and remove members from an Access Group

1. Navigate to your team's **Settings** tab and then **Access Groups** (`<team-name>/~/settings/access-groups`)
2. Press the **Edit Access Group** button for the Access Group you wish to edit from your list of Access Groups
3. Select the **Members** tab
4. Either:
   - Remove an Access Group member using the remove button to the right of a member
   - Add more members using the **Add more** button and the search controls

## Modifying Access Groups for a single team member

You can do this in two ways:

1. From within your team's members page using the **Manage Access** button (recommended for convenience). Access this by navigating to yo
2. By [editing each Access Group](#add-and-remove-members-from-an-access-group) using the **Edit Access Group** button and editing the **

## Access Group behavior

When configuring Access Groups, there are some key things to be aware of:

- Team roles cannot be overridden. An Access Group manages project roles only
- Only a subset of team role and project role combinations are valid:
  - **[Owner](/docs/rbac/access-roles#owner-role), [Member](/docs/rbac/access-roles#member-role), [Billing](/docs/rbac/access-roles#billi
  - **[Developer](/docs/rbac/access-roles#developer-role)**: [Admin](/docs/rbac/access-roles#project-administrators) assignment is valid
  - **[Contributor](/docs/rbac/access-roles#contributor-role)**: `Admin`, `Project Developer`, or `Project Viewer` roles are valid in sel
- When a `Contributor` belongs to **multiple** access groups the computed role will be:
  - `Admin` permissions in the project if any of the access groups they get assigned has a project mapping to `Admin`
  - `Project Developer` permissions in the project if any of the access groups they get assigned has a project mapping to `Project Develo
  - `Project Viewer` permissions in the project if any of the access groups they get assigned has a project mapping to `Project Viewer` a
- When a `Developer` belongs to **multiple** access groups the role assignation will be:
  - `Admin` permissions in the project if any of the access groups they get assigned has a project mapping to Admin
  - In all other cases the member will have `Developer` permissions
- Access Group assignations are not deleted when a team role gets changed. This allows a temporal increase of permissions without having
- Direct project assignations also affect member roles. Consider these examples:
  - A direct project assignment assigns a member as `Admin`. That member is within an Access Group that assigns `Developer`. The computed
  - A direct project assignment assigns a member as `Developer`. That member is within an Access Group that assigns `Admin`. The computed

> **💡 Note:** Contributors and Developers can increase their level of permissions in a
> project but they can never reduce their level of permissions

## Directory sync

If you use [Directory sync](/docs/security/directory-sync), you are able to map a Directory Group with an Access Group. This will grant a

Some things to note:

- The final role the user will have in a specific project will depend on the mappings of all Access Groups the user belongs to
- Assignations using directory sync can lead to `Owners`, `Members` `Billing` and `Viewers` being part of an Access Group dependent on th
- When a Directory Group is mapped to an Access Group, members of that group will default to `Contributor` role at team level. This is un

---
title: "Extended permissions"
description: "Learn about extended permissions in Vercel"
last_updated: "2026-01-16T02:19:35.022Z"
source: "https://vercel.com/docs/rbac/access-roles/extended-permissions"
---

# Extended permissions

Vercel's Role-Based Access Control (RBAC) system consists of three main components:

- **Team roles**: Core roles that define a user's overall access level within a team
- **Project roles**: Roles that apply to specific projects rather than the entire team
- **Extended permissions**: Granular permissions that can be combined with roles for fine-tuned access control

These components can be combined to create precise access patterns tailored to your organization's needs.

## Project roles for specific access

Project roles apply only to specific projects and include:

| Project Role | Compatible Team Roles |
| --- | --- |
| **[Admin](/docs/rbac/access-roles#project-administrators)** | [Contributor](/docs/rbac/access-roles#contributor-role), [Develope |
| **[Project Developer](/docs/rbac/access-roles#project-developer)** | [Contributor](/docs/rbac/access-roles#contributor-role) |
| **[Project Viewer](/docs/rbac/access-roles#project-viewer)** | [Contributor](/docs/rbac/access-roles#contributor-role) |

## Extended permissions for granular access

Extended permissions add granular capabilities that can be combined with roles:

| Extended permission | Description |
| --- | --- |
| | Allows the user to create a new project. | [Developer](/docs/rbac/acc |
| | Deploy to production from CLI, rollback and promote any deployment. | [Developer](/docs/rbac/access-roles#developer-role |
| | Read-only usage team-wide including prices and invoices. | [Developer](/docs/rbac |
| | Install and use Vercel integrations, marketplace integrations, and storage. | [Developer](/docs/rbac/access-roles# |
| | Create and manage project environments. | [Developer](/docs/rbac/access-roles# |
| Create and manage environment variables. | [Developer](/docs/rbac/access-roles#developer-role) |

Extended permissions work when the user has at least one compatible team role.

### How roles fit together

Team roles provide the foundation of access control. Each role has a specific scope of responsibilities:

| Team Role | Role Capabilities |

| ---------------------------------------------------------- | ---------------------------------------------------------------------- |
| **[Owner](/docs/rbac/access-roles#owner-role)**             | Complete control over all team and project settings                    |
| **[Member](/docs/rbac/access-roles#member-role)**           | Can manage projects but not team settings                              |
| **[Developer](/docs/rbac/access-roles#developer-role)**     | Can deploy and manage projects with limitations on production settings |
| **[Billing](/docs/rbac/access-roles#billing-role)**         | Manages financial aspects only                                         |
| **[Security](/docs/rbac/access-roles#security-role)**       | Manages security features team-wide                                    |
| **[Viewer](/docs/rbac/access-roles#viewer-role)**           | Read-only access to all projects                                       |
| **[Contributor](/docs/rbac/access-roles#contributor-role)** | Configurable role that can be assigned project-level roles             |

## How combinations work

The multi-role system allows users to have multiple roles simultaneously. When roles are combined:

- Users inherit the most permissive combination of all their assigned roles and permissions
- A user gets all the capabilities of each assigned role
- Extended permissions can supplement roles with additional capabilities
- Project roles can be assigned alongside team roles for project-specific access

The following table outlines various use cases and the role combinations that enable them. Each combination is designed to provide specif

| Use Case                | Role Combinations
| ----------------------- | ----------------------------------------------------------------------------------------------------------------------
| **DevOps engineer**     | [Developer](/docs/rbac/access-roles#developer-role) + [Environment Variable Manager](#environment-variable-ma
| **Technical team lead** | [Member](/docs/rbac/access-roles#member-role) + [Security](/docs/rbac/access-roles#security-role)
| **External contractor** | [Contributor](/docs/rbac/access-roles#contributor-role) + [Project Developer](/docs/rbac/access-roles#project
| **Finance manager**     | [Billing](/docs/rbac/access-roles#billing-role) + [Usage Viewer](#usage-viewer)
| **Product owner**       | [Viewer](/docs/rbac/access-roles#viewer-role) + [Create Project](#create-project) + [Environment Manager](#en

## Role compatibility and constraints

Not all roles and permissions can be meaningfully combined. For example:

- The **[Owner](/docs/rbac/access-roles#owner-role)** role already includes all permissions, so adding additional roles doesn't grant mor
- Some extended permissions are only compatible with specific roles (e.g. [Full Production Deployment](#full-production-deployment) works
- Project roles are primarily assigned to [Contributors](/docs/rbac/access-roles#contributor-role) or via Access Groups

---
title: "Access Roles"
description: "Learn about the different roles available for team members on a Vercel account."
last_updated: "2026-01-16T02:19:35.065Z"
source: "https://vercel.com/docs/rbac/access-roles"
---

# Access Roles

Vercel distinguishes between different roles to help manage team members' access levels and permissions. These roles are categorized into

The two groups are further divided into specific roles, each with its own set of permissions and responsibilities. These roles are design

- [**Team level roles**](#team-level-roles): Users who have access to all projects within a team
  - [Owner](#owner-role)
  - [Member](#member-role)
  - [Developer](#developer-role)
  - [Security](#security-role)
  - [Billing](#billing-role)
  - [Pro Viewer](#pro-viewer-role)
  - [Enterprise Viewer](#enterprise-viewer-role)
  - [Contributor](#contributor-role)
- [**Project level roles**](#project-level-roles): Users who have restricted access at the project level. Only contributors can have conf
  - [Project Administrator](#project-administrators)
  - [Project Developer](#project-developer)
  - [Project Viewer](#project-viewer)

## Team level roles

Team level roles are designed to provide a broad level of control and access to the team as a whole. These roles are assigned to individu

| Role                                          | Description
| --------------------------------------------- | ------------------------------------------------------------------------------
| [**Owner**](#owner-role)                      | Have the highest level of control. They can manage, modify, and oversee the team's s
| [**Member**](#member-role)                    | Have full control over projects and most team settings, but cannot invite or manage
| [**Developer**](#developer-role)              | Can deploy to projects and manage environment settings but lacks the comprehensive t
| [**Security**](#security-role)                | Can manage security features, IP blocking, firewall. Cannot create deployments by de
| [**Billing**](#billing-role)                  | Primarily responsible for the team's financial management and oversight. The billing
| [**Pro Viewer**](#pro-viewer-role)            | Has limited read-only access to projects and deployments, ideal for stakeholder coll
| [**Enterprise Viewer**](#enterprise-viewer-role) | Has read-only access to the team's resources and projects.
| [**Contributor**](#contributor-role)          | A unique role that can be configured to have any of the project level roles or none.

See the [Team Level Roles Reference](/docs/rbac/access-roles/team-level-roles) for a complete list of roles and their permissions.

### Owner role

| About                     | Details
| ------------------------- | ----------------------------------------------------------------------------------------------------------
| **Description**           | The owner role is the highest level of authority within a team, possessing comprehensive access and contro
| **Key Responsibilities**  | - Oversee and manage all team resources and projects  - Modify team settings, including [billing](#billing
| **Access and Permissions** | Owners have unrestricted access to all team functionalities, can modify all settings, and change other mem

Teams can have more than one owner. For continuity, we recommend that at least two individuals have owner permissions. Additional owners

See the [Team Level Roles Reference](/docs/rbac/access-roles/team-level-roles) for a complete list of roles and their permissions.

### Member role

Members play a pivotal role in team operations and project management.

**Key responsibilities**

- Create [deployments](/docs/deployments) and manage projects

- Set up [integrations](/docs/integrations) and manage project-specific [domains](/docs/domains)
- Handle [deploy hooks](/docs/deploy-hooks) and adjust [Vercel Function](/docs/functions) settings
- Administer security settings for their assigned projects

**Access and permissions**

Certain team-level settings remain exclusive to owners. Members cannot edit critical team settings like billing information or [invite ne

| About | Details |
| --- | --- |
| **Description** | Members play a pivotal role in team operations and project management. |
| **Key Responsibilities** | - Create [deployments](/docs/deployments) and manage projects - Set up [integrations](/docs/integrations) |
| **Access and Permissions** | Certain team-level settings remain exclusive to owners. Members cannot edit critical team settings like bi |

To assign the member role to a team member, refer to our [Adding team members and assigning roles](/docs/rbac/managing-team-members#addin

See the [Team Level Roles Reference](/docs/rbac/access-roles/team-level-roles) for a complete list of roles and their permissions.

### Developer role

| About | Details |
| --- | --- |
| **Description** | Central to the team's operational functionality, developers ensure a balance between project autonomy and |
| **Key Responsibilities** | - Create [deployments](/docs/deployments) and manage projects - Control [environment variables](/docs/env |
| **Access and Permissions** | While developers have significant access to project functionalities, they are restricted from altering pro |

Central to the team's operational functionality, developers ensure a balance between project autonomy and the safeguarding of essential s

**Key responsibilities**

- Create [deployments](/docs/deployments) and manage projects
- Control [environment variables](/docs/environment-variables), particularly for preview and development environments
- Manage project [domains](/docs/domains)
- Create a [production build](/docs/deployments/environments#production-environment) by committing to the `main` branch of a project. Not

**Access and permissions**

While Developers have significant access to project functionalities, they are restricted from altering production environment variables a

Developers can deploy to production through merging to the production branch for Git projects.

**Additional information**

To assign the developer role to a team member, refer to our [Adding team members and assigning roles](/docs/rbac/managing-team-members#ad

See the [Team Level Roles Reference](/docs/rbac/access-roles/team-level-roles) for a complete list of roles and their permissions.

### Contributor role

Contributors offer flexibility in access control at the project level. To limit team members' access at the project level, they must firs

Contributors may have project-specific role assignments, with the potential for comprehensive control over assigned projects only.

**Key responsibilities**

- Typically assigned to specific projects based on expertise and needs
- Initiate [deployments](/docs/deployments) - *Depending on their assigned [project role](#project-level-roles)*
- Manage [domains](/docs/domains) and set up [integrations](/docs/integrations) for projects if they have the [project administrator](#pr
- Adjust [Vercel functions](/docs/functions) and oversee [deploy hooks](/docs/deploy-hooks)

**Access and permissions**

Contributors can be assigned to specific projects and have the same permissions as [project administrators](#project-administrators), [pr

| About | Details |
| --- | --- |
| **Description** | Contributors offer flexibility in access control at the project level. To limit team members' access at th |
| **Key Responsibilities** | - Typically assigned to specific projects based on expertise and needs - Initiate [deployments](/docs/dep |
| **Access and Permissions** | Contributors can be assigned to specific projects and have the same permissions as [project administrators |

To assign the contributor role to a team member, refer to our [Adding team members and assigning roles](/docs/rbac/managing-team-members#

See the [Team Level Roles Reference](/docs/rbac/access-roles/team-level-roles) for a complete list of roles and their permissions.

### Security role

| About | Details |
| --- | --- |
| **Description** | Inspect and manage Vercel security features. |
| **Key Responsibilities** | - Manage Firewall - Rate Limiting - Deployment Protection |
| **Access and Permissions** | The security role is designed to provide focused access to security features and settings. This role also |

This role does not offer deployment permissions by default.

See the [Team Level Roles Reference](/docs/rbac/access-roles/team-level-roles) for a complete list of roles and their permissions.

### Billing role

| About | Details |
| --- | --- |
| **Description** | Specialized for financial operations, the billing role oversees financial operations and team resources ma |
| **Key Responsibilities** | - Oversee and manage the team's billing information - Review and manage team and project costs - Handle |
| **Access and Permissions** | The billing role is designed to provide financial oversight and management, with access to the team's bill |

The billing role can be assigned at no extra cost. For [Pro teams](/docs/plans/pro-plan), it's limited to one member while for [Enterpris

To assign the billing role to a team member, refer to our [Adding team members and assigning roles](/docs/rbac/managing-team-members#addi

**Compatible permission group:** `UsageViewer`.

See the [Team Level Roles Reference](/docs/rbac/access-roles/team-level-roles) for a complete list of roles and their permissions.

### Pro Viewer role

An observational role designed for Pro teams, Pro Viewer members can monitor team activities and collaborate on projects with limited adm

**Key responsibilities**

- Monitor and inspect all team [projects](/docs/projects/overview) and deployments
- Collaborate on [preview deployments](/docs/deployments/environments#preview-environment-pre-production) with commenting and feedback ca
- Review project-level performance data and analytics

**Access and permissions**

Pro Viewer members have read-only access to core project functionality but cannot view sensitive team data. They are restricted from:

- Viewing observability and log data
- Accessing team settings and configurations
- Viewing detailed usage data and billing information

Pro Viewer members cannot make changes to any settings or configurations.

**Additional information**

Pro Viewer seats are provided free of charge on Pro teams, making them ideal for stakeholders who need project visibility without full ad

To assign the Pro Viewer role to a team member, refer to the [adding team members and assigning roles](/docs/rbac/managing-team-members#a

See the [Team Level Roles Reference](/docs/rbac/access-roles/team-level-roles) for a complete list of roles and their permissions.

### Enterprise Viewer role

| About | Details |
| --- | --- |
| **Description** | An observational role, viewers are informed on team activities without direct intervention. |
| **Key Responsibilities** | - Monitor and inspect all team [projects](/docs/projects/overview)  - Review shared team resources  - Obse |
| **Access and Permissions** | Viewers have broad viewing privileges but are restricted from making changes. |

An observational role with enhanced visibility for Enterprise teams, Enterprise Viewer members have comprehensive read-only access to tea

**Key responsibilities**

- Monitor and inspect all team [projects](/docs/projects/overview) and deployments
- Collaborate on [preview deployments](/docs/deployments/environments#preview-environment-pre-production) with commenting and feedback ca
- Review project-level performance data and analytics
- Access observability and log data for troubleshooting and monitoring
- View team settings and configurations for governance and compliance
- Monitor usage data and resource consumption patterns

**Access and permissions**

Enterprise Viewer members have comprehensive read-only access across the team, including sensitive operational data that Pro viewers cann

Enterprise Viewer members cannot make changes to any settings or configurations but have visibility into all team operations.

**Additional information**

The enhanced access provided by Enterprise Viewer roles makes them ideal for compliance officers, auditors, and senior stakeholders who n

To assign the Enterprise Viewer role to a team member, refer to the [adding team members and assigning roles](/docs/rbac/managing-team-me

**Compatible permission group:** `UsageViewer`.

See the [Team Level Roles Reference](/docs/rbac/access-roles/team-level-roles) for a complete list of roles and their permissions.

## Project level roles

Project level roles provide fine-grained control and access to specific projects within a team. These roles are assigned to individuals a

| Role | Description |
| --- | --- |
| [**Project Administrator**](#project-administrators) | Team owners and members inherently act as project administrators for every proje |
| [**Project Developer**](#project-developer) | Can deploy to the project and manage its environment settings. Team developers i |
| [**Project Viewer**](#project-viewer) | Has read-only access to a specific project. Both team billing and viewer members |

See the [Project Level Roles Reference](/docs/rbac/access-roles/project-level-roles) for a complete list of roles and their permissions.

### Project administrators

| About | Details |
| --- | --- |
| **Description** | Project administrators hold significant authority at the project level, operating as the project-level cou |
| **Key Responsibilities** | - Govern [project settings](/docs/projects/overview#project-settings)  - Deploy to all [environments](/doc |
| **Access and Permissions** | Their authority doesn't extend across all [projects](/docs/projects/overview) within the team. Project adm |

To assign the project administrator role to a team member, refer to our [Assigning project roles](/docs/rbac/managing-team-members#assign

See the [Project Level Roles Reference](/docs/rbac/access-roles/project-level-roles) for a complete list of roles and their permissions.

### Project developer

| About | Details |
| --- | --- |
| **Description** | Project developers play a key role in working on projects, mirroring the functions of [team developers](#d |
| **Key Responsibilities** | - Initiate [deployments](/docs/deployments)  - Manage [environment variables](/docs/environment-variables) |
| **Access and Permissions** | Project developers have limited scope, with access restricted to only the projects they're assigned to. |

To assign the project developer role to a team member, refer to our [Assigning project roles](/docs/rbac/managing-team-members#assigning-

See the [Project Level Roles Reference](/docs/rbac/access-roles/project-level-roles) for a complete list of roles and their permissions.

### Project viewer

| About | Details |
| --- | --- |
| **Description** | Adopting an observational role within the project scope, they ensure transparency and understanding across |
| **Key Responsibilities** | - View and inspect all [deployments](/docs/deployments)  - Review [project settings](/docs/projects/overvi |
| **Access and Permissions** | They have a broad view but can't actively make changes. |

To assign the project viewer role to a team member, refer to our [Assigning project roles](/docs/rbac/managing-team-members#assigning-pro

See the [Project Level Roles Reference](/docs/rbac/access-roles/project-level-roles) for a complete list of roles and their permissions.

## Permission groups

Existing team roles can be combined with permission groups to create custom access configurations based on your team's specific needs. Th

| Permission | Description |
| --- | --- |
| **Create Project** | Allows the user to create a new project. |
| **Full Production Deployment** | Deploy to production from CLI, rollback and promote any deployment. |
| **Usage Viewer** | Read-only usage team-wide including prices and invoices. |
| **Environment Manager** | Create and manage project environments. |
| **Environment Variable Manager** | Create and manage environment variables. |
| **Deployment Protection Manager** | Configure password protection, deployment protection by pass, and Vercel Authentication for project |

See [project level roles](/docs/rbac/access-roles/project-level-roles) and [team level roles](/docs/rbac/access-roles/team-level-roles) f

---
title: "Project Level Roles"
description: "Learn about the project level roles and their permissions."
last_updated: "2026-01-16T02:19:35.078Z"
source: "https://vercel.com/docs/rbac/access-roles/project-level-roles"
---

# Project Level Roles

Project level roles are assigned to a team member on a project level. This means that the role is only valid for the project it is assign

## Equivalency roles

In the table below, the relationship between team and project roles is indicated by the column headers. For example, the team role "Devel

- The [**Developer**](/docs/rbac/access-roles#developer-role) team role is equivalent to the [**Project Developer**](/docs/rbac/access-ro
- The [**Viewer Pro**](/docs/rbac/access-roles#viewer-pro-role), [**Viewer Enterprise**](/docs/rbac/access-roles#viewer-enterprise-role),
- The [**Owner**](/docs/rbac/access-roles#owner-role) and [**Member**](/docs/rbac/access-roles#member-role) team roles are equivalent to

All project level roles can be assigned to those with the [**Contributor**](/docs/rbac/access-roles#team-level-roles) team role.

See our [Access roles docs](/docs/rbac/access-roles) for a more comprehensive breakdown of the different roles.

## Project level permissions

---
title: "Team Level Roles"
description: "Learn about the different team level roles and the permissions they provide."
last_updated: "2026-01-16T02:19:35.069Z"
source: "https://vercel.com/docs/rbac/access-roles/team-level-roles"
---

# Team Level Roles

Team level roles are designed to provide a comprehensive level of control and access to the team as a whole. These roles are assigned to

> **💡 Note:** While the [Enterprise](/docs/plans/enterprise) plan supports all the below
> roles, the [Pro](/docs/plans/pro-plan) plan only supports
> [Owner](/docs/rbac/access-roles#owner-role),
> [Member](/docs/rbac/access-roles#owner-role), and
> [Billing](/docs/rbac/access-roles#billing-role).

---
title: "Managing Team Members"
description: "Learn how to manage team members on Vercel, and how to assign roles to each member with role-based access control (RBAC)."
last_updated: "2026-01-16T02:19:35.089Z"
source: "https://vercel.com/docs/rbac/managing-team-members"
---

# Managing Team Members

As the team owner, you have the ability to manage your team's composition and the roles of its members, controlling the actions they can

## Adding team members and assigning roles

1. From the dashboard, select your team from the [scope selector](/docs/dashboard-features#scope-selector)

2. Select the **Settings** tab and go to the **Members** section

3. Enter the email address of the person you would like to invite, assign their [role](/docs/rbac/access-roles), and select the **Invite*

4. By default only the team level roles are visible in the dropdown. If you choose to assign the [contributor role](/docs/rbac/access-rol

5. You can view all pending invites in the **Pending Invitations** tab. When you issue an invite the recipient is not automatically added

6. Once a member has been accepted onto the team, you can edit their role using the **Manage Role** button located alongside their assign

### Invite link

Team owners can also share an invite link with others to allow them to join the team without needing to be invited individually.

To generate an invite link:

1. Ensure you have selected your team from the [scope selector](/docs/dashboard-features#scope-selector)
2. Select the **Settings** tab and go to the **Members** section
3. Select the **Invite Link** button and use the  icon to copy the invite link:
4. Optionally, you can select **Reset Invite Link** to generate a new link. After doing this, all other invite links will become invalid.
5. Share the link with others.
   Those who join from an invite link will be given the lowest permissions for that team. For the Enterprise plan, they will be assigned

## Assigning project roles

Team [owners](/docs/rbac/access-roles#owner-role) can assign project roles to team members with the [contributor role](/docs/rbac/access-

1. Ensure you have selected your team from the [scope selector](/docs/dashboard-features#scope-selector)
2. Select the project you want to assign a member to
3. Select **Access** from the left navigation, then inside the **Project Access** section select the team members email from the dropdown
4. Select the role you want to assign to the member on the project

## Delete a member

Team owners can delete members from a team. You can also remove yourself from a team.

1. Ensure you have selected your team from the [scope selector](/docs/dashboard-features#scope-selector)
2. Select the **Settings** tab and go to the **Members** section
3. Next to the name of the person you'd like to remove, select the ellipses (…) and then select **Remove from Team** from the menu

Vercel is also [SCIM](# "System for Cross-domain Identity Management") compliant. This means that if you are using SAML SSO, de-provision

---
title: "Role-based access control (RBAC)"
description: "Learn how to manage team members on Vercel, and how to assign roles to each member with role-based access control (RBAC)."
last_updated: "2026-01-16T02:19:35.096Z"
source: "https://vercel.com/docs/rbac"
---

# Role-based access control (RBAC)

Teams consist of members, and each member of a team can get assigned a role. These roles define what you can and cannot do within a team

As your project scales and you add more team members, you can assign them roles to ensure that they have the right permissions to work on

Vercel offers a range of roles for your team members. When deciding what role a member should have on your team, consider the following:

- What projects does this team member need to access?
- What actions does this team member need to perform on these projects?
- What actions does this team member need to perform on the team itself?

See the [Managing team members](/docs/rbac/managing-team-members) section for information on setting up and managing team members.

For specific information on the different access roles available on each plan, see the [Access Roles](/docs/rbac/access-roles) section.

## More resources

- [Managing team members](/docs/rbac/managing-team-members)
- [Access groups](/docs/rbac/access-groups)
- [Access roles](/docs/rbac/access-roles)

---
title: "Getting Started"
description: "Learn how to import thousands of simple redirects from CSV, JSON, or JSONL files."
last_updated: "2026-01-16T02:19:35.117Z"
source: "https://vercel.com/docs/redirects/bulk-redirects/getting-started"
---

# Getting Started

Bulk redirects can be specified either as part of a Vercel deployment or updated immediately through the UI, API, or CLI by settings redi

- [Deployment-time redirects](#deployment-time-redirects)
- [Project-level redirects](#project-redirects)

## Deployment-time redirects

Bulk redirects in deployments are specified in the `bulkRedirectsPath` field in `vercel.json`. `bulkRedirectsPath` can point to either a

Learn more about bulk redirects fields and file formats in the [project configuration documentation](/docs/projects/project-configuration

- ### Create your redirect file
  You can create fixed files of redirects, or generate them at build time as long as they end up in the location specified by  before the
  ```csv filename="redirects.csv"
  source,destination,permanent
  /old-blog,/blog,true
  /old-about,/about,false
  /legacy-contact,https://example.com/contact,true
  ```

- ### Configure bulkRedirectsPath
  Add the `bulkRedirectsPath` property to your `vercel.json` file, pointing to your redirect file. You can also point to a folder contain
  ```json filename="vercel.json"
  {
    "bulkRedirectsPath": "redirects.csv"
  }
  ```

- ### Deploy
  Deploy your project to Vercel. Your bulk redirects will be processed and applied automatically.

  ```bash
  vercel deploy
  ```
  Any errors processing the bulk redirects will appear in the build logs for the deployment.


---
title: "Bulk redirects"
description: "Learn how to import thousands of simple redirects from CSV, JSON, or JSONL files."
last_updated: "2026-01-16T02:19:35.124Z"
source: "https://vercel.com/docs/redirects/bulk-redirects"
---

# Bulk redirects

## Limits and pricing

Each project has a free configurable capacity of bulk redirects, and additional bulk redirect capacity can be purchased in groups of 25,0

- Bulk redirects do not support wildcard or header matching
- Bulk redirects do not work locally while using `vercel dev`
- A maximum of 1,000,000 bulk redirects can be configured per project.


---
title: "Configuration Redirects"
description: "Learn how to define static redirects in your framework configuration or vercel.json with support for wildcards, pattern mat
last_updated: "2026-01-16T02:19:35.149Z"
source: "https://vercel.com/docs/redirects/configuration-redirects"
---

# Configuration Redirects

Configuration redirects define routing rules that Vercel evaluates at build time. Use them for permanent redirects (`308`), temporary red

Define configuration redirects in your framework's config file or in the `vercel.json` file, which is located in the root of your applica

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "redirects": [
    { "source": "/me", "destination": "/profile.html" },
    { "source": "/user", "destination": "/api/user", "permanent": false },
    {
      "source": "/view-source",
      "destination": "https://github.com/vercel/vercel"
    },
    {
      "source": "/:path((?!uk/).*)",
      "has": [
        {
          "type": "header",
          "key": "x-vercel-ip-country",
          "value": "GB"
        }
      ],
      "destination": "/uk/:path*",
      "permanent": false
    }
  ]
}
```

View the full [API reference](/docs/projects/project-configuration#redirects) for the `redirects` property.

> **💡 Note:** Using `has` does not yet work locally while using `vercel dev`, but does work
> when deployed.

> For \["nextjs","nextjs-app"]:

When using Next.js, you do *not* need to use `vercel.json`. Instead, use the framework-native `next.config.js` to define configuration-ba

```js filename="next.config.js"
module.exports = {
  async redirects() {
    return [
      {
        source: '/about',
        destination: '/',
        permanent: true,
      },
      {
        source: '/old-blog/:slug',
        destination: '/news/:slug',
        permanent: true,
      },
      {
        source: '/:path((?!uk/).*)',
        has: [
          {
            type: 'header',
            key: 'x-vercel-ip-country',
            value: 'GB',
          },
        ],
        permanent: false,
        destination: '/uk/:path*',
      },
    ];
  },
```

};
```

Learn more in the [Next.js documentation](https://nextjs.org/docs/app/building-your-application/routing/redirecting).

> For \['sveltekit']:

Use `vercel.json`, see above.

> For \['nuxt']:

When using Nuxt, you do *not* need to use `vercel.json`. Instead, use the framework-native `nuxt.config.ts` to define configuration-based

```ts filename="nuxt.config.ts"
export default defineNuxtConfig({
  routeRules: {
    '/old-page': { redirect: '/new-page' },
    '/old-page2': { redirect: { to: '/new-page', statusCode: 308 } },
  },
});
```

> For \['other']:

Use `vercel.json`, see above.

When deployed, these redirect rules will be deployed to every [region](/docs/regions) in Vercel's CDN.

## Limits

The [/.well-known](# "The /.well-known directory") path is reserved and cannot be redirected or rewritten. Only
Enterprise teams can configure custom SSL. [Contact sales](/contact/sales) to
learn more.

If you are exceeding the limits below, we recommend using Middleware and Edge Config to [dynamically read redirect values](/docs/redirect

| Limit | Maximum |
| --- | --- |
| Number of redirects in the array | 2,048 |
| String length for `source` and `destination` | 4,096 |

---
title: "Redirects"
description: "Learn how to use redirects on Vercel to instruct Vercel"
last_updated: "2026-01-16T02:19:35.157Z"
source: "https://vercel.com/docs/redirects"
---

# Redirects

Redirects are rules that instruct Vercel to send users to a different URL than the one they requested. For example, if you rename a publi

With redirects on Vercel, you can define HTTP redirects in your application's configuration, regardless of the [framework](/docs/framewor

## Use cases

- **Moving to a new domain:** Redirects help maintain a seamless user experience when moving a website to a new domain by ensuring that v
- **Replacing a removed page:** If a page has been moved, temporarily or permanently, you can use redirects to send users to a relevant n
- **Canonicalization of multiple URLs:** If your website can be accessed through several URLs (e.g., `acme.com/home`, `home.acme.com`, or
- **Geolocation-based redirects:** Redirects can be configured to consider the source country of requests, enabling tailored experiences

We recommend using status code `307` or `308` to avoid the ambiguity of non `GET` methods, which is necessary when your application needs

## Implementing redirects

Review the table below to understand which redirect method best fits your use case:

| Redirect method | Use case |
| --- | --- |
| [Configuration redirects](/docs/redirects/configuration-redirects) | Support needed for wildcards, pattern matching, and geolocation-ba |
| [Bulk redirects](/docs/redirects/bulk-redirects) | For large-scale migrations or maintaining extensive redirect lists |
| [Vercel Functions](#vercel-functions) | For complex custom redirect logic. |
| [Middleware](#middleware) | Dynamic redirects that need to update without redeploying. |
| [Domain redirects](#domain-redirects) | Domain-level redirects such as www to apex domain. |
| [Firewall redirects](#firewall-redirects) | Emergency redirects that must execute before other redirects. |

### Vercel Functions

Use Vercel Functions to implement any redirect logic you need. This may not be optimal depending on the use case.

Any route can redirect requests like so:

```ts filename="pages/api/handler.ts" framework=nextjs
import { NextApiRequest, NextApiResponse } from 'next';

export default function handler(
  request: NextApiRequest,
  response: NextApiResponse,
) {
  // Use 308 for a permanent redirect, 307 for a temporary redirect
  return response.redirect(307, '/new-route');
}
```

```js filename="pages/api/handler.js" framework=nextjs
export default function handler(request, response) {
  // Use 308 for a permanent redirect, 307 for a temporary redirect
  return response.redirect(307, '/new-route');
}
```
```

```

```ts filename="app/api/route.ts" framework=nextjs-app
import { redirect } from 'next/navigation';

export async function GET(request: Request) {
  redirect('https://nextjs.org/');
}
```

```js filename="app/api/route.js" framework=nextjs-app
import { redirect } from 'next/navigation';

export async function GET(request) {
  redirect('https://nextjs.org/');
}
```

```ts filename="src/routes/user/+layout.server.ts" framework=sveltekit
import { redirect } from '@sveltejs/kit';
import type { LayoutServerLoad } from './$types';

export const load = (({ locals }) => {
  if (!locals.user) {
    throw redirect(307, '/login');
  }
}) satisfies LayoutServerLoad;
```

```js filename="src/routes/user/+layout.server.js" framework=sveltekit
import { redirect } from '@sveltejs/kit';

/** @type {import('./$types').LayoutServerLoad} */
export function load({ locals }) {
  if (!locals.user) {
    throw redirect(307, '/login');
  }
}
```

```ts filename="server/api/foo.get.ts" framework=nuxt
export default defineEventHandler((event) => {
  return sendRedirect(event, '/path/redirect/to', 307);
});
```

```js filename="server/api/foo.get.js" framework=nuxt
export default defineEventHandler((event) => {
  return sendRedirect(event, '/path/redirect/to', 307);
});
```

```ts filename="api/handler.ts" framework=other
import type { VercelRequest, VercelResponse } from '@vercel/node';

export default function handler(
  request: VercelRequest,
  response: VercelResponse,
) {
  // Use 308 for a permanent redirect, 307 for a temporary redirect
  return response.redirect(307, '/new-route');
}
```

```js filename="api/handler.js" framework=other
export default function handler(request, response) {
  // Use 308 for a permanent redirect, 307 for a temporary redirect
  return response.redirect(307, '/new-route');
}
```

### Middleware

For dynamic, critical redirects that need to run on every request, you can use [Middleware](/docs/routing-middleware) and [Edge Config](/

Redirects can be stored in an Edge Config and instantly read from Middleware. This enables you to update redirect values without having t

[Deploy a template](https://vercel.com/templates/next.js/maintenance-page) to get started.

### Domain Redirects

You can redirect a `www` subdomain to an apex domain, or other domain redirects, through the [Domains](/docs/projects/domains/deploying-a

### Firewall Redirects

In emergency situations, you can also define redirects using [Firewall rules](/docs/security/vercel-waf/examples#emergency-redirect) to r

## Redirect status codes

- **307 Temporary Redirect**: Not cached by client, the method and body never changed. This type of redirect does not affect SEO and sear
- **302 Found**: Not cached by client, the method may or may not be changed to `GET`.
- **308 Permanent Redirect**: Cached by client, the method and body never changed. This type of redirect does not affect SEO and search e
- **301 Moved Permanently**: Cached by client, the method may or may not be changed to `GET`.

## Observing redirects

You can observe your redirect performance using Observability. The **Edge Requests** tab shows request counts and cache status for your r

Learn more in the [Observability Insights](/docs/observability/insights#edge-requests) documentation.
```

## Draining redirects

You can export redirect data by draining logs from your application. Redirect events appear in your runtime logs, allowing you to analyze

To get started, configure a [logs drain](/docs/drains/using-drains).

## Best practices for implementing redirects

There are some best practices to keep in mind when implementing redirects in your application:

1. **Test thoroughly**: Test your redirects thoroughly to ensure they work as expected. Use a [preview deployment](/docs/deployments/envi
2. **Use relative paths**: Use relative paths in your `destination` field to avoid hardcoding your domain name
3. **Use permanent redirects**: Use [permanent redirects](#adding-redirects "Adding Redirects") for permanent URL changes and [temporary
4. **Use wildcards carefully**: Wildcards can be powerful but should be used with caution. For example, if you use a wildcard in a source
5. **Prioritize HTTPS**: Use redirects to enforce HTTPS for all requests to your domain

---
title: "Redis on Vercel"
description: "Learn how to use Redis stores through the Vercel Marketplace."
last_updated: "2026-01-16T02:19:35.143Z"
source: "https://vercel.com/docs/redis"
---

# Redis on Vercel

Vercel lets you connect external Redis databases through the [Marketplace](/marketplace), allowing you to integrate high-performance cach

> **💡 Note:** Vercel KV is no longer available. If you had an existing Vercel KV store, we automatically moved it to [Upstash Redis](htt

- Explore [Marketplace storage redis integrations](/marketplace?category=storage\&search=redis).
- Learn how to [add a Marketplace native integration](/docs/integrations/install-an-integration/product-integration).

## Connecting to the Marketplace

Vercel enables you to use Redis by integrating with external database providers. By using the Marketplace, you can:

- Select a [Redis provider](/marketplace?category=storage\&search=redis)
- Provision and configure a Redis database with minimal setup.
- Have credentials and [environment variables](/docs/environment-variables) injected into your Vercel project.

---
title: "Vercel Regions"
description: "View the list of regions supported by Vercel"
last_updated: "2026-01-16T02:19:35.180Z"
source: "https://vercel.com/docs/regions"
---

# Vercel Regions

**Vercel's CDN** is a globally distributed platform that stores content and runs compute close to your users and data, reducing latency a

## Global infrastructure

Vercel's CDN is built on a sophisticated global infrastructure designed to optimize performance and reliability:

- **Points of Presence (PoPs)**: We operate over 126 PoPs distributed across the globe. These PoPs serve as the first point of contact fo
- **Vercel Regions**: Behind these PoPs, we maintain  compute-capable regions where your code can run close to your data.
- **Private Network**: Traffic flows from PoPs to the nearest region through private, low-latency connections, ensuring fast and efficien

This architecture balances the benefits of widespread geographical distribution with the efficiency of concentrated caching and compute r

### Caching strategy

Our approach to caching is designed to maximize efficiency and performance:

- By maintaining fewer, dense regions, we increase cache hit probability. This means that popular content is more likely to be available
- The extensive PoP network ensures that users can quickly access regional caches, minimizing latency.
- This concentrated caching strategy results in higher cache hit ratios, reducing the need for requests to go back to the origin server a

## Region list

For information on different resource pricing based on region, see the [regional pricing](/docs/pricing/regional-pricing) page.

### Points of Presence (PoPs)

In addition to our  compute-capable regions, Vercel's CDN includes 126 PoPs distributed across the globe. These PoPs serve several crucia

1. Request routing: PoPs intelligently route requests to the nearest or most appropriate edge region with single-digit millisecond latenc
2. DDoS protection: They provide a first line of defense against distributed denial-of-service attacks.
3. SSL termination: PoPs handle SSL/TLS encryption and decryption, offloading this work from origin servers.

The extensive PoP network ensures that users worldwide can access your content with minimal latency, even if compute resources are concen

## Local development regions

When you use [the `vercel dev` CLI command to mimic your deployment environment locally](/docs/cli/dev), the region is assigned `dev1` to

| Region Code | Reference Location |
| ----------- | ------------------ |
| dev1        | localhost          |

## Compute defaults

- Vercel Functions default to running in the `iad1` (Washington, D.C., USA) region. Learn more about [changing function regions](/docs/fu

Functions should be executed in the same region as your database, or as close to it as possible, [for the lowest latency](/docs/functions

## Outage resiliency

Vercel's CDN is designed with high availability and fault tolerance in mind:

- In the event of regional downtime, application traffic is automatically rerouted to the next closest region. This ensures that your app
- Traffic will be rerouted to the next closest region in the following order:

* For Enterprise customers, Vercel functions can automatically failover to a different region if the region they are running in becomes u

This multi-layered approach to resiliency, combining our extensive PoP network with intelligent routing and regional failover capabilitie

---
title: "Release Phases for Vercel"
description: "Learn about the different phases of the Vercel Product release cycle and the requirements that a Product must meet before b
last_updated: "2026-01-16T02:19:35.186Z"
source: "https://vercel.com/docs/release-phases"
---

# Release Phases for Vercel

This page outlines the different phases of the Vercel product release cycle. Each phase has a different set of requirements that a produc

Although a product doesn't have to pass through each stage in sequential order, there is a default flow to how products are released:

- Alpha
- Beta
- General Availability (GA).

## Alpha

The Alpha phase is the first phase of the release cycle. A product in the Alpha phase lacks the essential features that are required to b
The product is considered to still be under development, and is being built to be ready for Beta phase.

> **💡 Note:** The product is under development.

## Beta

A Beta state generally means that the feature does **not** yet meet our quality standards for GA or limited availability.
An example of this is when there is a need for more information or feedback from external customers to validate that this feature solves

Releases in the Beta state have a committed timeline for getting to GA and are actively worked on.

> **⚠️ Warning:** Products in a Beta state, are  covered under the [Service
> Level Agreement](https://vercel.com/legal/sla) (SLA) for Enterprise plans.
> Vercel  recommend using Beta products in a full
> production environment.

### Private Beta

When a product is in Private Beta, it is still considered to be under development.
While some customers may have access, this access sometimes includes a Non-disclosure agreement (NDA)

> **💡 Note:** The product is under active development with limited customer access - may
> include an NDA.

### Limited Beta

A Limited Beta is still under active development, but has been publicly announced, and is potentially available to a limited number of cu

This phase is generally used when there is a need to control adoption of a feature.
For example, when underlying capacity is limited, if there are known severe caveats then additional guidance may be required.

> **💡 Note:** The product is under active development, and has been publicly announced.
> Limited customer access - may include an NDA.

### Public Beta

Once a product has been publicly announced, optionally tested in the field by selected customers, and meets Vercel's quality standards, i

Public Beta is the final phase of the release cycle before a product goes GA. At this stage the product can be used by a wider audience f

For a product to move from Public Beta to GA, the following requirements must be met. Note that these are general requirements, and that

- Fully load tested
- All bugs resolved
- Security analysis completed
- At least 10 customers have been on-boarded

> **💡 Note:** The product is under active development, and has been publicly announced.
> Available to the public without special invitation.

See the [Public Beta Agreement](/docs/release-phases/public-beta-agreement) for detailed information.

## General Availability

When the product reaches the General Availability (GA) phase, it is considered to be battle tested, and ready for use by the community.

> **💡 Note:** Publicly available with full support and guaranteed uptime.

## Deprecated and Sunset

A Deprecated state means that the product team is in the process of removing a product or feature.
Deprecated states are accompanied by documentation instructing existing users of remediation next steps, and information on when to expec

The ultimate state after Deprecation is Sunset. Sunset implies that there should be no customers using the Product and any artifacts with

---
title: "Public Beta Agreement"
description: "The following is the Public Beta Agreement for Vercel products in the Public Beta release phase, including any services or

last_updated: "2026-01-16T02:19:35.199Z"
source: "https://vercel.com/docs/release-phases/public-beta-agreement"
---

# Public Beta Agreement

This Public Beta Agreement (“Agreement”) is made and entered into effective as of the date You first agree to this Agreement (“Effective

If You are entering into these terms on behalf of a company or other legal entity, You represent that You have the legal authority to bin

## 1. Definitions

### 1.1 “Authorized User”

Any employee, contractor, or member of your organization (if applicable) who has been authorized to use the Services in accordance with t

### 1.2 “Public Beta Period”

The period commencing on the Effective Date and ending upon the release by Vercel of a generally available version of the Product or term

### 1.3 “Product”

The public beta version of any features, functionality, Software, SaaS, and all associated documentation (if any) (“Documentation”), coll

### 1.4 “Software”

The public beta version of Vercel's proprietary software, if any, provided hereunder.

### 1.5 “Terms”

Our Terms of Service or Enterprise Terms and Conditions, or any other agreements you have entered into with us for the provision of our s

## 2. License Grant

Subject to your compliance with the Terms and this Agreement, Vercel hereby grants You a non-exclusive, non-transferable, limited license

- (i) access and use the Product and/or any associated Software;
- (ii) use all associated Documentation in connection with such authorized use of the Product and/or Software; and
- (iii) make one copy of any Documentation solely for archival and backup purposes.

In all cases of (i) - (iii) solely for Your personal or internal business use purposes.

## 3. Open Source Software

The Software may contain open source software components (“Open Source Components”). Such Open Source Components are not licensed under t

## 4. Permissions and Restrictions

By agreeing to this Agreement, You allow the Product to connect to Your Vercel account. You must have a valid and active Vercel account i

- (i) reverse engineer, reverse assemble, or otherwise attempt to discover the source code of all or any portion of the Product;
- (ii) reproduce, modify, translate or create derivative works of all or any portion of the Product;
- (iii) export the Software or assist any third party to gain access, license, sublicense, resell distribute, assign, transfer or use the
- (iv) remove or destroy any proprietary notices contained on or in the Product or any copies thereof; or
- (v) publish or disclose the results of any benchmarking of the Product, or use such results for Your own competing software development

## 5. Disclaimer of Warranty

The Product made available to You is in "Beta” form, pre-release, and time limited. The Product may be incomplete and may contain errors o

YOU AGREE THAT VERCEL AND ITS LICENSORS PROVIDE THE PRODUCTS ON AN “AS IS” AND “WHERE IS” BASIS. NEITHER VERCEL NOR ITS LICENSORS MAKE AN

## 6. Intellectual Property Rights; Support and Feedback

### 6.1 Intellectual Property Rights

All rights, title and interest in and to the Product and any improved, updated, modified or additional parts thereof, shall at all times

### 6.2 Support

Notwithstanding the disclaimer of warranty above, Vercel may, but is not required to provide You with support on the use of the Product i

### 6.3 Feedback

You agree to use reasonable efforts to provide Vercel with oral feedback and/or written feedback related to Your use of the Product, incl

## 7. Limitation of Liability; Allocation of Risk

### 7.1 Limitation of Liability

NEITHER VERCEL NOR ITS LICENSORS SHALL BE LIABLE FOR SPECIAL, INCIDENTAL, CONSEQUENTIAL OR INDIRECT DAMAGES, RELATED TO THIS AGREEMENT, I

### 7.2 Allocation of Risk

You and Vercel agree that the foregoing Section 7.1 on limitation of liability and the Section 5 above on warranty disclaimer fairly allo

## 8. Term and Termination

### 8.1 Term and Termination

This Agreement will continue in effect until the expiration of the Public Beta Period, unless otherwise extended in writing by Vercel, in

### 8.2 Termination

You may terminate this Agreement at any time by ceasing use of the Product. This Agreement will terminate immediately upon written notice

## 9. Government End Users

Software provided under this Agreement is commercial computer software programs developed solely at private expense. As defined in U.S. F

## 10. General Provisions

All notices under this Agreement will be in writing and will be deemed to have been duly given when received, if personally delivered; wh

The parties agree that the United Nations Convention on Contracts for the International Sale of Goods is specifically excluded from appli

This Agreement may not be assigned, sublicensed or otherwise transferred by either party without the other party's prior written consent

---
title: "Request Collapsing"
description: "Learn how Vercel"
last_updated: "2026-01-16T02:19:35.284Z"
source: "https://vercel.com/docs/request-collapsing"
---

# Request Collapsing

Vercel uses **request collapsing** to protect uncached routes during high traffic. It reduces duplicate work by combining concurrent requ

## How request collapsing works

When a request for an uncached path arrives, Vercel invokes the origin [function](/docs/functions) and stores the response in the [cache]

However, if multiple requests arrive while the initial function is still processing, the cache is still empty. Instead of triggering addi

This prevents overwhelming the origin with duplicate work during traffic spikes and helps ensure faster, more stable performance.

Vercel also applies request collapsing when serving [STALE](/docs/headers/response-headers#stale) responses (with [stale-while-revalidate

### Example

Suppose a new blog post is published and receives 1,000 requests at once. Without request collapsing, each request would trigger a separa

With request collapsing, Vercel handles the first request, then holds the remaining 999 requests until the initial response is ready. Onc

## Supported features

Request collapsing is supported for:

- [Incremental Static Regeneration (ISR)](/docs/incremental-static-regeneration)
- [Image Optimization](/docs/image-optimization)

---
title: "Rewrites on Vercel"
description: "Learn how to use rewrites to send users to different URLs without modifying the visible URL."
last_updated: "2026-01-16T02:19:35.315Z"
source: "https://vercel.com/docs/rewrites"
---

# Rewrites on Vercel

A rewrite routes a request to a different destination without changing the URL in the browser. Unlike redirects, the user won't see the U

There are two main types:

1. **Same-application rewrites** – Route requests to different pages within your Vercel project.
2. **External rewrites** – Forward requests to an external API or website.

The [/.well-known](# "The /.well-known directory") path is reserved and cannot be redirected or rewritten. Only
Enterprise teams can configure custom SSL. [Contact sales](/contact/sales) to
learn more.

## Setting up rewrites

Rewrites are defined in a `vercel.json` file in your project's root directory:

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "rewrites": [
    {
      "source": "/source-path",
      "destination": "/destination-path"
    }
  ]
}
```

For all configuration options, see the [project configuration](/docs/project-configuration#rewrites) docs.

## Same-application rewrites

Same-application rewrites route requests to different destinations within your project. Common uses include:

- **Friendly URLs**: Transform `/products/t-shirts` into `/catalog?category=t-shirts`
- **Device-specific content**: Show different layouts based on device type
- **A/B testing**: Route users to different versions of a page
- **Country-specific content**: Show region-specific content based on the user's location

Example: Route image resize requests to a serverless function:

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "rewrites": [
    {
      "source": "/resize/:width/:height",
      "destination": "/api/sharp"
```

```
    }
  ]
}
```

This converts a request like `/resize/800/600` to `/api/sharp?width=800&height=600`.

Example: Route UK visitors to a UK-specific section:

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "rewrites": [
    {
      "source": "/:path((?!uk/).*)",
      "has": [
        { "type": "header", "key": "x-vercel-ip-country", "value": "GB" }
      ],
      "destination": "/uk/:path*"
    }
  ]
}
```

This routes a UK visitor requesting `/about` to `/uk/about`.

## External rewrites

External rewrites forward requests to APIs or websites outside your Vercel project, effectively allowing Vercel to function as a reverse

- **Proxy API requests**: Hide your actual API endpoint
- **Combine multiple services**: Merge multiple backends under one domain
- **Create microfrontends**: Combine multiple Vercel applications into a single website
- **Add caching**: Cache external API responses on the CDN
- **Serve externally hosted content**: Serve content that is not hosted on Vercel.

Example: Forward API requests to an external endpoint:

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "rewrites": [
    {
      "source": "/api/:path*",
      "destination": "https://api.example.com/:path*"
    }
  ]
}
```

A request to `/api/users` will be forwarded to `https://api.example.com/users` without changing the URL in the browser.

### Caching external rewrites

The CDN can cache external rewrites for better performance. There are three approaches to enable caching:

1. **Directly from your API (preferred)**: When you control the backend API, the API itself can return [`CDN-Cache-Control`](/docs/header

   ```
   CDN-Cache-Control: max-age=60
   ```

   This will cache API responses on the CDN for 60 seconds.

2. **Using Vercel Configuration**: When you can't modify the backend API, set the caching headers in your Vercel configuration:

   ```json filename="vercel.json"
   {
     "$schema": "https://openapi.vercel.sh/vercel.json",
     "rewrites": [
       {
         "source": "/api/:path*",
         "destination": "https://api.example.com/:path*"
       }
     ],
     "headers": [
       {
         "source": "/api/:path*",
         "headers": [
           {
             "key": "CDN-Cache-Control",
             "value": "max-age=60"
           }
         ]
       }
     ]
   }
   ```

   This will cache API responses on the CDN for 60 seconds.

3. **Using `x-vercel-enable-rewrite-caching` (fallback)**: Use this approach only when you cannot control the caching headers from the ex

   ```json filename="vercel.json"
   {
     "$schema": "https://openapi.vercel.sh/vercel.json",
     "headers": [
       {
         "source": "/api/:path*",
         "headers": [{ "key": "x-vercel-enable-rewrite-caching", "value": "1" }]
```

```
      }
    ]
  }
```

   This instructs Vercel to respect the `Cache-Control` header from the external API.

For more information on caching headers and detailed options, see the [Cache-Control headers documentation](/docs/headers/cache-control-h

### Draining external rewrites

You can export external rewrite data by draining logs from your application. External rewrite events appear in your runtime logs, allowin

To get started, configure a [logs drain](/docs/drains/using-drains).

### Observing external rewrites

You can observe your external rewrite performance using Observability. The **External Rewrites** tab shows request counts, connection lat

Learn more in the [Observability Insights](/docs/observability/insights#external-rewrites) documentation.

## Framework considerations

**External rewrites** work universally with all frameworks, making them ideal for API proxying, microfrontend architectures, and serving

For **same-application rewrites**, always prefer your framework's native routing capabilities:

- **Next.js**: [Next.js rewrites](https://nextjs.org/docs/api-reference/next.config.js/rewrites)
- **Astro**: [Astro routing](/docs/frameworks/astro#rewrites)
- **SvelteKit**: [SvelteKit routing](/docs/frameworks/sveltekit#rewrites)

Use `vercel.json` rewrites for same-application routing only when your framework doesn't provide native routing features. Always consult y

## Testing rewrites

Use Vercel's preview deployments to test your rewrites before going to production. Each pull request creates a unique preview URL where y

## Wildcard path forwarding

You can capture and forward parts of a path using wildcards:

```json
{
  "rewrites": [
    {
      "source": "/docs/:path*",
      "destination": "/help/:path*"
    }
  ]
}
```

A request to `/docs/getting-started/install` will be forwarded to `/help/getting-started/install`.

You can also capture multiple path segments:

```json
{
  "rewrites": [
    {
      "source": "/blog/:year/:month/:slug*",
      "destination": "/posts?date=:year-:month&slug=:slug*"
    }
  ]
}
```

## Using regular expressions

For more complex patterns, you can use regular expressions with capture groups:

```json
{
  "rewrites": [
    {
      "source": "^/articles/(\\d{4})/(\\d{2})/(.+)$",
      "destination": "/archive?year=$1&month=$2&slug=$3"
    }
  ]
}
```

This converts `/articles/2023/05/hello-world` to `/archive?year=2023&month=05&slug=hello-world`.

You can also use named capture groups:

```json
{
  "rewrites": [
    {
      "source": "^/products/(?<category>[a-z]+)/(?<id>\\d+)$",
      "destination": "/shop?category=$category&item=$id"
    }
  ]
}
```

This converts `/products/shirts/123` to `/shop?category=shirts&item=123`.

## When to use each type

- **Same-application rewrites**: Use when routing within your own application
- **External rewrites**: Use when connecting to external APIs, creating microfrontends, or using Vercel as a reverse proxy or standalone

---
title: "Rolling Releases"
description: "Learn how to use Rolling Releases for more cautious deployments."
last_updated: "2026-01-16T02:19:35.340Z"
source: "https://vercel.com/docs/rolling-releases"
---

# Rolling Releases

Rolling Releases allow you to roll out new deployments to a small fraction of your users before promoting them to everyone.

Once Rolling Releases is enabled, new deployments won't be immediately served to 100% of traffic. Instead, Vercel will direct a configura
your visitors, for example, 5%, to the new deployment. The rest of your traffic will be routed to your previous production deployment.

You can leave your rollout in this state for as long as you want, and Vercel will show you a breakdown of key metrics, such as [Speed Ins
between the canary and current deployment. You can also compare these deployments with other metrics you gather with your own observabili
or when a configurable period of time has passed, you can promote the prospective deployment to 100% of traffic. At any point, you can us
[Instant Rollback](/docs/instant-rollback) to revert from the current release candidate.

## Configuring Rolling Releases

1. From your [dashboard](/dashboard), navigate to your **Project Settings**.
2. Select **Build & Deployment** in the left sidebar.
3. Scroll to the **Rolling Releases** section.

Once you've enabled Rolling Releases, you need to configure two or more stages for your release. Stages are the distinct
traffic ratios you want to serve as your release candidate rolls out. Each stage must send a larger fraction of traffic
to the release candidate. The last stage must always be 100%, representing the full promotion of the
release candidate. Many projects only need two stages, with a single fractional stage before final promotion, but you can
configure more stages as needed.

> **💡 Note:** A stage configured for 0% of traffic is a special case. Vercel will not
> automatically direct any visitors to the release candidate in this case, but
> it can be accessed by forcing a value for the rolling release cookie. See
> [setting the rolling release cookie](#setting-the-rolling-release-cookie) for
> more information.

Once Rolling Releases are configured for the project, any subsequent rollout will use the project's current rolling
release configuration. Each new rollout clones the rolling release configuration. Therefore, editing the configuration
will not impact any rollouts that are currently in progress.

## Managing Rolling Releases

You can manage Rolling releases on the [project's settings page](/docs/project-configuration/project-settings) or via the API or CLI.

### Starting a rolling release

When you enable Rolling Releases in your [project's settings](/docs/project-configuration/project-settings), any action that promotes a d
a new rolling release. This includes:

- Pushing a commit to your git branch, if your project automatically promotes new commits.
- Selecting the **Promote** menu option on a deployment on the **Deployments** page.
- Promoting a deployment [via the CLI](/docs/cli/promote).

The rolling release will proceed to its first stage, sending a portion of traffic to the release candidate.

If a rolling release is in progress when one of the **promote** actions triggers, the project's
state won't change. The active rolling release must be resolved (either completed or aborted) before starting
a new one.

### Observability

While a rolling release is in progress, it will be prominently indicated in several locations:

- The Deployments page has a section summarizing the current rolling release status.
- The release candidate is badged "Canary" in the Deployments list, and indicates the fraction of traffic it is receiving.

Furthermore, the **Observability** tab for your project has a Rolling Releases section. This lets you examine Vercel-gathered
metrics about the actual traffic mix between your deployments and comparative performance differences between them.
You can use these metrics to help you decide whether you want to advance or abort a rolling release.

#### Metrics stored outside of Vercel

You may have observability metrics gathered by platforms other than Vercel. To leverage these metrics to help make
decisions about rolling releases, you will need to ensure that these metrics can distinguish between behaviors
observed on the base deployment and ones on the canary. The easiest way to do this is to propagate Vercel's deployment
ID to your other observability systems.

### Advancing a rolling release

Both the Deployments page and the Rolling Releases Observability tab have controls to change the state of the current release
with a button to advance the release to its next stage. If the next stage is the final stage, the release candidate will be fully
promoted to be your current production deployment, and the project exits the rolling release state.

### Aborting a rolling release

If the metrics on the release candidate are unacceptable to you, there are several ways to abort the rolling release:

- Use the Abort button on the Rolling Releases page.
- Use [Instant Rollback](/docs/instant-rollback) to roll back to any prior deployment, including the base deployment for the current roll

This will leave your project in a rolled-back state, as with Instant Rollback. When you're ready, you can select any deployment to promot
to initiate a new rolling release. The project will exit rollback status once that rolling release completes.

## Understanding Rolling Releases

Rolling Releases should work out-of-the-box for most projects, but the implementation details may be significant for some users.

When a user requests a page from a project's production deployment with an active rolling release, Vercel assigns this user to a random b
in a cookie on the client. We use client-identifying information such as the client's IP address to perform this bucket assignment. This
device to see the same deployment even when in incognito mode. It also ensures that in race conditions such as
multiple simultaneous requests from the same client, all requests resolve to the same target deployment.

Buckets are divided among the two releases at the fraction requested in the current rolling release stage. When the rolling release
advances to a later stage, clients assigned to some buckets will now be assigned to a different deployment, and will receive the new
deployment at that time.

Note that while we attempt to divide user sessions among the two deployments at the configured fraction, not all users behave the same.
If a particularly high-traffic user is placed into one bucket, the observed fraction of total requests between the two deployments may
not match the requested fraction. Likewise, note that randomized assignment based on hashing may not achieve precisely the desired
diversion rate, especially when the number of sessions is small.

### Why Rolling Releases needs Skew Protection

Rolling Releases impact which deployment a user gets when they make a page load. Skew Protection ensures that backend API requests made
from a particular deployment are served by a backend implementation from the same deployment.

When a new user loads a page from a project with an active rolling release, they might receive a page from either deployment. Skew
Protection ensures that, whichever deployment they are served, their backend calls are consistent with the page that they loaded.

If the rolling release stage is advanced, the user may be eligible for a new deployment. On their next page load or refresh, they
will fetch that page from the new deployment. Until they refresh, Skew Protection will continue to ensure that they use backends
consistent with the page they are currently on.

### Setting the Rolling Release cookie

You can modify the Rolling Release cookie on a client by issuing a request that includes a special query parameter.
Requests that include `vcrrForceStable=true` in the URL will always get the base release for the current rolling release.
Likewise, `vcrrForceCanary=true` will force the cookie to target the current canary, including for a rolling release stage
configured for 0% of traffic.

This forced cookie is good only for the duration of a single rolling release. When that rolling release is completed or aborted
and a new rolling release starts, the cookie will get re-processed to a random value.

## Manage rolling releases programmatically with the REST API

The Rolling Releases REST API allows you to programmatically manage rolling release configurations and monitor active releases. Common us

- **CI/CD integration**: Automate rolling release workflows as part of your deployment pipeline
- **Monitoring and observability**: Track the status and progress of active rolling releases
- **Update configuration**: Enable/disable rolling releases, add/remove stages, and more
- **Custom tooling**: Build internal dashboards or tools that interact with rolling release data

For detailed API specifications, request/response schemas, and code examples:

- [API reference](https://vercel.com/docs/rest-api/reference/endpoints/rolling-release)
- [Examples using the SDK](https://vercel.com/docs/rest-api/reference/examples/rolling-releases)


---
title: "Routing Middleware API"
description: "Learn how you can use Routing Middleware, code that executes before a request is processed on a site, to provide speed and
last_updated: "2026-01-16T02:19:35.384Z"
source: "https://vercel.com/docs/routing-middleware/api"
---

# Routing Middleware API

## Routing Middleware file location and name

The Routing Middleware file should be named  and placed at the root of your project, at the same level as your `package.json` file. This

The Routing Middleware must be a default export, with the function being named anything you like. For example, you can name it `router`,

```ts filename="middleware.ts"
export default function middleware() {}
```

## `config` object

Routing Middleware will be invoked for **every route in your project**. If you only want it to be run on specific paths, you can define t

You can also use the [`runtime` option](#config-properties) to [specify which runtime](#specify-runtime) you would like to use. The defau

While the `config` option is the preferred method, **as it does not get invoked on every request**, you can also use conditional statemen

### Match paths based on custom matcher config

To decide which route the Routing Middleware should be run on, you can use a custom matcher config to filter on specific paths. The match

> For \['nextjs']:

#### Match a single path

```ts filename="middleware.ts"
export const config = {
  matcher: '/about/:path*',
};
```

#### Match multiple paths

```ts filename="middleware.ts"
export const config = {
  matcher: ['/about/:path*', '/dashboard/:path*'],
};
```

```

#### Match using regex

The matcher config has full [regex](https://developer.mozilla.org/docs/Web/JavaScript/Guide/Regular_Expressions) support for cases such a

#### Match based on a negative lookahead

To match all request paths except for the ones starting with:

- `api` (API routes)
- `_next/static` (static files)
- `favicon.ico` (favicon file)

```ts filename="middleware.ts"
export const config = {
  matcher: ['/((?!api|_next/static|favicon.ico).*)'],
};
```

#### Match based on character matching

To match `/blog/123` but not `/blog/abc`:

```ts filename="middleware.ts"
export const config = {
  matcher: ['/blog/:slug(\\d{1,})'],
};
```

For help on writing your own regex path matcher, see [Path to regexp](https://github.com/pillarjs/path-to-regexp#path-to-regexp-1).

### Match paths based on conditional statements

```ts filename="middleware.ts"
import { rewrite } from '@vercel/functions';

export default function middleware(request: Request) {
  const url = new URL(request.url);

  if (url.pathname.startsWith('/about')) {
    return rewrite(new URL('/about-2', request.url));
  }

  if (url.pathname.startsWith('/dashboard')) {
    return rewrite(new URL('/dashboard/user', request.url));
  }
}
```

See the [helper methods](#routing-middleware-helper-methods) below for more information on using the `@vercel/functions` package.

### Specify runtime

To change the runtime from the `edge` default, update the `runtime` option as follows:

```ts filename="middleware.ts"
export const config = {
  runtime: 'nodejs', // or 'edge' (default)
};
```

To use the Bun runtime with Routing Middleware, set the [`bunVersion`](/docs/project-configuration#bunversion) property in your `vercel.j

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "bunVersion": "1.x"
}
```

### `config` properties

| Property | Type | Description |
| --- | --- | --- |
| `matcher` | `string / string[]` | A string or array of strings that define the paths the Middleware should be run on |
| `runtime` | `string` (`edge` or `nodejs`) | A string that defines the Middleware runtime and defaults to `edge` |

## Routing Middleware signature

The Routing Middleware signature is made up of two parameters: `request` and `context`. The `request` parameter is an instance of the [Re

| Parameter | Description | Next.js (/app) or (/p |
| --- | --- | --- |
| `request` | An instance of the [Request](/docs/functions/edge-functions/edge-functions-api#request) object | [`Request`](https://d |
| `context` | An extension to the standard [`Request`](https://developer.mozilla.org/docs/Web/API/Request) object | [`NextFetchEvent`](ht |

Routing Middleware comes with built in helpers that are based on the native [`FetchEvent`](https://developer.mozilla.org/docs/Web/API/Fet

[See the section on Routing Middleware helpers for more information](#routing-middleware-helper-methods).

```ts filename="middleware.ts" framework=nextjs-app
// config with custom matcher
export const config = {
  matcher: '/about/:path*',
};

export default function middleware(request: Request) {
  return Response.redirect(new URL('/about-2', request.url));
}
```
```

```

```js filename="middleware.js" framework=nextjs-app
// config with custom matcher
export const config = {
  matcher: '/about/:path*',
};

export default function middleware(request) {
  return Response.redirect(new URL('/about-2', request.url));
}
```

```ts filename="middleware.ts" framework=nextjs
// config with custom matcher
export const config = {
  matcher: '/about/:path*',
};

export default function middleware(request: Request) {
  return Response.redirect(new URL('/about-2', request.url));
}
```

```js filename="middleware.js" framework=nextjs
// config with custom matcher
export const config = {
  matcher: '/about/:path*',
};

export default function middleware(request) {
  return Response.redirect(new URL('/about-2', request.url));
}
```

```ts filename="middleware.ts" framework=other
// config with custom matcher
export const config = {
  matcher: '/about/:path*',
};

export default function middleware(request: Request) {
  return Response.redirect(new URL('/about-2', request.url));
}
```

```js filename="middleware.js" framework=other
// config with custom matcher
export const config = {
  matcher: '/about/:path*',
};

export default function middleware(request) {
  return Response.redirect(new URL('/about-2', request.url));
}
```

> **💡 Note:** If you're not using a framework, you must either add
> &amp;#x20;to your
> &amp;#x20;or change your JavaScript Functions'
> file extensions from  to
>

### Request

The `Request` object represents an HTTP request. It is a wrapper around the [Fetch API](https://developer.mozilla.org/docs/Web/API/Fetch_

#### Request properties

| Property | Type | Description |
| --- | --- | --- |
| `url` | `string` | The URL of the request |
| `method` | `string` | The HTTP method of the request |
| `headers` | `Headers` | The headers of the request |
| `body` | [`ReadableStream`](https://developer.mozilla.org/docs/Web/API/ReadableStream) | The body of the request |
| `bodyUsed` | `boolean` | Whether the body has been read |
| `cache` | `string` | The cache mode of the request |
| `credentials` | `string` | The credentials mode of the request |
| `destination` | `string` | The destination of the request |
| `integrity` | `string` | The integrity of the request |
| `redirect` | `string` | The redirect mode of the request |
| `referrer` | `string` | The referrer of the request |
| `referrerPolicy` | `string` | The referrer policy of the request |
| `mode` | `string` | The mode of the request |
| `signal` | [`AbortSignal`](https://developer.mozilla.org/docs/Web/API/AbortSignal) | The signal of the request |
| `arrayBuffer` | `function` | Returns a promise that resolves with |
| `blob` | `function` | Returns a promise that resolves with |
| `formData` | `function` | Returns a promise that resolves with |
| `json` | `function` | Returns a promise that resolves with |
| `text` | `function` | Returns a promise that resolves with |
| `clone` | `function` | Returns a clone of the request |

> For \["nextjs", "nextjs-app"]:

To learn more about the [`NextRequest`](https://nextjs.org/docs/api-reference/next/server#nextrequest) object and its properties, visit t

### `waitUntil()`

The `waitUntil()` method is from the [`ExtendableEvent`](https://developer.mozilla.org/docs/Web/API/ExtendableEvent/waitUntil) interface.
```

It can be used to keep the function running after a response has been sent. This is useful when you have an async task that you want to k

The example below will:

- Send a response immediately
- Keep the function running for ten seconds
- Fetch a product and log it to the console

> For \["other"]:

```ts filename="middleware.ts" framework=nextjs
import type { NextFetchEvent } from 'next/server';

export const config = {
  matcher: '/',
};

const wait = (ms: number) => new Promise((resolve) => setTimeout(resolve, ms));

async function getProduct() {
  const res = await fetch('https://api.vercel.app/products/1');
  await wait(10000);
  return res.json();
}

export default function middleware(request: Request, context: NextFetchEvent) {
  context.waitUntil(getProduct().then((json) => console.log({ json })));

  return new Response(JSON.stringify({ hello: 'world' }), {
    status: 200,
    headers: { 'content-type': 'application/json' },
  });
}
```

```js filename="middleware.js" framework=nextjs
export const config = {
  matcher: '/',
};

const wait = (ms) => new Promise((resolve) => setTimeout(resolve, ms));

async function getProduct() {
  const res = await fetch('https://api.vercel.app/products/1');
  await wait(10000);
  return res.json();
}

export default function middleware(request, context) {
  context.waitUntil(getProduct().then((json) => console.log({ json })));

  return new Response(JSON.stringify({ hello: 'world' }), {
    status: 200,
    headers: { 'content-type': 'application/json' },
  });
}
```

```ts filename="middleware.ts" framework=nextjs-app
import type { NextFetchEvent } from 'next/server';

export const config = {
  matcher: '/',
};

const wait = (ms: number) => new Promise((resolve) => setTimeout(resolve, ms));

async function getProduct() {
  const res = await fetch('https://api.vercel.app/products/1');
  await wait(10000);
  return res.json();
}

export default function middleware(request: Request, context: NextFetchEvent) {
  context.waitUntil(getProduct().then((json) => console.log({ json })));

  return new Response(JSON.stringify({ hello: 'world' }), {
    status: 200,
    headers: { 'content-type': 'application/json' },
  });
}
```

```js filename="middleware.js" framework=nextjs-app
import { NextResponse } from 'next/server';

export const config = {
  matcher: '/',
};

const wait = (ms) => new Promise((resolve) => setTimeout(resolve, ms));

async function getAlbum() {
  const res = await fetch('https://jsonplaceholder.typicode.com/albums/1');
  await wait(10000);
  return res.json();
}

export default function middleware(request, context) {
  context.waitUntil(getAlbum().then((json) => console.log({ json })));
```

```
  return new NextResponse(JSON.stringify({ hello: 'world' }), {
    status: 200,
    headers: { 'content-type': 'application/json' },
  });
}
```

```ts filename="middleware.ts" framework=other
import type { RequestContext } from '@vercel/functions';

export const config = {
  matcher: '/',
};

const wait = (ms: number) => new Promise((resolve) => setTimeout(resolve, ms));

async function getProduct() {
  const res = await fetch('https://api.vercel.app/products/1');
  await wait(10000);
  return res.json();
}

export default function middleware(request: Request, context: RequestContext) {
  context.waitUntil(getProduct().then((json) => console.log({ json })));

  return Response.json(
    { hello: 'world' },
    {
      status: 200,
      headers: { 'content-type': 'application/json' },
    },
  );
}
```

```js filename="middleware.js" framework=other
export const config = {
  matcher: '/',
};

const wait = (ms) => new Promise((resolve) => setTimeout(resolve, ms));

async function getProduct() {
  const res = await fetch('https://api.vercel.app/products/1');
  await wait(10000);
  return res.json();
}

export default function middleware(request, context) {
  context.waitUntil(getProduct().then((json) => console.log({ json })));

  return Response.json(
    { hello: 'world' },
    {
      status: 200,
      headers: { 'content-type': 'application/json' },
    },
  );
}
```

> **💡 Note:** If you're not using a framework, you must either add
>  to your
>  or change your JavaScript Functions'
> file extensions from  to
>

#### Context properties

| Property | Type | Description
| --- | --- | ---
| [`waitUntil`](https://developer.mozilla.org/docs/Web/API/ExtendableEvent/waitUntil) | `(promise: Promise<unknown>): void` | Prolongs th

## Routing Middleware helper methods

You can use Vercel-specific helper methods to access a request's [geolocation](#geolocation), [IP Address](/docs/functions/functions-api-

> For \['nextjs', 'nextjs-app']:

You can access these helper methods with the `request` and `response` objects in your middleware handler method.

> **💡 Note:** These helpers are exclusive to Vercel, and will not work on other providers,
> even if your app is built with Next.js.

> For \['other']:

Add the `@vercel/functions` package to your project with:

### Geolocation

> For \['nextjs', 'nextjs-app']:

The `geo` helper object returns geolocation information for the incoming request. It has the following properties:

> For \['other']:

The `geolocation()` helper returns geolocation information for the incoming request. It has the following properties:

| Property | Description |

| ---------- | ---------------------------------------------------- |
| `city`     | The city that the request originated from            |
| `country`  | The country that the request originated from         |
| `latitude` | The latitude of the client                           |
| `longitude` | The longitude of the client                          |
| `region`   | The [CDN region](/docs/regions) that received the request |

Each property returns a `string`, or `undefined`.

```ts filename="middleware.ts" framework=nextjs-app
import { NextResponse } from 'next/server';
import type { NextRequest } from 'next/server';

// The country to block from accessing the secret page
const BLOCKED_COUNTRY = 'SE';

// Trigger this middleware to run on the `/secret-page` route
export const config = {
  matcher: '/secret-page',
};

export default function middleware(request: NextRequest) {
  const country = request.geo?.country ?? 'US';

  console.log(`Visitor from ${country}`);

  const url = request.nextUrl.clone();
  url.pathname = country === BLOCKED_COUNTRY ? '/login' : '/secret-page';

  return NextResponse.rewrite(url);
}
```

```js filename="middleware.js" framework=nextjs-app
import { NextResponse } from 'next/server';
// The country to block from accessing the secret page
const BLOCKED_COUNTRY = 'SE';

// Trigger this middleware to run on the `/secret-page` route
export const config = {
  matcher: '/secret-page',
};

export default function middleware(request) {
  const country = request.geo?.country ?? 'US';

  console.log(`Visitor from ${country}`);

  const url = request.nextUrl.clone();
  url.pathname = country === BLOCKED_COUNTRY ? '/login' : '/secret-page';

  return NextResponse.rewrite(url);
}
```

```ts filename="middleware.ts" framework=nextjs
import { NextResponse } from 'next/server';
import type { NextRequest } from 'next/server';

// The country to block from accessing the secret page
const BLOCKED_COUNTRY = 'SE';

// Trigger this middleware to run on the `/secret-page` route
export const config = {
  matcher: '/secret-page',
};

export default function middleware(request: NextRequest) {
  const country = request.geo?.country ?? 'US';

  console.log(`Visitor from ${country}`);

  const url = request.nextUrl.clone();
  url.pathname = country === BLOCKED_COUNTRY ? '/login' : '/secret-page';

  return NextResponse.rewrite(url);
}
```

```js filename="middleware.js" framework=nextjs
import { NextResponse } from 'next/server';
// The country to block from accessing the secret page
const BLOCKED_COUNTRY = 'SE';

// Trigger this middleware to run on the `/secret-page` route
export const config = {
  matcher: '/secret-page',
};

export default function middleware(request) {
  const country = request.geo?.country ?? 'US';

  console.log(`Visitor from ${country}`);

  const url = request.nextUrl.clone();
  url.pathname = country === BLOCKED_COUNTRY ? '/login' : '/secret-page';

  return NextResponse.rewrite(url);
}
```

```ts filename="middleware.ts" framework=other
import { geolocation } from '@vercel/functions';

const BLOCKED_COUNTRY = 'US';

export const config = {
  // Only run the middleware on the home route
  matcher: '/',
};

export default function middleware(request: Request) {
  const url = new URL(request.url);
  const { country } = geolocation(request);
  // You can also get the country using dot notation on the function
  // const country = geolocation(request).country;
  if (country === BLOCKED_COUNTRY) {
    url.pathname = '/blocked.html';
  } else {
    url.pathname = '/index.html';
  }
  // Return a new redirect response
  return Response.redirect(url);
}
```

```js filename="middleware.js" framework=other
import { geolocation } from '@vercel/functions';

const BLOCKED_COUNTRY = 'US';

export const config = {
  // Only run the middleware on the home route
  matcher: '/',
};

export default function middleware(request) {
  const url = new URL(request.url);
  const { country } = geolocation(request);
  // You can also get the country using dot notation on the function
  // const country = geolocation(request).country;
  if (country === BLOCKED_COUNTRY) {
    url.pathname = '/blocked.html';
  } else {
    url.pathname = '/index.html';
  }
  // Return a new redirect response
  return Response.redirect(url);
}
```

### IP Address

> For \['nextjs', 'nextjs-app']:

The `ip` object returns the IP address of the request from the headers, or `undefined`.

> For \['other']:

The `ipAddress()` helper returns the IP address of the request from the headers, or `undefined`.

```ts filename="middleware.ts" framework=all
import { ipAddress } from '@vercel/functions';
import { next } from '@vercel/functions';

export default function middleware(request: Request) {
  const ip = ipAddress(request);
  return next({
    headers: { 'x-your-ip-address': ip || 'unknown' },
  });
}
```

```js filename="middleware.js" framework=all
import { ipAddress } from '@vercel/functions';
import { next } from '@vercel/functions';

export default function middleware(request) {
  const ip = ipAddress(request);
  return next({
    headers: { 'x-your-ip-address': ip || 'unknown' },
  });
}
```

### `RequestContext`

The `RequestContext` is an extension of the standard `Request` object, which contains the [`waitUntil`](#waitUntil) function. The followi

```ts filename="middleware.ts" framework=all
import type { RequestContext } from '@vercel/functions';

export default function handler(request: Request, context: RequestContext) {
  context.waitUntil(getAlbum().then((json) => console.log({ json })));

  return new Response(
    `Hello there, from ${request.url} I'm an Vercel Function!`,
  );
}
```

```
export const config = {
  matcher: '/',
};

const wait = (ms: number) => new Promise((resolve) => setTimeout(resolve, ms));

async function getAlbum() {
  const res = await fetch('https://jsonplaceholder.typicode.com/albums/1');
  await wait(10000);
  return res.json();
}
```

```js filename="middleware.js" framework=all
export default function handler(request, context) {
  context.waitUntil(getAlbum().then((json) => console.log({ json })));

  return new Response(
    `Hello there, from ${request.url} I'm an Vercel Function!`,
  );
}

export const config = {
  matcher: '/',
};

const wait = (number) => new Promise((resolve) => setTimeout(resolve, ms));

async function getAlbum() {
  const res = await fetch('https://jsonplaceholder.typicode.com/albums/1');
  await wait(10000);
  return res.json();
}
```

### Rewrites

> For \['nextjs', 'nextjs-app']:

The `NextResponse.rewrite()` helper returns a response that rewrites the request to a different URL.

> For \['other']:

The `rewrite()` helper returns a response that rewrites the request to a different URL.

```ts filename="middleware.ts" framework=nextjs-app
import { NextResponse } from 'next/server';
import type { NextRequest } from 'next/server';
// Trigger this middleware to run on the `/about` route
export const config = {
  matcher: '/about',
};

export default function middleware(request: NextRequest) {
  // Rewrite to URL
  return NextResponse.rewrite('/about-2');
}
```

```js filename="middleware.js" framework=nextjs-app
import { NextResponse } from 'next/server';
// Trigger this middleware to run on the `/about` route
export const config = {
  matcher: '/about',
};

export default function middleware(request) {
  // Rewrite to URL
  return NextResponse.rewrite('/about-2');
}
```

```ts filename="middleware.ts" framework=nextjs
import { NextResponse } from 'next/server';
import type { NextRequest } from 'next/server';
// Trigger this middleware to run on the `/about` route
export const config = {
  matcher: '/about',
};

export default function middleware(request: NextRequest) {
  // Rewrite to URL
  return NextResponse.rewrite('/about-2');
}
```

```js filename="middleware.js" framework=nextjs
import { NextResponse } from 'next/server';
// Trigger this middleware to run on the `/about` route
export const config = {
  matcher: '/about',
};

export default function middleware(request) {
  // Rewrite to URL
  return NextResponse.rewrite('/about-2');
}
```

```ts filename="middleware.ts" framework=other
```

```ts
import { rewrite } from '@vercel/functions';
// Trigger this middleware to run on the `/about` route
export const config = {
  matcher: '/about',
};

export default function middleware(request: Request) {
  return rewrite(new URL('/about-2', request.url));
}
```

```js filename="middleware.js" framework=other
import { rewrite } from '@vercel/functions';
// Trigger this middleware to run on the `/about` route
export const config = {
  matcher: '/about',
};

export default function middleware(request) {
  return rewrite(new URL('/about-2', request.url));
}
```

### Continuing the Routing Middleware chain

> For \['nextjs', 'nextjs-app']:

The `NextResponse.next()` helper returns a Response that instructs the function to continue the middleware chain. It takes the following

> For \['other']:

The `next()` helper returns a Response that instructs the function to continue the middleware chain. It takes the following optional para

| Parameter    | type                      | Description                |
| ------------ | ------------------------- | -------------------------- |
| `headers`    | `Headers[]` or `Headers` | The headers you want to set |
| `status`     | `number`                  | The status code            |
| `statusText` | `string`                  | The status text            |

The following example adds a custom header, then continues the Routing Middleware chain:

```ts filename="middleware.ts" framework=nextjs
import { NextResponse } from 'next/server';
import type { NextRequest } from 'next/server';

export function middleware(request: NextRequest) {
  // Clone the request headers and set a new header `x-hello-from-middleware1`
  const requestHeaders = new Headers(request.headers);
  requestHeaders.set('x-hello-from-middleware1', 'hello');

  // You can also set request headers in NextResponse.next
  const response = NextResponse.next({
    request: {
      // New request headers
      headers: requestHeaders,
    },
  });

  // Set a new response header `x-hello-from-middleware2`
  response.headers.set('x-hello-from-middleware2', 'hello');
  return response;
}
```

```js filename="middleware.js" framework=nextjs
import { NextResponse } from 'next/server'
import type { NextRequest } from 'next/server'

export function middleware(request: NextRequest) {
  // Clone the request headers and set a new header `x-hello-from-middleware1`
  const requestHeaders = new Headers(request.headers)
  requestHeaders.set('x-hello-from-middleware1', 'hello')

  // You can also set request headers in NextResponse.next
  const response = NextResponse.next({
    request: {
      // New request headers
      headers: requestHeaders,
    },
  })

  // Set a new response header `x-hello-from-middleware2`
  response.headers.set('x-hello-from-middleware2', 'hello')
  return response
}
```

```ts filename="middleware.ts" framework=nextjs-app
import { NextResponse } from 'next/server';
import type { NextRequest } from 'next/server';

export function middleware(request: NextRequest) {
  // Clone the request headers and set a new header `x-hello-from-middleware1`
  const requestHeaders = new Headers(request.headers);
  requestHeaders.set('x-hello-from-middleware1', 'hello');

  // You can also set request headers in NextResponse.next
  const response = NextResponse.next({
    request: {
      // New request headers
```

```
      headers: requestHeaders,
    },
  });

  // Set a new response header `x-hello-from-middleware2`
  response.headers.set('x-hello-from-middleware2', 'hello');
  return response;
}
```

```js filename="middleware.js" framework=nextjs-app
import { NextResponse } from 'next/server';

export function middleware(request) {
  // Clone the request headers and set a new header `x-hello-from-middleware1`
  const requestHeaders = new Headers(request.headers);
  requestHeaders.set('x-hello-from-middleware1', 'hello');

  // You can also set request headers in NextResponse.next
  const response = NextResponse.next({
    request: {
      // New request headers
      headers: requestHeaders,
    },
  });

  // Set a new response header `x-hello-from-middleware2`
  response.headers.set('x-hello-from-middleware2', 'hello');
  return response;
}
```

> For \['other']:

```js filename="middleware.js" framework=other
import { next } from '@vercel/functions';

export default function middleware(request) {
  // Clone the request headers
  const requestHeaders = new Headers(request.headers);

  // Set a new header `x-hello-from-middleware1`
  requestHeaders.set('x-hello-from-middleware1', 'hello');

  // Use the `next()` function to forward the request with modified headers
  return next({
    request: {
      headers: requestHeaders,
    },
    headers: {
      'x-hello-from-middleware2': 'hello',
    },
  });
}
```

```ts filename="middleware.ts" framework=other
import { next } from '@vercel/functions';

export default function middleware(request: Request) {
  // Clone the request headers
  const requestHeaders = new Headers(request.headers);

  // Set a new header `x-hello-from-middleware1`
  requestHeaders.set('x-hello-from-middleware1', 'hello');

  // Use the `next()` function to forward the request with modified headers
  return next({
    request: {
      headers: requestHeaders,
    },
    headers: {
      'x-hello-from-middleware2': 'hello',
    },
  });
}
```

#### `next()` no-op example

This no-op example will return a `200 OK` response with no further action:

```ts filename="middleware.ts" framework=nextjs
import { NextResponse } from 'next/server';
export default function middleware() {
  return NextResponse.next();
}
```

```js filename="middleware.js" framework=nextjs
import { NextResponse } from 'next/server';
export default function middleware() {
  return NextResponse.next();
}
```

```ts filename="middleware.ts" framework=nextjs-app
import { NextResponse } from 'next/server';
export default function middleware() {
```

  return NextResponse.next();
}
```

```js filename="middleware.js" framework=nextjs-app
import { NextResponse } from 'next/server';
export default function middleware() {
  return NextResponse.next();
}
```

```ts filename="middleware.ts" framework=other
import { next } from '@vercel/functions';
export default function middleware() {
  return next();
}
```

```js filename="middleware.js" framework=other
import { next } from '@vercel/functions';
export default function middleware() {
  return next();
}
```

## More resources

- [Redirect with unique tokens](/kb/guide/use-crypto-web-api)

---
title: "Getting Started with Routing Middleware"
description: "Learn how you can use Routing Middleware, code that executes before a request is processed on a site, to provide speed and
last_updated: "2026-01-16T02:19:35.239Z"
source: "https://vercel.com/docs/routing-middleware/getting-started"
---

# Getting Started with Routing Middleware

Routing Middleware lets you to run code before your pages load, giving you control over incoming requests. It runs close to your users fo

Routing Middleware is available on the [Node.js](/docs/functions/runtimes/node-js), [Bun](/docs/functions/runtimes/bun), and [Edge](/docs

## What you will learn

- Create your first Routing Middleware
- Redirect users based on URLs
- Add conditional logic to handle different scenarios
- Configure which paths your Routing Middleware runs on

## Prerequisites

- A Vercel project
- Basic knowledge of JavaScript/TypeScript

## Creating a Routing Middleware

The following steps will guide you through creating your first Routing Middleware.

- ### Create a new file for your Routing Middleware
  Create a file called `middleware.ts` in your project root (same level as your `package.json`) and add the following code:
  ```ts v0="build" filename="middleware.ts"
  export const config = {
    runtime: 'nodejs', // optional: use 'nodejs' or omit for 'edge' (default)
  };

  export default function middleware(request: Request) {
    console.log('Request to:', request.url);
    return new Response('Logging request URL from Middleware');
  }
  ```
  - Every request to your site will trigger this function
  - You log the request URL to see what's being accessed
  - You return a response to prove the middleware is running
  - The `runtime` config is optional and defaults to `edge`. To use Bun, set [`bunVersion`](/docs/project-configuration#bunversion) in `v
  Deploy your project and visit any page. You should see "Logging request URL from Middleware" instead of your normal page content.

- ### Redirecting users
  To redirect users based on their URL, add a new route to your project called `/blog`, and modify your `middleware.ts` to include a redi
  ```ts v0="build" filename="middleware.ts"
  export const config = {
    runtime: 'nodejs', // optional: use 'nodejs' or omit for 'edge' (default)
  };

  export default function middleware(request: Request) {
    const url = new URL(request.url);

    // Redirect old blog path to new one
    if (url.pathname === '/old-blog') {
      return new Response(null, {
        status: 302,
        headers: { Location: '/blog' },
      });
    }

    // Let other requests continue normally
    return new Response('Other pages work normally');
  }
  ```
  - You use `new URL(request.url)` to parse the incoming URL
  ```

  - You check if the path matches `/old-blog`
  - If it does, you return a redirect response (status 302)
  - The `Location` header tells the browser where to go
  Try visiting `/old-blog` - you should be redirected to `/blog`.

- ### Configure which paths trigger the middleware
  By default, Routing Middleware runs on every request. To limit it to specific paths, you can use the [`config`](/docs/routing-middlewar
  ```ts v0="build" filename="middleware.ts"
  export default function middleware(request: Request) {
    const url = new URL(request.url);

    // Only handle specific redirects
    if (url.pathname === '/old-blog') {
      return new Response(null, {
        status: 302,
        headers: { Location: '/blog' },
      });
    }

    return new Response('Middleware processed this request');
  }

  // Configure which paths trigger the Middleware
  export const config = {
    matcher: [
      // Run on all paths except static files
      '/((?!_next/static|_next/image|favicon.ico).*)',
      // Or be more specific:
      // '/blog/:path*',
      // '/api/:path*'
    ],
  };
  ```
  - The [`matcher`](/docs/routing-middleware/api#match-paths-based-on-custom-matcher-config) array defines which paths trigger your Routi
  - The regex excludes static files (images, CSS, etc.) for better performance
  - You can also use simple patterns like `/blog/:path*` for specific sections
  See the [API Reference](/docs/routing-middleware/api) for more details on the `config` object and matcher patterns.

- ### Debugging Routing Middleware
  When things don't work as expected:
  1. **Check the logs**: Use `console.log()` liberally and check your [Vercel dashboard](/dashboard) **Logs** tab
  2. **Test the matcher**: Make sure your paths are actually triggering the Routing Middleware
  3. **Verify headers**: Log `request.headers` to see what's available
  4. **Test locally**: Routing Middleware works in development too so you can debug before deploying
  ```ts filename="middleware.ts"
  export default function middleware(request: Request) {
    // Debug logging
    console.log('URL:', request.url);
    console.log('Method:', request.method);
    console.log('Headers:', Object.fromEntries(request.headers.entries()));

    // Your middleware logic here...
  }
  ```

## More resources

Learn more about Routing Middleware by exploring the following resources:

- [Routing Middleware](/docs/routing-middleware)
- [Routing Middleware API Reference](/docs/routing-middleware/api)


---
title: "Routing Middleware"
description: "Learn how you can use Routing Middleware, code that executes before a request is processed on a site, to provide speed and
last_updated: "2026-01-16T02:19:35.258Z"
source: "https://vercel.com/docs/routing-middleware"
---

# Routing Middleware

Routing Middleware **executes code *before* a request is processed on a site**, and are built on top of [fluid compute](/docs/fluid-compu

Because it runs globally before the cache, Routing Middleware is an effective way of providing personalization to statically generated co

The default runtime for Routing Middlewares is [Edge](/docs/functions/runtimes/edge). See [runtime options](#runtime-options) for informa

## Creating a Routing Middleware

You can use Routing Middleware with [**any framework**](/docs/frameworks). To add a Routing Middleware to your app, you need to create a

```ts v0="build" filename="middleware.ts" framework=all
export default function middleware(request: Request) {
  const url = new URL(request.url);

  // Redirect old paths
  if (url.pathname === '/old-page') {
    return new Response(null, {
      status: 302,
      headers: { Location: '/new-page' },
    });
  }

  // Continue to next handler
  return new Response('Hello from your Middleware!');
}
```

```js v0="build" filename="middleware.js" framework=all
```

```
export default function middleware(request) {
  const url = new URL(request.url);

  // Redirect old paths
  if (url.pathname === '/old-page') {
    return new Response(null, {
      status: 302,
      headers: { Location: '/new-page' },
    });
  }

  // Continue to next handler
  return new Response('Hello from your Middleware!');
}
```

> For \['nextjs', 'nextjs-app']:

## Logging

Routing Middleware has full support for the [`console`](https://developer.mozilla.org/docs/Web/API/Console) API, including `time`, `debug

## Using a database with Routing Middleware

If your Routing Middleware depends on a database far away from one of [our supported regions](/docs/regions), the overall latency of API

## Limits on requests

The following limits apply to requests processed by Routing Middleware:

| Name | Limit |
| --- | --- |
| Maximum URL length | 14 KB |
| Maximum request body length | 4 MB |
| Maximum number of request headers | 64 |
| Maximum request headers length | 16 KB |

## Runtime options

Routing Middleware is available on the [Node.js](/docs/functions/runtimes/node-js), [Bun](/docs/functions/runtimes/bun), and [Edge](/docs

To use the Bun runtime, set [`bunVersion`](/docs/project-configuration#bunversion) in your `vercel.json` file and your runtime config to

```ts filename="middleware.ts" framework=nextjs-app
export const config = {
  runtime: 'nodejs', // or 'edge' (default)
};
export default function middleware(request: Request) {
  // Your middleware logic here
  return new Response('Hello from your Middleware!');
}
```

```js filename="middleware.js" framework=nextjs-app
export const config = {
  runtime: 'nodejs' // or 'edge' (default)
}
export default function middleware(request: Request) {
  // Your middleware logic here
  return new Response('Hello from your Middleware!');
}
```

```ts filename="middleware.ts" framework=nextjs
export const config = {
  runtime: 'nodejs', // or 'edge' (default)
};
export default function middleware(request: Request) {
  // Your middleware logic here
  return new Response('Hello from your Middleware!');
}
```

```js filename="middleware.js" framework=nextjs
export const config = {
  runtime: 'nodejs', // or 'edge' (default)
};
export default function middleware(request) {
  // Your middleware logic here
  return new Response('Hello from your Middleware!');
}
```

```ts filename="middleware.ts" framework=other
export const config = {
  runtime: 'nodejs', // or 'edge' (default)
};
export default function middleware(request: Request) {
  // Your middleware logic here
  return new Response('Hello from your Middleware!');
}
```

```js filename="middleware.js" framework=other
export const config = {
  runtime: 'nodejs' // or 'edge' (default)
}
export default function middleware(request: Request) {
  // Your middleware logic here
  return new Response('Hello from your Middleware!');
```

}
```

## Pricing

Routing Middleware is priced using the [fluid compute](/docs/fluid-compute) model, which means you are charged by the amount of compute r

## Observability

The [Vercel Observability dashboard](/docs/observability) provides visibility into your routing middleware usage, including invocation co

- Analyze invocations by request path
- Break down actions by type, such as redirects or rewrites
- View rewrite targets and frequency
- Use the query builder for custom insights

## More resources

Learn more about Routing Middleware by exploring the following resources:

- [Getting Started with Routing Middleware](/docs/routing-middleware/getting-started)
- [Routing Middleware API Reference](/docs/routing-middleware/api)
- [Fluid compute](/docs/fluid-compute)
- [Runtimes](/docs/functions/runtimes)

---
title: "SAML Single Sign-On"
description: "Learn how to configure SAML SSO for your organization on Vercel."
last_updated: "2026-01-16T02:19:35.279Z"
source: "https://vercel.com/docs/saml"
---

# SAML Single Sign-On

To manage the [members](/docs/rbac/managing-team-members) of your team through a third-party identity provider like [Okta](https://www.ok

Once enabled, all team members will be able to log in or access [Preview](/docs/deployments/preview-deployments) and Production Deploymen

For Enterprise customers, you can also automatically manage team member roles and provisioning by setting up [Directory Sync](/docs/direc

## Configuring SAML SSO

1. To configure SAML SSO for your team, you must be an [owner](/docs/rbac/access-roles/team-level-roles) of the team
2. From your [dashboard](/dashboard), ensure your team is selected in the scope selector
3. Navigate to the **Settings** tab and select **Security & Privacy**
4. Navigate to the **SAML Single Sign-On** section. Click **Configure** and follow the walkthrough to configure SAML SSO for your team wi
5. As a further step, you may want to [enforce SAML SSO](#enforcing-saml) for your team

> **💡 Note:** Pro teams will first need to purchase the SAML SSO add-on from their [Billing settings](https://vercel.com/d?to=%2F%5Bteam

## Enforcing SAML

For additional security, SAML SSO can be enforced for a team so that all [team members](/docs/rbac/managing-team-members) **cannot access

1. To enforce SAML SSO for your team, you must be an [owner](/docs/rbac/access-roles/team-level-roles) and currently be authenticated wit
2. From your [dashboard](/dashboard), navigate to the **Settings** tab and select **Security & Privacy**. Then go to the **SAML Single Si
3. Toggle the **Require Team Members to login with SAML** switch to **Enabled**

> **💡 Note:** When modifying your SAML configuration, the option for enforcing will
> automatically be turned off. Please verify your new configuration is working
> correctly by re-authenticating with SAML SSO before re-enabling the option.

## Authenticating with SAML SSO

Once you have configured SAML, your [team members](/docs/rbac/managing-team-members) can use SAML SSO to log in or sign up to Vercel. To

1. Select the **Continue with SAML SSO** button on the authentication page, then enter your team's URL.
   Your team slug is the identifier in the URLs for your team. For example, the identifier for vercel.com/acme is `acme`.
2. Select **Continue with SAML SSO** again to be redirected to the third-party authentication provider to finish authenticating. Once com

SAML SSO sessions last for 24 hours before users must re-authenticate with the third-party SAML provider.

### Customizing the login page

You can choose to share a Vercel login page that only shows the option to log in with SAML SSO. This prevents your team members from logg

To use this page, you can set the `saml` query param to your team URL. For example:

```text
https://vercel.com/login?saml=team_id
```

## Managing team members

When using SAML SSO, team members can authenticate through your identity provider, but team membership must be managed manually through t

For automatic provisioning and de-provisioning of team members based on your identity provider, consider upgrading to [Directory Sync](/d

## SAML providers

Vercel supports the following third-party SAML providers:

- [Okta](https://www.okta.com/)
- [Auth0](https://auth0.com/)
- [Google](https://accounts.google.com/)
- [Microsoft Entra (formerly Azure Active Directory)](https://www.microsoft.com/en-in/security/business/identity-access/microsoft-entra-s
- [Microsoft ADFS](https://docs.microsoft.com/en-us/windows-server/identity/active-directory-federation-services)
- [OneLogin](https://onelogin.com/)
- [Duo](https://duo.com/product/single-sign-on-sso/)
```

- [JumpCloud](https://jumpcloud.com/)
- [PingFederate](https://www.pingidentity.com/en/platform/capabilities/single-sign-on.html)
- [ADP](https://apps.adp.com/en-US/home)
- [Keycloak](https://www.keycloak.org/)
- [Cyberark](https://www.cyberark.com/products/single-sign-on/)
- [OpenID](https://openid.net/)
- [VMware](https://kb.vmware.com/s/article/2034918)
- [LastPass](https://www.lastpass.com/)
- [miniOrange](https://www.miniorange.com/products/single-sign-on-sso)
- [NetIQ](https://www.microfocus.com/en-us/cyberres/identity-access-management/secure-login)
- [Oracle Cloud](https://docs.oracle.com/en/cloud/paas/content-cloud/administer/enable-single-sign-sso.html)
- [Salesforce](https://help.salesforce.com/s/articleView?id=sf.sso_about.htm\&type=5)
- [CAS](https://www.apereo.org/projects/cas)
- [ClassLink](https://www.classlink.com/)
- [Cloudflare](https://developers.cloudflare.com/cloudflare-one/applications/configure-apps/dash-sso-apps/)
- [SimpleSAMLphp](https://simplesamlphp.org/)

---
title: "Access Control"
description: "Learn about the protection and compliance measures Vercel takes to ensure the security of your data, including DDoS mitigat
last_updated: "2026-01-16T02:19:35.388Z"
source: "https://vercel.com/docs/security/access-control"
---

# Access Control

Deployments can be protected with [Password protection](/docs/security/deployment-protection/methods-to-protect-deployments/password-prot

## Password protection

Password protection applies to Preview deployments and Production deployments. This feature can be enabled through the Teams Project dash

## Vercel Authentication

Vercel Authentication protection applies to Preview deployments and Production deployments. When enabled, a person with a Personal Accoun

Both Password protection, and Vercel Authentication can be enabled at the same time. When this is the case, the person trying to access t

[Read more about in the documentation here](/docs/security/deployment-protection/methods-to-protect-deployments/vercel-authentication).

---
title: "Security & Compliance Measures"
description: "Learn about the protection and compliance measures Vercel takes to ensure the security of your data, including DDoS mitigat
last_updated: "2026-01-16T02:19:35.404Z"
source: "https://vercel.com/docs/security/compliance"
---

# Security & Compliance Measures

This page covers the protection and compliance measures Vercel takes to ensure the security of your data, including [DDoS mitigation](/do

To understand how security responsibilities are divided between you (the customer) and Vercel, see the [shared responsibility model](/doc

## Compliance

### SOC 2 Type 2

System and Organization Control 2 Type 2 ([SOC 2](https://www.aicpa-cima.com/topic/audit-assurance/audit-and-assurance-greater-than-soc-2

**Vercel has a SOC 2 Type 2 attestation for Security, Confidentiality, and Availability**.

More information is available at [security.vercel.com](https://security.vercel.com/).

### ISO 27001:2022

ISO 27001 is an internationally recognized standard, developed by the International Organization for Standardization (ISO) and Internatio

**Vercel is ISO 27001:2022 certified**. Our certificate is available [here](https://www.schellman.com/certificate-directory?certificateNu

### GDPR

The EU General Data Protection Regulation (GDPR), is a comprehensive data protection law that governs the use, sharing, transfer, and pro

Vercel supports GDPR compliance, which means that we commit to the following:

- Implement and maintain appropriate technical and organizational security measures surrounding customer data
- Notify our customers without undue delay of any data breaches
- Impose similar data protection obligations on our sub-processors as we do for ourselves
- Respond to applicable [data subjects rights](/legal/privacy-policy#eea), including requests for access, correction, and/or deletion of
- Rely on the EU Standard Contractual Clauses and the UK Addendum as valid data transfer mechanisms when transferring personal data outsi

For more information on how Vercel protects your personal data, and the data of your customers, refer to our [Privacy Policy](/legal/priv

### PCI DSS

Payment Card Industry Data Security Standard (PCI DSS) is a standard that defines the security and privacy requirements for payment card

In alignment with Vercel's [shared responsibility model](/docs/security/shared-responsibility), Vercel serves as a service provider to cu

[Learn about PCI DSS iframe integration](/docs/security/pci-dss).

Vercel provides both a Self-Assessment Questionnaire D (SAQ-D) Attestation of Compliance (AOC) for service providers and a Self-Assessmen

PCI DSS compliance is a shared responsibility between Vercel and its customers. To help customers better understand their responsibilitie

A copy of our PCI DSS compliance documentation can be obtained through our [Trust Center](https://security.vercel.com).

[Contact us](https://vercel.com/contact/sales/security) for more details about our SAQ-D and SAQ-A AOC reports or Responsibility Matrix.

### HIPAA

Certain businesses, covered entities, and business associates, are required to comply with these regulations to ensure that health data i
The [Health Information Portability and Accountability Act](https://www.hhs.gov/hipaa/) (HIPAA) is one of the most important sectoral reg

Vercel supports HIPAA compliance as a **business associate** by committing to the following:

- Implementing and maintaining appropriate technical and organizational security measures designed to safeguard a customer's [Protected H
- Notifying customers of any data breaches without undue delay
- Signing Business Associate Agreements (BAAs) with enterprise customers

#### Additional protection

Customers subject to HIPAA may enable [Vercel Secure Compute (available on Enterprise plans)](/docs/secure-compute) for additional layers

- Private, isolated cloud environments
- Dedicated outgoing IP addresses

[VPC peering and VPN support](/docs/secure-compute#vpn-support) (built on top of Secure Compute) allows customers to create fewer entry p

[Learn](https://security.vercel.com/?itemUid=aec41c33-0f3a-4030-ac59-49adfd4a975b\&source=click) about how Vercel supports HIPAA complian

[Contact us](https://vercel.com/contact/sales/security) to request a **BAA** or to add Secure Compute to your plan.

### EU-U.S Data Privacy Framework

The EU-U.S [Data Privacy Framework](https://www.dataprivacyframework.gov) (DPF) provides U.S. organizations a reliable mechanism for tran

The International Trade Administration (ITA) within the U.S. Department of Commerce administers the DPF program, enabling eligible U.S.-b

**Vercel is certified under the EU-U.S. Data Privacy Framework.** To view our public listing, visit the [Data Privacy Framework website](

Vercel's certification provides adequate data protection for transferring personal data outside of the EU, UK, and Switzerland under the

[Learn more](https://security.vercel.com/?itemName=data_privacy\&source=click) about Vercel's data privacy practices or visit our [Privac

### TISAX

The [Trusted Information Security Assessment Exchange](https://enx.com/tisax) (TISAX) is a recognized standard in the automotive industry

Vercel has achieved TISAX Assessment Level 2 (AL2), which covers requirements for handling information with a high need for protection. T

- Reducing the time and cost of third party service provider security and privacy reviews
- Aligning with Original Equipment Manufacturer (OEM) and various automotive supply chain requirements
- Supporting compliance across regulated environments

TISAX results are not intended for the general public. Vercel's assessment results are available to registered ENX participants through t

[Contact us](https://vercel.com/contact/sales/security) for more information.

## Infrastructure

The Vercel CDN and deployment platform primarily uses Amazon Web Services (AWS), and currently has [19 different regions](/docs/regions)

We use a multi-layered security approach that combines people, processes, and technology, including centralized [IAM](# "What is IAM?"),

We use cloud security processes to develop and implement procedures for provisioning, configuring, managing, monitoring, and accessing cl

To ensure always-on security, Vercel's edge infrastructure uses a combination of cloud-native and vendor tooling, including cloud securit

When an AWS outage occurs in a region, Vercel will automatically route traffic to the nearest available edge, ensuring network resilience

### Where does my data live?

Vercel operates on a shared responsibility model with customers. Customers have the ability to select their preferred region for deployin

Additionally, Vercel may transfer data to and in the United States and anywhere else in the world where Vercel or its service providers m

### Failover strategy

- Vercel uses [AWS Global Accelerator](https://aws.amazon.com/global-accelerator/) and our Anycast network to automatically reroute traff
- [Vercel Functions](/docs/functions/configuring-functions/region#automatic-failover) have multiple availability zone redundancy by defau
- Our core database and data plane is a globally replicated database with rapid manual failover, using multiple availability zones

#### Regional failover

With region-based failover, Vercel data is replicated across multiple regions, and a failover is triggered when an outage occurs in a reg

#### Resiliency testing

To meet [RTO/RPO](# "What is RTO/RPO?") goals, Vercel conducts recurring resiliency testing. This testing simulates regional failures. Th

### Data encryption

Vercel encrypts data at rest (when on disk) with 256 bit Advanced Encryption Standard (AES-256). While data is in transit (on route betwe

> **💡 Note:** If you need isolated runtime infrastructure, you can use [Vercel Secure
> Compute](/docs/secure-compute) to create a private, isolated cloud environment
> with dedicated outgoing IP addresses.

### Data backup

Vercel backs-up customer data at an interval of every two hours, each backup is persisted for 30 days, and is globally replicated for res

All backups are stored separately in a storage service. If a database instance is deleted, all associated backups are also automatically

> **💡 Note:** These backups are **not available** to customers and are created for Vercel's
> infrastructure's use in case of disaster.

### Do Enterprise accounts run on a different infrastructure?

Enterprise Teams on Vercel have their own build infrastructure ensuring isolation from Hobby/Pro accounts on Vercel.

### Penetration testing and Audit scans

Vercel conducts regular penetration testing through third-party penetration testers, and has daily code reviews and static analysis check

---
title: "Content Warning Interstitial FAQ"
description: "Learn what the Content Warning page means when visiting a site on Vercel, why it appears, and what you can do if you see it
last_updated: "2026-01-16T02:19:35.412Z"
source: "https://vercel.com/docs/security/faq-content-warning-interstitial"
---

# Content Warning Interstitial FAQ

When you see a **Content Warning** page while visiting a site hosted on Vercel, it means our systems detected signs that the site might p

These warnings protect visitors from accessing a potentially harmful site.

## What this warning means

Vercel may show an interstitial page when our automated systems or trusted reports suggest that a site may be unsafe.

Common examples of potential risks include:

- Deceptive or misleading pages (for example, fake login forms or impersonation attempts)
- Unsafe downloads or embedded code
- Other signals that indicate risky or harmful behavior

You're in control: you can **close the page** to return to safety or **continue to the site** if you trust it.

## Why we use these warnings

Our goal is to help users make safer decisions when visiting sites hosted on Vercel.

Warnings appear when either automated detection or human review indicate a site might be deceptive, harmful, or insecure.

We don't share the exact detection details publicly - that information could be misused to evade detection.

However, we continuously refine our internal models to minimize false positives and ensure accuracy.

## What you can do

- **Go back to safety** - safest choice if you're unsure
- **Continue (not recommended)** - proceed if you're confident the site is legitimate
- **Report an error** - if you believe this warning is incorrect, you can [contact us through our review form](https://vercel.com/account

## How we review reports

When a review request is submitted, our Safety team re-evaluates the site using automated checks and/or human review. If the site is conf

## Our commitment to transparency

We believe in protecting users and empowering developers with clear information.

Even though we can’t share specific security signals, Vercel:

- Uses content warnings to address trust & safety risks
- Offers clear next steps for both users and site owners
- Accepts flagging and feedback on content warning accuracy
- Works to continually improve our detection accuracy

## For site owners: appeal a content warning

If your site shows a **Content Warning** interstitial (the warning page before entry), it means our systems identified potential security

This section explains what that means, why it happens, and how to request a review.

### Why a warning might appear

A warning appears when a site or project exhibits behavior that could put users at risk.

While we don't disclose internal detection rules, common triggers include:

- Misleading branding or impersonation patterns
- Unsafe downloads or embedded code
- Unsecured HTTPS connections or certificate issues
- Redirects or cloaking that misrepresent the destination
- Multiple credible abuse reports

These warnings are not punitive - they're a proactive protection measure for the platform and its users.

### How to request a review

If you are the site owner and you believe your site was flagged incorrectly, you can request a re-evaluation using our secure form:

[Submit a review request](https://vercel.com/accountrecovery?userType=existing\&problemType=content-warning)

**Steps:**

1. Visit the form above and include:
   - The site URL or project ID
   - A short explanation of the site's purpose and your authorization to use material that may be trademarked or copyrighted
   - Confirmation that your content follows Vercel's Terms of Service
2. Submit the form
3. Our Safety team will review your case and follow up

If the site is found to be safe, the warning will be removed.

### Best practices to prevent future warnings

Adopt strong web-safety and transparency practices:

- Keep SSL/TLS certificates valid and up to date
- Avoid designs or domains that mimic other brands
- Clearly identify your organization or ownership
- Regularly patch software and dependencies
- Review redirects, forms, and scripts for potential misuse

### If you disagree with a review decision

If you still believe your site was incorrectly flagged after review, you can submit a **secondary appeal** within 14 days.

Reply to your review email and include new evidence or steps you've taken to address potential risks.

### Our broader commitment

Vercel's content warning system is one part of our overall safety approach.

We aim to balance openness with accountability - helping users make informed choices while allowing legitimate developers to build freely

### Related resources

- [Vercel Terms of Service](https://vercel.com/legal/terms)
- [Vercel Fair Use Guidelines](https://vercel.com/docs/limits/fair-use-guidelines)

---
title: "Vercel security overview"
description: "Vercel provides built-in and customizable features to ensure that your site is secure."
last_updated: "2026-01-16T02:19:35.417Z"
source: "https://vercel.com/docs/security"
---

# Vercel security overview

Cloud-deployed web applications face constant security threats, with attackers launching millions of malicious attacks weekly. Your appli

A comprehensive security strategy requires both active protection, robust policies, and compliance frameworks:

- [Security governance and policies](#governance-and-policies) ensure long-term organizational safety, maintain regulatory adherence, and
- A [Multi-layered protection](#multi-layered-protection) system provides active security against immediate threats and attacks.

## Governance and policies

### Compliance measures

Learn about the [protection and compliance measures](/docs/security/compliance) Vercel takes to ensure the security of your data, includi

### Shared responsibility model

A [shared responsibility model](/docs/security/shared-responsibility) is a framework designed to split tasks and obligations between two

### Encryption

Out of the box, every Deployment on Vercel is served over an [HTTPS connection](/docs/security/encryption). The SSL certificates for thes

## Multi-layered protection

Understand how Vercel protects every incoming request with [multiple layers](/docs/security/firewall-concepts#how-vercel-secures-requests

### Vercel firewall

The Vercel firewall helps to protect your applications and websites from malicious attacks and unauthorized access through:

- An enterprise-grade platform-wide firewall available for free for all customers with no configuration required that includes automatic
- A [Web Application Firewall (WAF)](/docs/security/vercel-waf) that supports custom rules, managed rulesets, and allows customers to cha
- [Observability](/docs/vercel-firewall/firewall-observability) into network traffic and firewall activity, including the access to firew

---
title: "PCI DSS iframe Integration"
description: "Learn how to integrate an iframe into your application to support PCI DSS compliance."
last_updated: "2026-01-16T02:19:35.423Z"
source: "https://vercel.com/docs/security/pci-dss"
---

# PCI DSS iframe Integration

## Benefits of using an `iframe`

When you use an [\<InlineCode>iframe\</InlineCode>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/iframe "What is an <I

In accordance with Vercel's [shared responsibility model](/docs/security/shared-responsibility), this approach facilitates:

- **Data isolation**: The payment card information entered in the `iframe` is isolated from Vercel’s environment and **does not** pass th
- **Direct data transmission**: Information entered in the `iframe` is sent directly to your payment processor so that Vercel never proce
- **Reduced PCI DSS scope**: With isolation and direct data transmission, the scope of PCI DSS compliance is reduced. This simplifies com

## Integrate an `iframe` for payment processing

1. Select a [payment provider](https://www.pcisecuritystandards.org/glossary/payment-processor/) that offers the following:
   - End-to-end encryption
   - Data tokenization
   - Built-in fraud detection
   - 3DS authentication protocol
   - Compliance with latest PCI DSS requirements

2. Embed the provider's `iframe` in your application's payment page

   This is an example code for a payment processor's `iframe`:

   ```tsx filename="paymentProcessor.tsx" framework=all
   const PaymentProcessorIframe = (): JSX.Element => {
     const paymentProcessorIframeURL = `https://${PAYMENT_PROCESSOR_BASE_URL}.com/secure-payment-form`;

     return (
       <div className="container mx-auto my-10 rounded bg-white p-5 shadow-md">
         <iframe
           src={paymentProcessorIframeURL}
           frameBorder="0"
           width="100%"
           height="500px"
           sandbox="allow-forms allow-top-navigation allow-same-origin"
           className="h-auto w-full"
         />
       </div>
     );
   };

   export default PaymentProcessorIframe;
   ```

   ```jsx filename="paymentProcessor.jsx" framework=all
   const PaymentProcessorIframe = () => {
     const paymentProcessorIframeURL = `https://${PAYMENT_PROCESSOR_BASE_URL}.com/secure-payment-form`;

     return (
       <div className="container mx-auto my-10 rounded bg-white p-5 shadow-md">
         <iframe
           src={paymentProcessorIframeURL}
           frameBorder="0"
           width="100%"
           height="500px"
           sandbox="allow-forms allow-top-navigation allow-same-origin"
           className="h-auto w-full"
         />
       </div>
     );
   };

   export default PaymentProcessorIframe;
   ```

   The `sandbox` attribute and its values are often required by the payment processor:

   - `allow-forms`: Enables form submissions in the `iframe`, essential for payment data entry
   - `allow-top-navigation`: Allows the `iframe` to change the full page URL. This is useful for post-transaction redirections
   - `allow-same-origin`: Permits the `iframe` to interact with resources from the hosting page's origin. This is important for functiona


---
title: "Reverse Proxy Servers and Vercel"
description: "Learn why reverse proxy servers are not recommended with Vercel"
last_updated: "2026-01-16T02:19:35.430Z"
source: "https://vercel.com/docs/security/reverse-proxy"
---

# Reverse Proxy Servers and Vercel

**We do not recommend** placing a reverse proxy server in front of your Vercel project as it affects the Vercel's firewall in the followi

- Vercel's CDN **loses visibility** into the traffic, which reduces the effectiveness of the firewall in identifying suspicious activity.
- Real end-user IP addresses cannot be accurately identified.
- If the reverse proxy undergoes a malicious attack, this traffic can be forwarded to the Vercel project and cause usage spikes.
- If the reverse proxy is compromised, Vercel's firewall cannot automatically purge the cache.

## Using a reverse proxy server

However, you may still need to use a reverse proxy server. For example, your organization has legacy web applications protected by a reve

In such a case, you want to ensure that Vercel's [platform-wide firewall](/docs/vercel-firewall#platform-wide-firewall) does not block th

### Prerequisites

- **TLS setup:** Disable HTTP→HTTPS redirection for `http://<DOMAIN>/.well-known/acme-challenge/*` on port 80
- **Cache control:** Never cache `https://<DOMAIN>/.well-known/vercel/*` paths
- **Plan eligibility:**
  - Hobby/Pro: Verified Proxy Lite only
  - Enterprise: Lite + Advanced (self-hosted/geolocation preservation)

### Automatic vs. Manual enablement

Verified Proxy is automatically enabled for the providers listed below on all plans. Any other provider or a self-hosted proxy (for examp

### Supported providers (Verified Proxy Lite)

| Provider | Required Header | Configuration |
| --- | --- | --- |
| Fastly | `Fastly-Client-IP` | A built-in header. No additional configuration required. |
| Google Cloud Load Balancing | `X-GCP-Connecting-IP` | Add a custom header: `X-GCP-Connecting-IP: {client_ip_address}` using their |
| Cloudflare | `CF-Connecting-IP` | A built-in header. No additional configuration required. |
| AWS CloudFront | `CloudFront-Viewer-Address` | Enable the header via the [Origin Request Policy](https://docs.aws.amazon.c |
| Imperva CDN (Cloud WAF) | `Incap-Client-IP` | A built-in header. No additional configuration required. |
| Akamai | `True-Client-IP` | Enable the header via the property manager. Clients may be able to spoof th |
| Azure Front Door | `X-Azure-ClientIP` | A built-in header. No additional configuration required. |
| F5 | `X-F5-True-Client-IP` | Add a custom header: `X-F5-True-Client-IP: {client_ip_address}` |

### Self-hosted reverse proxies (Verified Proxy Advanced)

Ensure that the following requirements are met if you are running self-hosted reverse proxies:

- Your proxy must have static egress IP addresses assigned. We cannot support dynamic IP addresses.
- Your proxy must send a custom request header that carries the real client IP (e.g. `x-${team-slug}-connecting-ip`).
- Your proxy must enable SNI (Server Name Indication) on outbound TLS connections.
- Use consistent and predictable Vercel project domains for onboarding. For example, use *.vercel.example.com and ensure your Proxy alwa

For detailed setup instructions, please contact your Customer Success Manager (CSM) or Account Executive (AE).

## More resources

- [Can I use Vercel as a reverse proxy?](/kb/guide/vercel-reverse-proxy-rewrites-external)

---
title: "Shared Responsibility Model"
description: "Discover the essentials of our Shared Responsibility Model, outlining the key roles and responsibilities for customers, Ver
last_updated: "2026-01-16T02:19:35.440Z"
source: "https://vercel.com/docs/security/shared-responsibility"
---

# Shared Responsibility Model

A shared responsibility model is a framework designed to split tasks and obligations between two groups in cloud computing. The model div

When using a cloud platform such as Vercel, it is important to understand where your security responsibilities lie, and where Vercel take

The customer handles their data, applications, and user access management. This includes data encryption, safeguarding sensitive informat

Vercel manages infrastructure components, such as compute, storage, and networking. Our role is to guarantee that the platform is secure,

## Customer responsibilities

- **Security Requirements Assessment**: Customers are responsible for evaluating and deciding whether Vercel's platform and the security
  - **Handling Malicious Traffic**: Customers are responsible for addressing any costs and resource consumption related to malicious traf
  - **Payment Transactions**: Customers subject to PCI DSS compliance are responsible for choosing an appropriate payment gateway provide
- **Client-side Data**: Customers are responsible for the security and management of data on their clients' devices
- **Source Code**: Customers are responsible for securely storing, and maintaining their source code at all times
- **Server-side Encryption**: Customers are responsible for encrypting their server-side data, whether it's stored in the file system or
- **Identity & Access Management (IAM)**: Customers choose and implement their desired level of access control regarding their IAM config
- **Region Selection for Compute**: Customers are responsible for selecting the appropriate regions for their compute resources based on
- **Production Checklist**: Customers are responsible for implementing and adhering to recommended best practices provided in [Vercel's p
- **Spend Management**: Customers are responsible for enabling [Spend Management](/docs/spend-management) to set a reasonable spend amoun

## Shared responsibilities

- **Information and Data**: Customers control and own their data. By design, customers determine the access to their data and are respons
  - **Integrations**: Customers are responsible for deciding which Vercel services to use and the data that is collected or needed to pro
- **Encryption & Data Integrity**: Vercel is responsible for [encryption](/docs/security/encryption) and data integrity for data in trans
  - **User Code & Environment Variables**: Customers are responsible for managing their application's code, including the exposure of [en
- **Authentication**: Customers handle their app's authentication with tools like [NextAuth.js](https://next-auth.js.org/getting-started/
- **Log Management**: While Vercel provides access to short-term [runtime logs](/docs/runtime-logs) for debugging purposes, it is the cus

## Vercel responsibilities

- **Infrastructure**: Vercel is responsible for the security and availability of the underlying infrastructure used to provide our servic
  - **Multiple Availability Zones and Globally Located Edge Locations**: Vercel makes use of [19 different regions](/docs/regions), which
- **Compute**: Vercel provides a compute environment for customer applications that utilizes Vercel Functions and containers to ensure th
- **Storage**: Vercel is responsible for the security and reliability of storage environments for customer data. This includes the storag
- **Networking**: Vercel is responsible for providing a secure and reliable networking environment for customer applications. This includ

---
title: "Authorization Server API"
description: "Learn how to use the Authorization Server API"
last_updated: "2026-01-16T02:19:35.456Z"
source: "https://vercel.com/docs/sign-in-with-vercel/authorization-server-api"
---

# Authorization Server API

The Authorization Server API exposes a set of endpoints which are used by your application for obtaining, refreshing, revoking, and intro

| Endpoint | URL |
| --- | --- |
| Authorization Endpoint | https://vercel.com/oauth/authorize |
| Token Endpoint | https://api.vercel.com/login/oauth/token |
| Revoke Token Endpoint | https://api.vercel.com/login/oauth/token/revoke |
| Token Introspection Endpoint | https://api.vercel.com/login/oauth/token/introspect |
| User Info Endpoint | https://api.vercel.com/login/oauth/userinfo |

These endpoints and other features of the authorization server are advertised at the following well-known URL:

```
https://vercel.com/.well-known/openid-configuration
```

## Authorization Endpoint

When the user clicks your Sign in with Vercel button, your application should redirect the user to the Authorization Endpoint (`https://v

If the user is not logged in, Vercel will show a login screen and then the consent page to grant or deny the requested [permissions](/doc

The Authorization Endpoint supports the following parameters:

| Parameter | Required | Description |
| --- | --- | --- |

| `client_id`             | **Yes**  | The ID of the App, located in the **Manage** page of the App.
| `scope`                 | **No**   | A space-separated list of [scopes](/docs/sign-in-with-vercel/scopes-and-permissions) you're reques
| `redirect_uri`          | **Yes**  | The URL used to redirect users back to the application after granting authorization, located in th
| `response_type`         | **Yes**  | Must be `code`.
| `response_mode`         | **No**   | Specifies how the authorization response is delivered. Defaults to `query` (redirect with query pa
| `nonce`                 | No       | A random string generated by the application that is used to protect against replay attacks. The s
| `state`                 | No       | A random string generated by the application that is used to protect against [CSRF](# "What is CSR
| `code_challenge`        | **Yes**  | A random string generated by the application for additional protection, based on the [PKCE specifi
| `code_challenge_method` | **Yes**  | Must be `S256`.

In your application create an API Route that saves the `state`, `nonce` and `code_verifier` in cookies and redirects the user to the Auth

After Vercel redirects the user back to your application's `redirect_uri` with a `code`, your application should call the [Token Endpoint

```ts {54} filename="app/api/auth/authorize/route.ts"
import crypto from 'node:crypto';
import { type NextRequest, NextResponse } from 'next/server';
import { cookies } from 'next/headers';

function generateSecureRandomString(length: number) {
  const charset =
    'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-._~';
  const randomBytes = crypto.getRandomValues(new Uint8Array(length));
  return Array.from(randomBytes, (byte) => charset[byte % charset.length]).join(
    '',
  );
}

export async function GET(req: NextRequest) {
  const state = generateSecureRandomString(43);
  const nonce = generateSecureRandomString(43);
  const code_verifier = crypto.randomBytes(43).toString('hex');
  const code_challenge = crypto
    .createHash('sha256')
    .update(code_verifier)
    .digest('base64url');
  const cookieStore = await cookies();

  cookieStore.set('oauth_state', state, {
    maxAge: 10 * 60, // 10 minutes
    secure: true,
    httpOnly: true,
    sameSite: 'lax',
  });
  cookieStore.set('oauth_nonce', nonce, {
    maxAge: 10 * 60, // 10 minutes
    secure: true,
    httpOnly: true,
    sameSite: 'lax',
  });
  cookieStore.set('oauth_code_verifier', code_verifier, {
    maxAge: 10 * 60, // 10 minutes
    secure: true,
    httpOnly: true,
    sameSite: 'lax',
  });

  const queryParams = new URLSearchParams({
    client_id: process.env.NEXT_PUBLIC_VERCEL_APP_CLIENT_ID as string,
    redirect_uri: `${req.nextUrl.origin}/api/auth/callback`,
    state,
    nonce,
    code_challenge,
    code_challenge_method: 'S256',
    response_type: 'code',
    scope: 'openid email profile offline_access',
  });

  const authorizationUrl = `https://vercel.com/oauth/authorize?${queryParams.toString()}`;
  return NextResponse.redirect(authorizationUrl);
}
```

## Token Endpoint

The Token Endpoint is used to exchange the `code` returned from the Authorization Endpoint, or a Refresh Token for a new [Access Token](/

| Parameter       | Required     | Description
| --------------- | ------------ | ----------------------------------------------------------------------------------------------------
| `grant_type`    | **Yes**      | Either `authorization_code` or `refresh_token`.- If the user signs in from the application then `autho
| `client_id`     | **Yes**      | The ID of the App located in the [**Manage**](/docs/sign-in-with-vercel/manage-from-dashboard) page.
| `client_secret` | **Optional** | The client secret generated in the [**Manage**](/docs/sign-in-with-vercel/manage-from-dashboard) page.
| `code`          | No           | If `grant_type` is `authorization_code` then this parameter is required. The value is obtained during
| `code_verifier` | No           | If `grant_type` is `authorization_code` then this parameter is required. It should be the code verifie
| `redirect_uri`  | No           | If `grant_type` is `authorization_code` then this parameter is required. It should be the same value u
| `refresh_token` | No           | If `grant_type` is `refresh_token` then this parameter is required. This is the Refresh Token which wi

The example below shows how to exchange the `code` for tokens in Next.js, validating the `state` and `nonce` before setting the authentic

```ts {91} filename="app/api/auth/callback/route.ts"
import type { NextRequest } from 'next/server';
import { cookies } from 'next/headers';

interface TokenData {
  access_token: string;
  token_type: string;
  id_token: string;
  expires_in: number;
  scope: string;
  refresh_token: string;
```

```
}

export async function GET(request: NextRequest) {
  try {
    const url = new URL(request.url);
    const code = url.searchParams.get('code');
    const state = url.searchParams.get('state');

    if (!code) {
      throw new Error('Authorization code is required');
    }

    const storedState = request.cookies.get('oauth_state')?.value;
    const storedNonce = request.cookies.get('oauth_nonce')?.value;
    const codeVerifier = request.cookies.get('oauth_code_verifier')?.value;

    if (!validate(state, storedState)) {
      throw new Error('State mismatch');
    }

    const tokenData = await exchangeCodeForToken(
      code,
      codeVerifier,
      request.nextUrl.origin,
    );
    const decodedNonce = decodeNonce(tokenData.id_token);

    if (!validate(decodedNonce, storedNonce)) {
      throw new Error('Nonce mismatch');
    }

    await setAuthCookies(tokenData);

    const cookieStore = await cookies();

    // Clear the state, nonce, and oauth_code_verifier cookies
    cookieStore.set('oauth_state', '', { maxAge: 0 });
    cookieStore.set('oauth_nonce', '', { maxAge: 0 });
    cookieStore.set('oauth_code_verifier', '', { maxAge: 0 });

    // Redirect the user to the profile page, your application may have a different page
    return Response.redirect(new URL('/profile', request.url));
  } catch (error) {
    console.error('OAuth callback error:', error);
    // Redirect the user to the error page, your application may have a different page
    return Response.redirect(new URL('/auth/error', request.url));
  }
}

function validate(
  value: string | null,
  storedValue: string | undefined,
): boolean {
  if (!value || !storedValue) {
    return false;
  }
  return value === storedValue;
}

function decodeNonce(idToken: string): string {
  const payload = idToken.split('.')[1];
  const decodedPayload = Buffer.from(payload, 'base64').toString('utf-8');
  const nonceMatch = decodedPayload.match(/"nonce":"([^"]+)"/);
  return nonceMatch ? nonceMatch[1] : '';
}

async function exchangeCodeForToken(
  code: string,
  code_verifier: string | undefined,
  requestOrigin: string,
): Promise<TokenData> {
  const params = new URLSearchParams({
    grant_type: 'authorization_code',
    client_id: process.env.NEXT_PUBLIC_VERCEL_APP_CLIENT_ID as string,
    client_secret: process.env.VERCEL_APP_CLIENT_SECRET as string,
    code: code,
    code_verifier: code_verifier || '',
    redirect_uri: `${requestOrigin}/api/auth/callback`,
  });

  const response = await fetch('https://api.vercel.com/login/oauth/token', {
    method: 'POST',
    body: params,
  });

  if (!response.ok) {
    const errorData = await response.json();
    throw new Error(
      `Failed to exchange code for token: ${JSON.stringify(errorData)}`,
    );
  }

  return await response.json();
}

async function setAuthCookies(tokenData: TokenData) {
  const cookieStore = await cookies();

  cookieStore.set('access_token', tokenData.access_token, {
    httpOnly: true,
```

```
    secure: process.env.NODE_ENV === 'production',
    sameSite: 'lax',
    maxAge: tokenData.expires_in,
  });

  cookieStore.set('refresh_token', tokenData.refresh_token, {
    httpOnly: true,
    secure: process.env.NODE_ENV === 'production',
    sameSite: 'lax',
    maxAge: 60 * 60 * 24 * 30, // 30 days
  });
}
```

The expected response from the Token Endpoint is a JSON object with the following properties:

```json filename="Token Endpoint response example"
{
  "access_token": "vca_...",
  "token_type": "Bearer",
  "id_token": "...", // The ID Token is a JWT
  "expires_in": 3600,
  "scope": "openid email profile offline_access", // The scopes that were granted to the application
  "refresh_token": "vcr_..." // Present if offline_access scope is requested
}
```

## Revoke Token Endpoint

Both the Access and Refresh Token can be revoked before expiration if needed. If the Access Token is revoked, the Refresh Token is also r

```ts {14} filename="app/api/auth/signout/route.ts"
import { cookies } from 'next/headers';

export async function POST() {
  const cookieStore = await cookies();
  const accessToken = cookieStore.get('access_token')?.value;

  if (!accessToken) {
    return Response.json({ error: 'No access token found' }, { status: 401 });
  }

  const credentials = `${process.env.NEXT_PUBLIC_VERCEL_APP_CLIENT_ID}:${process.env.VERCEL_APP_CLIENT_SECRET}`;

  const response = await fetch(
    'https://api.vercel.com/login/oauth/token/revoke',
    {
      method: 'POST',
      headers: {
        Authorization: `Basic ${Buffer.from(credentials).toString('base64')}`,
      },
      body: new URLSearchParams({
        token: accessToken,
      }),
    },
  );

  if (!response.ok) {
    const errorData = await response.json();
    console.error('Error revoking token:', errorData);
    return Response.json(
      { error: 'Failed to revoke access token' },
      { status: response.status },
    );
  }

  cookieStore.set('access_token', '', { maxAge: 0 });
  cookieStore.set('refresh_token', '', { maxAge: 0 });

  return Response.json({}, { status: response.status });
}
```

## Token Introspection Endpoint

The token introspection endpoint validates an Access Token or Refresh Token and returns metadata about its state. Use this endpoint to ch

| Parameter | Required | Description                                                  |
| --------- | -------- | ------------------------------------------------------------ |
| `token`   | **Yes**  | The token to validate (either Access Token or Refresh Token). |

The endpoint returns a JSON response with token metadata:

```json filename="Token Introspection response"
{
  "active": true,
  "client_id": "cl_p4M3ExwwNx2qfEMWQHZfoajUbbYiTR4i",
  "token_type": "bearer",
  "exp": 1757367451,
  "iat": 1757363851,
  "sub": "XLrCnEgbKhsyfbiNR7E849p",
  "iss": "https://vercel.com",
  "jti": "6cd20f0f-0ce2-408b-a21b-63445bccb69a",
  "session_id": "44c44cd9-6b1a-4a16-9296-cc9aea3f1800"
}
```

The example below shows how to validate a token in Next.js:

```ts {26} filename="app/api/validate-token/route.ts"
```

```
import { cookies } from 'next/headers';

interface IntrospectionResponse {
  active: boolean;
  aud?: string;
  client_id?: string;
  token_type?: 'bearer';
  exp?: number;
  iat?: number;
  sub?: string;
  iss?: string;
  jti?: string;
  session_id?: string;
}

export async function GET(): Promise<Response> {
  try {
    const cookieStore = await cookies();
    const token = cookieStore.get('access_token')?.value;

    if (!token) {
      return Response.json({ error: 'No access token found' }, { status: 401 });
    }

    const introspectResponse = await fetch(
      'https://api.vercel.com/login/oauth/token/introspect',
      {
        method: 'POST',
        body: new URLSearchParams({ token }),
      },
    );

    if (!introspectResponse.ok) {
      return Response.json(
        { error: 'Failed to introspect token' },
        { status: 500 },
      );
    }

    const introspectionData: IntrospectionResponse =
      await introspectResponse.json();

    if (!introspectionData.active) {
      return Response.json({ error: 'Token is not active' }, { status: 401 });
    }

    return Response.json({
      message: 'Token is valid',
      tokenInfo: introspectionData,
    });
  } catch (error) {
    console.error('Token validation error:', error);
    return Response.json({ error: 'Internal server error' }, { status: 500 });
  }
}
```

## User Info Endpoint

The user info endpoint returns the consented OpenID claims about the signed-in user. You must authenticate to this endpoint by including

The endpoint returns a JSON response with consented OpenID claims:

```json filename="User Info Endpoint response"
{
  "sub": "345e869043f1e55f8bdc837c",
  "email": "user@example.com",
  "email_verified": true,
  "name": "John Doe",
  "preferred_username": "john-doe",
  "picture": "https://api.vercel.com/www/avatar/avatar-42…"
}
```

The example below shows how to request user info in Next.js:

```ts {23} filename="app/api/user-info/route.ts"
import { cookies } from 'next/headers';

interface UserInfoResponse {
  sub: string;
  email?: string;
  email_verified?: boolean;
  name?: string;
  preferred_username?: string;
  picture?: string;
}

export async function GET(): Promise<Response> {
  try {
    const cookieStore = await cookies();
    const token = cookieStore.get('access_token')?.value;

    if (!token) {
      return Response.json({ error: 'No access token found' }, { status: 401 });
    }

    const userInfoResponse = await fetch(
      // User Info
      'https://api.vercel.com/login/oauth/userinfo',
```

```
      {
        method: 'POST',
        headers: {
          Authorization: `Bearer ${token}`,
        },
      },
    );

    if (!userInfoResponse.ok) {
      return Response.json(
        { error: 'Failed to fetch user info' },
        { status: 500 },
      );
    }

    const userInfoData: UserInfoResponse = await userInfoResponse.json();

    return Response.json({
      userInfo: userInfoData,
    });
  } catch (error) {
    console.error('Error fetching user info:', error);
    return Response.json({ error: 'Internal server error' }, { status: 500 });
  }
}
```

---
title: "Consent Page"
description: "Learn how the consent page works when users authorize your app"
last_updated: "2026-01-16T02:19:35.460Z"
source: "https://vercel.com/docs/sign-in-with-vercel/consent-page"
---

# Consent Page

When users sign in to your application for the first time, Vercel shows them a consent page that displays:

- Your app's name and logo
- The permissions your app requests
- Two actions: **Allow** or **Cancel**

Users review these permissions before deciding whether to authorize your app.

## When users click Allow

When a user clicks **Allow**, Vercel redirects them to your authorization callback URL with a `code` query parameter:

```plaintext
https://example.com/callback?code=abc123...
```

Your application exchanges this code for tokens using the [Token Endpoint](/docs/sign-in-with-vercel/authorization-server-api#token-endpo

## When users click Cancel

When a user clicks **Cancel**, Vercel redirects them to your authorization callback URL with error parameters:

```bash
https://example.com/callback?
  error=access_denied&
  error_description=The user canceled the authorization process
```

Your application should handle this error and display an appropriate message to the user.

## Returning users

Users only see the consent page the first time they authorize your app, and if you add new scopes and permissions to your app. On subsequ

To force users to see the consent page again, include `prompt=consent` in your authorization request. Learn more in the [Authorization En

---
title: "Getting started with Sign in with Vercel"
description: "Learn how to get started with Sign in with Vercel"
last_updated: "2026-01-16T02:19:35.480Z"
source: "https://vercel.com/docs/sign-in-with-vercel/getting-started"
---

# Getting started with Sign in with Vercel

This guide uses Next.js App Router. You'll create a Sign in with Vercel button that redirects to the authorization endpoint, add a callba

> **💡 Note:** View a live version of this tutorial to see the sign in flow in action.

### Prerequisites

- A Vercel account
- A project deployed to Vercel
- An App [created from the dashboard](/docs/sign-in-with-vercel/manage-from-dashboard#create-an-app)
- A client secret [generated from the dashboard](/docs/sign-in-with-vercel/manage-from-dashboard#generate-a-client-secret)
- An authorization callback URL [configured from the dashboard](/docs/sign-in-with-vercel/manage-from-dashboard#configure-the-authorizati
  - This should be configured to be
    - `http://localhost:3000/api/auth/callback` for running the application locally
    - `https://<your-apps-domain>/api/auth/callback` for running the application in production
- The necessary permissions [configured from the dashboard](/docs/sign-in-with-vercel/manage-from-dashboard#configure-the-necessary-permi

- ### Add environment variables

  Add the following variables to your `.env.local` at your project's root:
  ```env filename=".env.local"
  NEXT_PUBLIC_VERCEL_APP_CLIENT_ID="your-client-id-from-the-dashboard"
  VERCEL_APP_CLIENT_SECRET="your-client-secret-from-the-dashboard"
  ```
  > **💡 Note:** When you are ready to go to production, add your [environment
  > variables](/docs/environment-variables) to your project from the dashboard. If
  > you have [Vercel CLI](/docs/cli) installed, you can run [`vercel env
  >   pull`](/docs/cli/env) to pull the values from your project settings into your
  > local file.

- ### Create your folder structure for the API routes
  Create a folder structure for the API routes in your project. Each API route will be in a folder with the name of the route.
  - `app/api/auth/authorize`: This route will be used to redirect the user to the authorization endpoint.
  - `app/api/auth/callback`: This route will be used to exchange the `code` for tokens.
  - `app/api/auth/signout`: This route will be used to sign the user out.
  - `app/api/validate-token`: This route is **optional** and will be used to validate the access token.

- ### Create an `authorize` API route
  Use the `authorize` route to redirect the user to the authorization endpoint.
  - Generate a secure random string for the `state`, `nonce`, and `code_verifier`.
  - Create a cookie for the `state`, `nonce`, and `code_verifier`.
  - Redirect the user to the authorization endpoint with the required parameters.
  ```ts filename="app/api/auth/authorize/route.ts"
  import crypto from 'node:crypto';
  import { type NextRequest, NextResponse } from 'next/server';
  import { cookies } from 'next/headers';

  function generateSecureRandomString(length: number) {
    const charset =
      'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-._~';
    const randomBytes = crypto.getRandomValues(new Uint8Array(length));
    return Array.from(randomBytes, (byte) => charset[byte % charset.length]).join(
      '',
    );
  }

  export async function GET(req: NextRequest) {
    const state = generateSecureRandomString(43);
    const nonce = generateSecureRandomString(43);
    const code_verifier = crypto.randomBytes(43).toString('hex');
    const code_challenge = crypto
      .createHash('sha256')
      .update(code_verifier)
      .digest('base64url');
    const cookieStore = await cookies();

    cookieStore.set('oauth_state', state, {
      maxAge: 10 * 60, // 10 minutes
      secure: true,
      httpOnly: true,
      sameSite: 'lax',
    });
    cookieStore.set('oauth_nonce', nonce, {
      maxAge: 10 * 60, // 10 minutes
      secure: true,
      httpOnly: true,
      sameSite: 'lax',
    });
    cookieStore.set('oauth_code_verifier', code_verifier, {
      maxAge: 10 * 60, // 10 minutes
      secure: true,
      httpOnly: true,
      sameSite: 'lax',
    });

    const queryParams = new URLSearchParams({
      client_id: process.env.NEXT_PUBLIC_VERCEL_APP_CLIENT_ID as string,
      redirect_uri: `${req.nextUrl.origin}/api/auth/callback`,
      state,
      nonce,
      code_challenge,
      code_challenge_method: 'S256',
      response_type: 'code',
      scope: 'openid email profile offline_access',
    });

    const authorizationUrl = `https://vercel.com/oauth/authorize?${queryParams.toString()}`;
    return NextResponse.redirect(authorizationUrl);
  }
  ```

- ### Create a `callback` API route
  Use the `callback` route to exchange the authorization code for tokens.
  - Validate the `state` and `nonce`.
  - Exchange the `code` for tokens using the stored `code_verifier`.
  - Set the authentication cookies.
  - Clear the temporary cookies (`state`, `nonce`, and `code_verifier`).
  - Redirect the user to the profile page.
  ```ts filename="app/api/auth/callback/route.ts"
  import type { NextRequest } from 'next/server';
  import { cookies } from 'next/headers';

  interface TokenData {
    access_token: string;
    token_type: string;
    id_token: string;
    expires_in: number;
    scope: string;
    refresh_token: string;
  ```

```
}

export async function GET(request: NextRequest) {
  try {
    const url = new URL(request.url);
    const code = url.searchParams.get('code');
    const state = url.searchParams.get('state');

    if (!code) {
      throw new Error('Authorization code is required');
    }

    const storedState = request.cookies.get('oauth_state')?.value;
    const storedNonce = request.cookies.get('oauth_nonce')?.value;
    const codeVerifier = request.cookies.get('oauth_code_verifier')?.value;

    if (!validate(state, storedState)) {
      throw new Error('State mismatch');
    }

    const tokenData = await exchangeCodeForToken(
      code,
      codeVerifier,
      request.nextUrl.origin,
    );
    const decodedNonce = decodeNonce(tokenData.id_token);

    if (!validate(decodedNonce, storedNonce)) {
      throw new Error('Nonce mismatch');
    }

    await setAuthCookies(tokenData);

    const cookieStore = await cookies();

    // Clear the state, nonce, and oauth_code_verifier cookies
    cookieStore.set('oauth_state', '', { maxAge: 0 });
    cookieStore.set('oauth_nonce', '', { maxAge: 0 });
    cookieStore.set('oauth_code_verifier', '', { maxAge: 0 });

    return Response.redirect(new URL('/profile', request.url));
  } catch (error) {
    console.error('OAuth callback error:', error);
    return Response.redirect(new URL('/auth/error', request.url));
  }
}

function validate(
  value: string | null,
  storedValue: string | undefined,
): boolean {
  if (!value || !storedValue) {
    return false;
  }
  return value === storedValue;
}

function decodeNonce(idToken: string): string {
  const payload = idToken.split('.')[1];
  const decodedPayload = Buffer.from(payload, 'base64').toString('utf-8');
  const nonceMatch = decodedPayload.match(/"nonce":"([^"]+)"/);
  return nonceMatch ? nonceMatch[1] : '';
}

async function exchangeCodeForToken(
  code: string,
  code_verifier: string | undefined,
  requestOrigin: string,
): Promise<TokenData> {
  const params = new URLSearchParams({
    grant_type: 'authorization_code',
    client_id: process.env.NEXT_PUBLIC_VERCEL_APP_CLIENT_ID as string,
    client_secret: process.env.VERCEL_APP_CLIENT_SECRET as string,
    code: code,
    code_verifier: code_verifier || '',
    redirect_uri: `${requestOrigin}/api/auth/callback`,
  });

  const response = await fetch('https://api.vercel.com/login/oauth/token', {
    method: 'POST',
    body: params.toString(),
  });

  if (!response.ok) {
    const errorData = await response.json();
    throw new Error(
      `Failed to exchange code for token: ${JSON.stringify(errorData)}`,
    );
  }

  return await response.json();
}

async function setAuthCookies(tokenData: TokenData) {
  const cookieStore = await cookies();

  cookieStore.set('access_token', tokenData.access_token, {
    httpOnly: true,
    secure: process.env.NODE_ENV === 'production',
    sameSite: 'lax',
```

```
      maxAge: tokenData.expires_in,
    });

    cookieStore.set('refresh_token', tokenData.refresh_token, {
      httpOnly: true,
      secure: process.env.NODE_ENV === 'production',
      sameSite: 'lax',
      maxAge: 60 * 60 * 24 * 30, // 30 days
    });
  }
  ```

- ### Create a profile page
  Create a profile page to display the user's information.
  ```tsx filename="app/profile/page.tsx"
  import { cookies } from 'next/headers';
  import Link from 'next/link';
  import SignOutButton from '../components/sign-out-button';

  export default async function Profile() {
    const cookieStore = await cookies();
    const token = cookieStore.get('access_token')?.value;
    const result = await fetch('https://api.vercel.com/v2/user', {
      method: 'GET',
      headers: {
        Authorization: `Bearer ${token}`,
      },
    });
    const data = await result.json();
    const user = data.user;

    if (!user) {
      return (
        <main className="p-10">
          <h1 className="text-3xl font-semibold">Error</h1>
          <p className="mt-4">
            An error occurred while trying to fetch your profile.
          </p>
          Go{' '}
          <Link className="underline" href="/">
            back to the home page
          </Link>{' '}
          and sign in again.
        </main>
      );
    }

    return (
      <main className="p-10">
        <h1 className="text-3xl font-semibold">Profile</h1>
        <p className="mt-4">
          Welcome to your profile page <strong>{user.name}</strong>.
        </p>
        <div>
          <h2 className="text-xl font-semibold mt-8">User Details</h2>
          <ul className="mt-4">
            <li>
              <strong>Name:</strong> {user.name}
            </li>
            <li>
              <strong>Email:</strong> {user.email}
            </li>
            <li>
              <strong>Username:</strong> {user.username}
            </li>
          </ul>
        </div>
        <SignOutButton />
      </main>
    );
  }
  ```

- ### Create an error page
  Create an error page to display when an error occurs.
  ```tsx filename="app/auth/error/page.tsx"
  import Link from 'next/link';

  export default function ErrorPage() {
    return (
      <div>
        <h1>Error</h1>
        <p>An error occurred while trying to sign in.</p>
        <Link href="/">Back to the home page</Link>
      </div>
    );
  }
  ```

- ### Create a `signout` API route
  Use the `signout` route to revoke the token and sign the user out.
  - Revoke the access token.
  - Clear the `access_token` and `refresh_token` cookies.
  - Return a JSON response.
  ```ts filename="app/api/auth/signout/route.ts"
  import { cookies } from 'next/headers';

  export async function POST() {
    const cookieStore = await cookies();
    const accessToken = cookieStore.get('access_token')?.value;
```

````
    if (!accessToken) {
      return Response.json({ error: 'No access token found' }, { status: 401 });
    }

    const credentials = `${process.env.NEXT_PUBLIC_VERCEL_APP_CLIENT_ID}:${process.env.VERCEL_APP_CLIENT_SECRET}`;

    const response = await fetch(
      'https://api.vercel.com/login/oauth/token/revoke',
      {
        method: 'POST',
        headers: {
          Authorization: `Basic ${Buffer.from(credentials).toString('base64')}`,
        },
        body: new URLSearchParams({
          token: accessToken,
        }),
      },
    );

    if (!response.ok) {
      const errorData = await response.json();
      console.error('Error revoking token:', errorData);
      return Response.json(
        { error: 'Failed to revoke access token' },
        { status: response.status },
      );
    }

    cookieStore.set('access_token', '', { maxAge: 0 });

    return Response.json({}, { status: response.status });
  }
  ```

- ### Add Sign in and Sign out buttons
  Add two components to start the OAuth flow (and sign in) and to sign out:
  ```tsx filename="app/components/sign-in-with-vercel-button.tsx"
  import Link from 'next/link';

  export default function SignInWithVercelButton() {
    return <Link href="/api/auth/authorize">Sign in with Vercel</Link>;
  }
  ```
  ```tsx filename="app/components/sign-out-button.tsx"
  'use client';
  import { useTransition } from 'react';

  export default function SignOutButton() {
    const [isPending, startTransition] = useTransition();

    return (
      <button
        onClick={() =>
          startTransition(async () => {
            await fetch('/api/auth/signout', { method: 'POST' });
          })
        }
        disabled={isPending}
      >
        {isPending ? 'Signing out...' : 'Sign out'}
      </button>
    );
  }
  ```

- ### Run your application
  Run your application locally using the following command:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm run dev
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn run dev
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm run dev
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun run dev
      ```
    </Code>
  </CodeBlock>
  Open <http://localhost:3000> and Sign in with Vercel. You will be redirected to the [consent page](/docs/sign-in-with-vercel/consent-pa

- ### Create a token introspection API route (Optional)
  The `validate-token` API route can be used to validate the access token. This is optional, but it can be useful to validate the access
  ```ts filename="app/api/validate-token/route.ts"
  import { cookies } from 'next/headers';

  interface IntrospectionResponse {
    active: boolean;
    aud?: string;
````

```
  client_id?: string;
  token_type?: 'bearer';
  exp?: number;
  iat?: number;
  sub?: string;
  iss?: string;
  jti?: string;
  session_id?: string;
}

export async function GET(): Promise<Response> {
  try {
    const cookieStore = await cookies();
    const token = cookieStore.get('access_token')?.value;

    if (!token) {
      return Response.json({ error: 'No access token found' }, { status: 401 });
    }

    const introspectResponse = await fetch(
      'https://api.vercel.com/login/oauth/token/introspect',
      {
        method: 'POST',
        body: new URLSearchParams({ token }),
      },
    );

    if (!introspectResponse.ok) {
      return Response.json(
        { error: 'Failed to introspect token' },
        { status: 500 },
      );
    }

    const introspectionData: IntrospectionResponse =
      await introspectResponse.json();

    if (!introspectionData.active) {
      return Response.json({ error: 'Token is not active' }, { status: 401 });
    }

    return Response.json({
      message: 'Token is valid!',
      tokenInfo: introspectionData,
    });
  } catch (error) {
    console.error('Token validation error:', error);
    return Response.json({ error: 'Internal server error' }, { status: 500 });
  }
}
```

- ### Create a token introspection component (Optional)
  The `TokenIntrospection` component can be used to validate the access token. This is optional, but it can be useful to validate the acc
  ```tsx filename="app/components/token-introspection.tsx"
  'use client';

  import { useState } from 'react';

  interface ValidationResponse {
    message: string;
    tokenInfo: {
      active: boolean;
      clientId?: string;
      tokenType?: string;
      subject?: string;
      issuer?: string;
      tokenId?: string;
      sessionId?: string;
      expiresAt?: number;
      issuedAt?: number;
    };
  }

  export default function TokenIntrospection() {
    const [validationData, setValidationData] =
      useState<ValidationResponse | null>(null);
    const [loading, setLoading] = useState(false);
    const [error, setError] = useState<string | null>(null);

    const handleValidateToken = async () => {
      setLoading(true);
      setError(null);
      setValidationData(null);

      try {
        const response = await fetch('/api/validate-token');

        const data = await response.json();

        if (!response.ok) {
          throw new Error(data.error || `HTTP error! status: ${response.status}`);
        }

        setValidationData(data as ValidationResponse);
      } catch (err) {
        setError(err instanceof Error ? err.message : 'An error occurred');
      } finally {
        setLoading(false);
      }
```

```
  };

  const formatTimestamp = (timestamp?: number) => {
    if (!timestamp) return 'N/A';
    return new Date(timestamp * 1000).toLocaleString();
  };

  return (
    <div className="mt-8">
      <h2 className="text-xl font-semibold">Token Introspection</h2>
      <p className="mt-2 text-sm">
        Validate your access token using Vercel&apos;s token introspection
        endpoint.
      </p>

      <button
        onClick={handleValidateToken}
        disabled={loading}
        className={`mt-4 px-4 py-2 rounded-lg border ${
          loading
            ? 'opacity-60 cursor-not-allowed'
            : 'cursor-pointer hover:opacity-80'
        }`}
      >
        {loading ? 'Validating Token...' : 'Validate Token'}
      </button>

      {error && (
        <div className="mt-4 p-4 bg-red-50 border border-red-200 rounded-lg">
          <p className="text-red-600">Error: {error}</p>
        </div>
      )}

      {validationData && (
        <div className="mt-4 p-4 border rounded-lg">
          <h3 className="font-semibold mb-2 text-green-800">
            ✓ {validationData.message}
          </h3>
          <div className="space-y-2 text-sm">
            <div>
              <strong>Token Active:</strong>{' '}
              <span className="text-green-600">
                {validationData.tokenInfo.active ? 'Yes' : 'No'}
              </span>
            </div>

            {validationData.tokenInfo.active && (
              <>
                <div>
                  <strong>Client ID:</strong>{' '}
                  {validationData.tokenInfo.clientId || 'N/A'}
                </div>
                <div>
                  <strong>Token Type:</strong>{' '}
                  {validationData.tokenInfo.tokenType || 'N/A'}
                </div>
                <div>
                  <strong>Subject (User ID):</strong>{' '}
                  {validationData.tokenInfo.subject || 'N/A'}
                </div>
                <div>
                  <strong>Issuer:</strong>{' '}
                  {validationData.tokenInfo.issuer || 'N/A'}
                </div>
                <div>
                  <strong>Token ID (JTI):</strong>{' '}
                  {validationData.tokenInfo.tokenId || 'N/A'}
                </div>
                <div>
                  <strong>Session ID:</strong>{' '}
                  {validationData.tokenInfo.sessionId || 'N/A'}
                </div>
                <div>
                  <strong>Issued At:</strong>{' '}
                  {formatTimestamp(validationData.tokenInfo.issuedAt)}
                </div>
                <div>
                  <strong>Expires At:</strong>{' '}
                  {formatTimestamp(validationData.tokenInfo.expiresAt)}
                </div>
              </>
            )}
          </div>
        </div>
      )}
    </div>
  );
}
```

Add this component to your profile page.

```tsx filename="app/profile/page.tsx"
import TokenIntrospection from '../components/token-introspection';

export default function Profile() {
  //...rest of your profile page code
  return (
    //...rest of your profile page code
    <TokenIntrospection />
  );
}
```

```

---
title: "Manage Sign in with Vercel from the Dashboard"
description: "Learn how to manage Sign in with Vercel from the Dashboard"
last_updated: "2026-01-16T02:19:35.494Z"
source: "https://vercel.com/docs/sign-in-with-vercel/manage-from-dashboard"
---

# Manage Sign in with Vercel from the Dashboard

## Create an App

To manage any third-party apps, or create a new one yourself, you need to create an App. An App acts as an intermediary that requests and

To create an App, follow these steps:

1. Navigate to your teams [**Settings**](https://vercel.com/d?to=%2F%5Bteam%5D%2F%7E%2Fsettings\&title=Go+to+team+settings) tab
2. Scroll down and select **Apps**, and click **Create**
3. Choose a name for your app
4. Choose a slug for your app (The slug is automatically generated from the name if you don't provide one)
5. Optionally add a logo for your app
6. Click **Save**

| Field | Required | Description
| ----- | -------- | ----------
| Name  | Yes      | The name of your app. It must be unique across all Vercel applications. Example: `My App`
| Slug  | Yes      | The slug of your app. A URL friendly name that uniquely identifies your app. Defaults to the name if not provided. E
| Logo  | Optional | The logo that represents your app.

## Choose your client authentication method

The client authentication method determines how your app will authenticate with the Vercel Authorization Server. You can enable multiple

| Field | Description | Usage
| ---------- | ---------- | ----------
| `client_secret_basic` | HTTP Basic Authentication Scheme | Client credentials are sent via HTTP Basic Authenticati
| `client_secret_post` | HTTP request body as a form parameter | Client credentials are included as form parameters in t
| `client_secret_jwt` | JSON Web Token (JWT) | Client authenticates using a JWT signed with the shared
| `none` | For public, unauthenticated, non-confidential clients | No client authentication required - suitable for public

## Generate a client secret

Client secrets are used to authenticate your app with the Vercel Authorization Server. You can generate a client secret by clicking the *

> **💡 Note:** You can have up to two active client secrets at a time. This lets you rotate
> secrets without downtime.

## Configure the authorization callback URL

The authorization callback URL is where Vercel redirects users after they authorize your app. This URL must be registered to prevent unau

To add a callback URL:

1. Navigate to the **Manage** page for your app
2. Scroll to **Authorization Callback URLs**
3. Enter your callback URL
4. Click **Add**

For local development, add `http://localhost:3000/api/auth/callback`. For production, add `https://your-domain.com/api/auth/callback`. Fo

When a user authorizes your app, Vercel redirects them to this URL with a `code` query parameter. Your application exchanges this code fo

## Configure the necessary permissions

Permissions control what data your app can access. Configure them from the **Permissions** page in your app settings.

To configure permissions:

1. Navigate to the **Manage** page for your app
2. Select the **Permissions** tab
3. Enable the scopes and permissions your app needs:
   - **openid**: Required to issue an ID Token for user identification
   - **email**: Access the user's email address in the ID Token
   - **profile**: Access the user's name, username, and profile picture in the ID Token
   - **offline\_access**: Issue a Refresh Token to get new Access Tokens without re-authentication
4. Click **Save**

When users authorize your app, they'll see these permissions on the consent page and decide whether to grant access.

Learn more about scopes and permissions in the [scopes and permissions](/docs/sign-in-with-vercel/scopes-and-permissions) documentation.

---
title: "Sign in with Vercel"
description: "Learn how to Sign in with Vercel"
last_updated: "2026-01-16T02:19:35.501Z"
source: "https://vercel.com/docs/sign-in-with-vercel"
---

# Sign in with Vercel

Sign in with Vercel lets people use their Vercel account to log in to your application. Your application doesn't need to handle passwords

Vercel's IdP uses the [OAuth 2.0](https://auth0.com/intro-to-iam/what-is-oauth-2 "What is the OAuth 2.0 protocol?") authorization framewo

> **💡 Note:** For users to be able to use Sign in with Vercel in your application, they must
> have a Vercel account.
```

To learn how to set up Sign in with Vercel, see the [getting started with Sign in with Vercel](/docs/sign-in-with-vercel/getting-started)

## When to use Sign in with Vercel

Sign in with Vercel should be used when you want to offer your users an easy way to sign in to your application.

In the same way that you can sign in with Google, GitHub, or other providers on the web, you can use Sign in with Vercel to authenticate

When configuring the app you will be able to choose which user information will be shared to your application, and users will have to [co

## High level overview

Sign in with Vercel is based on the OAuth 2.0 authorization framework, which allows your application to request access to user data from '

1. A user clicks the Sign in with Vercel button in your application
2. Your application redirects the user to Vercel's IdP consent page (or opens it in a popup window)
3. They review the permissions and click **Allow**
4. After approval by the user, Vercel sends your application a short lived `code` to your pre-registered callback URL
5. Your application swaps the `code` for tokens
6. Your application uses those tokens to identify the user and log them into your application

### Tokens

- **ID Token**: A signed JWT that proves who the user is. Your application verifies its signature and read claims to identify the user
- **Access Token**: A bearer token your application uses to call the Vercel REST API for the permissions the user grants. This lasts for
- **Refresh Token**: This token lets your application get a new Access Token without asking the user to sign in again. This lasts for 30

Learn more about each token in the [tokens](/docs/sign-in-with-vercel/tokens) documentation.

### Scopes and permissions

Scopes decide what identity information from the user goes into the ID Token and whether to issue a Refresh Token.

Learn more about scopes and permissions in the [scopes and permissions](/docs/sign-in-with-vercel/scopes-and-permissions) documentation.

### Consent page

The first time someone tries to sign in to your application, Vercel will show them a consent page to review the permissions your applicat

If the user grants access, they are redirected back to your application where you can use the tokens to identify the user and log them in

If they cancel the sign in, they are redirected back to your application where you can handle the failed sign in state in your applicatio

Learn more about the consent page in the [consent page](/docs/sign-in-with-vercel/consent-page) documentation.

## More resources

- [Getting started with Sign in with Vercel](/docs/sign-in-with-vercel/getting-started)
- [Tokens](/docs/sign-in-with-vercel/tokens)
- [Scopes and permissions](/docs/sign-in-with-vercel/scopes-and-permissions)
- [Authorization Server API](/docs/sign-in-with-vercel/authorization-server-api)
- [Manage Sign in with Vercel from the dashboard](/docs/sign-in-with-vercel/manage-from-dashboard)
- [Consent page](/docs/sign-in-with-vercel/consent-page)
- [Troubleshooting](/docs/sign-in-with-vercel/troubleshooting)

--------------------------------------------------------------------------------
title: "Scopes and Permissions"
description: "Learn how to manage scopes and permissions for Sign in with Vercel"
last_updated: "2026-01-16T02:19:35.505Z"
source: "https://vercel.com/docs/sign-in-with-vercel/scopes-and-permissions"
--------------------------------------------------------------------------------

# Scopes and Permissions

Scopes define what data is included in the [ID Token](/docs/sign-in-with-vercel/tokens#id-token) and whether to issue a [Refresh Token](/

## Scopes

The following scopes are available:

| Scope | Description |
| --- | --- |
| `openid` | Required permission, needed to issue an [ID Token](/docs/sign-in-with-vercel/tokens#id-token) for user identificatio |
| `email` | Enabling this scope grants access to the user's email address in the [ID Token](/docs/sign-in-with-vercel/tokens#id- |
| `profile` | Enabling this scope grants access to the user's basic profile information, including name, username, and profile pic |
| `offline_access` | Enabling this scope issues a [Refresh Token](/docs/sign-in-with-vercel/tokens#refresh-token). |

## Permissions

> **💡 Note:** Permissions for issuing API requests and interacting with team resources are
> currently in private beta.

--------------------------------------------------------------------------------
title: "Tokens"
description: "Learn how to Sign in with Vercel"
last_updated: "2026-01-16T02:19:35.514Z"
source: "https://vercel.com/docs/sign-in-with-vercel/tokens"
--------------------------------------------------------------------------------

# Tokens

There are three tokens your application will work with when using Sign in with Vercel:

- [ID Token](#id-token)
- [Access Token](#access-token)
- [Refresh Token](#refresh-token)

## ID Token

The ID Token is a signed JWT that contains information about the user who is signing in. When using ID Token claims, your application sho

```json filename="ID Token payload example"
{
  "iss": "https://vercel.com",
  "sub": "345e869043f1e55f8bdc837c",
  "aud": "cl_be6c3c8b9f340d4a20feefab2862a49a",
  "exp": 1519948800,
  "iat": 1519945200,
  "nbf": 1519945200,
  "jti": "50e67781-c8b6-4391-98d1-89d755bb095a",
  "name": "John Doe",
  "preferred_username": "john-doe",
  "picture": "https://api.vercel.com/www/avatar/00159aa4c88348dedc91a456b457d1baa48df6d",
  "email": "user@example.com",
  "nonce": "a4a522fa63f9cea6eeb1"
}
```

The code below shows how to decode and validate an ID token using the [jose](https://www.npmjs.com/package/jose) library:

```ts
import { jwtVerify, createRemoteJWKSet } from 'jose';

const jwkSet = createRemoteJWKSet(
  new URL('https://vercel.com/.well-known/jwks'),
);

async function decodeIdToken(idToken: string) {
  const { payload } = await jwtVerify(idToken, jwkSet, {
    issuer: 'https://vercel.com',
    audience: [process.env.NEXT_PUBLIC_VERCEL_APP_CLIENT_ID],
  });

  return payload;
}
```

### JWT claims in ID Tokens

Vercel's IdP generates OpenID Connect tokens that contain various JWT claims depending on the requested scopes:

| Claim | Type | Description | Example |
| --- | --- | --- | --- |
| `iss` | string | **Issuer** - The server that issued the token | `"https://vercel.com"` |
| `sub` | string | **Subject** - Unique identifier for the authenticated user | `"345e869043f1e55f8bdc837c"` |
| `aud` | string | **Audience** - The ID of the Vercel application | `"cl_be6c3c8b9f340d4a20feefab2862a49a"` |
| `exp` | number | **Expiration time** - Unix timestamp when the token expires | `1519948800` |
| `iat` | number | **Issued at** - Unix timestamp when the token was issued | `1519945200` |
| `nbf` | number | **Not before** - Unix timestamp before which the token is invalid | `1519945200` |
| `jti` | string | **JWT ID** - Unique identifier for this specific token | `"50e67781-c8b6-4391-98d1-89d755bb095a"` |
| `nonce` | string | Cryptographic nonce for replay protection | `"a4a522fa63f9cea6eeb1"` |

### Scope dependent claims

Depending on the scopes requested the following claims will be included in the ID Token:

| Scope | Claims | Description | Example |
| --- | --- | --- | --- |
| `profile` | `name` | The user's full display name | `"John Doe"` |
| `profile` | `preferred_username` | The user's username on Vercel | `"john-doe"` |
| `profile` | `picture` | URL to the user's avatar image (only if user has an avatar) | `"https://api.vercel.com/www/avatar/av |
| `email` | `email` | The user's email address | `"user@example.com"` |

## Access Token

The Access Token grants your application permission to access specific resources on Vercel on behalf of the user trying to sign in. It is

```plaintext filename="Access Token example"
vca_BQuu9ChDu3n6Pfh6YQnCshpoYkWDSFKogLqmBtQ0tC8NAA5rXt340sjz
```

Access Tokens are valid for one hour. Refresh Tokens can be exchanged to receive new Access Tokens when they expire. Refresh Tokens are v

When using the Access Token in your application code to fetch the user's data, it must be included in the `Authorization` header as a Bea

```ts filename="Fetching the users data with the Access Token"
const result = await fetch('https://api.vercel.com/v2/user', {
  method: 'GET',
  headers: {
    Authorization: `Bearer ${token}`,
  },
});
```

## Refresh Token

Refresh Tokens allow your application to get a new Access Token without asking the user to sign in again. The token lasts for 30 days and

Each Refresh Token is single use and automatically rotated on exchange, invalidating the previous token.

Refresh Tokens use an opaque format that ensures they are not readable by humans, are secure, and have server side validation to ensure t

```plaintext filename="Refresh Token example"
vcr_BQuu9ChDu3n6Pfh6YQnCshpoYkWDSFKogLqmBtQ0tC8NAA5rXt340sjz
```

## Securing your tokens

Access and Refresh Tokens are sensitive credentials and should be stored securely. Never expose them to the client side of your applicati

- They can be stored in cookies with the `HttpOnly`, `Secure` and `SameSite=Strict` attributes
- They can be stored in a database with encryption
- Revoke tokens immediately if you suspect they have been compromised, either by calling the [Revoke Token Endpoint](/docs/sign-in-with-v

---
title: "Troubleshooting Sign in with Vercel"
description: "Learn how to troubleshoot common errors with Sign in with Vercel"
last_updated: "2026-01-16T02:19:35.520Z"
source: "https://vercel.com/docs/sign-in-with-vercel/troubleshooting"
---

# Troubleshooting Sign in with Vercel

When users try to authorize your app, several errors can occur. Common troubleshooting steps include:

- Checking that all required parameters are included in your requests
- Verifying your app configuration in the dashboard
- Reviewing the [Authorization Server API](/docs/sign-in-with-vercel/authorization-server-api) documentation
- Checking the [Getting Started](/docs/sign-in-with-vercel/getting-started) guide for implementation examples

## Error handling patterns

Vercel handles authorization errors in two ways:

- **Error page**: Shown when critical parameters are missing or invalid
- **Redirect with error**: User redirected to your callback URL with error parameters

When errors redirect to your callback URL, your application must handle them and show users an appropriate message.

## Authorization endpoint errors

These errors occur when users navigate to the authorization endpoint with invalid parameters.

### Missing or invalid client_id

When the `client_id` parameter is missing or references a non-existent app, Vercel shows an error page.

**Fix**: Verify your `client_id` matches the ID shown in your app's **Manage** page.

### Missing or invalid redirect_uri

When the `redirect_uri` parameter is missing or doesn't match a registered callback URL, Vercel shows an error page.

**Fix**: Add the redirect URL to your app's **Authorization Callback URLs** in the **Manage** page.

### Missing response_type

When the `response_type` parameter is missing, Vercel redirects to your callback URL with an error:

```plaintext
https://example.com/api/auth/callback?
  error=invalid_request&
  error_description=Parameter 'response_type'. Required
```

**Fix**: Include `response_type=code` in your authorization request.

### Invalid response_type

When the `response_type` parameter has an invalid value, Vercel redirects to your callback URL with an error:

```plaintext
https://example.com/api/auth/callback?
  error=invalid_request&
  error_description=Parameter 'response_type'. Invalid enum value. Expected 'code', received 'test'
```

**Fix**: Set `response_type=code`. This is the only supported value.

### Invalid code_challenge length

When the `code_challenge` parameter is provided but not between 43 and 128 characters, Vercel redirects to your callback URL with an erro

```plaintext
https://example.com/api/auth/callback?
  error=invalid_request&
  error_description=Parameter 'code_challenge'. code_challenge must be at least 43 characters
```

**Fix**: Generate a `code_challenge` that's between 43 and 128 characters long. Follow the [PKCE specification](https://datatracker.ietf.

### Invalid code_challenge_method

When the `code_challenge_method` parameter has an invalid value, Vercel redirects to your callback URL with an error:

```plaintext
https://example.com/api/auth/callback?
  error=invalid_request&
  error_description=Parameter 'code_challenge_method'. Invalid enum value. Expected 'S256', received 'test'
```

**Fix**: Set `code_challenge_method=S256`. This is the only supported value.

### Invalid prompt parameter

When the `prompt` parameter has an invalid value, Vercel redirects to your callback URL with an error:

```plaintext
https://example.com/api/auth/callback?
  error=invalid_request&
  error_description=Parameter 'prompt'. Invalid enum value. Expected 'consent' | 'login', received 'test'
```

**Fix**: Use only `consent` or `login` for the `prompt` parameter. Leave it out if you don't need to control the authorization behavior.

---
title: "Vercel Enterprise Managed Infrastructure"

last_updated: "2026-01-16T02:19:35.588Z"
source: "https://vercel.com/docs/sitecore/managed-infrastructure"
---

# Vercel Enterprise Managed Infrastructure

Vercel prices its [CDN](/docs/cdn) resources by region to help optimize costs and performance for your projects. This is to ensure you ar

### Managed Infrastructure Units

Managed Infrastructure Units (MIUs) serve as both a financial commitment and a measurement of the infrastructure consumption of an Enterp

**MIUs are billed monthly and do not roll over from month to month**.

### Regional pricing

The following table lists the usage amounts for each resource in Managed Infrastructure Units. Resources that depend on the region of you

Use the dropdown to select the region you are interested in.

### Fluid compute regional pricing

The following table shows the regional pricing for fluid compute resources on Vercel. The prices are per hour for CPU and per GB-hr for m

| Region | Active CPU time (per hour) | Provisioned Memory (GB-hr) |
| ------------------------------ | -------------------------- | -------------------------- |
| Washington, D.C., USA (iad1) | 0.128 MIUs | 0.0106 MIUs |
| Cleveland, USA (cle1) | 0.128 MIUs | 0.0106 MIUs |
| San Francisco, USA (sfo1) | 0.177 MIUs | 0.0147 MIUs |
| Portland, USA (pdx1) | 0.128 MIUs | 0.0106 MIUs |
| Cape Town, South Africa (cpt1) | 0.200 MIUs | 0.0166 MIUs |
| Hong Kong (hkg1) | 0.176 MIUs | 0.0146 MIUs |
| Mumbai, India (bom1) | 0.140 MIUs | 0.0116 MIUs |
| Osaka, Japan (kix1) | 0.202 MIUs | 0.0167 MIUs |
| Seoul, South Korea (icn1) | 0.169 MIUs | 0.0140 MIUs |
| Singapore (sin1) | 0.160 MIUs | 0.0133 MIUs |
| Sydney, Australia (syd1) | 0.180 MIUs | 0.0149 MIUs |
| Tokyo, Japan (hnd1) | 0.202 MIUs | 0.0167 MIUs |
| Frankfurt, Germany (fra1) | 0.184 MIUs | 0.0152 MIUs |
| Dublin, Ireland (dub1) | 0.168 MIUs | 0.0139 MIUs |
| London, UK (lhr1) | 0.177 MIUs | 0.0146 MIUs |
| Paris, France (cdg1) | 0.177 MIUs | 0.0146 MIUs |
| Stockholm, Sweden (arn1) | 0.160 MIUs | 0.0133 MIUs |
| Dubai, UAE (dxb1) | 0.185 MIUs | 0.0153 MIUs |
| São Paulo, Brazil (gru1) | 0.221 MIUs | 0.0183 MIUs |
| Montréal, Canada (yul1) | 0.147 MIUs | 0.0121 MIUs |

### Additional usage based products

The following table lists the MIUs for additional usage based products in Managed Infrastructure.

---
title: "Skew Protection"
description: "Learn how Vercel"
last_updated: "2026-01-16T02:19:35.538Z"
source: "https://vercel.com/docs/skew-protection"
---

# Skew Protection

[Version skew](https://www.industrialempathy.com/posts/version-skew/) occurs when different versions of your application run on client an

Vercel's Skew Protection resolves this problem at the platform and framework layer by using [version locking](https://www.industrialempat

By implementing Skew Protection, you can reduce user-facing errors during new rollouts and boost developer productivity, minimizing conce

## Enable Skew Protection

Projects created after November 19th 2024 using one of the [supported frameworks](#supported-frameworks) already have Skew Protection ena

For older projects, you can enable Skew Protection in your project's settings.

1. Ensure your project has the [Automatically expose system environment variables](/docs/environment-variables/system-environment-variabl
2. Ensure your deployment method is not using the `vercel deploy --prebuilt` option. To learn more, see [When not to use --prebuilt](/doc
3. Select the project in the Vercel dashboard
4. Select the **Settings** tab in the top menu
5. Select the **Advanced** tab in the side menu
6. Scroll down to **Skew Protection** and enable the switch
7. You can optionally set a custom maximum age (see [limitations](#limitations))
8. [Redeploy](/docs/deployments/managing-deployments#redeploy-a-project) your latest production deployment.

## Custom Skew Protection Threshold

In some cases, you may have problematic deployments you want to ensure no longer resolves requests from any other active clients.

Once you deploy a fix, you can set a Skew Protection threshold with the following:

1. Select the deployment that fixed the problem in the deployment list
2. Select the  button (near the **Visit** button)
3. Click **Skew Protection Threshold**
4. Click **Set** to apply the changes

This ensure that deployments created before the fixed deployment will no longer resolve requests from outdated clients.

## Monitor Skew Protection

You can observe how many requests are protected from version skew by visiting the [Monitoring](/docs/observability/monitoring) page in th

For example, on the `requests` event, filter where `skew_protection = 'active'`.

You can view Edge Requests that are successfully fulfilled without the need for skew protection by using `skew_protection = 'inactive'`.

## Supported frameworks

Skew Protection is available with zero configuration when using the following frameworks:

- [Next.js](#skew-protection-with-next.js)
- [SvelteKit](#skew-protection-with-sveltekit)
- [Qwik](#skew-protection-with-qwik)
- [Astro](#skew-protection-with-astro)
- Nuxt ([coming soon](https://github.com/nitrojs/nitro/issues/2311))

Other frameworks can implement Skew Protection by checking if `VERCEL_SKEW_PROTECTION_ENABLED` has value `1`
and then appending the value of `VERCEL_DEPLOYMENT_ID` to each request using **one of the following** options.

- `dpl` query string parameter:

  ```ts filename="option1.ts"
  const query =
    process.env.VERCEL_SKEW_PROTECTION_ENABLED === '1'
      ? `?dpl=${process.env.VERCEL_DEPLOYMENT_ID}`
      : '';

  const res = await fetch(`/get${query}`);
  ```

- `x-deployment-id` header:

  ```ts filename="option2.ts"
  const headers =
    process.env.VERCEL_SKEW_PROTECTION_ENABLED === '1'
      ? { 'x-deployment-id': process.env.VERCEL_DEPLOYMENT_ID }
      : {};

  const res = await fetch('/get', { headers });
  ```

- `__vdpl` cookie:

  ```ts filename="option3.ts"
  export default function handler(req, res) {
    if (
      process.env.VERCEL_SKEW_PROTECTION_ENABLED === '1' &&
      req.headers['sec-fetch-dest'] === 'document'
    ) {
      res.setHeader('Set-Cookie', [
        `__vdpl=${process.env.VERCEL_DEPLOYMENT_ID}; HttpOnly`,
      ]);
    }
    res.end('<h1>Hello World</h1>');
  }
  ```

### Skew Protection with Next.js

> **⚠️ Warning:** If you're building outside of Vercel and then deploying using the `vercel
>   deploy --prebuilt` command, Skew Protection will not be enabled by default
> because the Deployment ID is not known at build time.For more information, see [When not to use --prebuilt](/docs/cli/deploy#when-not-t

If you are using Next.js 14.1.4 or newer, there is no additional configuration needed to [enable Skew Protection](#enable-skew-protection

Older versions of Next.js require additional [`next.config.js`](https://nextjs.org/docs/app/api-reference/next-config-js) configuration.

### Skew Protection with SvelteKit

If you are using SvelteKit, you will need to install `@sveltejs/adapter-vercel` version 5.2.0 or newer in order to [enable Skew Protectio

Older versions can be upgraded by running `npm i -D @sveltejs/adapter-vercel@latest`.

### Skew Protection with Qwik

If you are using Qwik 1.5.3 or newer, there is no additional configuration needed to [enable Skew Protection](#enable-skew-protection).

Older versions can be upgraded by running `npm i @builder.io/qwik@latest`.

### Skew Protection with Astro

If you are using Astro, you will need to install `@astrojs/vercel` version 9.0.0 or newer in order to [enable Skew Protection](#enable-sk

```js {8} filename="astro.config.mjs"
import { defineConfig } from 'astro/config';
import vercel from '@astrojs/vercel';

export default defineConfig({
  // ...
  output: 'server',
  adapter: vercel({
```

```
    skewProtection: true,
  }),
});
```

Older versions can be upgraded by running `npm i -D @astrojs/vercel@latest`.

## Limitations

Skew Protection is only available for Pro and Enterprise, not for Hobby teams. You can configure a custom maximum age up to, but not exce

Vercel automatically adjusts the maximum age to 60 days for requests from Googlebot and Bingbot in order to handle any delay between docu

Deployments that have been deleted either manually or automatically using a [retention policy](/docs/deployment-retention) will not be ac

## More resources

- [Version Skew Protection blog](/blog/version-skew-protection)
- [Version Skew](https://www.industrialempathy.com/posts/version-skew/)

---
title: "Speed Insights Intake API"
description: "Learn how to use Speed Insights in Vercel with any frontend framework or project through the Speed Insights intake API."
last_updated: "2026-01-16T02:19:35.558Z"
source: "https://vercel.com/docs/speed-insights/api"
---

# Speed Insights Intake API

Vercel Speed Insights supports Next.js, Nuxt, and Gatsby with zero configuration through build plugins. You can use Speed Insights with a

## Getting Started

To use the Speed Insights API, you'll need to retrieve the analytics ID for your Vercel project. This value is exposed during the build a

Inside your framework or Node.js script, you can then use this value in the `body` of your request to the Vercel Speed Insights API.

> **💡 Note:**  does not pull
>  as the Vercel Analytics ID
> environment variable is inlined during the build process. It is not part of your
> project Environment Variables, which can be pulled locally using Vercel CLI.

## Example

You can view an example of the following code implemented inside our [Create React App](https://github.com/vercel/vercel/tree/main/exampl

```javascript filename="vitals.js"
import { getCLS, getFCP, getFID, getLCP, getTTFB } from 'web-vitals';

const vitalsUrl = 'https://vitals.vercel-analytics.com/v1/vitals';

function getConnectionSpeed() {
  return 'connection' in navigator &&
    navigator['connection'] &&
    'effectiveType' in navigator['connection']
    ? navigator['connection']['effectiveType']
    : '';
}

function sendToAnalytics(metric, options) {
  const page = Object.entries(options.params).reduce(
    (acc, [key, value]) => acc.replace(value, `[${key}]`),
    options.path,
  );

  const body = {
    dsn: options.analyticsId, // qPgJqYH9LQX5o31Ormk8iWhCxZO
    id: metric.id, // v2-1653884975443-1839479248192
    page, // /blog/[slug]
    href: location.href, // https://{my-example-app-name-here}/blog/my-test
    event_name: metric.name, // TTFB
    value: metric.value.toString(), // 60.20000000298023
    speed: getConnectionSpeed(), // 4g
  };

  if (options.debug) {
    console.log('[Analytics]', metric.name, JSON.stringify(body, null, 2));
  }

  const blob = new Blob([new URLSearchParams(body).toString()], {
    // This content type is necessary for `sendBeacon`
    type: 'application/x-www-form-urlencoded',
  });
  if (navigator.sendBeacon) {
    navigator.sendBeacon(vitalsUrl, blob);
  } else
    fetch(vitalsUrl, {
      body: blob,
      method: 'POST',
      credentials: 'omit',
      keepalive: true,
    });
}

export function webVitals(options) {
  try {
    getFID((metric) => sendToAnalytics(metric, options));
    getTTFB((metric) => sendToAnalytics(metric, options));
    getLCP((metric) => sendToAnalytics(metric, options));
```

```
    getCLS((metric) => sendToAnalytics(metric, options));
    getFCP((metric) => sendToAnalytics(metric, options));
  } catch (err) {
    console.error('[Analytics]', err);
  }
}
```

---
title: "Limits and Pricing for Speed Insights"
description: "Learn about our limits and pricing when using Vercel Speed Insights. Different limitations are applied depending on your pl
last_updated: "2026-01-16T02:19:35.567Z"
source: "https://vercel.com/docs/speed-insights/limits-and-pricing"
---

# Limits and Pricing for Speed Insights

## Pricing

Speed Insights is available on the Hobby, Pro, and Enterprise plans.

On the Hobby plan, Speed Insights is free and can be enabled on **one** project with a [set allotment](/docs/speed-insights/limits-and-pr

On the Pro plan, the **base** fee for Speed Insights is $10 per-project, per-month.

The following table outlines the price for each resource according to the plan you are on.

Pro teams can [set up Spend Management](/docs/spend-management#managing-your-spend-amount) to get notified or to automatically take actio

## Limitations

Once you've enabled Speed Insights, different limitations are applied depending on your plan:

| | Hobby | Pro | Enterprise |
| --- | --- | --- | --- |
| Reporting Window for Data Points | 7 Day | 30 Days | 90 Days |
| Maximum Number of Data Points per Month | 10,000 | None | None |

Once the maximum limit of data points is reached, no more data points will be recorded until the current day has passed. On the next day,

You can reduce the number of data points collected by adjusting the [Sample Rate](#sample-rate) at the project level by using the `@verce

## Sample rate

By default, all incoming data points are used to calculate the scores you're being presented with on the Speed Insights view.

To reduce cost, you can change the sample rate at a project level by using the `@vercel/speed-insights` package as explained in [Sample r

## Prorating

Teams on the Pro or Enterprise plan will immediately be charged the base fee when enabling Speed Insights for each project. However, you

- If there ten days are remaining in your current billing cycle — *that's roughly 30% of your billing cycle* - you will only pay around 3
- If you disable Speed Insights before the billing cycle ends Vercel will continue to show the already collected data points until the en
- Once the billing cycle is over, Speed Insights will automatically turn off, and you will lose access to existing data. You won't be ref
- If you decide to re-enable the feature after cancellation, you won't be charged when you enable it. Instead, the usual 10 USD base fee

## Usage

The table below shows the metrics for the [**Observability**](/docs/pricing/observability) section of the **Usage** dashboard where you c

To view information on managing each resource, select the resource link in the **Metric** column. To jump straight to guidance on optimiz

See the [manage and optimize Observability usage](/docs/pricing/observability) section for more information on how to optimize your usage

> **💡 Note:** Speed Insights and Web Analytics require scripts to do collection of [data
> points](/docs/speed-insights/metrics#understanding-data-points). These scripts
> are loaded on the client-side and therefore may incur additional usage and
> costs for [Data Transfer](/docs/manage-cdn-usage#fast-data-transfer) and [Edge
> Requests](/docs/manage-cdn-usage#edge-requests).

---
title: "Speed Insights Metrics"
description: "Learn what each performance metric on Speed Insights means and how the scores are calculated."
last_updated: "2026-01-16T02:19:35.583Z"
source: "https://vercel.com/docs/speed-insights/metrics"
---

# Speed Insights Metrics

## Real Experience Score (RES)

### Real user monitoring

While many performance measurement tools, like [Lighthouse](https://web.dev/measure/), estimate user experience based on lab simulations,

As a result, RES shows how real users experience your application. This real-time data helps you understand your application's performanc

You can use these insights to see how new deployments affect performance, helping you improve your application's user experience.

> **💡 Note:** The timestamps in the Speed Insights view are in local time (not UTC).

## Core Web Vitals explained

The Core Web Vitals, as defined by Google and the [Web Performance Working Group](https://www.w3.org/webperf/ "What is the Web Performanc

> **💡 Note:** Speed Insights now uses Lighthouse 10 scoring criteria instead of Lighthouse 6
> criteria as explained in [Updated Scoring
> Criteria](/docs/speed-insights/migrating-from-legacy#updated-scoring-criteria)

| Metric | Description |
| --- | --- |
| [Largest Contentful Paint (LCP)](#largest-contentful-paint-lcp) | Measures the time from page start to when the largest content eleme |
| [Cumulative Layout Shift (CLS)](#cumulative-layout-shift-cls) | Quantifies the fraction of layout shift experienced by the user ove |
| [Interaction to Next Paint (INP)](#interaction-to-next-paint-inp) | Measures the time from user interaction to when the browser renders |
| [First Contentful Paint (FCP)](#first-contentful-paint-fcp) | Measures the time from page start to the rendering of the first pie |
| [First Input Delay (FID)](#first-input-delay-fid) | Measures the time from a user's first interaction to the time the b |
| [Total Blocking Time (TBT)](#total-blocking-time-tbt) | Measures the total amount of time between FCP and TTI where the mai |
| [Time to First Byte (TTFB)](#time-to-first-byte-ttfb) | Measures the time from the request of a resource to when the first |

### Largest Contentful Paint (LCP)

[Largest Contentful Paint](https://web.dev/articles/lcp) (LCP) is a performance metric that measures the time from when the page starts l

**A good LCP time is considered to be 2.5 seconds or less**.

### Cumulative Layout Shift (CLS)

[Cumulative Layout Shift](https://web.dev/articles/cls) (CLS) is a performance metric that quantifies the fraction of layout shift experi

The score is calculated from the product of two measures:

- The impact fraction - the area of the viewport impacted by the shift
- The distance fraction - the distance the elements have moved relative to the viewport between frames

**A good CLS score is considered to be 0.1 or less**.

### Interaction to Next Paint (INP)

[Interaction to Next Paint](https://web.dev/articles/inp) (INP) is a metric that measures the time from when a user interacts with your s

This metric is used to gauge the responsiveness of a page to user interactions. The quicker the page responds to user input, the better t

**Lower INP times are better, with an INP time of 200 milliseconds or less being considered good**.

### First Contentful Paint (FCP)

[First Contentful Paint](https://web.dev/articles/fcp) (FCP) is a performance metric that measures the time from the moment the page star

**Lower FCP times are better, with an FCP time of 1.8 seconds or less being considered good**.

## Other metrics

### Time to First Byte (TTFB)

Time to First Byte (TTFB) measures the time between the request for a resource and when the first byte of a response begins to arrive.

**Lower TTFB times are better, with a good TTFB time being considered as under 800 milliseconds**.

### First Input Delay (FID)

[First Input Delay](https://web.dev/articles/fid) (FID) measures the time from when a user first interacts with your site (by selecting a

**A good FID score is 100 milliseconds or less**.

As [stated by Google](https://web.dev/vitals/#lab-tools-to-measure-core-web-vitals), simulating an environment to measure Web Vitals nece

### Total Blocking Time (TBT)

Total Blocking Time (TBT) quantifies how non-interactive a page is. It measures the total time between the First Contentful Paint (FCP) a

**Lower TBT times are better, with a good TBT time being considered as under 800 milliseconds**.

> **💡 Note:** For more in-depth information related to performance metrics, visit the
> PageSpeed Insights [
> documentation](https://developers.google.com/speed/docs/insights/v5/about).

## How the scores are determined

Vercel calculates performance scores using real-world data obtained from the [HTTP Archive](https://httparchive.org/). This process invol

For instance, if [HTTP Archive](https://httparchive.org/) data shows that the top-performing sites render the Largest Contentful Paint (L

The Real Experience Score is a weighted average of all individual metric scores. Vercel has assigned each metric a specific weighting, wh

## Understanding data points

In the context of Vercel's Speed Insights, a data point is a single unit of information that represents a measurement of a specific Web V

Data points are collected on hard navigations, which in the case of Next.js apps, are only the first-page view in a session. During a use

As of now, up to 6 data points can be potentially tracked per visit:

- On page load: Time to First Byte ([TTFB](#time-to-first-byte-ttfb)) and First Contentful Paint ([FCP](#first-contentful-paint-fcp))
- On interaction: First Input Delay ([FID](#first-input-delay-fid)) and Largest Contentful Paint ([LCP](#largest-contentful-paint-lcp))
- On leave: Interaction to Next Paint ([INP](#interaction-to-next-paint-inp)), Cumulative Layout Shift ([CLS](#cumulative-layout-shift-cl

The collection of metrics may vary depending on how users interact with or exit the page. On average, you can expect to collect between 3

These data points provide insights into various performance aspects of your website, such as the time it takes to display the first conte

### How the percentages are calculated?

By default, the user experience percentile is set to P75, which offers a balanced overview of the majority of user experiences. You can v

The chosen percentile corresponds to the proportion of users who experience a load time faster than a specific value. Here's how each per

- **P75**: Represents the experience of the fastest 75% of your users, excluding the slowest 25%.
- **P90**: Represents the experience of the fastest 90% of your users, excluding the slowest 10%.
- **P95**: Represents the experience of the fastest 95% of your users, excluding the slowest 5%.
- **P99**: Represents the experience of the fastest 99% of your users, excluding the slowest 1%.

For instance, a P75 score of 1 second for [First Contentful Paint (FCP)](#first-contentful-paint-fcp) means that 75% of your users experi

## Interpreting performance scores

Performance metrics, including the [Real Experience Score](#real-user-monitoring), the [Virtual Experience Score](#predictive-performance

- **0 to 49 (red)**: Poor
- **50 to 89 (orange)**: Needs Improvement
- **90 to 100 (green)**: Good

Aim for 'Good' scores (90 to 100) for both Real and Virtual Experience Scores. Keep in mind that reaching a score of 100 is extremely cha

### Implications of scores for the end-user experience

Higher Real Experience and Virtual Experience Scores generally translate to better end-user experiences, making it worthwhile to strive f

If you aim to boost your site's search ranking, aim to move your scores into a higher color-coded category, for instance, from 'Needs Imp

## Predictive performance metrics with Virtual Experience Score

The Real Experience Score ([RES](#real-user-monitoring)) displayed in the Speed Insights tab is derived from actual data points collected

In contrast, the Virtual Experience Score (VES) is a predictive performance metric that allows you to anticipate the impact of changes on

Setting up an integration supporting performance checks enables these checks to run for each deployment. These checks assess whether the

Like RES, the VES draws from four separate Speed Insights, albeit with some variations:

- In place of the First Input Delay ([FID](#first-input-delay-fid)) Core Web Vital, the Virtual Experience Score utilizes Total Blocking
- The specific device type used for checks depends on the Integration you've set up. For example, Checkly only uses "Desktop" for determi

## Breaking down data in Speed Insights

Speed Insights offers a variety of views to help you analyze your application's performance data. This allows you to identify areas that

---
title: "Migrating to the latest Speed Insights package"
description: "Understand the transition from Speed Insights to the new version – know the differences and how they affect you."
last_updated: "2026-01-16T02:19:35.598Z"
source: "https://vercel.com/docs/speed-insights/migrating-from-legacy"
---

# Migrating to the latest Speed Insights package

The new Speed Insights brings a few changes to the UI and the ingestion mechanism. You find a list of changes below and understand how th

## Changes to the integration

### New package: `@vercel/speed-insights`

Vercel introduced a **package** titled [`@vercel/speed-insights`](/docs/speed-insights/package) as an iteration
from the automatic install process. This shift is intended to offer more flexibility
and broader framework support.

By migrating to the new Speed Insights package, you benefit from the following features:

- **First-Party Ingestion**: Data is processed directly through your own domain, eliminating the third-party domain lookup
- **Enhanced Route Support**: Dynamic route segment is supported in more frameworks such as Next.js `app` router, Nuxt, Remix, and Svelte
- **Advanced Customization**: The updated package provides tools for more granular control, such as the ability to [intercept requests](/

You should become familiar with the `@vercel/speed-insights` [configuration options](/docs/speed-insights/package) and upgrade. However,

### Sample rate

Sample rate configurations have been relocated from team settings to the [@vercel/speed-insights package](/docs/speed-insights/package),

### First-Party intake

Data ingestion now utilizes a first-party intake during your deployment. Here's how it works:

- The script is now sourced from your own domain at this endpoint: `https://yourdomain.com/_vercel/speed-insights/script.js`.
- Data points are also ingested through your own domain at this endpoint: `https://yourdomain.com/_vercel/speed-insights/vitals`.

With this change, the script becomes less affected by content blockers and performs fewer DNS lookups, resulting in a faster and more rel
Policy](https://developer.mozilla.org/docs/Web/HTTP/CSP) to allow the third-party script.

## Changes to the UI

### Emphasis on P75

Our revamped dashboard emphasizes the 75th percentile, a [recommendation](https://web.dev/articles/defining-core-web-vitals-thresholds#ch

In other terms, the **score is now determined by the experience of the fastest 75% of your users**.

This percentile was chosen because it represents the performance experienced by the majority of visits and is not significantly affected

For deeper insights, it is now possible to view multiple percentiles at once, without affecting the score.

### Updated Scoring Criteria

Speed Insights now uses scoring criteria that are inspired by the improvements found in Lighthouse 10. Below, you'll find a comprehensive

> **💡 Note:** All previous (prior to the new Speed Insights) and new data points use this
> updated scoring criteria.

**Comparison table between the new and old scoring criteria**

| Metric | Old Thresholds | **New Thresholds** | Old Weights | **New Weights** |
| ------ | -------------- | ------------------ | ----------- | --------------- |
| RES | 90~50 | 90~50 | Not applicable | Not applicable |
| FCP | 0.9~1.6s (Desktop)  2.3~4s (Mobile) | **1.8~3s** | 20% | **15%** |
| LCP | 1.2~2.4s (Desktop)  2.5~4s (Mobile) | **2.5~4s** | 35% | **30%** |
| INP | Not applicable | **200~500ms** | - | **30%** |
| FID | 100~300ms | 100~300ms | 30% | **Not applicable** |
| CLS | 0.1~0.25 | 0.1~0.25 | 15% | **25%** |
| TTFB | Not applicable | 0.8~1.8s | - | - |

The **CLS** metric is given more weight in the new version,
and the **FID** metric is replaced with **INP**. The **FCP**
and **LCP** metrics now have the same thresholds for both desktop and mobile.

### New Metric: TTFB

We've introduced a new metric, [**Time to First Byte** (TTFB)](/docs/speed-insights/metrics#time-to-first-byte-ttfb), which measures the

---
title: "Speed Insights Configuration with @vercel/speed-insights"
description: "Learn how to configure your application to capture and send web performance metrics to Vercel using the @vercel/speed-insig
last_updated: "2026-01-16T02:19:35.673Z"
source: "https://vercel.com/docs/speed-insights/package"
---

# Speed Insights Configuration with @vercel/speed-insights

With the `@vercel/speed-insights` npm package, you're able to configure your application to capture and send web performance metrics to V

## Getting started

To get started with Speed Insights, refer to our [Quickstart](/docs/speed-insights/quickstart) guide which will walk you through the proc

## `sampleRate`

> **💡 Note:** In prior versions of Speed Insights this was managed in the UI. This option is
> now managed through code with the package.

This parameter determines the percentage of events that are sent to the server. By default, all events are sent. Lowering this parameter

To learn more about how to configure the `sampleRate` option, see the [Sending a sample of events to Speed Insights](/kb/guide/sending-sa

## `beforeSend`

With the `beforeSend` function, you can modify or filter out the event data before it's sent to Vercel. You can use this to redact sensit

For instance, if you wish to ignore events from a specific URL or modify them, you can do so with this option.

```tsx
// Example usage of beforeSend
beforeSend: (data) => {
  if (data.url.includes('/sensitive-path')) {
    return null; // this will ignore the event
  }
  return data; // this will send the event as is
};
```

## `debug`

With the debug mode, you can view all Speed Insights events in the browser's console. This option is especially useful during development

This option is **automatically enabled** if the `NODE_ENV` environment variable is available and either `development` or `test`.

You can manually disable it to prevent debug messages in your browsers console.

## `route`

The `route` option allows you to specify the current dynamic route (such as `/blog/[slug]`). This is particularly beneficial when you nee

This option is **automatically set** when using a framework specific import such as for Next.js, Nuxt, SvelteKit and Remix.

## `endpoint`

The `endpoint` option allows you to report the collected metrics to a different url than the default: `https://yourdomain.com/_vercel/spe

This is useful when deploying several projects under the same domain, as it allows you to keep each application isolated.

For example, when `yourdomain.com` is managed outside of Vercel:

1. "alice-app" is deployed under `yourdomain.com/alice/*`, vercel alias is `alice-app.vercel.sh`
2. "bob-app" is deployed under `yourdomain.com/bob/*`, vercel alias is `bob-app.vercel.sh`
3. `yourdomain.com/_vercel/*` is routed to `alice-app.vercel.sh`

Both applications are sending their metrics to `alice-app.vercel.sh`. To restore the isolation, "bob-app" should use:

```tsx
<SpeedInsights endpoint="https://bob-app.vercel.sh/_vercel/speed-insights/vitals" />
```

## `scriptSrc`

The `scriptSrc` option allows you to load the Speed Insights script from a different URL than the default one.

```tsx
<SpeedInsights scriptSrc="https://bob-app.vercel.sh/_vercel/speed-insights/script.js" />
```

## More resources

- [Sending a sample of your events](/kb/guide/sending-sample-to-speed-insights)

---
title: "Speed Insights Overview"
description: "This page lists out and explains all the performance metrics provided by Vercel"
last_updated: "2026-01-16T02:19:35.681Z"
source: "https://vercel.com/docs/speed-insights"
---

# Speed Insights Overview

Vercel **Speed Insights** provides you with a detailed view of your website's performance [metrics](/docs/speed-insights/metrics), based

The **Speed Insights** dashboard offers in-depth information about scores and individual metrics without the need for code modifications

To get started, follow the quickstart to [enable Speed Insights](/docs/speed-insights/quickstart) and learn more about the [dashboard vie

> **💡 Note:** When you enable Speed Insights, data will be tracked on all deployed
> environments, including
> [preview](/docs/deployments/environments#preview-environment-pre-production)
> and [production](/docs/deployments/environments#production-environment)
> deployments.

## Dashboard view

Once you [enable Speed Insights](/docs/speed-insights/quickstart), you can access the dashboard by selecting your project in the Vercel [

The Speed Insights dashboard displays data that you can sort and inspect based on a variety of parameters:

- **Device type**: Toggle between mobile and desktop.
- **Environment**: Filter by preview, production, or all environments.
- **Time range**: Select the timeframe dropdown in the top-right of the page to choose a predefined timeframe. Alternatively, select the
- [**Performance metric**](/docs/speed-insights/metrics): Switch between parameters that include Real Experience Score (RES), First Conte
- **Performance metric views**: When you select a performance metric, the dashboard displays three views:
  - **Time-based line graph** that, by default, shows the P75 [percentile of data](/docs/speed-insights/metrics#how-the-percentages-are-c
  - **Kanban board** that shows which routes, paths, or HTML elements need improvement (URLs that make up less than 0.5% of visits are no
  - **Geographical map** showing the experience metric by country:

The data in the Kanban and map views is selectable so that you can filter by
country, route, path and HTML element. The red, orange and green colors in the
map view indicate the P75 score.

- [Quickstart](/docs/speed-insights/quickstart)
- [Usage and pricing](/docs/speed-insights/limits-and-pricing#pricing)
- [Data points](/docs/speed-insights/metrics#understanding-data-points)
- [Metrics](/docs/speed-insights/metrics)

## More resources

- [How Core Web Vitals affect SEO: Understand your application's Google page experience ranking and Lighthouse scores](https://www.youtub

---
title: "Vercel Speed Insights Privacy & Compliance"
description: "Learn how Vercel follows the latest privacy and data compliance standards with its Speed Insights feature."
last_updated: "2026-01-16T02:19:35.602Z"
source: "https://vercel.com/docs/speed-insights/privacy-policy"
---

# Vercel Speed Insights Privacy & Compliance

To ensure that the Speed Insights feature can be used despite many different regulatory limitations around the world, we've designed it i

The recording of data points is anonymous and the Speed Insights feature does not collect or store information that would enable us to re

The following information is stored with every data point:

| Collected Value | Example Value |
| --- | --- |
| Route | /blog/\[slug] |
| URL | /blog/nextjs-10 |
| Network Speed | 4g (or slow-2g, 2g, 3g) |
| Browser | Chrome 86 (Blink) |
| Device Type | Mobile (or Desktop/Tablet) |
| Device OS | Android 10 |
| Country (ISO 3166-1 alpha-2) | US |
| Web Vital | FCP 1.0s |
| Web Vital Attribution | html>body img.header |
| SDK Information | @vercel/speed-insights 0.1.0 |
| Server-Received Event Time | 2023-10-29 09:06:30 |

See our [Privacy Notice](/legal/privacy-policy) for more information, including how Vercel Speed Insights complies with the GDPR.

## How the data points are tracked

Once you've followed the dashboard's instructions for enabling Speed Insights and installed the `@vercel/speed-insights` package, it will

The package injects a script that retrieves the visitor's [Web Vitals](/docs/speed-insights/metrics) by invoking native browser APIs and

Learn more about the [first-party intake data ingestion method](/docs/speed-insights/migrating-from-legacy#first-party-intake), which ena

---

title: "Getting started with Speed Insights"
description: "Vercel Speed Insights provides you detailed insights into your website"
last_updated: "2026-01-16T02:19:35.736Z"
source: "https://vercel.com/docs/speed-insights/quickstart"
---

# Getting started with Speed Insights

This guide will help you get started with using Vercel Speed Insights on your project, showing you how to enable it, add the package to y

To view instructions on using the Vercel Speed Insights in your project for your framework, use the **Choose a framework** dropdown on th

## Prerequisites

- A Vercel account. If you don't have one, you can [sign up for free](https://vercel.com/signup).
- A Vercel project. If you don't have one, you can [create a new project](https://vercel.com/new).
- The Vercel CLI installed. If you don't have it, you can install it using the following command:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i vercel
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i vercel
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i vercel
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i vercel
      ```
    </Code>
  </CodeBlock>

- ### Enable Speed Insights in Vercel
  On the [Vercel dashboard](/dashboard), select your Project followed by the **Speed Insights** tab. You can also select the button below
  > **💡 Note:** Enabling Speed Insights will add new routes (scoped
  > at`/_vercel/speed-insights/*`) after your next deployment.

- ### Add `@vercel/speed-insights` to your project
  > For \['nextjs', 'nextjs-app', 'sveltekit', 'remix',  'create-react-app', 'nuxt', 'vue', 'other', 'astro']:
  Using the package manager of your choice, add the `@vercel/speed-insights` package to your project:
  > For \['html']:

- > For \[
  >     'nextjs',
  >     'nextjs-app',
  >     'remix',
  >     'create-react-app',
  >     'nuxt',
  >     'vue',
  >     'astro',
  >   ]:
  ### Add the `SpeedInsights` component to your app
  > For \['sveltekit', 'other']:
  ### Call the `injectSpeedInsights` function in your app
  > For \['html']:
  ### Add the `script` tag to your site
  > For \['nextjs']:
  The `SpeedInsights` component is a wrapper around the tracking script, offering more seamless integration with Next.js.

  The instructions differ based on which version of Next.js you're deploying.
  > For \['nextjs-app']:
  The `SpeedInsights` component is a wrapper around the tracking script, offering more seamless integration with Next.js.

  Add the following component to the root layout:
  > For \['create-react-app']:
  The `SpeedInsights` component is a wrapper around the tracking script, offering more seamless integration with React.

  Add the following component to the main app file.
  ```ts {1, 7} filename="App.tsx" framework=create-react-app
  import { SpeedInsights } from '@vercel/speed-insights/react';

  export default function App() {
    return (
      <div>
        {/* ... */}
        <SpeedInsights />
      </div>
    );
  }
  ```
  ```js {1, 7} filename="App.jsx" framework=create-react-app
  import { SpeedInsights } from '@vercel/speed-insights/react';

  export default function App() {
    return (
      <div>
        {/* ... */}
        <SpeedInsights />
      </div>
    );
  }
  ```

> For \['remix']:
The `SpeedInsights` component is a wrapper around the tracking script, offering a seamless integration with Remix.

Add the following component to your root file:
```ts {1, 8} filename="app/root.tsx" framework=remix
import { SpeedInsights } from '@vercel/speed-insights/remix';

export default function App() {
  return (
    <html lang="en">
      <body>
        {/* ... */}
        <SpeedInsights />
      </body>
    </html>
  );
}
```
```js {1, 8} filename="app/root.jsx" framework=remix
import { SpeedInsights } from '@vercel/speed-insights/remix';

export default function App() {
  return (
    <html lang="en">
      <body>
        {/* ... */}
        <SpeedInsights />
      </body>
    </html>
  );
}
```
> For \['sveltekit']:
Add the following component to your root file:
```ts filename="src/routes/+layout.ts" framework=sveltekit
import { injectSpeedInsights } from '@vercel/speed-insights/sveltekit';

injectSpeedInsights();
```
```js filename="src/routes/+layout.js" framework=sveltekit
import { injectSpeedInsights } from '@vercel/speed-insights/sveltekit';

injectSpeedInsights();
```
> For \['html']:
Add the following scripts before the closing tag of the `<body>`:
```ts filename="index.html" framework=html
<script>
  window.si = window.si || function () { (window.siq = window.siq || []).push(arguments); };
</script>
<script defer src="/_vercel/speed-insights/script.js"></script>
```
```js filename="index.html" framework=html
<script>
  window.si = window.si || function () { (window.siq = window.siq || []).push(arguments); };
</script>
<script defer src="/_vercel/speed-insights/script.js"></script>
```
> For \['vue']:
The `SpeedInsights` component is a wrapper around the tracking script, offering more seamless integration with Vue.

Add the following component to the main app template.
```ts {2, 6} filename="src/App.vue" framework=vue
<script setup lang="ts">
import { SpeedInsights } from '@vercel/speed-insights/vue';
</script>

<template>
  <SpeedInsights />
</template>
```
```js {2, 6} filename="src/App.vue" framework=vue
<script setup>
import { SpeedInsights } from '@vercel/speed-insights/vue';
</script>

<template>
  <SpeedInsights />
</template>
```
> For \['nuxt']:
The `SpeedInsights` component is a wrapper around the tracking script, offering more seamless integration with Nuxt.

Add the following component to the default layout.
```ts {2, 6} filename="layouts/default.vue" framework=nuxt
<script setup lang="ts">
import { SpeedInsights } from '@vercel/speed-insights/vue';
</script>

<template>
  <SpeedInsights />
</template>
```
```js {2, 6} filename="layouts/default.vue" framework=nuxt
<script setup>
import { SpeedInsights } from '@vercel/speed-insights/vue';
</script>

<template>
  <SpeedInsights />
</template>
```

```
  </template>
  ```
  > For \['other']:
  Import the `injectSpeedInsights` function from the package, which will add the tracking script to your app. **This should only be calle

  Add the following code to your main app file:
  ```ts filename="main.ts" framework=other
  import { injectSpeedInsights } from '@vercel/speed-insights';

  injectSpeedInsights();
  ```
  ```js filename="main.js" framework=other
  import { injectSpeedInsights } from '@vercel/speed-insights';

  injectSpeedInsights();
  ```
  > For \['astro']:
  Speed Insights is available for both [static](/docs/frameworks/astro#static-rendering) and [SSR](/docs/frameworks/astro#server-side-ren

  To enable this feature, declare the `<SpeedInsights />` component from `@vercel/speed-insights/astro` near the bottom of one of your la
  ```tsx filename="BaseHead.astro" framework=astro
  ---
  import SpeedInsights from '@vercel/speed-insights/astro';
  const { title, description } = Astro.props;
  ---
  <title>{title}</title>
  <meta name="title" content={title} />
  <meta name="description" content={description} />

  <SpeedInsights />
  ```
  ```jsx filename="BaseHead.astro" framework=astro
  ---
  import SpeedInsights from '@vercel/speed-insights/astro';
  const { title, description } = Astro.props;
  ---
  <title>{title}</title>
  <meta name="title" content={title} />
  <meta name="description" content={description} />

  <SpeedInsights />
  ```
  Optionally, you can remove sensitive information from the URL by adding a `speedInsightsBeforeSend` function to the global `window` obj
  ```tsx filename="BaseHead.astro" framework=astro
  ---
  import SpeedInsights from '@vercel/speed-insights/astro';
  const { title, description } = Astro.props;
  ---
  <title>{title}</title>
  <meta name="title" content={title} />
  <meta name="description" content={description} />

  <script is:inline>
    function speedInsightsBeforeSend(data){
      console.log("Speed Insights before send", data)
      return data;
    }
  </script>
  <SpeedInsights />
  ```
  ```jsx filename="BaseHead.astro" framework=astro
  ---
  import SpeedInsights from '@vercel/speed-insights/astro';
  const { title, description } = Astro.props;
  ---
  <title>{title}</title>
  <meta name="title" content={title} />
  <meta name="description" content={description} />

  <script is:inline>
    function speedInsightsBeforeSend(data){
      console.log("Speed Insights before send", data)
      return data;
    }
  </script>
  <SpeedInsights />
  ```
  [Learn more about `beforeSend`](/docs/speed-insights/package#beforesend).

- ### Deploy your app to Vercel
  You can deploy your app to Vercel's global [CDN](/docs/cdn) by running the following command from your terminal:
  ```bash filename="terminal"
  vercel deploy
  ```
  Alternatively, you can [connect your project's git repository](/docs/git#deploying-a-git-repository), which will enable Vercel to deplo

  Once your app is deployed, it's ready to begin tracking performance metrics.
  > **💡 Note:** If everything is set up correctly, you should be able to find the
  > `/_vercel/speed-insights/script.js` script inside the body tag of your page.

- ### View your data in the dashboard
  Once your app is deployed, and users have visited your site, you can view the data in the dashboard.

  To do so, go to your [dashboard](/dashboard), select your project, and click the **Speed Insights** tab.

  After a few days of visitors, you'll be able to start exploring your metrics. For more information on how to use Speed Insights, see [U

Learn more about how Vercel supports [privacy and data compliance standards](/docs/speed-insights/privacy-policy) with Vercel Speed Insig

## Next steps

Now that you have Vercel Speed Insights set up, you can explore the following topics to learn more:

- [Learn how to use the `@vercel/speed-insights` package](/docs/speed-insights/package)
- [Learn about metrics](/docs/speed-insights/metrics)
- [Read about privacy and compliance](/docs/speed-insights/privacy-policy)
- [Explore pricing](/docs/speed-insights/limits-and-pricing)
- [Troubleshooting](/docs/speed-insights/troubleshooting)

---
title: "Troubleshooting Vercel Speed Insights"
description: "Learn about common issues and how to troubleshoot Vercel Speed Insights."
last_updated: "2026-01-16T02:19:35.605Z"
source: "https://vercel.com/docs/speed-insights/troubleshooting"
---

# Troubleshooting Vercel Speed Insights

## No data visible in Speed Insights dashboard

If you are experiencing a situation where data is not visible in the Speed Insights dashboard, it could be due to a couple of reasons.

**How to fix**:

1. Double check if you followed the quickstart instructions correctly
2. Check if your adblocker is interfering with the Speed Insights script. If so, consider disabling it

## Requests are not getting called

If `/_vercel/speed-insights/script.js` is correctly loading but not sending any data (e.g. no `vitals` request), ensure that you're check

## Speed Insights is not working with proxy

We do not recommend placing a reverse proxy in front of Vercel, as it may interfere with the proper functioning of Speed Insights.

**How to fix**:

1. Check your proxy configuration to make sure that all desired pages are correctly proxied to the deployment
2. Additionally, forward all requests to `/_vercel/speed-insights/*` to the deployments to ensure proper functioning of Speed Insights th

---
title: "Using Speed Insights"
description: "Learn how to use Speed Insights to analyze your application"
last_updated: "2026-01-16T02:19:35.625Z"
source: "https://vercel.com/docs/speed-insights/using-speed-insights"
---

# Using Speed Insights

## Accessing Speed Insights

To access Speed Insights:

1. Select a project from your dashboard and navigate to the **Speed Insights** tab.
2. Select the [timeframe](/docs/analytics/using-web-analytics#specifying-a-timeframe) and [environment](/docs/analytics/using-web-analyti
3. Use the panels to [filter](/docs/analytics/filtering) the page or event data you want to view.

## Breaking down data in Speed Insights

Speed Insights offers a variety of views to help you analyze your application's performance data. This allows you to identify areas that

### Breakdown by route or path

To view metrics for a specific route or path:

1. Select a project from your dashboard and navigate to the **Speed Insights** tab.
2. From the left-hand panel, select the [metric](/docs/speed-insights/metrics) you want to view data for.
3. From the URL view, select the corresponding tab to view by the **Route** (the actual pages you built), or by **Path** (the URLs reques
4. The information is organized by performance score and sorted by data points. Scroll the list to view more all paths or routes, or clic
5. You can also edit the [timeframe](/docs/analytics/using-web-analytics#specifying-a-timeframe) and [environment](/docs/analytics/using-

### Breakdown by HTML elements

To view a detailed breakdown of the performance of individual HTML elements on your site:

1. Select a project from your dashboard and navigate to the **Speed Insights** tab.
2. From the left-hand panel, select the [metric](/docs/speed-insights/metrics) you want to view data for. HTML element attribution is onl
   - **Interaction to Next Paint** (INP)
   - **First Input Delay** (FID)
   - **Cumulative Layout Shift** (CLS)
   - **Largest Contentful Paint** (LCP)
3. From the URL view, select the **Selectors** tab.
4. The information is organized by performance score and sorted by data points. Scroll the list to view more all elements, or click the *
5. You can also edit the [timeframe](/docs/analytics/using-web-analytics#specifying-a-timeframe) and [environment](/docs/analytics/using-

This view is particularly useful for identifying specific elements that may be causing performance issues.

### Breakdown by country

This view is helpful for identifying regions where your application may be underperforming.

To view a geographical breakdown of your application's performance:

1. Select a project from your dashboard and navigate to the **Speed Insights** tab.
2. From the left-hand panel, select the [metric](/docs/speed-insights/metrics) you want to view data for.
3. Scroll down to the **Countries** section.
4. The map is colored based on the experience metric per country. Click on a country to view more detailed data.

## Disabling Speed Insights

You may want to disable Speed Insights in your project if you find you no longer need it. You can disable Speed Insights from within the

> **💡 Note:** If you transfer a project with Speed Insights enabled from a Hobby team to a
> Pro plan, it will continue to be enabled but with increased limits, as
> documented in the [pricing docs](/docs/speed-insights/limits-and-pricing).
> This means that Speed Insights will be added to your Pro plan invoice
> automatically.

1. Select a project from your [dashboard](/dashboard).
2. Navigate to the **Speed Insights** tab.
3. Click on the ellipsis on the top-right of the Speed Insights page and select **Disable Speed Insights**.

When you disable Speed Insights in the middle of your billing cycle, it will not be removed instantly. Instead it will stop collecting ne

> **💡 Note:** If you are on an Enterprise plan, check your contract entitlements as you may
> have custom limits included. If you have any questions about your
> billing/contract regarding Speed Insights you can reach out to your Customer
> Success Manager (CSM) or Account Executive (AE) for further clarification.

## Identifying if Speed Insights is enabled

If you have many projects on your Vercel account and are not sure which of them has Speed Insights enabled, you can see this from the [da

If Speed Insights is not enabled, then the circle will be gray, with the speed insights logo. For example:

If Speed Insights is enabled but no data points have been collected yet then it will show an empty circle, like the below:

If Speed Insights is enabled and data points have been collected then the circle will be colored with a number inside, similar to the bel

---
title: "Spend Management"
description: "Learn how to get notified about your account spend and configure a webhook."
last_updated: "2026-01-16T02:19:35.649Z"
source: "https://vercel.com/docs/spend-management"
---

# Spend Management

Spend management is a way for you to notify or to automatically take action on your account when your team hits a [set spend amount](#wha

- [Receive a notification](/docs/spend-management#managing-alert-threshold-notifications)
- [Trigger a webhook](/docs/spend-management#configuring-a-webhook)
- [Pause the production deployment of all your projects](/docs/spend-management#pausing-projects)

> **⚠️ Warning:** Setting a spend amount does not automatically stop usage. If you want to pause
> all your projects at a certain amount, you must [enable the
> option](#pausing-projects).

The spend amount is set per billing cycle.

Setting the amount halfway through a billing cycle considers your current spend. You can increase or decrease your spend amount as needed

## What does Spend Management include?

The spend amount that you set covers [metered resources](/docs/limits#additional-resources) that go beyond your Pro plan [credits and usa

It **does not** include seats, integrations (such as Marketplace), or separate [add-ons](/docs/pricing#pro-plan-add-ons), which Vercel ch

### How Vercel checks your spend amount

Vercel checks your metered resource usage often to determine if you are approaching or have exceeded your spend amount. This check happen

## Managing your spend amount

1. To enable spend management, you must have an [Owner](/docs/rbac/access-roles#owner-role) or [Billing](/docs/rbac/access-roles#billing-
2. From your team's [dashboard](/dashboard), select the **Settings** tab
3. Select **Billing** from the list
4. Under **Spend Management**, toggle the switch to enabled:
5. Set the amount in USD at which you would like to receive a notification or trigger an action
6. Select the action(s) to happen when your spend amount is reached: [pause all your projects](#pausing-projects), [send notifications](#

## Managing alert threshold notifications

When you set a spend amount, Vercel automatically enables web and email notifications for your team. These get triggered when spending on

1. You must have an [Owner](/docs/rbac/access-roles#owner-role) or [Billing](/docs/rbac/access-roles#billing-role) role on your [Pro](/do
2. From your team's [dashboard](/dashboard), select the **Settings** tab
3. Select **My Notifications** from the list
4. Under **Team**, ensure that **Spend Management** is selected
5. Select the  icon and select the thresholds for which you would like to receive web and email notification, as described in [Notificati
6. Repeat the previous step for the Web, Email, and SMS notification sections

> **💡 Note:** Following these steps only configures  notifications. Team members
> with the Owner or Billing role can configure their own preferences

### SMS notifications

In addition to web and email notifications, you can enable SMS notifications for Spend Management. They are only triggered when you reach

To enable SMS notifications:

1. You must have an [Owner](/docs/rbac/access-roles#owner-role) or [Billing](/docs/rbac/access-roles#billing-role) role on your [Pro](/do
2. Set your [spend amount](#managing-your-spend-amount)
3. From your team's [dashboard](/dashboard), select the **Settings** tab
4. Select **My Notifications** from the list, scroll to **SMS** at the bottom of the page and toggle the switch to Enabled. If your perso
5. Under **Team**, ensure that **Spend Management** is selected
6. Enter your phone number and follow the steps to verify it

## Pausing projects

Vercel provides an option to automatically pause the production deployment for all of your projects when your spend amount is reached.

1. In the **Spend Management** section of your team's settings, enable and set your [spend amount](#managing-your-spend-amount)
2. Ensure the **Pause production deployment** switch is **Enabled**
3. Confirm the action by entering the team name and select **Continue**. Your changes save automatically
4. When your team reaches the spend amount, Vercel automatically pauses the production deployment for **all projects** on your team

When visitors access your production deployment while it is paused, they will see a [503 DEPLOYMENT_PAUSED error](/docs/errors/DEPLOYMEN

### Unpausing projects

Projects need to be resumed on an individual basis, either [through the dashboard](/docs/projects/overview#resuming-a-project) or the [Ve

Projects won't automatically unpause if you increase the spend amount, you must resume each project manually.

## Configuring a webhook

You can configure a webhook URL to trigger events such as serving a static version of your site, [pausing a project](/docs/projects/overv

Vercel will send a [HTTPS POST request](#webhook-payload) to the URL that you provide when the following events happen:

- [When a spend amount reaches 100%](#spend-amount)
- [At the end of your billing cycle](#end-of-billing-cycle)

To configure a webhook for spend management:

1. In the **Spend Management** section of your team's settings, set your [spend amount](#managing-your-spend-amount)
2. Enter the webhook URL for the endpoint that will receive a POST request. In order to be accessible, make sure your endpoints are publi
3. Secure your webhooks by comparing the [`x-vercel-signature`](/docs/headers/request-headers#x-vercel-signature) request header to the S

### Webhook payload

The webhook URL receives an HTTP POST request with the following JSON payload for each event:

#### Spend amount

Sent when the team hits 50%, 75%, and 100% of their spend amount. For budgets created before September 2025, this is only sent at 100%.

| Parameters | Type | Description |
| --- | --- | --- |
| `budgetAmount` | | The [spend amount](/docs/spend-management#managing-your-spend-amount) that you have set |
| `currentSpend` | | The [total cost](/docs/spend-management#managing-your-spend-amount) that your team [has accrued](/docs/spend |
| `teamId` | | Your Vercel Team ID |
| `thresholdPercent` | | The percentage of the total budget amount for the threshold that triggered this alert |

```json filename="webhook-payload.json"
{
  "budgetAmount": 500,
  "currentSpend": 500,
  "teamId": "team_jkT8yZ3oE1u6xLo8h6dxfNc3",
  "thresholdPercent": 100
}
```

### End of billing cycle

Sent when the billing cycle ends. You can use this event to resume paused projects.

| Parameters | Type | Description |
| --- | --- | --- |
| `teamId` | | Your Vercel Team ID |
| `type` | | The type of event |

```json filename="webhook-payload.json"
{
  "teamId": "team_jkT8yZ3oE1u6xLo8h6dxfNc3",
  "type": "endOfBillingCycle"
}
```

## Spend Management activity

Vercel displays all spend management activity in the **Activity** tab of your [team's dashboard](/docs/observability/activity-log). This

## More resources

For more information on Vercel's pricing, guidance on optimizing consumption, and invoices, see the following resources:

- [How are resources used on Vercel?](/docs/pricing/how-does-vercel-calculate-usage-of-resources)
- [Manage and optimize usage](/docs/pricing/manage-and-optimize-usage)
- [Understanding my invoice](/docs/pricing/understanding-my-invoice)
- [Spend limits for Vercel](https://youtu.be/-_vpoayWTps?si=Jv6b8szx68lVHGYz)

---
title: "Vercel Storage overview"
description: "Store large files and global configuration with Vercel"
last_updated: "2026-01-16T02:19:35.666Z"
source: "https://vercel.com/docs/storage"
---

# Vercel Storage overview

Vercel offers a suite of managed, serverless storage products that integrate with your frontend framework.

- [**Vercel Blob**](/docs/storage/vercel-blob): Large file storage
- [**Vercel Edge Config**](/docs/edge-config): Global, low-latency data store

You can also find storage solutions in the [Vercel Marketplace](https://vercel.com/marketplace/category/storage).

- Explore [Marketplace Redis (KV) integrations](https://vercel.com/marketplace?category=storage\&search=redis)
- Explore [Marketplace Postgres integrations](https://vercel.com/marketplace?category=storage\&search=postgres)

## Choosing a storage product

The right storage solution depends on your needs for latency, durability, and consistency. This table summarizes the key differences:

| Product | Reads | Writes | Use Case | Limits |
| --- | --- | --- | --- | --- |
| [Blob](/docs/storage/vercel-blob) | Fast | Milliseconds | Large, content-addressable files ("blobs") | [Learn more](/docs/storag |
| [Edge Config](/docs/edge-config) | Ultra-fast | Seconds | Runtime configuration (e.g., feature flags) | [Learn more](/docs/edge-c |

See [best practices](#best-practices) for optimizing your storage usage.

## Vercel Blob

Vercel Blob offers optimized storage for images, videos, and other files.

You should use Vercel Blob if you need to:

- **Store images**: For example, storing user avatars or product images
- **Store videos**: For example, storing user-generated video content

### Explore Vercel Blob

- [Overview](/docs/storage/vercel-blob)
- [Quickstart](/docs/storage/vercel-blob/server-upload)

## Edge Config

An Edge Config is a global data store that enables you to read data in the region closest to the user without querying an external databa

You should use Edge Config if you need to:

- **Fetch data at ultra-low latency**: For example, you should store feature flags in an Edge Config store.
- **Store data that is read often but changes rarely**: For example, you should store critical redirect URLs in an Edge Config store.
- **Read data in every region**: Edge Config data is actively replicated to all regions in the Vercel CDN.

### Explore Edge Config

- [Overview](/docs/edge-config)
- [Quickstart](/docs/edge-config/get-started)
- [Limits & Pricing](/docs/edge-config/edge-config-limits)

## Best practices

Follow these best practices to get the most from your storage:

### Locate your data close to your functions

Deploy your databases in [regions](/docs/regions) closest to your Functions. This minimizes network roundtrips and keeps response times l

### Optimize for high cache hit rates

Vercel's CDN caches content in every region globally. Cache data fetched from your data store on the CDN using [cache headers](/docs/cdn-

[Incremental Static Regeneration](/docs/concepts/incremental-static-regeneration/overview) sets up caching headers automatically and stor

You can also configure cache-control headers manually with [Vercel Functions](/docs/cdn-cache#using-vercel-functions) to cache responses

## Transferring your store

You can bring your Blob or Edge Config stores along with your account as you upgrade from Hobby to Pro, or downgrade from Pro to Hobby. T

1. Navigate to the [dashboard](/dashboard) and select the **Storage** tab
2. Select the store that you would like to transfer
3. Select **Settings**, then select **Transfer Store**
4. Select a destination account or team. If you're upgrading to Pro, select your new Pro team. If downgrading, select your Hobby team

When successful, you'll be taken to the **Storage** tab of the account or team you transferred the store to.

---
title: "Instrumentation"
description: "Learn how to instrument your application to understand performance and infrastructure details."
last_updated: "2026-01-16T02:19:35.700Z"
source: "https://vercel.com/docs/tracing/instrumentation"
---

# Instrumentation

Observability is crucial for understanding and optimizing the behavior and performance of your app. Vercel supports OpenTelemetry instrum

## Getting started

To get started, install the following packages:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i @opentelemetry/api @vercel/otel
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @opentelemetry/api @vercel/otel
    ```

    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @opentelemetry/api @vercel/otel
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i @opentelemetry/api @vercel/otel
    ```
  </Code>
</CodeBlock>

Next, create a `instrumentation.ts` (or `.js`) file in the root directory of the project, or, on Next.js [it must be placed](https://next

```ts filename="instrumentation.ts" framework=nextjs-app
import { registerOTel } from '@vercel/otel';

export function register() {
  registerOTel({ serviceName: 'your-project-name' });
}
// NOTE: You can replace `your-project-name` with the actual name of your project
```

```js filename="instrumentation.js" framework=nextjs-app
import { registerOTel } from '@vercel/otel';

export function register() {
  registerOTel({ serviceName: 'your-project-name' });
}
// NOTE: You can replace `your-project-name` with the actual name of your project
```

```ts filename="instrumentation.ts" framework=nextjs
import { registerOTel } from '@vercel/otel';

export function register() {
  registerOTel({ serviceName: 'your-project-name' });
}
// NOTE: You can replace `your-project-name` with the actual name of your project
```

```js filename="instrumentation.js" framework=nextjs
import { registerOTel } from '@vercel/otel';

export function register() {
  registerOTel({ serviceName: 'your-project-name' });
}
// NOTE: You can replace `your-project-name` with the actual name of your project
```

```ts filename="instrumentation.ts" framework=other
import { registerOTel } from '@vercel/otel';

registerOTel({ serviceName: 'your-project-name' });
// NOTE: You can replace `your-project-name` with the actual name of your project
```

```js filename="instrumentation.js" framework=other
import { registerOTel } from '@vercel/otel';

registerOTel({ serviceName: 'your-project-name' });
// NOTE: You can replace `your-project-name` with the actual name of your project
```

## Configuring context propagation

Context propagation connects operations across service boundaries so you can trace a request through your entire system. When your app ca

Without context propagation, each service generates isolated spans you can't connect. With it, you see exactly how a request flows throug

For more details on how context propagation works, see the [OpenTelemetry context propagation documentation](https://opentelemetry.io/doc

### For outgoing requests

You can configure context propagation by configuring the `fetch` option in the `instrumentationConfig` option.

```ts filename="instrumentation.ts" framework=nextjs-app
import { registerOTel } from '@vercel/otel';

export function register() {
  registerOTel({
    serviceName: `your-project-name`,
    instrumentationConfig: {
      fetch: {
        // This URLs will have the tracing context propagated to them.
        propagateContextUrls: [
          'your-service-domain.com',
          'your-database-domain.com',
        ],
        // This URLs will not have the tracing context propagated to them.
        dontPropagateContextUrls: [
          'some-third-party-service-domain.com',
        ],
        // This URLs will be ignored and will not be traced.
        ignoreUrls: ['my-internal-private-tool.com'],
      },
    },
  });
```
```

```
}
// NOTE: You can replace `your-project-name` with the actual name of your project
```

```js filename="instrumentation.js" framework=nextjs-app
import { registerOTel } from '@vercel/otel';

export function register() {
  registerOTel({
    serviceName: `your-project-name`,
    instrumentationConfig: {
      fetch: {
        // This URLs will have the tracing context propagated to them.
        propagateContextUrls: [
          'your-service-domain.com',
          'your-database-domain.com',
        ],
        // This URLs will not have the tracing context propagated to them.
        dontPropagateContextUrls: [
          'some-third-party-service-domain.com',
        ],
        // This URLs will be ignored and will not be traced.
        ignoreUrls: ['my-internal-private-tool.com'],
      },
    },
  });
}
// NOTE: You can replace `your-project-name` with the actual name of your project
```

```ts filename="instrumentation.ts" framework=nextjs
import { registerOTel } from '@vercel/otel';

export function register() {
  registerOTel({
    serviceName: `your-project-name`,
    instrumentationConfig: {
      fetch: {
        // This URLs will have the tracing context propagated to them.
        propagateContextUrls: [
          'your-service-domain.com',
          'your-database-domain.com',
        ],
        // This URLs will not have the tracing context propagated to them.
        dontPropagateContextUrls: [
          'some-third-party-service-domain.com',
        ],
        // This URLs will be ignored and will not be traced.
        ignoreUrls: ['my-internal-private-tool.com'],
      },
    },
  });
}
// NOTE: You can replace `your-project-name` with the actual name of your project
```

```js filename="instrumentation.js" framework=nextjs
import { registerOTel } from '@vercel/otel';

export function register() {
  registerOTel({
    serviceName: `your-project-name`,
    instrumentationConfig: {
      fetch: {
        // This URLs will have the tracing context propagated to them.
        propagateContextUrls: [
          'your-service-domain.com',
          'your-database-domain.com',
        ],
        // This URLs will not have the tracing context propagated to them.
        dontPropagateContextUrls: [
          'some-third-party-service-domain.com',
        ],
        // This URLs will be ignored and will not be traced.
        ignoreUrls: ['my-internal-private-tool.com'],
      },
    },
  });
}
// NOTE: You can replace `your-project-name` with the actual name of your project
```

```ts filename="instrumentation.ts" framework=other
import { registerOTel } from '@vercel/otel';

registerOTel({
  serviceName: `your-project-name`,
  instrumentationConfig: {
    fetch: {
      // This URLs will have the tracing context propagated to them.
      propagateContextUrls: [
        'your-service-domain.com',
        'your-database-domain.com',
      ],
      // This URLs will not have the tracing context propagated to them.
      dontPropagateContextUrls: [
        'some-third-party-service-domain.com',
      ],
      // This URLs will be ignored and will not be traced.
      ignoreUrls: ['my-internal-private-tool.com'],
```

```
    },
  },
});
// NOTE: You can replace `your-project-name` with the actual name of your project
```

```js filename="instrumentation.js" framework=other
import { registerOTel } from '@vercel/otel';

registerOTel({
  serviceName: `your-project-name`,
  instrumentationConfig: {
    fetch: {
      // This URLs will have the tracing context propagated to them.
      propagateContextUrls: [
        'your-service-domain.com',
        'your-database-domain.com',
      ],
      // This URLs will not have the tracing context propagated to them.
      dontPropagateContextUrls: [
        'some-third-party-service-domain.com',
      ],
      // This URLs will be ignored and will not be traced.
      ignoreUrls: ['my-internal-private-tool.com'],
    },
  },
});
// NOTE: You can replace `your-project-name` with the actual name of your project
```

### From incoming requests

Next.js 13.4+ supports automatic OpenTelemetry context propagation for incoming requests. For other frameworks, that do not support autom

```ts filename="api-handler.ts"
import { propagation, context, trace } from "@opentelemetry/api";

const tracer = trace.getTracer('custom-tracer');

// This function injects the inbound context into the request handler
function injectInboundContext(f: (request: Request) => Promise<Response>): (request: Request) => Promise<Response> {
  return (req) => {
    const c = propagation.extract(context.active(), Object.fromEntries(req.headers))
    return context.with(c, async () => {
      return await f(req);
    })
  }
}

export const GET = injectInboundContext(async (req: Request) => {
  const span = tracer.startSpan('your-operation-name');
  // The above ^ span will be automatically attached to incoming tracing context (if any)
  try {
    // Your operation logic here
    span.setAttributes({
      'custom.attribute': 'value',
    });
    return new Response('Hello, world!');
  } finally {
    span.end();
  }
});
```

## Adding custom spans

After installing `@vercel/otel`, you can add custom spans to your traces to capture additional visibility into your application. Custom s

Use the `@opentelemetry/api` package to instrument specific operations:

```ts filename="custom-span.ts" {3, 6, 9-11, 13}
import { trace } from '@opentelemetry/api';

const tracer = trace.getTracer('custom-tracer');

async function performOperation() {
  const span = tracer.startSpan('operation-name');
  try {
    // Your operation logic here
    span.setAttributes({
      'custom.attribute': 'value',
    });
  } finally {
    span.end();
  }
}
```

Custom spans from functions using the [Edge runtime](/docs/functions/runtimes/edge) are not supported.

## OpenTelemetry configuration options

For the full list of configuration options, see the [@vercel/otel documentation](https://github.com/vercel/otel/blob/main/packages/otel/R

## Limitations

- If your app uses manual OpenTelemetry SDK configuration without the usage of `@vercel/otel`, you will not be able to use [Session Traci

---

title: "Tracing"
description: "Learn how to trace your application to understand performance and infrastructure details."
last_updated: "2026-01-16T02:19:35.751Z"
source: "https://vercel.com/docs/tracing"
---

# Tracing

In observability, tracing is the process of collecting and analyzing how a request or operation flows through your application and throug

You can think of a trace as the story of a single request:

**Request arrives at Vercel CDN -> Middleware executes -> Function handler processes request -> Database query runs -> Response returns t

Each step in this process is a **span**. A span is a single unit of work in a trace. Spans are used to measure the performance of each st

## Automatic instrumentation

Vercel automatically instruments your application without needing any additional code changes. When you have set up [Trace Drains](/docs/

- **Vercel infrastructure**: You'll be able to view spans showing the lifecycle of each invocation of your Vercel Functions and how it mo
- **Outbound HTTP calls**: The HTTP requests made from your function will be displayed as fetch spans, displaying information on the leng

For additional tracing, such as framework spans, you can install the [@vercel/otel](/docs/tracing/instrumentation) package to use the Ope

## Session tracing

To visualize traces in your dashboard, you need to enable session tracing using the Vercel toolbar. Session tracing captures infrastructu

You can initiate a session trace in two ways:

- **Page Trace**: Trace a single page load to see how that specific request flows through your application.
- **Session Trace**: Start an ongoing trace that captures all requests from your browser until you stop it or clear cookies.

For detailed instructions on starting traces, managing active sessions, and viewing previous traces, see the [Session Tracing](/docs/trac

## Using OpenTelemetry

Vercel uses [OpenTelemetry](https://opentelemetry.io/), an open standard for collecting traces from your application. In order to capture

See the [Instrumentation](/docs/tracing/instrumentation) guide to set up OpenTelemetry for your project.

## Viewing traces in the dashboard

Once you have enabled session tracing, you can visualize traces in your dashboard:

1. Select your team from the scope selector and select your project.
2. Select the [**Logs** tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Flogs\&title=Go+to+Logs).
3. Use the tracing icon  in the filter bar to filter to traces. You can filter traces using [all the same filters available](/docs/runtim
4. Find the request you want to view traces for and click the **Trace** button at the bottom of the request details panel. This will open

### Anatomy of a trace

When you view a trace in the dashboard, you see a timeline visualization of how a request flows through your application and Vercel's inf

When session tracing is enabled, your traces display the following types of spans:

| Span type | Visual appearance | Description |
| --- | --- | --- |
| **Infrastructure spans** | Black and white with a triangle icon | Capture how requests move through Vercel's infrastructure, including |
| **Fetch spans** | Green | Represent HTTP requests made from your functions. |
| **Framework spans** | Blue | Appear when you [instrument your application](/docs/tracing/instrumen |
| **Custom spans** | Blue | [Custom instrumentation](/docs/tracing#adding-custom-spans) you can a |

To view details of a span, click on the span in the trace. The sidebar will display the span's details. For infrastructure spans, a "what

To view trace spans in more detail, click and drag to zoom in on a specific area of the trace. You can also use the zoom controls in the

## Exporting traces to a third party

You can export traces to a third party observability provider using [Vercel Drains](/docs/drains). This can be done either by sending tra

See the [Vercel Drains](/docs/drains) page to learn how to set up a Drain to export traces to a third party observability provider.

### Using custom OpenTelemetry setup with Sentry

If you want to trace your Vercel application using `@vercel/otel` while also using Sentry SDK v8+, you need to configure them to work tog

To use both together, configure Sentry to work with your custom OpenTelemetry setup by following the [Sentry custom setup documentation](

> **💡 Note:** **Using Vercel OTel instead of Sentry:** If you prefer to use Vercel's
> OpenTelemetry setup instead of Sentry's OTel instrumentation, add
> `skipOpenTelemetrySetup: true` to your Sentry initialization in your
> `instrumentation.ts` file. This resolves conflicts between Vercel's OTel and
> Sentry v8+ that can prevent traces from reaching downstream providers.

## More resources

- [Using Vercel Drains](/docs/drains)
- [Trace Drains](/docs/drains/reference/traces)
- [Learn about the Vercel toolbar](/docs/vercel-toolbar)
- [Session Tracing](/docs/tracing/session-tracing)

---
title: "Session tracing"
description: "Learn how to trace your sessions to understand performance and infrastructure details."
last_updated: "2026-01-16T02:19:35.829Z"
source: "https://vercel.com/docs/tracing/session-tracing"
---

# Session tracing

With session tracing, you can use the Vercel toolbar to trace **your** sessions and view the corresponding spans in the logs dashboard. T

A session trace is initiated through the Vercel toolbar, either through a [Page Trace](/docs/tracing/session-tracing#run-a-page-trace) or

## Prerequisites

- A Vercel account. If you don't have one, you can [sign up for free](https://vercel.com/signup).
- A Vercel project that is deployed to preview or production. You cannot create and run a session trace for a local deployment.
- [The toolbar enabled](/docs/vercel-toolbar/in-production-and-localhost) in your preview or production environment.

## Run a session trace

1. In the Vercel toolbar on your deployment, click (or search for) **Tracing**.
2. Select **Start Tracing Session**. Once enabled, the page will reload to activate the session trace.
3. From the toolbar, you can then using the **Tracing** icon to select any of the following options:
   - **View Page Trace**: View the trace for the current page. Selecting this option will open the trace for the current page in a new ta
   - **View Session Traces**: View all traced requests from your active session. Selecting this option will open the dashboard to the **L
   - **Stop Tracing Session**: Stop tracing the current session.
   - **Restart Tracing Session**: Restart tracing the current session.

## Run a page trace

To run a trace on a specific page, you can run a **Page Trace**:

1. In your deployment, open the Vercel toolbar and scroll down to **Tracing**.
2. Select **Run Page Trace**.
3. The page will reload, and a toast will indicate the status of the trace. Once the trace has propagated, the toast will indicate that t
4. Click the toast to view the trace in a new browser tab under the **Logs** tab of the dashboard.

## View previous session traces

1. In the Vercel toolbar on your deployment, click (or search for) **Tracing**.
2. Select **View Previous Session Traces**.
3. The dashboard will open to the **Logs** tab, filtered to the session ID, and the tracing filter applied - indicated by the Traces icon

You can filter traces using [all the same filters available](/docs/runtime-logs#log-filters) in the **Logs** tab of the dashboard. To vie

## Usage and pricing

Tracing is available on all plans with a limit up to **1 million spans per month, per team**.

| Plan | Monthly span limit per team |
| --- | --- |
| Hobby | 1 million |
| Pro | 1 million |
| Enterprise | 1 million |

## Limitations

Custom spans from functions using the [Edge runtime](/docs/functions/runtimes/edge) are not supported.

## More resources

- [Learn about the Vercel toolbar](/docs/vercel-toolbar)
- [Explore Observability on Vercel](/docs/observability)

---
title: "Two-factor Authentication"
description: "Learn how to configure two-factor authentication for your Vercel account."
last_updated: "2026-01-16T02:19:35.836Z"
source: "https://vercel.com/docs/two-factor-authentication"
---

# Two-factor Authentication

To add an additional layer of security to your Vercel account, you can enable two-factor authentication (2FA).
This feature requires you to provide a second form of verification when logging in to your account. There are two
methods available for 2FA on Vercel:

- **Authenticator App**: Use an authenticator app like Google Authenticator to generate a time-based one-time password (TOTP).
- **Passkey**: Authenticate using any WebAuthN compatible device, such as a security key or biometric key.

## Enabling Two-factor Authentication

1. Navigate to your [account settings](https://vercel.com/account/settings/authenticate#two-factor-authentication) on Vercel
2. Toggle the switch to enable 2FA
3. Set up your 2FA methods
4. Confirm your setup
5. Save your recovery codes

### Configuring an Authenticator App (TOTP)

Scan the QR code with your authenticator app or manually enter the provided key.
Once added, enter the generated 6-digit code to verify your setup.

### Configuring a Passkey

See the [Login with passkeys](/docs/accounts/create-an-account#login-with-passkeys) for more information on setting up a security key or

### Recovery Codes

After setting up two-factor authentication (2FA), you will be prompted to save your recovery codes.
Store these codes in a safe place, as they can be used to access your account if you lose access to your 2FA methods.

Each recovery code can only be used once, and you can generate a new set of codes at any time.

## Enforcing Two-Factor Authentication

Teams can enforce two-factor authentication (2FA) for all members. Once enabled, team members must configure 2FA before accessing team re
Visit the [Two-Factor Enforcement](/docs/two-factor-enforcement) documentation for more information on how to enforce 2FA for your team.


---
title: "Two-factor enforcement"
description: "Learn how to enforce two-factor authentication (2FA) for your Vercel team members to enhance security."
last_updated: "2026-01-16T02:19:35.821Z"
source: "https://vercel.com/docs/two-factor-enforcement"
---

# Two-factor enforcement

To enhance the security of your Vercel team, you can enforce two-factor authentication (2FA) for all team members. When enabled, members

What to expect:

- Team members will not be able to access team resources until they have 2FA enabled.
- Team members will continue to occupy a team seat.
- Any CI/CD pipeline tokens associated with users without 2FA will cease to work.
- Managed accounts, like service accounts or bots, will also need to have 2FA enabled.
- Members without 2FA will be prompted to enable it when visiting the team dashboard.
- Builds will fail for members without 2FA.
- Notifications will continue to be sent to members without 2FA.

For more information on how to set up two-factor authentication for your account, see the [two-factor authentication](/docs/two-factor-au

## Viewing team members' 2FA status

Team owners can view the two-factor authentication status of all team members in the [team members page](/docs/rbac/managing-team-members

## Enabling team 2FA enforcement

Before enabling 2FA enforcement for your team, you must have 2FA enabled on your own account. To prevent workflow disruptions, we recomme

Steps to follow:

1. Go to **Team Settings** then **Security & Privacy** and scroll to **Two-Factor Authentication Enforcement**
2. Toggle the switch to enforce 2FA
3. Click the **Save** button to confirm the action


---
title: "Blocked Blob Store"
description: "The Blob Store you are trying to access has been paused."
last_updated: "2026-01-16T02:19:35.753Z"
source: "https://vercel.com/docs/vercel-blob/blocked-store"
---

# Blocked Blob Store

## The Blob Store you are trying to access has been paused

This can happen for one of these reasons:

- the Blob Store reached the usage limits for its plan
- the Blob Store has been paused by the Vercel team

Visit the [Vercel Blob Dashboard](https://vercel.com/dashboard/storage) to check the status of the Blob Store.
If you think Vercel wrongly paused your store, reach out to our support team at https://vercel.com/help.


---
title: "Client Uploads with Vercel Blob"
description: "Learn how to upload files larger than 4.5 MB directly from the browser to Vercel Blob"
last_updated: "2026-01-16T02:19:35.852Z"
source: "https://vercel.com/docs/vercel-blob/client-upload"
---

# Client Uploads with Vercel Blob

In this guide, you'll learn how to do the following:

- Use the Vercel dashboard to create a Blob store connected to a project
- Upload a file using the Blob SDK from a browser

## Prerequisites

Vercel Blob works with any frontend framework. First, install the package:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i @vercel/blob
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @vercel/blob
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @vercel/blob
    ```
  </Code>
  <Code tab="bun">
    ```bash
    ```

```
    bun i @vercel/blob
    ```
  </Code>
</CodeBlock>

- ### Create a Blob store
  Navigate to the [Project](/docs/projects/overview) you'd like to add the blob store to. Select the **Storage** tab, then select the **C

  Under the **Create New** tab, select **Blob** and then the **Continue** button.

  Use the name "Images" and select **Create a new Blob store**. Select the environments where you would like the read-write token to be i

  Once created, you are taken to the Vercel Blob store page.

- ### Prepare your local project
  Since you created the Blob store in a project, we automatically created and added the following Environment Variable to the project for
  - `BLOB_READ_WRITE_TOKEN`
  To use this Environment Variable locally, we recommend pulling it with the Vercel CLI:
  ```bash
  vercel env pull
  ```

When you need to upload files larger than 4.5 MB, you can use client uploads. In this case, the file is sent directly from the client (a

- ### Create a client upload page
  This page allows to upload files to Vercel Blob. The files will go directly from the browser to Vercel Blob without going through your

  Behind the scenes, the upload is done securely by exchanging a token with your server before uploading the file.
  ```tsx filename="src/app/avatar/upload/page.tsx" framework=nextjs-app
  'use client';

  import { type PutBlobResult } from '@vercel/blob';
  import { upload } from '@vercel/blob/client';
  import { useState, useRef } from 'react';

  export default function AvatarUploadPage() {
    const inputFileRef = useRef<HTMLInputElement>(null);
    const [blob, setBlob] = useState<PutBlobResult | null>(null);
    return (
      <>
        <h1>Upload Your Avatar</h1>

        <form
          onSubmit={async (event) => {
            event.preventDefault();

            if (!inputFileRef.current?.files) {
              throw new Error('No file selected');
            }

            const file = inputFileRef.current.files[0];

            const newBlob = await upload(file.name, file, {
              access: 'public',
              handleUploadUrl: '/api/avatar/upload',
            });

            setBlob(newBlob);
          }}
        >
          <input name="file" ref={inputFileRef} type="file" required />
          <button type="submit">Upload</button>
        </form>
        {blob && (
          <div>
            Blob url: <a href={blob.url}>{blob.url}</a>
          </div>
        )}
      </>
    );
  }
  ```
  ```jsx filename="app/avatar/upload/page.jsx" framework=nextjs-app
  'use client';

  import { upload } from '@vercel/blob/client';
  import { useState, useRef } from 'react';

  export default function AvatarUploadPage() {
    const inputFileRef = useRef(null);
    const [blob, setBlob] = useState(null);
    return (
      <>
        <h1>Upload Your Avatar</h1>

        <form
          onSubmit={async (event) => {
            event.preventDefault();

            const file = inputFileRef.current.files[0];

            const newBlob = await upload(file.name, file, {
              access: 'public',
              handleUploadUrl: '/api/avatar/upload',
            });

            setBlob(newBlob);
          }}
        >
          <input
```

```
          name="file"
          ref={inputFileRef}
          type="file"
          accept="image/jpeg, image/png, image/webp"
          required
        />
        <button type="submit">Upload</button>
      </form>
      {blob && (
        <div>
          Blob url: <a href={blob.url}>{blob.url}</a>
        </div>
      )}
    </>
  );
}
```
```tsx filename="pages/avatar/upload.tsx" framework=nextjs
import { type PutBlobResult } from '@vercel/blob';
import { upload } from '@vercel/blob/client';
import { useState, useRef } from 'react';

export default function AvatarUploadPage() {
  const inputFileRef = useRef<HTMLInputElement>(null);
  const [blob, setBlob] = useState<PutBlobResult | null>(null);
  return (
    <>
      <h1>Upload Your Avatar</h1>

      <form
        onSubmit={async (event) => {
          event.preventDefault();

          if (!inputFileRef.current?.files) {
            throw new Error('No file selected');
          }

          const file = inputFileRef.current.files[0];

          const newBlob = await upload(file.name, file, {
            access: 'public',
            handleUploadUrl: '/api/avatar/upload',
          });

          setBlob(newBlob);
        }}
      >
        <input
          name="file"
          ref={inputFileRef}
          type="file"
          accept="image/jpeg, image/png, image/webp"
          required
        />
        <button type="submit">Upload</button>
      </form>
      {blob && (
        <div>
          Blob url: <a href={blob.url}>{blob.url}</a>
        </div>
      )}
    </>
  );
}
```
```jsx filename="pages/avatar/upload.jsx" framework=nextjs
import { upload } from '@vercel/blob/client';
import { useState, useRef } from 'react';

export default function AvatarUploadPage() {
  const inputFileRef = useRef(null);
  const [blob, setBlob] = useState(null);
  return (
    <>
      <h1>Upload Your Avatar</h1>

      <form
        onSubmit={async (event) => {
          event.preventDefault();

          const file = inputFileRef.current.files[0];

          const newBlob = await upload(file.name, file, {
            access: 'public',
            handleUploadUrl: '/api/avatar/upload',
          });

          setBlob(newBlob);
        }}
      >
        <input
          name="file"
          ref={inputFileRef}
          type="file"
          accept="image/jpeg, image/png, image/webp"
          required
        />
        <button type="submit">Upload</button>
      </form>
      {blob && (
```

```
          <div>
            Blob url: <a href={blob.url}>{blob.url}</a>
          </div>
        )}
      </>
    );
  }
```

- ### Create a client upload route
  The responsibility of this client upload route is to:
  1. Generate tokens for client uploads
  2. Listen for completed client uploads, so you can update your database with the URL of the uploaded file for example
  The `@vercel/blob` npm package exposes a helper to implement said responsibilities.
  ```ts filename="src/app/api/avatar/upload/route.ts" framework=nextjs-app
  import { handleUpload, type HandleUploadBody } from '@vercel/blob/client';
  import { NextResponse } from 'next/server';

  export async function POST(request: Request): Promise<NextResponse> {
    const body = (await request.json()) as HandleUploadBody;

    try {
      const jsonResponse = await handleUpload({
        body,
        request,
        onBeforeGenerateToken: async (
          pathname,
          /* clientPayload */
        ) => {
          // Generate a client token for the browser to upload the file
          // Make sure to authenticate and authorize users before generating the token.
          // Otherwise, you're allowing anonymous uploads.

          return {
            allowedContentTypes: ['image/jpeg', 'image/png', 'image/webp'],
            addRandomSuffix: true,
            // callbackUrl: 'https://example.com/api/avatar/upload',
            // optional, `callbackUrl` is automatically computed when hosted on Vercel
            tokenPayload: JSON.stringify({
              // optional, sent to your server on upload completion
              // you could pass a user id from auth, or a value from clientPayload
            }),
          };
        },
        onUploadCompleted: async ({ blob, tokenPayload }) => {
          // Called by Vercel API on client upload completion
          // Use tools like ngrok if you want this to work locally

          console.log('blob upload completed', blob, tokenPayload);

          try {
            // Run any logic after the file upload completed
            // const { userId } = JSON.parse(tokenPayload);
            // await db.update({ avatar: blob.url, userId });
          } catch (error) {
            throw new Error('Could not update user');
          }
        },
      });

      return NextResponse.json(jsonResponse);
    } catch (error) {
      return NextResponse.json(
        { error: (error as Error).message },
        { status: 400 }, // The webhook will retry 5 times waiting for a 200
      );
    }
  }
  ```
  ```js filename="src/app/api/avatar/upload/route.js" framework=nextjs-app
  import { handleUpload } from '@vercel/blob/client';
  import { NextResponse } from 'next/server';

  export async function POST(request) {
    const body = await request.json();

    try {
      const jsonResponse = await handleUpload({
        body,
        request,
        onBeforeGenerateToken: async (pathname /*, clientPayload */) => {
          // Generate a client token for the browser to upload the file
          // Make sure to authenticate and authorize users before generating the token.
          // Otherwise, you're allowing anonymous uploads.

          return {
            allowedContentTypes: ['image/jpeg', 'image/png', 'image/webp'],
            addRandomSuffix: true,
            // callbackUrl: 'https://example.com/api/avatar/upload',
            // optional, `callbackUrl` is automatically computed when hosted on Vercel
            tokenPayload: JSON.stringify({
              // optional, sent to your server on upload completion
              // you could pass a user id from auth, or a value from clientPayload
            }),
          };
        },
        onUploadCompleted: async ({ blob, tokenPayload }) => {
          // Called by Vercel API on client upload completion
          // Use tools like ngrok if you want this to work locally
```

```
        console.log('blob upload completed', blob, tokenPayload);

        try {
          // Run any logic after the file upload completed
          // const { userId } = JSON.parse(tokenPayload);
          // await db.update({ avatar: blob.url, userId });
        } catch (error) {
          throw new Error('Could not update user');
        }
      },
    });

    return NextResponse.json(jsonResponse);
  } catch (error) {
    return NextResponse.json(
      { error: error.message },
      { status: 400 }, // The webhook will retry 5 times waiting for a status 200
    );
  }
}
```

```ts filename="pages/api/avatar/upload.ts" framework=nextjs
import { handleUpload, type HandleUploadBody } from '@vercel/blob/client';
import type { NextApiResponse, NextApiRequest } from 'next';

export default async function handler(
  request: NextApiRequest,
  response: NextApiResponse,
) {
  const body = request.body as HandleUploadBody;

  try {
    const jsonResponse = await handleUpload({
      body,
      request,
      onBeforeGenerateToken: async (
        pathname,
        /* clientPayload */
      ) => {
        // Generate a client token for the browser to upload the file
        // Make sure to authenticate and authorize users before generating the token.
        // Otherwise, you're allowing anonymous uploads.

        return {
          allowedContentTypes: ['image/jpeg', 'image/png', 'image/webp'],
          addRandomSuffix: true,
          // callbackUrl: 'https://example.com/api/avatar/upload',
          // optional, `callbackUrl` is automatically computed when hosted on Vercel
          tokenPayload: JSON.stringify({
            // optional, sent to your server on upload completion
            // you could pass a user id from auth, or a value from clientPayload
          }),
        };
      },
      onUploadCompleted: async ({ blob, tokenPayload }) => {
        // Called by Vercel API on client upload completion
        // Use tools like ngrok if you want this to work locally

        console.log('blob upload completed', blob, tokenPayload);

        try {
          // Run any logic after the file upload completed
          // const { userId } = JSON.parse(tokenPayload);
          // await db.update({ avatar: blob.url, userId });
        } catch (error) {
          throw new Error('Could not update user');
        }
      },
    });

    return response.status(200).json(jsonResponse);
  } catch (error) {
    // The webhook will retry 5 times waiting for a 200
    return response.status(400).json({ error: (error as Error).message });
  }
}
```

```js filename="pages/api/avatar/upload.js" framework=nextjs
import { handleUpload } from '@vercel/blob/client';

export default async function handler(request, response) {
  const body = await request.json();

  try {
    const jsonResponse = await handleUpload({
      body,
      request,
      onBeforeGenerateToken: async (pathname /*, clientPayload */) => {
        // Generate a client token for the browser to upload the file
        // Make sure to authenticate and authorize users before generating the token.
        // Otherwise, you're allowing anonymous uploads.

        return {
          allowedContentTypes: ['image/jpeg', 'image/png', 'image/webp'],
          addRandomSuffix: true,
          // callbackUrl: 'https://example.com/api/avatar/upload',
          // optional, `callbackUrl` is automatically computed when hosted on Vercel
          tokenPayload: JSON.stringify({
            // optional, sent to your server on upload completion
            // you could pass a user id from auth, or a value from clientPayload
```

```
          }),
        };
      },
      onUploadCompleted: async ({ blob, tokenPayload }) => {
        // Called by Vercel API on client upload completion
        // Use tools like ngrok if you want this to work locally

        console.log('blob upload completed', blob, tokenPayload);

        try {
          // Run any logic after the file upload completed
          // const { userId } = JSON.parse(tokenPayload);
          // await db.update({ avatar: blob.url, userId });
        } catch (error) {
          throw new Error('Could not update user');
        }
      },
    });

    return response.status(200).json(jsonResponse);
  } catch (error) {
    // The webhook will retry 5 times waiting for a 200
    return response.status(400).json({ error: error.message });
  }
}
```
```ts filename="api/avatar/upload.ts" framework=other
import { handleUpload, type HandleUploadBody } from '@vercel/blob/client';

export default async function handler(request: Request) {
  const body = (await request.json()) as HandleUploadBody;

  try {
    const jsonResponse = await handleUpload({
      body,
      request,
      onBeforeGenerateToken: async (
        pathname,
        /* clientPayload */
      ) => {
        // Generate a client token for the browser to upload the file
        // Make sure to authenticate and authorize users before generating the token.
        // Otherwise, you're allowing anonymous uploads.

        return {
          allowedContentTypes: ['image/jpeg', 'image/png', 'image/webp'],
          addRandomSuffix: true,
          // callbackUrl: 'https://example.com/api/avatar/upload',
          // optional, `callbackUrl` is automatically computed when hosted on Vercel
          tokenPayload: JSON.stringify({
            // optional, sent to your server on upload completion
            // you could pass a user id from auth, or a value from clientPayload
          }),
        };
      },
      onUploadCompleted: async ({ blob, tokenPayload }) => {
        // Called by Vercel API on client upload completion
        // Use tools like ngrok if you want this to work locally

        console.log('blob upload completed', blob, tokenPayload);

        try {
          // Run any logic after the file upload completed
          // const { userId } = JSON.parse(tokenPayload);
          // await db.update({ avatar: blob.url, userId });
        } catch (error) {
          throw new Error('Could not update user');
        }
      },
    });

    return Response.json(jsonResponse);
  } catch (error) {
    return Response.json(
      { error: (error as Error).message },
      { status: 400 }, // The webhook will retry 5 times waiting for a 200
    );
  }
}
```
```js filename="api/avatar/upload.js" framework=other
import { handleUpload } from '@vercel/blob/client';

export default async function handler(request) {
  const body = await request.json();

  try {
    const jsonResponse = await handleUpload({
      body,
      request,
      onBeforeGenerateToken: async (pathname /*, clientPayload */) => {
        // Generate a client token for the browser to upload the file
        // Make sure to authenticate and authorize users before generating the token.
        // Otherwise, you're allowing anonymous uploads.

        return {
          allowedContentTypes: ['image/jpeg', 'image/png', 'image/webp'],
          addRandomSuffix: true,
          // callbackUrl: 'https://example.com/api/avatar/upload',
          // optional, `callbackUrl` is automatically computed when hosted on Vercel
```

```
            tokenPayload: JSON.stringify({
              // optional, sent to your server on upload completion
              // you could pass a user id from auth, or a value from clientPayload
            }),
          };
        },
        onUploadCompleted: async ({ blob, tokenPayload }) => {
          // Called by Vercel API on client upload completion
          // Use tools like ngrok if you want this to work locally

          console.log('blob upload completed', blob, tokenPayload);

          try {
            // Run any logic after the file upload completed
            // const { userId } = JSON.parse(tokenPayload);
            // await db.update({ avatar: blob.url, userId });
          } catch (error) {
            throw new Error('Could not update user');
          }
        },
      });

      return Response.json(jsonResponse);
    } catch (error) {
      return Response.json(
        { error: error.message },
        { status: 400 }, // The webhook will retry 5 times waiting for a 200
      );
    }
  }
  ```

## Testing your page

- ### Run your application locally
  Run your application locally and visit `/avatar/upload` to upload the file to your store. The browser will display the unique URL creat

- ### Review the Blob object metadata
  - Go to the Vercel Project where you created the store
  - Select the **Storage** tab and select your new store
  - Paste the blob object URL returned in the previous step in the **Blob URL** input box in the **Browser** section and select **Lookup*
  - The following blob object metadata will be displayed: file name, path, size, uploaded date, content type and HTTP headers
  - You also have the option to download and delete the file from this page

You have successfully uploaded an object to your Vercel Blob store and are able to review it's metadata, download, and delete it from you

### `onUploadCompleted` callback behavior

The `onUploadCompleted` callback is called by Vercel API when a client upload completes. For this to work, `@vercel/blob` computes the co

We use the following environment variables to compute the callback URL:

- `VERCEL_BRANCH_URL` in preview environments
- `VERCEL_URL` in preview environments where `VERCEL_BRANCH_URL` is not set
- `VERCEL_PROJECT_PRODUCTION_URL` in production environments

These variables are automatically set by Vercel through [System Environment Variables](/docs/environment-variables/system-environment-var
If you're not using System Environment Variables, use the `callbackUrl` option at the [`onBeforeGenerateToken`](/docs/vercel-blob/using-b

#### Local development

When running your application locally, the `onUploadCompleted` callback will not work as Vercel Blob cannot contact your localhost. Inste

When using ngrok in local development, you can configure the domain to call for onUploadCompleted by using the `VERCEL_BLOB_CALLBACK_URL`

```bash
VERCEL_BLOB_CALLBACK_URL=https://abc123.ngrok-free.app
```

## Next steps

- Learn how to [use the methods](/docs/storage/vercel-blob/using-blob-sdk) available with the `@vercel/blob` package


---
title: "Vercel Blob examples"
description: "Examples on how to use Vercel Blob in your applications"
last_updated: "2026-01-16T02:19:35.778Z"
source: "https://vercel.com/docs/vercel-blob/examples"
---

# Vercel Blob examples

## Range requests

Vercel Blob supports [range requests](https://developer.mozilla.org/docs/Web/HTTP/Range_requests) for partial downloads. This means you c

```bash filename="Terminal"
# First 4 bytes
curl -r 0-3 https://1sxstfwepd7zn41q.public.blob.vercel-storage.com/pi.txt
# 3.14

# Last 5 bytes
curl -r -5 https://1sxstfwepd7zn41q.public.blob.vercel-storage.com/pi.txt
# 58151

# Bytes 3-6
curl -r 3-6 https://1sxstfwepd7zn41q.public.blob.vercel-storage.com/pi.txt
# 4159
```

## Upload progress

You can track the upload progress when uploading blobs with the `onUploadProgress` callback:

```js
const blob = await upload('big-file.mp4', file, {
  access: 'public',
  handleUploadUrl: '/api/upload',
  onUploadProgress: (progressEvent) => {
    console.log(`Loaded ${progressEvent.loaded} bytes`);
    console.log(`Total ${progressEvent.total} bytes`);
    console.log(`Percentage ${progressEvent.percentage}%`);
  },
});
```

`onUploadProgress` is available on `put` and `upload` methods.

## Aborting requests

Every Vercel Blob operation can be canceled, just like a fetch call. This is useful when you want to abort an ongoing operation, for exam

```ts
const abortController = new AbortController();

try {
  const blobPromise = vercelBlob.put('hello.txt', 'Hello World!', {
    access: 'public',
    abortSignal: abortController.signal,
  });

  const timeout = setTimeout(() => {
    // Abort the request after 1 second
    abortController.abort();
  }, 1000);

  const blob = await blobPromise;

  console.info('blob put request completed', blob);

  clearTimeout(timeout);

  return blob.url;
} catch (error) {
  if (error instanceof vercelBlob.BlobRequestAbortedError) {
    // Handle the abort
    console.info('canceled put request');
  }

  // Handle other errors
}
```

## Deleting all blobs

If you want to delete all the blobs in your store you can use the following code snippet to delete them in batches.
This is useful if you have a lot of blobs and you want to avoid hitting the rate limits.

Either execute this code in a [Vercel Cron Job](/docs/cron-jobs), as a serverless function or on your local machine.

```ts
import { list, del, BlobServiceRateLimited } from '@vercel/blob';
import { setTimeout } from 'node:timers/promises';

async function deleteAllBlobs() {
  let cursor: string | undefined;
  let totalDeleted = 0;

  // Batch size to respect rate limits (conservative approach)
  const BATCH_SIZE = 100; // Conservative batch size
  const DELAY_MS = 1000; // 1 second delay between batches

  do {
    const listResult = await list({
      cursor,
      limit: BATCH_SIZE,
    });

    if (listResult.blobs.length > 0) {
      const batchUrls = listResult.blobs.map((blob) => blob.url);

      // Retry logic with exponential backoff
      let retries = 0;
      const maxRetries = 3;

      while (retries <= maxRetries) {
        try {
          await del(batchUrls);
          totalDeleted += listResult.blobs.length;
          console.log(
            `Deleted ${listResult.blobs.length} blobs (${totalDeleted} total)`,
          );
          break; // Success, exit retry loop
        } catch (error) {
          retries++;

          if (retries > maxRetries) {
            console.error(
              `Failed to delete batch after ${maxRetries} retries:`,
```

```
              error,
            );
            throw error; // Re-throw after max retries
          }

          // Exponential backoff: wait longer with each retry
          let backoffDelay = 2 ** retries * 1000;

          if (error instanceof BlobServiceRateLimited) {
            backoffDelay = error.retryAfter * 1000;
          }

          console.warn(
            `Retry ${retries}/${maxRetries} after ${backoffDelay}ms delay`,
          );

          await setTimeout(backoffDelay);
        }

        await setTimeout(DELAY_MS);
      }
    }

    cursor = listResult.cursor;
  } while (cursor);

  console.log(`All blobs were deleted. Total: ${totalDeleted}`);
}

deleteAllBlobs().catch((error) => {
  console.error('An error occurred:', error);
});
```

## Backups

While there's no native backup system for Vercel Blob, here are two ways to backup your blobs:

1. **Continuous backup**: When using [Client Uploads](/docs/storage/vercel-blob/using-blob-sdk#client-uploads) you can leverage the `onUp
2. **Periodic backup**: Using [Cron Jobs](/docs/cron-jobs) and the [Vercel Blob SDK](/docs/storage/vercel-blob/using-blob-sdk) you can pe

Here's an example implementation of a periodic backup as a Cron Job:

```ts
import { Readable } from 'node:stream';
import { S3Client } from '@aws-sdk/client-s3';
import { list } from '@vercel/blob';
import { Upload } from '@aws-sdk/lib-storage';
import type { NextRequest } from 'next/server';
import type { ReadableStream } from 'node:stream/web';

export async function GET(request: NextRequest) {
  const authHeader = request.headers.get('authorization');
  if (authHeader !== `Bearer ${process.env.CRON_SECRET}`) {
    return new Response('Unauthorized', {
      status: 401,
    });
  }

  const s3 = new S3Client({
    region: 'us-east-1',
  });

  let cursor: string | undefined;

  do {
    const listResult = await list({
      cursor,
      limit: 250,
    });

    if (listResult.blobs.length > 0) {
      await Promise.all(
        listResult.blobs.map(async (blob) => {
          const res = await fetch(blob.url);
          if (res.body) {
            const parallelUploads3 = new Upload({
              client: s3,
              params: {
                Bucket: 'vercel-blob-backup',
                Key: blob.pathname,
                Body: Readable.fromWeb(res.body as ReadableStream),
              },
              leavePartsOnError: false,
            });

            await parallelUploads3.done();
          }
        }),
      );
    }

    cursor = listResult.cursor;
  } while (cursor);

  return new Response('Backup done!');
}
```

This script optimizes the process by streaming the content directly from Vercel Blob to the backup storage, avoiding buffering all the co

You can split your backup process into smaller chunks if you're hitting an execution limit. In this case you would save the `cursor` to a

---
title: "Vercel Blob"
description: "Vercel Blob is a scalable, and cost-effective object storage service for static assets, such as images, videos, audio files
last_updated: "2026-01-16T02:19:35.809Z"
source: "https://vercel.com/docs/vercel-blob"
---

# Vercel Blob

## Use cases

[Vercel Blob](/storage/blob) is a great solution for storing [blobs](https://developer.mozilla.org/docs/Web/API/Blob "Blob object") that

- Files that are programmatically uploaded or generated at build time, for display and download such as avatars, screenshots, cover image
- Large files such as videos and audios to take advantage of the global network
- Files that you would normally store in an external file storage solution like Amazon S3. With your project hosted on Vercel, you can re

> **💡 Note:** Stored files are referred to as "blobs" once they're in the storage system,
> following cloud storage terminology.

## Getting started

```js
import { put } from '@vercel/blob';

const blob = await put('avatar.jpg', imageFile, {
  access: 'public',
});
```

You can create and manage your Vercel Blob stores from your [account dashboard](/dashboard) or the [Vercel CLI](/docs/cli/blob). You can

To get started, see the [server-side](/docs/storage/vercel-blob/server-upload), or [client-side](/docs/storage/vercel-blob/client-upload)

## Using Vercel Blob in your workflow

If you'd like to know whether or not Vercel Blob can be integrated into your workflow, it's worth knowing the following:

- You can have one or more Vercel Blob stores per Vercel account
- You can use multiple Vercel Blob stores in one Vercel project
- Each Vercel Blob store can be accessed by multiple Vercel projects
  Vercel Blob URLs are publicly accessible, but you can make them [unguessable](/docs/vercel-blob/security).
- To add to or remove from the content of a Blob store, a valid [token](/docs/storage/vercel-blob/using-blob-sdk#read-write-token) is req

### Transferring to another project

If you need to transfer your blob store from one project to another project in the same or different team, review [Transferring your stor

## Viewing and downloading blobs

Each Blob is served with a `content-disposition` header. Based on the MIME type of the uploaded blob, it is either set to `attachment` (f
This is done to prevent hosting specific files on `@vercel/blob` like HTML web pages. Your browser will automatically download the blob i

Currently `text/plain`, `text/xml`, `application/json`, `application/pdf`, `image/*`, `audio/*` and `video/*` resolve to a `content-dispo

All other MIME types default to `content-disposition: attachment`.

If you need a blob URL that always forces a download you can use the `downloadUrl` property on the blob object. This URL always has the `

```js
import { list } from '@vercel/blob';

export default async function Page() {
  const response = await list();

  return (
    <>
      {response.blobs.map((blob) => (
        <a key={blob.pathname} href={blob.downloadUrl}>
          {blob.pathname}
        </a>
      ))}
    </>
  );
}
```

Alternatively the SDK exposes a helper function called `getDownloadUrl` that returns the same URL.

## Caching

When you request a blob URL using a browser, the content is cached in two places:

1. Your browser's cache
2. Vercel's [cache](/docs/cdn-cache)

Both caches store blobs for up to 1 month by default to ensure optimal performance when serving content. While both systems aim to respec

Vercel will cache blobs up to [512 MB](/docs/vercel-blob/usage-and-pricing#size-limits). Bigger blobs will always be served from the orig

### Configuring cache duration

You can customize the caching duration using the `cacheControlMaxAge` option in the [`put()`](/docs/storage/vercel-blob/using-blob-sdk#pu

The minimum configurable value is 60 seconds (1 minute). This represents the maximum time needed for our cache to update content behind a

### Important considerations when updating blobs

When you delete or update (overwrite) a blob, the changes may take up to 60 seconds to propagate through our cache. However, browser cach

- While our cache can update to serve the latest content, browsers will continue serving the cached version
- To force browsers to fetch the updated content, add a unique query parameter to the blob URL:

```html
<img
  src="https://1sxstfwepd7zn41q.public.blob.vercel-storage.com/blob-oYnXSVczoLa9yBYMFJOSNdaiiervF5.png?v=123456"
/>
```

For more information about updating existing blobs, see the [Overwriting blobs](#overwriting-blobs) section.

### Best practice: Treat blobs as immutable

For optimal performance and to avoid caching issues, consider treating blobs as immutable objects:

- Instead of updating existing blobs, create new ones with different pathnames (or use `addRandomSuffix: true` option)
- This approach avoids unexpected behaviors like outdated content appearing in your application

There are still valid use cases for mutable blobs with shorter cache durations, such as a single JSON file that's updated every 5 minutes

## Overwriting blobs

By default, Vercel Blob prevents you from accidentally overwriting existing blobs by using the same pathname twice. When you attempt to u

### Using `allowOverwrite`

To explicitly allow overwriting existing blobs, you can use the `allowOverwrite` option:

```js
const blob = await put('user-profile.jpg', imageFile, {
  access: 'public',
  allowOverwrite: true, // Enable overwriting an existing blob with the same pathname
});
```

This option is available in these methods:

- `put()`
- In client uploads via the `onBeforeGenerateToken()` function

### When to use overwriting

Overwriting blobs can be appropriate for certain use cases:

1. **Regularly updated files**: For files that need to maintain the same URL but contain updated content (like JSON data files or configu
2. **Content with predictable update patterns**: For data that changes on a schedule and where consumers expect updates at the same URL

When overwriting blobs, be aware that due to [caching](#caching), changes won't be immediately visible. The minimum time for changes to p

### Alternatives to overwriting

If you want to avoid overwriting existing content (recommended for most use cases), you have two options:

1. **Use `addRandomSuffix: true`**: This automatically adds a unique random suffix to your pathnames:

```js
const blob = await put('avatar.jpg', imageFile, {
  access: 'public',
  addRandomSuffix: true, // Creates a pathname like 'avatar-oYnXSVczoLa9yBYMFJOSNdaiiervF5.jpg'
});
```

2. **Generate unique pathnames programmatically**: Create unique pathnames by adding timestamps, UUIDs, or other identifiers:

```js
const timestamp = Date.now();
const blob = await put(`user-profile-${timestamp}.jpg`, imageFile, {
  access: 'public',
});
```

## Blob Data Transfer

Vercel Blob delivers content through a specialized network optimized for static assets:

- **Region-based distribution**: Content is served from 19 regional hubs strategically located around the world
- **Optimized for non-critical assets**: Well-suited for content "below the fold" that isn't essential for initial page rendering metrics
- **Cost-optimized for large assets**: 3x more cost-efficient than [Fast Data Transfer](/docs/cdn) on average
- **Great for media delivery**: Ideal for large media files like images, videos, and documents

While [Fast Data Transfer](/docs/manage-cdn-usage#fast-data-transfer) provides city-level, ultra-low latency, Blob Data Transfer prioriti

Blob Data Transfer fees apply only to downloads (outbound traffic), not uploads. See [pricing documentation](/docs/vercel-blob/usage-and-

## Upload charges

Upload charges depend on your implementation method:

- [Client Uploads](/docs/vercel-blob/client-upload): No data transfer charges for uploads
- [Server Uploads](/docs/vercel-blob/server-upload): [Fast Data Transfer](/docs/manage-cdn-usage#fast-data-transfer) transfer charges app

## SEO and search engine indexing

### Search engine visibility of blobs

While Vercel Blob URLs can be designed to be unique and unguessable (when using `addRandomSuffix: true`), they can still be indexed by se

- If you link to blob URLs from public webpages
- If you embed blob content (images, PDFs, etc.) in indexed content
- If you share blob URLs publicly, even in contexts outside your application

By default, Vercel Blob does not provide a `robots.txt` file or other indexing controls. This means search engines like Google may discov

### Preventing search engine indexing

If you want to prevent search engines from indexing your blob content, you need to upload a `robots.txt` file directly to your blob store

1. Go to your [**Storage** page](https://vercel.com/d?to=%2F%5Bteam%5D%2F~%2Fstores\&title=Go+to+Storage) and select your blob store
2. Upload a `robots.txt` file to the root of your blob store with appropriate directives

Example `robots.txt` content to block all crawling of your blob store:

```
User-agent: *
Disallow: /
```

### Removing already indexed blob content

If your blob content has already been indexed by search engines:

1. Verify your website ownership in [Google Search Console](https://search.google.com/search-console/)
2. Upload a `robots.txt` file to your blob store as described above
3. Use the "Remove URLs" tool in Google Search Console to request removal

## Choosing your Blob store region

You can create Blob stores in any of the 19 [regions](/docs/regions#region-list). Use the region selector in the dashboard at blob store

Select a region close to your customers and functions to minimize upload time. Region selection also helps meet data regulatory requireme

You cannot change the region once the store is created.

## Simple operations

Simple operations in Vercel Blob are specific read actions counted for billing purposes:

- When the [`head()`](/docs/vercel-blob/using-blob-sdk#head) method is called to retrieve blob metadata
- When a blob is accessed by its URL and it's a cache MISS

A cache MISS occurs when the blob is accessed for the first time or when its previously cached version has expired. Note that blob URL ac

## Advanced operations

Advanced operations in Vercel Blob are write, copy, and listing actions counted for billing purposes:

- When the [`put()`](/docs/vercel-blob/using-blob-sdk#put) method is called to upload a blob
- When the [`upload()`](/docs/vercel-blob/using-blob-sdk#upload) method is used for client-side uploads
- When the [`copy()`](/docs/vercel-blob/using-blob-sdk#copy) method is called to copy an existing blob
- When the [`list()`](/docs/vercel-blob/using-blob-sdk#list) method is called to list blobs in your store

### Dashboard usage counts as operations

Using the Vercel Blob file browser in your dashboard will count as operations. Each time you refresh the blob list, upload files through

Common dashboard actions that count as operations:

- **Refreshing the file browser**: Uses `list()` to display your blobs
- **Uploading files via dashboard**: Uses `put()` for each file uploaded
- **Viewing blob details**: May trigger additional API calls
- **Navigating folders**: Uses `list()` with different prefixes

If you notice unexpected increases in your operations count, check whether team members are browsing your blob store through the Vercel d

For [multipart uploads](#multipart-uploads), multiple advanced operations are counted:

- One operation when starting the upload
- One operation for each part uploaded
- One operation for completing the upload

Delete operations using the [`del()`](/docs/vercel-blob/using-blob-sdk#del) are free of charge. They are considered advanced operations f

## Storage calculation

Vercel Blob measures your storage usage by taking snapshots of your blob store size every 15 minutes and averages these measurements over

The Vercel dashboard displays two metrics:

- **Latest value**: The most recent measurement of your blob store size
- **Monthly average**: The average of all measurements throughout the billing period (this is what you're billed for)

**Example:**

1. Day 1: Upload a 2GB file → Store size: 2GB
2. Day 15: Add 1GB file → Store size: 3GB
3. Day 25: Delete 2GB file → Store size: 1GB

Month end billing:

- Latest value: 1GB
- Monthly average: ~2GB (billed amount)

If no changes occur in the following month (no new uploads or deletions), each 15-minute measurement would consistently show 1 GB. In thi

## Multipart uploads

Vercel Blob supports [multipart uploads](/docs/vercel-blob/using-blob-sdk#multipart-uploads) for large files, which provides significant

Multipart uploads work by splitting large files into smaller chunks (parts) that are uploaded independently and then reassembled on the s

- **Improved upload reliability**: If a network issue occurs during upload, only the affected part needs to be retried instead of restart
- **Better performance**: Multiple parts can be uploaded in parallel, significantly increasing transfer speed
- **Progress tracking**: More granular upload progress reporting as each part completes

We recommend using multipart uploads for files larger than 100 MB. Both the [`put()`](/docs/vercel-blob/using-blob-sdk#put) and [`upload(

For billing purposes, multipart uploads count as multiple advanced operations:

- One operation when starting the upload
- One operation for each part uploaded
- One operation for completing the upload

This approach ensures reliable handling of large files while maintaining the performance and efficiency expected from modern cloud storag

## Durability and availability

Vercel Blob leverages [Amazon S3](https://aws.amazon.com/s3/) as its underlying storage infrastructure, providing industry-leading durabi

- **Durability**: Vercel Blob offers 99.999999999% (11 nines) durability. This means that even with one billion objects, you could expect
- **Availability**: Vercel Blob provides 99.99% (4 nines) availability in a given year, ensuring that your data is accessible when you ne

These guarantees are backed by [S3's robust architecture](https://docs.aws.amazon.com/AmazonS3/latest/userguide/DataDurability.html), whi

## Folders and slashes

Vercel Blob has folders support to organize your blobs:

```js
const blob = await put('folder/file.txt', 'Hello World!', { access: 'public' });
```

The path `folder/file.txt` creates a folder named `folder` and a blob named `file.txt`. To list all blobs within a folder, use the [`list

```js
const listOfBlobs = await list({
  cursor,
  limit: 1000,
  prefix: 'folder/',
});
```

You don't need to create folders. Upload a file with a path containing a slash `/`, and Vercel Blob will interpret the slashes as folder

In the Vercel Blob file browser on the Vercel dashboard, any pathname with a slash `/` is treated as a folder. However, these are not act

## Blob sorting and organization

Blobs are returned in **lexicographical order** by pathname (not creation date) when using [`list()`](/docs/vercel-blob/using-blob-sdk#li

**Sort by creation date:** Include timestamps in pathnames:

```js
const timestamp = new Date().toISOString().split('T')[0]; // YYYY-MM-DD
await put(`reports/${timestamp}-quarterly-report.pdf`, file, {
  access: 'public',
});
```

**Use prefixes for search:** Consider lowercase pathnames for consistent matching:

```js
await put('user-uploads/avatar.jpg', file, { access: 'public' });
const userUploads = await list({ prefix: 'user-uploads/' });
```

For complex sorting, sort results client-side using `uploadedAt` or other properties.

## More resources

- [Client Upload Quickstart](/docs/storage/vercel-blob/client-upload)
- [Server Upload Quickstart](/docs/storage/vercel-blob/server-upload)
- [Vercel Blob SDK](/docs/storage/vercel-blob/using-blob-sdk)
- [Vercel Blob CLI](/docs/cli/blob)
- [Vercel Blob Pricing](/docs/vercel-blob/usage-and-pricing)
- [Vercel Blob Security](/docs/storage/vercel-blob/security)
- [Vercel Blob Examples](/docs/storage/vercel-blob/examples)
- [Observability](/docs/observability)

---
title: "Security"
description: "Learn how your Vercel Blob store is secured"
last_updated: "2026-01-16T02:19:35.763Z"
source: "https://vercel.com/docs/vercel-blob/security"
---

# Security

Vercel Blob URLs, although publicly accessible, are unique and hard to guess when you use the `addRandomSuffix: true` option. They consis

> **💡 Note:** This is similar to [Share a file
> publicly](https://support.google.com/drive/answer/2494822?hl=en\&co=GENIE.Platform%3DDesktop#zippy=%2Cshare-a-file-publicly)
> in Google Docs. You should ensure that the URLs are only shared to authorized
> users

Headers that enhance security by preventing unauthorized downloads, blocking external content from being embedded, and protecting against

- `content-security-policy`: `default-src "none"`
- `x-frame-options`: `DENY`
- `x-content-type-options`: `nosniff`
- `content-disposition`: `attachment/inline; filename="filename.extension"`

### Encryption

All files stored on Vercel Blob are secured using AES-256 encryption. This encryption process is applied at rest and is transparent, ensu

### Firewall and WAF integration

Vercel Blob is protected by Vercel's [platform-wide firewall](/docs/vercel-firewall#platform-wide-firewall) which provides DDoS mitigatio

Vercel Blob does not currently support [Vercel WAF](/docs/vercel-firewall/vercel-waf). If you need WAF rules on your blob URLs, consider

---
title: "Server Uploads with Vercel Blob"
description: "Learn how to upload files to Vercel Blob using Server Actions and Route Handlers"
last_updated: "2026-01-16T02:19:35.862Z"
source: "https://vercel.com/docs/vercel-blob/server-upload"
---

# Server Uploads with Vercel Blob

In this guide, you'll learn how to do the following:

- Use the Vercel dashboard to create a Blob store connected to a project
- Upload a file using the Blob SDK from the server

> **⚠️ Warning:** Vercel has a [4.5 MB request body size
> limit](/docs/functions/runtimes#request-body-size) on Vercel Functions. If you
> need to upload larger files, use [client
> uploads](/docs/storage/vercel-blob/client-upload).

## Prerequisites

Vercel Blob works with any frontend framework. First, install the package:

- ### Create a Blob store
  Navigate to the [Project](/docs/projects/overview) you'd like to add the blob store to. Select the **Storage** tab, then select the **C

  Under the **Create New** tab, select **Blob** and then the **Continue** button.

  Use the name "Images" and select **Create a new Blob store**. Select the environments where you would like the read-write token to be i

  Once created, you are taken to the Vercel Blob store page.

- ### Prepare your local project
  Since you created the Blob store in a project, we automatically created and added the following Environment Variable to the project for
  - `BLOB_READ_WRITE_TOKEN`
  To use this Environment Variable locally, we recommend pulling it with the Vercel CLI:
  ```bash
  vercel env pull
  ```

Server uploads are perfectly fine as long as you do not need to upload files larger than [4.5 MB on Vercel](/docs/functions/runtimes#requ

## Upload a file using Server Actions

## Upload a file using a server upload page and route

You can upload files to Vercel Blob using Route Handlers/API Routes. The following example shows how to upload a file to Vercel Blob usin

- ### Create a server upload page
  This page will upload files to your server. The files will then be sent to Vercel Blob.

- ### Create a server upload route
  This route forwards the file to Vercel Blob and returns the URL of the uploaded file to the browser.

### Testing your page

- ### Run your application locally
  Run your application locally and visit `/avatar/upload` to upload the file to your store. The browser will display the unique URL creat

- ### Review the Blob object metadata
  - Go to the Vercel Project where you created the store
  - Select the **Storage** tab and select your new store
  - Paste the blob object URL returned in the previous step in the **Blob URL** input box in the **Browser** section and select **Lookup*
  - The following blob object metadata will be displayed: file name, path, size, uploaded date, content type and HTTP headers
  - You also have the option to download and delete the file from this page

You have successfully uploaded an object to your Vercel Blob store and are able to review it's metadata, download, and delete it from you

## Next steps

- Learn how to [use the methods](/docs/storage/vercel-blob/using-blob-sdk) available with the `@vercel/blob` package

---
title: "Vercel Blob Pricing"
description: "Learn about the pricing for Vercel Blob."
last_updated: "2026-01-16T02:19:35.878Z"
source: "https://vercel.com/docs/vercel-blob/usage-and-pricing"
---

# Vercel Blob Pricing

## Usage

Vercel Blob usage is measured based on the following:

- **Storage Size**: Monthly average of your blob store size (GB-month)
- **Simple Operations**: Counts when a blob is accessed by its URL and it's a cache MISS or when using the [`head()`](/docs/vercel-blob/u
- **Advanced Operations**: Counts when using [`put()`](/docs/vercel-blob/using-blob-sdk#put), [`copy()`](/docs/vercel-blob/using-blob-sdk
- **Blob Data Transfer**: Charged when blobs are downloaded or viewed
- **[Edge Requests](/docs/pricing/networking#edge-requests)**: Each blob access by its URL counts as one Edge Request, regardless if it's
- **[Fast Origin Transfer](/docs/pricing/networking#fast-origin-transfer)**: Applied only for cache MISS scenarios

See the [usage details](#usage-details) and [pricing example](#pricing-example) sections for more information on how usage is calculated.

> **💡 Note:** Stored files are referred to as "blobs" once they're in the storage system,
> following cloud storage terminology.

## Pricing

> **💡 Note:** [Edge Requests](/docs/pricing/networking#edge-requests) and [Fast Origin
> Transfer](/docs/pricing/networking#fast-origin-transfer) for blobs are billed
> at standard [CDN rates](/docs/cdn#pricing). The included resource usage for
> the Hobby plan is shared across all Vercel services in your project.

## Usage details

- Cache HITs do not count as Simple Operations
- Cache HITs do not incur Fast Origin Transfer charges
- The maximum size of a blob in cache is [512 MB](/usage-and-pricing#size-limits). Any blob larger than this will generate a cache MISS o
- Uploads do not incur data transfer charges when using [Client Uploads](/docs/vercel-blob/client-upload)
- Uploads incur [Fast Data Transfer](/docs/manage-cdn-usage#fast-data-transfer) charges when using [Server Uploads](/docs/vercel-blob/ser
- [Multipart uploads](/docs/vercel-blob/using-blob-sdk#multipart-uploads) count as multiple Advanced Operations: one when starting, one p
- [`del()`](/docs/vercel-blob/using-blob-sdk#del) operations are free
- **Dashboard interactions count as operations**: Each time you interact with the Vercel dashboard to browse your blob store, upload file

## Hobby

Vercel Blob is free for Hobby users within the [usage limits](#pricing).

Vercel will send you emails as you are nearing your usage limits. You **will not pay for any additional usage**. However, you will not be

## Pro

You pay for usage using your [monthly credit allocation](/docs/plans/pro-plan#credit-and-usage-allocation) which switches to on-demand on

Pro teams can [set up Spend Management](/docs/spend-management#managing-your-spend-amount) to get notified or to automatically take actio

## Enterprise

Vercel Blob is available for all Enterprise teams at the same price as Pro. Contact your account team for pricing or support questions.

## Pricing Example

Let's say during one month of usage, you upload 120,000 blobs of which 30% (36,000 blobs) are uploaded using multipart uploads with 5 par
Your storage averages 50 GB and your blobs are downloaded 2.5 million times, with a 70% cache HIT ratio (meaning 30% or 750,000 downloads

Here's the cost breakdown:

- **Storage**: 50 GB total - 5 GB included = 45 GB extra at $0.023/GB = $1.04
- **Simple Operations**: 750K (30% cache MISSes of 2.5M downloads + head calls) - 100K included = 650K extra at $0.40/1M = $0.26
- **Advanced Operations**:
  - Single uploads: 84K (70% of 120K blobs)
  - Multipart uploads: 36K × (1 start + 5 parts + 1 completion) = 252K operations
  - Total: 336K - 10K included = 326K extra at $5.00/1M = $1.63
- **Data Transfer** (iad1): 350 GB total - 100 GB included = 250 GB extra at $0.050/GB = $12.50
- **Edge Requests**: 2.5M requests (all downloads) - 10M included = $0.00
- **Fast Origin Transfer** (iad1): 105 GB (30% cache MISSes of 350 GB) - 100 GB included = 5 GB extra at $0.06/GB = $0.30

**Total**: $15.73/month

## Limits

Vercel Blob has certain [limits](/docs/limits) that you should be aware of when designing your application.

### Operation rate limits

| Plan | Simple Operations | Advanced Operations |
| --- | --- | --- |
| Hobby | 1,200/min (20/s) | 900/min (15/s) |
| Pro | 7,200/min (120/s) | 4,500/min (75/s) |
| Enterprise | 9,000/min (150/s) | 7,500/min (125/s) |

**Note:** Rate limits are based on the number of operations, not HTTP requests. For example, when using `del([pathnames])` to delete mult

### Size limits

- **Cache Size Limit**: 512 MB per blob
  - Blobs larger than 512 MB will not be cached
  - Accessing these blobs will always count as a cache MISS (generating one [Simple Operation](/docs/vercel-blob#simple-operations))
  - These large blobs will also incur [Fast Origin Transfer](/docs/manage-cdn-usage#fast-origin-transfer) charges for each access

- **Maximum File Size**: 5TB (5,000GB)
  - This is the absolute maximum size for any individual file uploaded to Vercel Blob
  - For files larger than 100MB, we recommend using [multipart uploads](/docs/vercel-blob#multipart-uploads)

## Observability

You can monitor and analyze your Vercel Blob usage with the [Observability tab](https://vercel.com/d?to=%2F%5Bteam%5D%2F~%2Fobservability

---
title: "@vercel/blob"

description: "Learn how to use the Vercel Blob SDK to access your blob store from your apps."
last_updated: "2026-01-16T02:19:36.093Z"
source: "https://vercel.com/docs/vercel-blob/using-blob-sdk"
---

# @vercel/blob

## Getting started

To start using [Vercel Blob](/storage/blob) SDK, follow the steps below:

> **💡 Note:** You can also interact with Vercel Blob using the [Vercel CLI](/docs/cli/blob)
> for command-line operations. For example, you might want to quickly upload
> assets during local development without writing additional code.

Vercel Blob works with any frontend framework. begin by installing the package:

- ### Create a Blob store
  Navigate to the [Project](/docs/projects/overview) you'd like to add the blob store to. Select the **Storage** tab, then select the **C

  Under the **Create New** tab, select **Blob** and then the **Continue** button.

  Choose a name for your store and select **Create a new Blob store**. Select the environments where you would like the read-write token

  Once created, you are taken to the Vercel Blob store page.

- ### Prepare your local project
  Since you created the Blob store in a project, environment variables are automatically created and added to the project for you.
  - `BLOB_READ_WRITE_TOKEN`
  To use this environment variable locally, use the Vercel CLI to [pull the values into your local project](/docs/cli/env#exporting-develo
  ```bash
  vercel env pull
  ```

## Read-write token

A read-write token is required to interact with the Blob SDK. When you create a Blob store in your Vercel Dashboard, an environment varia

- If you deploy your application in the same Vercel project where your Blob store is located, you *do not* need to specify the `token` pa
- If you deploy your application in a different Vercel project or scope, you can create an environment variable there and assign the toke
- If you deploy your application outside of Vercel, you can copy the `token` value from the store settings and pass it as the `token` par

## Using the SDK methods

In the examples below, we use [Fluid compute](/docs/fluid-compute) for optimal performance and scalability.

## Upload a blob

This example creates a Function that accepts a file from a `multipart/form-data` form and uploads it to the Blob store. The function retu

### `put()`

The `put` method uploads a blob object to the Blob store.

It accepts the following parameters:

- `pathname`: (Required) A string specifying the base value of the return URL
- `body`: (Required) A blob object as `ReadableStream`, `String`, `ArrayBuffer` or `Blob` based on these [supported body types](https://d
- `options`: (Required) A `JSON` object with the following required and optional parameters:

| Parameter | Required | Values |
| --- | --- | --- |
| `access` | Yes | `public` |
| `addRandomSuffix` | No | A boolean specifying whether to add a random suffix to the `pathname`. It defaults to `false`. **We r |
| `allowOverwrite` | No | A boolean to allow overwriting blobs. By default an error will be thrown if you try to overwrite a bl |
| `cacheControlMaxAge` | No | A number in seconds to configure how long Blobs are cached. Defaults to one month. Cannot be set to a |
| `contentType` | No | A string indicating the [media type](https://developer.mozilla.org/docs/Web/HTTP/Headers/Content-Type |
| `token` | No | A string specifying the token to use when making requests. It defaults to `process.env.BLOB_READ_WRIT |
| `multipart` | No | Pass `multipart: true` when uploading large files. It will split the file into multiple parts, upload |
| `abortSignal` | No | An [AbortSignal](https://developer.mozilla.org/en-US/docs/Web/API/AbortSignal) to cancel the operatio |
| `onUploadProgress` | No | Callback to track upload progress: `onUploadProgress({loaded: number, total: number, percentage: numb |

#### Example code with folder output

To upload your file to an existing [folder](#folders) inside your blob storage, pass the folder name in the `pathname` as shown below:

#### Example responses

`put()` returns a `JSON` object with the following data for the created blob object:

```json
{
  "pathname": "string",
  "contentType": "string",
  "contentDisposition": "string",
  "url": "string",
  "downloadUrl": "string"
}
```

An example blob (uploaded with `addRandomSuffix: true`) is:

```json
{
  "pathname": "profilesv1/user-12345-NoOVGDVcqSPc7VYCUAGnTzLTG2qEM2.txt",
  "contentType": "text/plain",
  "contentDisposition": "attachment; filename=\"user-12345-NoOVGDVcqSPc7VYCUAGnTzLTG2qEM2.txt\"",
  "url": "https://ce0rcu23vrrdzqap.public.blob.vercel-storage.com/profilesv1/user-12345-NoOVGDVcqSPc7VYCUAGnTzLTG2qEM2.txt",
  "downloadUrl": "https://ce0rcu23vrrdzqap.public.blob.vercel-storage.com/profilesv1/user-12345-NoOVGDVcqSPc7VYCUAGnTzLTG2qEM2.txt?downlo
}
```

```

An example blob uploaded without `addRandomSuffix: true` (default) is:

```json
{
  "pathname": "profilesv1/user-12345.txt",
  "contentType": "text/plain",
  "contentDisposition": "attachment; filename=\"user-12345.txt\"",
  // no automatic random suffix added 👇
  "url": "https://ce0rcu23vrrdzqap.public.blob.vercel-storage.com/profilesv1/user-12345.txt",
  "downloadUrl": "https://ce0rcu23vrrdzqap.public.blob.vercel-storage.com/profilesv1/user-12345.txt?download=1"
}
```

## Multipart Uploads

When uploading large files you should use multipart uploads to have a more reliable upload process. A multipart upload splits the file in
This process consists of three phases: creating a multipart upload, uploading the parts and completing the upload. `@vercel/blob` offers

### Automatic

This method has everything baked in and is easiest to use. It's part of the `put` and `upload` API's. Under the hood it will start the up

### Manual

This method gives you full control over the multipart upload process. It consists of three phases:

**Phase 1: Create a multipart upload**

`createMultipartUpload` accepts the following parameters:

- `pathname`: (Required) A string specifying the path inside the blob store. This will be the base value of the return URL and includes t
- `options`: (Required) A `JSON` object with the following required and optional parameters:

| Parameter | Required | Values |
| --- | --- | --- |
| `access` | Yes | `public` |
| `contentType` | No | The [media type](https://developer.mozilla.org/docs/Web/HTTP/Headers/Content-Type) for the file. If n |
| `token` | No | A string specifying the token to use when making requests. It defaults to `process.env.BLOB_READ_WRIT |
| `addRandomSuffix` | No | A boolean specifying whether to add a random suffix to the pathname. It defaults to `true`. |
| `cacheControlMaxAge` | No | A number in seconds to configure the edge and browser cache. Defaults to one year. See the [caching]( |
| `abortSignal` | No | An [AbortSignal](https://developer.mozilla.org/en-US/docs/Web/API/AbortSignal) to cancel the operatio |

`createMultipartUpload()` returns a `JSON` object with the following data for the created upload:

```json
{
  "key": "string",
  "uploadId": "string"
}
```

**Phase 2: Upload all the parts**

> **⚠️ Warning:** In the multipart uploader process, it's necessary for you to manage both
> memory usage and concurrent upload requests. Additionally, each part must be a
> minimum of 5MB, except the last one which can be smaller, and all parts should
> be of equal size.

`uploadPart` accepts the following parameters:

- `pathname`: (Required) Same value as the `pathname` parameter passed to `createMultipartUpload`
- `chunkBody`: (Required) A blob object as `ReadableStream`, `String`, `ArrayBuffer` or `Blob` based on these [supported body types](http
- `options`: (Required) A `JSON` object with the following required and optional parameters:

| Parameter | Required | Values |
| --- | --- | --- |
| `access` | Yes | `public` |
| `partNumber` | Yes | A number identifying which part is uploaded |
| `key` | Yes | A string returned from `createMultipartUpload` which identifies the blob object |
| `uploadId` | Yes | A string returned from `createMultipartUpload` which identifies the multipart upload |
| `token` | No | A string specifying the token to use when making requests. It defaults to `process.env.BLOB_READ_WRITE_TOKEN |
| `abortSignal` | No | An [AbortSignal](https://developer.mozilla.org/en-US/docs/Web/API/AbortSignal) to cancel the operation |

`uploadPart()` returns a `JSON` object with the following data for the uploaded part:

```json
{
  "etag": "string",
  "partNumber": "string"
}
```

**Phase 3: Complete the multipart upload**

`completeMultipartUpload` accepts the following parameters:

- `pathname`: (Required) Same value as the `pathname` parameter passed to `createMultipartUpload`
- `parts`: (Required) An array containing all the uploaded parts
- `options`: (Required) A `JSON` object with the following required and optional parameters:

| Parameter | Required | Values |
| --- | --- | --- |
| `access` | Yes | `public` |
| `key` | Yes | A string returned from `createMultipartUpload` which identifies the blob object |
| `uploadId` | Yes | A string returned from `createMultipartUpload` which identifies the multipart upload |
| `contentType` | No | The [media type](https://developer.mozilla.org/docs/Web/HTTP/Headers/Content-Type) for the file. If n |
| `token` | No | A string specifying the token to use when making requests. It defaults to `process.env.BLOB_READ_WRIT |
| `addRandomSuffix` | No | A boolean specifying whether to add a random suffix to the pathname. It defaults to `true`. |
```

| `cacheControlMaxAge` | No | A number in seconds to configure the edge and browser cache. Defaults to one year. See the [caching](
| `abortSignal` | No | An [AbortSignal](https://developer.mozilla.org/en-US/docs/Web/API/AbortSignal) to cancel the operatio

`completeMultipartUpload()` returns a `JSON` object with the following data for the created blob object:

```json
{
  "pathname": "string",
  "contentType": "string",
  "contentDisposition": "string",
  "url": "string",
  "downloadUrl": "string"
}
```

### Uploader

A less verbose way than the manual process is the multipart uploader method. It's a wrapper around the manual multipart upload process an
This results in a simpler API, but still requires you to handle memory usage and concurrent upload requests.

**Phase 1: Create the multipart uploader**

`createMultipartUploader` accepts the following parameters:

- `pathname`: (Required) A string specifying the path inside the blob store. This will be the base value of the return URL and includes t
- `options`: (Required) A `JSON` object with the following required and optional parameters:

| Parameter | Required | Values
| --- | --- | ---
| `access` | Yes | `public`
| `contentType` | No | The [media type](https://developer.mozilla.org/docs/Web/HTTP/Headers/Content-Type) for the file. If n
| `token` | No | A string specifying the token to use when making requests. It defaults to `process.env.BLOB_READ_WRIT
| `addRandomSuffix` | No | A boolean specifying whether to add a random suffix to the pathname. It defaults to `true`.
| `cacheControlMaxAge` | No | A number in seconds to configure the edge and browser cache. Defaults to one year. See the [caching](
| `abortSignal` | No | An [AbortSignal](https://developer.mozilla.org/en-US/docs/Web/API/AbortSignal) to cancel the operatio

`createMultipartUploader()` returns an `Uploader` object with the following attributes and methods:

**Phase 2: Upload all the parts**

> **⚠️ Warning:** In the multipart uploader process, it's necessary for you to manage both
> memory usage and concurrent upload requests. Additionally, each part must be a
> minimum of 5MB, except the last one which can be smaller, and all parts should
> be of equal size.

`uploader.uploadPart` accepts the following parameters:

- `partNumber`: (Required) A number identifying which part is uploaded
- `chunkBody`: (Required) A blob object as `ReadableStream`, `String`, `ArrayBuffer` or `Blob` based on these [supported body types](http

`uploader.uploadPart()` returns an object with the following data for the uploaded part:

**Phase 3: Complete the multipart upload**

`uploader.complete` accepts the following parameters:

- `parts`: (Required) An array containing all the uploaded parts

`uploader.complete()` returns an object with the following data for the created blob object:

## Deleting blobs

This example creates a function that deletes a blob object from the Blob store. You can delete multiple blob objects in a single request

```ts filename="app/delete/route.ts" framework=nextjs-app
import { del } from '@vercel/blob';

export async function DELETE(request: Request) {
  const { searchParams } = new URL(request.url);
  const urlToDelete = searchParams.get('url') as string;
  await del(urlToDelete);

  return new Response();
}
```

```js filename="app/delete/route.js" framework=nextjs-app
import { del } from '@vercel/blob';

export async function DELETE(request) {
  const { searchParams } = new URL(request.url);
  const urlToDelete = searchParams.get('url');
  await del(urlToDelete);

  return new Response();
}
```

```ts filename="app/delete/route.ts" framework=nextjs
import { del } from '@vercel/blob';

export async function DELETE(request: Request) {
  const { searchParams } = new URL(request.url);
  const urlToDelete = searchParams.get('url') as string;
  await del(urlToDelete);

  return new Response();
}
```

```js filename="app/delete/route.js" framework=nextjs
import { del } from '@vercel/blob';

export async function DELETE(request) {
  const { searchParams } = new URL(request.url);
  const urlToDelete = searchParams.get('url');
  await del(urlToDelete);

  return new Response();
}
```

```ts filename="api/blob.ts" framework=other
import { del } from '@vercel/blob';

export async function DELETE(request: Request) {
  const { searchParams } = new URL(request.url);
  const urlToDelete = searchParams.get('url') as string;
  await del(urlToDelete);

  return new Response();
}
```

```js filename="api/blob.js" framework=other
import { del } from '@vercel/blob';

export async function DELETE(request) {
  const { searchParams } = new URL(request.url);
  const urlToDelete = searchParams.get('url');
  await del(urlToDelete);

  return new Response();
}
```

### `del()`

The `del` method deletes one or multiple blob objects from the Blob store.

Since blobs are cached, it may take up to one minute for them to be fully removed from the Vercel CDN cache.

It accepts the following parameters:

- `urlOrPathname`: (Required) A string or array of strings specifying the URL(s) or pathname(s) of the blob object(s) to delete.
- `options`: (Optional) A `JSON` object with the following optional parameter:

| Parameter | Required | Values
| ---------- | -------- | ----------
| `token` | No | A string specifying the read-write token to use when making requests. It defaults to `process.env.BLOB_READ_
| `abortSignal` | No | An [AbortSignal](https://developer.mozilla.org/en-US/docs/Web/API/AbortSignal) to cancel the operation

`del()` returns a `void` response. A delete action is always successful if the blob url exists. A delete action won't throw if the blob u

## Get blob metadata

This example creates a Function that returns a blob object's metadata.

```ts filename="app/get-blob/route.ts" framework=nextjs-app
import { head } from '@vercel/blob';

export async function GET(request: Request) {
  const { searchParams } = new URL(request.url);
  const blobUrl = searchParams.get('url');
  const blobDetails = await head(blobUrl);

  return Response.json(blobDetails);
}
```

```js filename="app/get-blob/route.js" framework=nextjs-app
import { head } from '@vercel/blob';

export async function GET(request) {
  const { searchParams } = new URL(request.url);
  const blobUrl = searchParams.get('url');
  const blobDetails = await head(blobUrl);

  return Response.json(blobDetails);
}
```

```ts filename="app/get-blob/route.ts" framework=nextjs
import { head } from '@vercel/blob';

export async function GET(request: Request) {
  const { searchParams } = new URL(request.url);
  const blobUrl = searchParams.get('url');
  const blobDetails = await head(blobUrl);

  return Response.json(blobDetails);
}
```

```js filename="app/get-blob/route.js" framework=nextjs
import { head } from '@vercel/blob';

export async function GET(request) {
  const { searchParams } = new URL(request.url);
  const blobUrl = searchParams.get('url');
```

```
  const blobDetails = await head(blobUrl);

  return Response.json(blobDetails);
}
```

```ts filename="/api/blob.ts" framework=other
import { head } from '@vercel/blob';

export async function GET(request: Request) {
  const { searchParams } = new URL(request.url);
  const blobUrl = searchParams.get('url');
  const blobDetails = await head(blobUrl);

  return Response.json(blobDetails);
}
```

```js filename="/api/blob.js" framework=other
import { head } from '@vercel/blob';

export async function GET(request) {
  const { searchParams } = new URL(request.url);
  const blobUrl = searchParams.get('url');
  const blobDetails = await head(blobUrl);

  return Response.json(blobDetails);
}
```

### `head()`

The `head` method returns a blob object's metadata.

It accepts the following parameters:

- `urlOrPathname`: (Required) A string specifying the URL or pathname of the blob object to read.
- `options`: (Optional) A `JSON` object with the following optional parameter:

| Parameter | Required | Values |
| --- | --- | --- |
| `token` | No | A string specifying the read-write token to use when making requests. It defaults to `process.env.BLOB_READ_
| `abortSignal` | No | An [AbortSignal](https://developer.mozilla.org/en-US/docs/Web/API/AbortSignal) to cancel the operation

`head()` returns one of the following:

- a `JSON` object with the requested blob object's metadata
- throws a `BlobNotFoundError` if the blob object was not found

## List blobs

This example creates a Function that returns a list of blob objects in a Blob store.

```ts filename="app/get-blobs/route.ts" framework=nextjs-app
import { list } from '@vercel/blob';

export async function GET(request: Request) {
  const { blobs } = await list();
  return Response.json(blobs);
}
```

```js filename="app/get-blobs/route.js" framework=nextjs-app
import { list } from '@vercel/blob';

export async function GET(request) {
  const { blobs } = await list();
  return Response.json(blobs);
}
```

```ts filename="app/get-blobs/route.ts" framework=nextjs
import { list } from '@vercel/blob';

export async function GET(request: Request) {
  const { blobs } = await list();
  return Response.json(blobs);
}
```

```js filename="app/get-blobs/route.js" framework=nextjs
import { list } from '@vercel/blob';

export async function GET(request) {
  const { blobs } = await list();
  return Response.json(blobs);
}
```

```ts filename="api/blob.ts" framework=other
import { list } from '@vercel/blob';

export async function GET(request: Request) {
  const { blobs } = await list();
  return Response.json(blobs);
}
```

```js filename="api/blob.js" framework=other
import { list } from '@vercel/blob';
```

```
export async function GET(request) {
  const { blobs } = await list();
  return Response.json(blobs);
}
```

### `list()`

The `list` method returns a list of blob objects in a Blob store.

It accepts the following parameters:

- `options`: (Optional) A `JSON` object with the following optional parameters:

| Parameter | Required | Values
| --- | --- | ---
| `token` | No | A string specifying the read-write token to use when making requests. It defaults to `process.env.BLOB_READ_
| `limit` | No | A number specifying the maximum number of blob objects to return. It defaults to 1000
| `prefix` | No | A string used to filter for blob objects contained in a specific folder assuming that the folder name was us
| `cursor` | No | A string obtained from a previous `list` response to be used for reading the next page of results
| `mode` | No | A string specifying the response format. Can either be `expanded` (default) or `folded`. In `folded` mode al
| `abortSignal` | No | An [AbortSignal](https://developer.mozilla.org/en-US/docs/Web/API/AbortSignal) to cancel the operation

`list()` returns a `JSON` object in the following format:

### Pagination

For a long list of blob objects (the default list `limit` is 1000), you can use the `cursor` and `hasMore` parameters to paginate through

### Folders

To retrieve the folders from your blob store, alter the `mode` parameter to modify the response format of the list operation.
The default value of `mode` is `expanded`, which returns all blobs in a single array of objects.

Alternatively, you can set `mode` to `folded` to roll up all blobs located inside a folder into a single entry.
These entries will be included in the response as `folders`. Blobs that are not located in a folder will still be returned in the blobs p

By using the `folded` mode, you can efficiently retrieve folders and subsequently list the blobs inside them by using the returned `folde
Omitting the `prefix` parameter entirely, will return all folders in the root of your store. Be aware that the blobs pathnames and the fo

## Copy a blob

This example creates a Function that copies an existing blob to a new path in the store.

```ts filename="app/copy-blob/route.ts" framework=nextjs-app
import { copy } from '@vercel/blob';

export async function PUT(request: Request) {
  const form = await request.formData();

  const fromUrl = form.get('fromUrl') as string;
  const toPathname = form.get('toPathname') as string;

  const blob = await copy(fromUrl, toPathname, { access: 'public' });

  return Response.json(blob);
}
```

```js filename="app/copy-blob/route.js" framework=nextjs-app
import { copy } from '@vercel/blob';

export async function PUT(request) {
  const form = await request.formData();

  const fromUrl = form.get('fromUrl');
  const toPathname = form.get('toPathname');

  const blob = await copy(fromUrl, toPathname, { access: 'public' });

  return Response.json(blob);
}
```

```ts filename="app/copy-blob/route.ts" framework=nextjs
import { copy } from '@vercel/blob';

export async function PUT(request: Request) {
  const form = await request.formData();

  const fromUrl = form.get('fromUrl') as string;
  const toPathname = form.get('toPathname') as string;

  const blob = await copy(fromUrl, toPathname, { access: 'public' });

  return Response.json(blob);
}
```

```js filename="app/copy-blob/route.js" framework=nextjs
import { copy } from '@vercel/blob';

export async function PUT(request) {
  const form = await request.formData();

  const fromUrl = form.get('fromUrl');
  const toPathname = form.get('toPathname');

  const blob = await copy(fromUrl, toPathname, { access: 'public' });
```

```
  return Response.json(blob);
}
```

```ts filename="api/copy-blob.ts" framework=other
import { copy } from '@vercel/blob';

export async function PUT(request: Request) {
  const form = await request.formData();

  const fromUrl = form.get('fromUrl') as string;
  const toPathname = form.get('toPathname') as string;

  const blob = await copy(fromUrl, toPathname, { access: 'public' });

  return Response.json(blob);
}
```

```js filename="api/copy-blob.js" framework=other
import { copy } from '@vercel/blob';

export async function PUT(request) {
  const form = await request.formData();

  const fromUrl = form.get('fromUrl');
  const toPathname = form.get('toPathname');

  const blob = await copy(fromUrl, toPathname, { access: 'public' });

  return Response.json(blob);
}
```

### `copy()`

The `copy` method copies an existing blob object to a new path inside the blob store.

The `contentType` and `cacheControlMaxAge` will not be copied from the source blob. If the values should be carried over to the copy, they

Contrary to `put()`, `addRandomSuffix` is false by default. This means no automatic random id suffix is added to your blob url, unless yo

It accepts the following parameters:

- `fromUrlOrPathname`: (Required) A blob URL or pathname identifying an already existing blob
- `toPathname`: (Required) A string specifying the new path inside the blob store. This will be the base value of the return URL
- `options`: (Required) A `JSON` object with the following required and optional parameters:

| Parameter | Required | Values |
| --- | --- | --- |
| `access` | Yes | `public` |
| `contentType` | No | A string indicating the [media type](https://developer.mozilla.org/docs/Web/HTTP/Headers/Content-Type |
| `token` | No | A string specifying the token to use when making requests. It defaults to `process.env.BLOB_READ_WRIT |
| `addRandomSuffix` | No | A boolean specifying whether to add a random suffix to the pathname. It defaults to `false`. |
| `cacheControlMaxAge` | No | A number in seconds to configure the edge and browser cache. Defaults to one year. See the [caching]( |
| `abortSignal` | No | An [AbortSignal](https://developer.mozilla.org/en-US/docs/Web/API/AbortSignal) to cancel the operatio |

`copy()` returns a `JSON` object with the following data for the copied blob object:

An example blob is:

```json
{
  "pathname": "profilesv1/user-12345-copy.txt",
  "contentType": "text/plain",
  "contentDisposition": "attachment; filename=\"user-12345-copy.txt\"",
  "url": "https://ce0rcu23vrrdzqap.public.blob.vercel-storage.com/profilesv1/user-12345-copy.txt",
  "downloadUrl": "https://ce0rcu23vrrdzqap.public.blob.vercel-storage.com/profilesv1/user-12345-copy.txt?download=1"
}
```

## Client uploads

As seen in the [client uploads quickstart docs](/docs/storage/vercel-blob/client-upload), you can upload files directly from clients (lik

All client uploads related methods are available under `@vercel/blob/client`.

### `upload()`

The `upload` method is dedicated to [client uploads](/docs/storage/vercel-blob/client-upload). It fetches a client token on your server u

```js
upload(pathname, body, options);
```

It accepts the following parameters:

- `pathname`: (Required) A string specifying the base value of the return URL
- `body`: (Required) A blob object as `ReadableStream`, `String`, `ArrayBuffer` or `Blob` based on these [supported body types](https://d
- `options`: (Required) A `JSON` object with the following required and optional parameters:

| Parameter | Required | Values |
| --- | --- | --- |
| `access` | Yes | `public` |
| `contentType` | No | A string indicating the [media type](https://developer.mozilla.org/docs/Web/HTTP/Headers/Content-Type). |
| `handleUploadUrl` | Yes* | A string specifying the route to call for generating client tokens for [client uploads](/docs/storage/v |
| `clientPayload` | No | A string to be sent to your `handleUpload` server code. Example use-case: attaching the post id an imag |
| `multipart` | No | Pass `multipart: true` when uploading large files. It will split the file into multiple parts, upload t |
| `abortSignal` | No | An [AbortSignal](https://developer.mozilla.org/en-US/docs/Web/API/AbortSignal) to cancel the operation |

| `onUploadProgress` | No | Callback to track upload progress: `onUploadProgress({loaded: number, total: number, percentage: number

`upload()` returns a `JSON` object with the following data for the created blob object:

```ts
{
  pathname: string;
  contentType: string;
  contentDisposition: string;
  url: string;
  downloadUrl: string;
}
```

An example `url` is:

```
https://ce0rcu23vrrdzqap.public.blob.vercel-storage.com/profilesv1/user-12345-NoOVGDVcqSPc7VYCUAGnTzLTG2qEM2.txt
```

### `handleUpload()`

A server-side route helper to manage client uploads, it has two responsibilities:

1. Generate tokens for client uploads
2. Listen for completed client uploads, so you can update your database with the URL of the uploaded file for example

```js
handleUpload(options);
```

It accepts the following parameters:

- `options`: (Required) A `JSON` object with the following parameters:

| Parameter | Required | Values
| --- | --- | ---
| `token` | No | A string specifying the read-write token to use when making requests. It
| `request` | Yes | An `IncomingMessage` or `Request` object to be used to determine the act
| [`onBeforeGenerateToken`](#onbeforegeneratetoken) | Yes | A function to be called right before generating client tokens for client
| [`onUploadCompleted`](#onuploadcompleted) | Yes | A function to be called by Vercel Blob when the client upload finishes.
| `body` | Yes | The request body

`handleUpload()` returns:

```ts
Promise<
  | { type: 'blob.generate-client-token'; clientToken: string }
  | { type: 'blob.upload-completed'; response: 'ok' }
>;
```

#### `onBeforeGenerateToken()`

The `onBeforeGenerateToken` function receives the following arguments:

- `pathname`: The destination path for the blob
- `clientPayload`: A string payload specified on the client when calling `upload()`
- `multipart`: A boolean specifying whether the file is a multipart upload.

The function must return an object with the following properties:

| Parameter | Required | Values
| --- | --- | ---
| `addRandomSuffix` | No | A boolean specifying whether to add a random suffix to the `pathname`. It defaults to `false`. **We
| `allowedContentTypes` | No | An array of strings specifying the [media type](https://developer.mozilla.org/docs/Web/HTTP/Headers/
| `maximumSizeInBytes` | No | A number specifying the maximum size in bytes that can be uploaded. The maximum is 5TB.
| `validUntil` | No | A number specifying the timestamp in ms when the token will expire. By default, it's now + 1 hour.
| `allowOverwrite` | No | A boolean to allow overwriting blobs. By default an error will be thrown if you try to overwrite a b
| `cacheControlMaxAge` | No | A number in seconds to configure how long Blobs are cached. Defaults to one month. Cannot be set to
| `callbackUrl` | No | A string specifying the URL that Vercel Blob will call when the upload completes. See [client upload
| `tokenPayload` | No | A string specifying a payload to be sent to your server on upload completion.

#### `onUploadCompleted()`

The `onUploadCompleted` function receives the following arguments:

- `blob`: The blob that was uploaded. See the return type of [`put()`](#put) for more details.
- `tokenPayload`: The payload that was defined in the [`onBeforeGenerateToken()`](#onbeforegeneratetoken) function.

### Client uploads routes

Here's an example Next.js App Router route handler that uses `handleUpload()`:

```ts filename="app/api/post/upload/route.ts"
import { handleUpload, type HandleUploadBody } from '@vercel/blob/client';
import { NextResponse } from 'next/server';

// Use-case: uploading images for blog posts
export async function POST(request: Request): Promise<NextResponse> {
  const body = (await request.json()) as HandleUploadBody;

  try {
    const jsonResponse = await handleUpload({
      body,
      request,
      onBeforeGenerateToken: async (pathname, clientPayload) => {
        // Generate a client token for the browser to upload the file
        // ⚠️ Authenticate and authorize users before generating the token.
        // Otherwise, you're allowing anonymous uploads.
```

```
        // ⚠️ When using the clientPayload feature, make sure to validate it
        // otherwise this could introduce security issues for your app
        // like allowing users to modify other users' posts

        return {
          allowedContentTypes: [
            // optional, default to all content types
            'image/jpeg',
            'image/png',
            'image/webp',
            'text/*',
          ],
          // callbackUrl: 'https://example.com/api/avatar/upload',
          // optional, `callbackUrl` is automatically computed when hosted on Vercel
        };
      },
      onUploadCompleted: async ({ blob, tokenPayload }) => {
        // Get notified of client upload completion
        // ⚠️ This will not work on `localhost` websites,
        // Use ngrok or similar to get the full upload flow

        console.log('blob upload completed', blob, tokenPayload);

        try {
          // Run any logic after the file upload completed,
          // If you've already validated the user and authorization prior, you can
          // safely update your database
        } catch (error) {
          throw new Error('Could not update post');
        }
      },
    });

    return NextResponse.json(jsonResponse);
  } catch (error) {
    return NextResponse.json(
      { error: error instanceof Error ? error.message : String(error) },
      { status: 400 }, // The webhook will retry 5 times waiting for a 200
    );
  }
}
```

## Handling errors

When you make a request to the SDK using any of the above methods, they will return an error if the request fails due to any of the follo

- Missing required parameters
- An invalid token or a token that does have access to the Blob object
- Suspended Blob store
- Blob file or Blob store not found
- Unforeseen or unknown errors

To catch these errors, wrap your requests with a `try/catch` statement as shown below:

---
title: "Attack Challenge Mode"
description: "Learn how to use Attack Challenge Mode to help control who has access to your site when it"
last_updated: "2026-01-16T02:19:35.926Z"
source: "https://vercel.com/docs/vercel-firewall/attack-challenge-mode"
---

# Attack Challenge Mode

Attack Challenge Mode is a security feature that protects your site during DDoS attacks. When enabled, visitors must complete a [security

The Vercel Firewall automatically [mitigates against DDoS attacks](/docs/security/ddos-mitigation), but Attack Challenge Mode provides an

Attack Challenge Mode is available for [free](#pricing) on all [plans](/docs/plans) and requests blocked by challenge mode do not count t

## Known bots support

Vercel maintains and continuously updates a comprehensive directory of known legitimate bots from across the internet. Attack Challenge M

Review [Verified bots](/docs/bot-management#verified-bots) for examples of bot categories and services that are automatically allowed. [I

Vercel's bot directory is regularly updated to include new legitimate services as they emerge, ensuring your SEO, analytics, integrations

> **💡 Note:** To block specific known bots instead of allowing them through, you can create
> a [Custom Rule](/docs/security/vercel-waf/custom-rules) that matches their
> User Agent.

## Internal requests

When Attack Challenge Mode is enabled, requests from your own [Functions](/docs/functions) and [Cron Jobs](/docs/cron-jobs) are automatic

For example, if you have multiple projects in your Vercel account:

- Your projects can communicate with each other without being challenged
- Only requests from outside your account will be challenged
- Each Vercel account has its own secure boundary

Other Vercel accounts cannot bypass Attack Challenge Mode on your projects. The security is strictly enforced per account, ensuring that

## Enabling attack challenge mode

While Vercel's Firewall [automatically monitors for and mitigates attacks](/docs/security/ddos-mitigation#what-to-do-in-case-of-a-ddos-at

To enable:

1. Select your project from the [Dashboard](/dashboard).
2. Navigate to the **Firewall** tab.
3. Click **Bot Management**.
4. Under **Attack Challenge Mode**, select **Enable**.

All traffic initiated by web browsers, including API traffic, is supported. For example, a Next.js frontend calling a Next.js API in the

> **💡 Note:** Standalone APIs, other backend frameworks, and non-recognized automated
> services may not be able to pass challenges and could be blocked. If you need
> more control over what traffic is challenged, consider using [Custom Rules
> with the Vercel WAF](/docs/security/vercel-waf/custom-rules).

## How long to keep it enabled

Attack Challenge Mode can be safely used for extended periods without affecting search engine indexing or webhook functionality. However,

## Disabling attack challenge mode

When you no longer need the additional protection:

1. Select your project from the [Dashboard](/dashboard)
2. Navigate to the **Firewall** tab.
3. Click **Bot Management**.
4. Under **Attack Challenge Mode**, select **Disable**.

## Challenging with custom rules

For more granular control, define a [Custom Rule with the Vercel WAF](/docs/security/vercel-waf/custom-rules) to challenge specific web t

## Search indexing

Search engine crawlers like Googlebot are automatically allowed through Attack Challenge Mode without being challenged. This means enabli

## Pricing

Attack Challenge Mode is available for free on all plans.

All mitigations by Attack Challenge Mode are free and unlimited, and there are zero costs associated with traffic blocked by Attack Chall

---
title: "DDoS Mitigation"
description: "Learn how the Vercel Firewall mitigates against DoS and DDoS attacks"
last_updated: "2026-01-16T02:19:35.937Z"
source: "https://vercel.com/docs/vercel-firewall/ddos-mitigation"
---

# DDoS Mitigation

Vercel provides automatic DDoS mitigation for all deployments, regardless of your plan. We block incoming traffic if we identify abnormal

> **💡 Note:** Vercel does not charge customers for traffic that gets blocked with DDoS
> mitigation.

It works by:

- **Monitoring traffic:** Vercel Firewall continuously analyzes incoming traffic to detect signs of DDoS attacks. This helps to identify
- **Blocking traffic:** Vercel Firewall filters out malicious traffic while allowing legitimate requests to pass through
- **Scaling resources:** During a DDoS attack, Vercel Firewall dynamically scales resources to absorb the increased traffic, preventing y

If you need further control over incoming traffic, you can temporarily enable [Attack Challenge Mode](/docs/attack-challenge-mode) to cha

Learn more about [DoS, DDoS and the Open System Interconnection model](/docs/security/firewall-concepts#understanding-ddos).

## Responding to DDoS attacks

Vercel mitigates against L3, L4, and L7 DDoS attacks regardless of the plan you are on. The Vercel Firewall uses hundreds of signals and

However, there are other steps you can take to protect your site during DDoS attacks:

- Enable [Attack Challenge Mode](/docs/attack-challenge-mode) to challenge all visitors to your site. This is a temporary measure and pro
- You can set up [IP Blocking](/docs/security/vercel-waf/ip-blocking) to block specific IP addresses from accessing your projects. Enterp
- You can add [Custom Rules](/docs/security/vercel-waf/custom-rules) to deny or challenge specific traffic to your site based on the cond
- You can also use Middleware to [block requests](https://github.com/vercel/examples/tree/main/edge-middleware/geolocation-country-block)

Pro teams can [set up Spend Management](/docs/spend-management#managing-your-spend-amount) to get notified or to automatically take actio

## Bypass System-level Mitigations

While Vercel's system-level mitigations (such as [DDoS protection](/docs/security/ddos-mitigation)) safeguards your websites and applicat

To temporarily pause all automatic mitigations for a specific project:

1. Click the menu button with the ellipsis icon  at the top right of the **Firewall** tab for your project.
2. Select **Pause System Mitigations**.
3. Review the warning in the **Pause System Mitigations** dialog and confirm that you would like to pause all automatic mitigations for t

To resume the automatic mitigations **before** the 24 hour period ends:

1. Click the menu button with the ellipsis icon  at the top right of the **Firewall** tab for your project.
2. Select **Resume System Mitigations**.
3. Select **Resume** from the **Resume System Mitigations** dialog.

> **⚠️ Warning:** You are responsible for all usage fees incurred when using this feature,
> including illegitimate traffic that may otherwise have been blocked.

### System Bypass Rules

In situations where you need a more granular and permanent approach, you can use [System Bypass Rules](/docs/security/vercel-waf/system-by

This feature is available for Pro and Enterprise customers. Learn how to [set up a System Bypass rule](/docs/security/vercel-waf/system-by

## Dedicated DDoS support for Enterprise teams

For larger, distributed attacks on Enterprise Teams, we collaborate with you to keep your site(s) online during an attack. Automated prev

## DDoS and billing

[Vercel automatically mitigates against L3, L4, and L7 DDoS attacks](/docs/security/ddos-mitigation) at the platform level for all plans.

Usage will be incurred for requests that are successfully served prior to us automatically mitigating the event. Usage will also be incur

For an additional layer of security, we recommend that you enable [Attack Challenge Mode](/docs/attack-challenge-mode) when you are under

You can monitor usage in the [Vercel Dashboard](/dashboard) under the **Usage** tab, although you will [receive notifications](/docs/noti

---
title: "Using the REST API with the Firewall"
description: "Learn how to interact with the security endpoints of the Vercel REST API programmatically."
last_updated: "2026-01-16T02:19:35.942Z"
source: "https://vercel.com/docs/vercel-firewall/firewall-api"
---

# Using the REST API with the Firewall

The security section of the [Vercel REST API](/docs/rest-api) allows you to programmatically interact with some of the functionality of t

You can use the REST API programmatically as follows:

- Install the [Vercel SDK](/docs/rest-api/sdk) and use the [security methods](https://github.com/vercel/sdk/blob/HEAD/docs/sdks/security/
- [Call the endpoints directly](/docs/rest-api) and use the [security endpoints](/docs/rest-api/reference/endpoints/security).

To define firewall rules in code that apply across multiple projects, you can use the [Vercel Terraform provider](https://registry.terraf

After [setting up Terraform](/kb/guide/integrating-terraform-with-vercel), you can use the following rules:

- [vercel_firewall_config](https://registry.terraform.io/providers/vercel/vercel/latest/docs/resources/firewall_config)
- [vercel_firewall_bypass](https://registry.terraform.io/providers/vercel/vercel/latest/docs/resources/firewall_bypass)

## Examples

Learn how to use some of these endpoints with specific examples for the Vercel WAF.

- [Challenge `cURL` requests](/kb/guide/challenge-curl-requests)
- [Challenge cookieless requests on a specific path](/kb/guide/challenge-cookieless-requests-on-a-specific-path)
- [Deny non-browser traffic or blocklisted ASNs](/kb/guide/deny-non-browser-traffic-or-blocklisted-asns)
- [Deny traffic from a set of IP addresses](/kb/guide/deny-traffic-from-a-set-of-ip-addresses)
- [Vercel Firewall Terraform configuration](/kb/guide/firewall-terraform-configuration)

---
title: "Firewall concepts"
description: "Understand the fundamentals behind the Vercel Firewall."
last_updated: "2026-01-16T02:19:35.971Z"
source: "https://vercel.com/docs/vercel-firewall/firewall-concepts"
---

# Firewall concepts

## How Vercel secures requests

To safeguard your application against malicious activity, Vercel's platform-wide firewall is the first line of defense, inspecting reques

If allowed to go through, the request is subject to the rules that you configured with the [Web Application Firewall (WAF)](/docs/securit

If you [enabled a persistent action](/docs/security/vercel-waf/custom-rules#persistent-actions) for a WAF rule and it blocks the request,

## Firewall actions

The Vercel Firewall allows several possible actions to be taken when traffic matches a rule. These actions, that can be taken by custom r

### Log

The log action allows you to monitor and record specific traffic patterns without affecting the request. When a request matches a rule wi

- The request is allowed to proceed normally.
- Details about the request are logged and displayed in the Firewall and Monitoring tabs, and sent to log drains for analysis.
- There is no impact on the visitor's experience.

This is useful for monitoring suspicious patterns or gathering data about specific types of traffic before implementing stricter actions.

### Deny

The deny action blocks requests immediately when they match a rule. When a request is denied:

- A `403 Forbidden` response is returned.
- The request does not reach your application.
- Usage is incurred only for the edge request and ingress [Fast Data Transfer](/docs/manage-cdn-usage#fast-data-transfer) which are requi

This is the most restrictive action and you should use it for known malicious traffic patterns or IP addresses.

### Challenge

A security challenge verifies that incoming traffic originates from a real web browser with JavaScript capabilities. Only human visitors

Use the challenge action when you want to block automated traffic while allowing legitimate users to access your content. It offers a mid

When the challenge action is applied:

- ### Initial challenge
  During this process, visitors see a **Vercel Security Checkpoint** screen:
  - The browser must execute JavaScript code to prove it's a real browser.
  - The code computes and submits a challenge solution.
  - The system validates browser characteristics to prevent automated tools from passing.
  - If the challenge succeeds, the [WAF](/docs/vercel-firewall/vercel-waf) validates the request as a legitimate browser client and conti
  - If the challenge fails, the request is blocked before reaching your application.
  The checkpoint page localizes to a language based on the visitor's browser settings and respects their preferred color scheme, ensuring

- ### Challenge session
  - Upon successful verification, a challenge session is created in the browser.
  - Sessions are valid for 1 hour.
  - All subsequent requests within the session are allowed.
  - Challenge sessions are tied to the browser that completed the challenge, ensuring secure session management.
  - After session expiration, the client must re-solve the challenge.

#### Challenge subrequests and APIs

If your application makes additional requests (e.g., API calls) during a valid session, they automatically succeed. This is particularly

#### Challenge limitations

- API routes that are protected by a challenge rule can only be accessed within a valid challenge session.
- Direct API calls (e.g., from scripts, cURL, or Postman) will fail if they require challenge validation.
- Direct API calls from outside a valid challenge session will not succeed.
- If a user hasn't completed a challenge session through your website first, they cannot access challenged API routes.
- Automated tools and scripts cannot establish challenge sessions. For legitimate automation needs, use [Bypass](#bypass) to allow specif

### Bypass

The bypass action allows specific traffic to skip any subsequent firewall rules. When a request matches a bypass rule:

- For custom rule bypasses, the request is allowed through any custom or managed rules.
- For system bypasses, the request is allowed through any system-level mitigations.
- The request proceeds directly to your application.

This is useful for trusted traffic sources, internal tools, or critical services that should never be blocked.

## Understanding DDoS

A Denial of Service (DoS) attack happens when one device attempts to exhaust the resources of a system using methods such as sending a la

A Distributed Denial of Service (DDoS) attack happens when multiple connected devices are used to simultaneously overwhelm a site with ta

In addition to built-in systems like [rate limits](/docs/limits#rate-limits), you can protect yourself against such attacks with [WAF cus

### Open System Interconnection (OSI) model

The OSI model is a concept that outlines the different communication steps of a networking system. Different attack types can target diff

DDoS attacks target either the [network](#layer-3-ddos) (layer 3), the [transport](#layer-4-ddos) (layer 4) or the [application](#layer-7

### Layer 3 DDoS

The goal of a layer 3 (L3) DDoS attack is to slow down and ultimately crash applications, servers, and entire networks. These attacks are

### Layer 4 DDoS

The goal of a layer 4 (L4) DDoS attack is to crash and slow down applications. They target the 3-way-handshake used to establish a reliab

### Layer 7 DDoS

The goal of a Layer 7 (L7) DDoS attack is to crash and slow down software at the application layer by targeting protocols such as HTTP, w

## JA3 and JA4 TLS fingerprints

Vercel Firewall leverages [JA3](#ja3) and [JA4](#ja4) TLS fingerprints to identify and restrict malicious traffic. TLS fingerprints allow

### TLS fingerprinting

TLS fingerprinting is a process used to identify and categorize encrypted network traffic.

It creates a unique identifier from the details of a [TLS client hello packet](https://serializethoughts.com/2014/07/27/dissecting-tls-cl

- TLS fingerprints allow the unique identification of user session
- JA3 and JA4 transform the TLS handshake details into a hash
- The hash is used as a fingerprint to monitor and restrict access
- The hash can then be read from your Functions through the request headers

### Why track TLS fingerprints?

Controlling access by TLS fingerprint allows us to mitigate malicious actors that use sophisticated methods of attack.
For example, a DDoS attack that is spread across multiple user agents, IPs, or geographic locations might share the same TLS fingerprint.
With fingerprinting, the Vercel Firewall can block all of the traffic that matches that TLS fingerprint.

#### JA4

JA4 is part of the [JA4+ suite](https://github.com/FoxIO-LLC/ja4?tab=readme-ov-file#ja4-details). It offers a more granular and flexible

With JA4, it's possible to identify, track, and categorize server-side encrypted network traffic. This is crucial in detecting and mitiga

#### JA3

JA3 is a tool that uses TLS fingerprinting to track and identify potential security threats. It specifically focuses on the details of th

#### Monitor JA4 signatures

In the **Allowed Requests** view of the [Vercel WAF monitoring page](/docs/security/vercel-waf#traffic-monitoring), you can group the web

### Request headers

The following headers are sent to each deployment and can be used to process the [request](https://developer.mozilla.org/en-US/docs/Web/A

#### `x-vercel-ja4-digest` (preferred)

Unique client fingerprint hash generated by the JA4 algorithm. JA4 is preferred as it offers a more granular and flexible approach to net

#### `x-vercel-ja3-digest`

Unique client fingerprint hash generated by the JA3 algorithm.

---
title: "Firewall Observability"
description: "Learn how firewall traffic monitoring and alerts help you react quickly to potential security threats."
last_updated: "2026-01-16T02:19:35.997Z"
source: "https://vercel.com/docs/vercel-firewall/firewall-observability"
---

# Firewall Observability

The project **Firewall** page of your Vercel dashboard provides a consolidated view of traffic and event analysis across Vercel's [platfo

## Overview

The **Overview** page provides a summary of active rules with associated events and mitigations that apply to your project. This page dis

The default time period for the traffic view is the past hour. From a drop-down on the top left, you can adjust this time period to show

The **Alerts** section displays recent firewall alerts such as detected attacks against your project. When large volume attacks are detec

The **Rules** section breaks down incoming traffic by the rule that applied. This gives you a quick view of which rules are protecting yo

The **Events** section provides insight into actions Vercel's platform-wide firewall has applied to your project. Selected events can be

The **Denied IPs** section shows the most commonly blocked malicious sources.

Discrete events and alerts can be inspected from the Overview page to view request and time data from malicious sources.

## Traffic

The **Traffic** page lets you drill down into top traffic sources and signals. You can view all traffic or have the following ways to fil

- By a specific rule with the drop down above the graph
- By an action using the action tab within the graph to see only the traffic that matched this filter

You can also review incoming requests grouped by the following dimensions:

- **Client IP Addresses**: View traffic grouped by source IP address
- **User Agents**: Inspect clients by user agent strings
- **Request Paths**: Monitor traffic patterns across different URL paths
- **ASNs (Autonomous System Numbers)**: Track traffic by source network provider
- **JA4 (TLS Fingerprints)**: Identify clients by their [JA4](/docs/vercel-firewall/firewall-concepts#ja4) TLS fingerprints
- **Country**: Geographic distribution of traffic by country

## Firewall Alerts

### How alerts work

To help protect your site effectively, you can configure alerts to be notified of potential security threats and firewall actions. To do

### DDoS attack alerts

When Vercel's [DDoS Mitigation](/docs/security/ddos-mitigation) detects malicious traffic on your site that exceeds 100,000 requests over

To receive notifications from these alerts, you can use one of the following methods:

- Create a [webhook](/docs/webhooks) and subscribe to the URL to receive notifications
  1. Follow the [configure a webhook](/docs/webhooks#configure-a-webhook) guide to create a webhook with the **Attack Detected Firewall E
  2. Subscribe to the created webhook URL
- Use the [Vercel Slack app](https://vercel.com/integrations/slack) to enable notifications for Attack Detected Firewall Events
  1. Add the Slack app for your team by following the [Use the Vercel Slack app](/docs/comments/integrations#use-the-vercel-slack-app) gu
  2. Subscribe your team to DDoS attack alerts using your [`team_id`](/docs/accounts#find-your-team-id)
     - Use the command `/vercel subscribe {team_id} firewall_attack`
  3. Review the [Vercel Slack app command reference](/docs/comments/integrations#vercel-slack-app-command-reference) for additional optio

---
title: "Vercel Firewall"
description: "Learn how Vercel Firewall helps protect your applications and websites from malicious attacks and unauthorized access."
last_updated: "2026-01-16T02:19:36.005Z"
source: "https://vercel.com/docs/vercel-firewall"
---

# Vercel Firewall

The Vercel Firewall is a robust, multi-layered security system designed to protect your applications from a wide range of threats. Every

- [Platform-wide firewall](#platform-wide-firewall): With [DDoS mitigation](/docs/security/ddos-mitigation), it protects against large-sc
- [Web Application Firewall (WAF)](#vercel-waf): A customizable layer for fine-tuning security measures with logic tailored to your needs

### Concepts

Understand the fundamentals:

- How [Vercel protects every request](/docs/security/firewall-concepts#how-vercel-secures-requests).
- Why [DDoS](/docs/security/firewall-concepts#understanding-ddos) needs to be mitigated.
- How the firewall decides [which rule to apply first](#rule-execution-order).

- How the firewall uses [JA3 and JA4 TLS fingerprints](/docs/security/firewall-concepts#ja3-and-ja4-tls-fingerprints) to identify and res

## Rule execution order

The automatic rules of the platform-wide firewall and the custom rules of the WAF work together in the following execution order:

1. [DDoS mitigation rules](/docs/security/ddos-mitigation)
2. [WAF IP blocking rules](/docs/security/vercel-waf/ip-blocking)
3. [WAF custom rules](/docs/security/vercel-waf/custom-rules)
4. [Managed rulesets](/docs/security/vercel-waf/managed-rulesets)

When you have more than one custom rule, you can [customize](/docs/security/vercel-waf/custom-rules#custom-rule-configuration) their orde

## Platform-wide firewall

Vercel provides automated [DDoS mitigation](/docs/security/ddos-mitigation) for all deployments, regardless of the plan that you are on. \

## Vercel WAF

The [Vercel WAF](/docs/security/vercel-waf) complements the platform-wide firewall by allowing you to define custom protection strategies

- [Custom Rules](/docs/security/vercel-waf/custom-rules)
- [IP Blocking](/docs/security/vercel-waf/ip-blocking)
- [Managed Rulesets](/docs/security/vercel-waf/managed-rulesets)
- [Attack Challenge Mode](/docs/attack-challenge-mode)

## Observability

You can use the following tools to [monitor the internet traffic](/docs/vercel-firewall/firewall-observability) at your team or project l

- The [Monitoring](/docs/observability/monitoring) feature at the team level allows you to create [queries](/docs/observability/monitorin
- The **Firewall** tab of the Vercel dashboard on every project allows you to monitor the internet traffic to your deployments with a [tr
- [Firewall alerts](/docs/vercel-firewall/firewall-observability#firewall-alerts) allow you to react quickly to potential security threat
- Use [Log Drains](/docs/drains/using-drains) to send your application logs to a Security Information and Event Management (SIEM) system.

---
title: "WAF Custom Rules"
description: "Learn how to add and manage custom rules to configure the Vercel Web Application Firewall (WAF)."
last_updated: "2026-01-16T02:19:36.037Z"
source: "https://vercel.com/docs/vercel-firewall/vercel-waf/custom-rules"
---

# WAF Custom Rules

You can [configure](#custom-rule-configuration) specific rules to log, deny, challenge, bypass, or [rate limit](/docs/security/vercel-waf

[Get started](#get-started) by reviewing the [Best practices for applying rules](#best-practices-for-applying-rules) section.

## Access roles

- You need to be a [Developer](/docs/rbac/access-roles#developer-role) or viewer ([Viewer Pro](/docs/rbac/access-roles#viewer-pro-role) o
- You need to be a [Project administrator](/docs/rbac/access-roles#project-administrators) or [Team member](/docs/rbac/access-roles#membe

## Custom Rule configuration

You can create multiple Custom Rules for the same project. Each rule can perform the following actions according to one or more logical c

- [log](/docs/vercel-firewall/firewall-concepts#log)
- [deny](/docs/vercel-firewall/firewall-concepts#deny)
- [challenge](/docs/vercel-firewall/firewall-concepts#challenge)
- [bypass](/docs/vercel-firewall/firewall-concepts#bypass)
- redirect

You can **save**, **delete**, or **disable** a rule at any time and these actions have immediate effect. You also have the ability to re-

## Custom Rule execution

When a rule denies or challenges the traffic to your site and the client has not previously solved the challenge (in the case of challeng

After a **Log** rule runs, the rule execution continues. If no other rule matches and acts on the request, the **Log** rule that is last r

When you apply a [rate limiting](/docs/security/vercel-waf/rate-limiting) rule, you need to include a follow up action that will log, den

## Persistent actions

When a custom rule blocks a client's request, future requests that do not match the rule's condition from the same client, are allowed th

With persistent actions, you can automatically block potential bad actors by adding a time-based block to the **Challenge** or **Deny** a

Notes about this time-based block:

- It is applied to the IP address of the client that originally triggered the rule to match.
- It happens before the firewall processes the request, so that none of the requests blocked by persistent actions count towards your [CD

### Enable persistent actions

You can enable persistent actions for any challenge, deny or rate limit action when you create or edit a custom rule. From your project's

1. Select the **Firewall** tab followed by **Configure** on the top right of the Firewall overview page.
2. Select a Custom Rule you would like to edit from the list or select **+ New Rule** and follow the [steps](#get-started) for configurin

When you select challenge, deny or rate limit for the [action](/docs/vercel-firewall/vercel-waf/rule-configuration#actions) dropdown (**T

3. Select a time value from the available options
4. Remove the timeframe (If you don't want to enable persistent actions)

Once you're happy with the changes:

5. Select **Save Rule** to apply it

6. Apply the changes with the **Review Changes** button

## Best practices for applying rules

To ensure your Custom Rule behaves as intended:

1. Test a Custom Rule by setting it up with a **log** action
2. Observe the 10-minute live traffic to check the behavior
3. Update the Custom Rule condition if needed. Once you're happy with the behavior, update the rule with a
   **challenge**, **deny**, or **bypass**, or **rate limit** action

## Get started

Learn how to create, test, and apply a Custom Rule.

1. From your dashboard, select the project that you'd like to configure a rule for and then select the **Firewall** tab
2. Select **Configure** on the top right of the Firewall overview page
3. Select **+ New Rule**
4. Type a name to help you identify the purpose of this rule for future reference
5. In the **Configure** section, add as many **If** conditions as needed. For each condition you add, choose how you will combine it with
6. Select **Log** for the **Then** action
   - For **Rate Limit**, review [WAF Rate Limiting](/docs/security/vercel-waf/rate-limiting#get-started)
7. Select **Save Rule** to apply it
8. Apply the changes:
   - When you make any change, you will see a **Review Changes** button appear or update on the top right with the number of changes requ
   - Select **Review Changes** and review the changes to be applied
   - Select **Publish** to apply the changes to your production deployment
9. Go to the Firewall overview page, select your Custom Rule from the traffic grouping drop-down and select the paramater(s) related to t
10. If you are satisfied with the traffic behavior, select **Configure**
11. Select the Custom Rule that you created
12. Update the **Then** action to **Challenge**, **Deny** or **Bypass** as needed
13. Select **Save Rule** to apply it
14. Apply the changes with the **Review Changes** button

Review [Common Examples](/docs/security/vercel-waf/examples) for the application of specific rules in common situations.

## Configuration in vercel.json

You can configure custom WAF rules directly in your `vercel.json` file using the `routes` property. This allows you to define firewall ru

### Supported actions

When configuring WAF rules in `vercel.json`, you can use the following actions:

- **challenge**: Challenge the request with a security check
- **deny**: Block the request entirely

> **💡 Note:** This is a subset of the actions available in the dashboard - `log`, `bypass`,
> and `redirect` actions are not supported in `vercel.json` configuration.

### Example configuration

The following example shows how to deny requests that contain a specific header:

```json filename="vercel.json"
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "routes": [
    {
      "src": "/(.*)",
      "has": [
        {
          "type": "header",
          "key": "x-react-router-prerender-data"
        }
      ],
      "mitigate": {
        "action": "deny"
      }
    }
  ]
}
```

In this example:

- The route matches all paths (`/(.*)`)
- The `has` condition checks for the presence of a specific header
- The `mitigate` property specifies the action to take (deny the request)

### Route configuration

For complete documentation on route configuration options, including `has`, `missing`, and other conditional matching properties, see the

---
title: "WAF Examples"
description: "Learn how to use Vercel WAF to protect your site in specific situations."
last_updated: "2026-01-16T02:19:35.989Z"
source: "https://vercel.com/docs/vercel-firewall/vercel-waf/examples"
---

# WAF Examples

| Example | Category |
| --- | --- |
| [Suspicious traffic in specific countries](/kb/guide/suspicious-traffic-in-specific-countries) | [Custom Rule](/docs/se |
| [Emergency redirect](/kb/guide/emergency-redirect) | [Custom Rule](/docs/se |
| [Limit abuse with rate limiting](/kb/guide/limit-abuse-with-rate-limiting) | [Custom Rule](/docs/se |
| [Block AI bots](/docs/vercel-waf/managed-rulesets#configure-ai-bots-managed-ruleset) | [Managed Ruleset](/doc |

| [Block `.php` requests](/kb/guide/block-php-requests) | [Custom Rule](/docs/se
| [Block traffic from a specific IP address](/kb/guide/traffic-spikes) | [IP Blocking](/docs/se
| [Challenge `cURL` requests](/kb/guide/challenge-curl-requests) | [Firewall REST API](/d
| [Challenge cookieless requests on a specific path](/kb/guide/challenge-cookieless-requests-on-a-specific-path) | [Firewall REST API](/d
| [Deny non-browser traffic or blocklisted ASNs](/kb/guide/deny-non-browser-traffic-or-blocklisted-asns) | [Firewall REST API](/d
| [Deny traffic from a set of IP addresses](/kb/guide/deny-traffic-from-a-set-of-ip-addresses) | [Firewall REST API](/d

---
title: "WAF IP Blocking"
description: "Learn how to customize the Vercel WAF to restrict access to certain IP addresses."
last_updated: "2026-01-16T02:19:36.105Z"
source: "https://vercel.com/docs/vercel-firewall/vercel-waf/ip-blocking"
---

# WAF IP Blocking

You can create custom rules to block a specific IP address or multiple IP addresses by [CIDR](# "What is CIDR?"), effectively preventing

Common use cases for IP blocking on Vercel include:

- Blocking known malicious IP addresses
- Preventing competitors or scrapers from accessing your content

In cases such as blocking based on complying with specific laws and regulations or to restrict access to or from a particular geographic

## Access roles

- You need to be a [Developer](/docs/rbac/access-roles#developer-role) or viewer ([Viewer Pro](/docs/rbac/access-roles#viewer-pro-role) o
- You need to be a [Project administrator](/docs/rbac/access-roles#project-administrators) or [Team member](/docs/rbac/access-roles#membe

## Project level IP Blocking

To block an IP address, navigate to the **Firewall** tab of your project and follow these steps:

1. Select **Configure** on the top right of the Firewall overview page
2. Scroll down to the **IP Blocking** section
3. Select the **+ Add IP** button
4. Complete the required **IP Address Or CIDR** and **Host** fields in the **Configure New Domain Protection** modal
   - The host is the domain name of the site you want to block the IP address from accessing. It should match the domain(s) associated wi
   - You can copy this value from the URL of the site you want to block **without the `https` prefix**
   - It must match the exact domain you want to block, for example `my-site.com`, `www.my-site.com` or `docs.my-site.com`
   - You should add an entry for all subdomains that you wish block, such as `blog.my-site.com` and `docs.my-site.com`
5. Select the **Create IP Block Rule** button
6. Apply the changes:
   - When you make any change, you will see a **Review Changes** button appear or update on the top right with the number of changes requ
   - Select **Review Changes** and review the changes to be applied
   - Select **Publish** to apply the changes to your production deployment

## Account-level IP Blocking

### How to add an IP block rule

To block an IP address, you can create an IP Blocking rule in your dashboard:

1. On your Team's [dashboard](/dashboard), navigate to **Settings** and select the **Security** tab
2. On the **IP Blocking** section, select **Create New Rule** to create a new rule set
3. Add the IP address you want to block and the host you want to block it from. The host is the domain name of the site you want to block
   - You can copy this value from the URL of the site you want to block **without the `https` prefix**
   - It must match the exact domain you want to block, for example `my-site.com`, `www.my-site.com` or `docs.my-site.com`
   - You should add a separate entry for each subdomain that you wish to block, such as `blog.my-site.com` and `docs.my-site.com`
4. Select the **Create IP Block Rule** button

## More resources

- [Geolocation region block](/kb/guide/suspicious-traffic-in-specific-countries)

---
title: "WAF Managed Rulesets"
description: "Learn how to use managed rulesets with the Vercel Web Application Firewall (WAF)"
last_updated: "2026-01-16T02:19:36.216Z"
source: "https://vercel.com/docs/vercel-firewall/vercel-waf/managed-rulesets"
---

# WAF Managed Rulesets

Managed rulesets are collections of predefined WAF rules based on standards such as [Open Worldwide Application Security Project (OWASP) T

The following ruleset(s) are currently available:

- [OWASP core ruleset](#configure-owasp-core-ruleset)
- [Bot protection managed ruleset](#configure-bot-protection-managed-ruleset)
- [AI Bots managed ruleset](#configure-ai-bots-managed-ruleset)

## Access roles

- You need to be a [Developer](/docs/rbac/access-roles#developer-role) or viewer ([Viewer Pro](/docs/rbac/access-roles#viewer-pro-role) o
- You need to be a [Project administrator](/docs/rbac/access-roles#project-administrators) or [Team member](/docs/rbac/access-roles#membe

## Configure OWASP core ruleset

To enable and configure [OWASP Core Ruleset](https://owasp.org/www-project-top-ten/) for your project, follow these steps:

1. From your [project's dashboard](/docs/projects/project-dashboard), select the **Firewall** tab
2. Select the **Rules** tab
3. From the **Managed Rulesets** section, enable **OWASP Core Ruleset**
4. You can apply the changes with the OWASP rules enabled by default:
   - When you make any change, you will see a **Review Changes** button appear or update on the top right with the number of changes requ
   - Select **Review Changes** and review the changes to be applied

   - Select **Publish** to apply the changes to your production deployment
5. Or select what OWASP rules to enable first by selecting **Configure** from the **OWASP Core Ruleset** list item
6. For the **OWASP Core Ruleset** configuration page, enable or disable the rule that you would like to apply
7. For each enabled rule, select **Log** or **Deny** from the action drop-down
   - Use **Log** first and monitor the live traffic on the **Firewall** overview page to check that the rule has the desired effect when
8. Apply the changes
9. Monitor the live traffic on the **Firewall** overview page

## Configure bot protection managed ruleset

To enable and configure [bot protection](/docs/bot-management#bot-protection-managed-ruleset) for your project, follow these steps:

1. From your [project's dashboard](/docs/projects/project-dashboard), select the [**Firewall**](https://vercel.com/d?to=%2F%5Bteam%5D%2F%
2. Select the **Rules** tab.
3. From the **Bot Management** section, select **Log** or **Challenge** on the **Bot Protection** rule to choose what action should be pe
   - When enabled in challenge mode, the Vercel WAF will serve a JavaScript challenge to traffic that is unlikely to be a browser.
4. You can then apply as follows:
   - When you make any change, you will see a **Review Changes** button appear or update on the top right with the number of changes requ
   - Select **Review Changes** and review the changes to be applied
   - Select **Publish** to apply the changes to your production deployment

## Configure AI Bots managed ruleset

To manage AI bots for your project, follow these steps:

1. From your [project's dashboard](/docs/projects/project-dashboard), select the [**Firewall**](https://vercel.com/d?to=%2F%5Bteam%5D%2F%
2. Select the **Rules** tab.
3. From the **Bot Management** section, select **Log** or **Deny** on the **AI Bots** rule to choose what action should be performed when
   - **Log**: This action records AI bot traffic without blocking it. Its useful for monitoring.
   - **Deny**: This action blocks all traffic identified as coming from AI bots.
4. You can then apply as follows:
   - When you make any change, you will see a **Review Changes** button appear or update on the top right with the number of changes requ
   - Select **Review Changes** and review the changes to be applied
   - Select **Publish** to apply the changes to your production deployment

## Bypassing rulesets

Sometimes, you may need to allow specific requests that a managed ruleset is blocking. For example, [Bot Protection](/docs/bot-management
In this case, use the [bypass](/docs/vercel-firewall/firewall-concepts#bypass) [action](/docs/vercel-firewall/vercel-waf/rule-configuratio
In the case of the custom user agent, you would use the "User Agent" parameter with a value of the user agent name in the custom rule.

### Bypassing custom rules

If you need to allow requests being blocked by your own custom rule set up in your project, you can add another custom rule with a bypass

### Rules execution order

The Vercel WAF executes rules on incoming traffic in the following order:

1. Custom rules set up in the project
2. Managed rulesets configured in the project

---
title: "Vercel WAF"
description: "Learn how to secure your website with the Vercel Web Application Firewall (WAF)"
last_updated: "2026-01-16T02:19:36.123Z"
source: "https://vercel.com/docs/vercel-firewall/vercel-waf"
---

# Vercel WAF

The Vercel WAF, part of the [Firewall](/docs/vercel-firewall), provides security controls to [monitor](/docs/vercel-firewall/firewall-obs

- [Configure your first Custom Rule](/docs/security/vercel-waf/custom-rules)
- [Add IP Blocks](/docs/security/vercel-waf/ip-blocking)
- [Explore Managed Rulesets](/docs/security/vercel-waf/managed-rulesets)

## Traffic control

You can control the internet traffic to your website in the following ways:

- **IP blocking**: Learn how to [configure IP blocking](/docs/security/vercel-waf/ip-blocking)
- **Custom rules**: Learn how to [configure custom rules](/docs/security/vercel-waf/custom-rules) for your project
- **Managed rulesets**: Learn how to [enable managed rulesets](/docs/security/vercel-waf/managed-rulesets) for your project (Enterprise p

## Instant rollback

You can quickly revert to a previous version of your firewall configuration. This can be useful in situations that require a quick recove

To restore to a previous version:

1. From your dashboard, select the project that you'd like to configure a rule for and then select the **Firewall** tab
2. Select the **View Audit Log** option by clicking on the ellipsis menu at the top right
3. Find the version that you would like to restore to by using the date and time selectors
4. Select **Restore** and then **Restore Configuration** on the confirmation modal

## Limits

Depending on your plan, there are limits for each Vercel WAF feature.

| Feature | Hobby | Pro | Enterprise |
| --- | --- | --- | --- |
| [Project level IP Blocking](/docs/security/vercel-waf/ip-blocking#project-level-ip-blocking) | Up to 10 | Up to 100 | Custom |
| [Account-level IP Blocking](/docs/security/vercel-waf/ip-blocking#account-level-ip-blocking) | N/A | N/A | Custom |
| [Custom Rules](/docs/security/vercel-waf/custom-rules) | Up to 3 | Up to 40 | Up to 1000 |
| [Custom Rule Parameters](/docs/security/vercel-waf/rule-configuration#parameters) | All | All | All |
| [Managed Rulesets](/docs/security/vercel-waf/managed-rulesets) | N/A | N/A | Contact sales |

- For **Account-level IP Blocking**, CIDR rules are limited to `/16` for IPv4 and `/48` for IPv6

- For **Custom Rule Parameters**, JA3 (Legacy) is available on Enterprise plans

---
title: "WAF Rate Limiting"
description: "Learn how to configure custom rate limiting rules with the Vercel Web Application Firewall (WAF)."
last_updated: "2026-01-16T02:19:36.134Z"
source: "https://vercel.com/docs/vercel-firewall/vercel-waf/rate-limiting"
---

# WAF Rate Limiting

Rate limiting allows you to control the number of times that a request from the same source can hit your application within a specific ti

The use of rate limiting rules helps ensure that only intended traffic reaches your resources such as API endpoints or external services,

## Get started

1. From your [dashboard](https://vercel.com/dashboard/), select the project that you'd like to configure rate limiting for. Then select t
2. Select **Configure** on the top right of the Firewall overview page. Then, select **+ New Rule**
3. Complete the fields for the rule as follows
   1. Type a name to help you identify the purpose of this rule for future reference
   2. In the **Configure** section, add as many **If** conditions as needed:
      > **💡 Note:** All conditions must be true for the action to happen.
   3. For the **Then** action, select **Rate Limit**
      - If this is the first time you are creating a rate limit rule, you will need to review the **Rate Limiting Pricing** dialog and se
   4. Select [Fixed Window (all plans)](# "About the Fixed Window algorithm") or [Token Bucket (Enterprise)](# "About the Token Bucket al
   5. Update the **Time Window** field as needed (defaults to 60s) and the **Request Limit** field as needed (defaults to 100 requests)
      - The **Request Limit** defines the maximum number of requests allowed in the selected time window from a common source
   6. Select the key(s) from the request's source that you want to match against
   7. For the **Then** action, you can leave the **Default (429)** action or choose between **Log**, **Deny** and **Challenge**
      > **💡 Note:** The **Log** action will not perform any blocks. You can use it to first
      > monitor the effect before applying a rate limit or block action.
4. Select **Save Rule**
5. Apply the changes:
   - When you make any change, you will see a **Review Changes** button appear or update on the top right with the number of changes requ
   - Select **Review Changes** and review the changes to be applied
   - Select **Publish** to apply the changes to your production deployment
6. Go to the Firewall overview page, select your Custom Rule from the traffic grouping drop-down and select the paramater(s) related to t

## Limits

| Resource | Hobby | Pro | Enterprise |
| --- | --- | --- | --- |
| Included counting keys | IP, JA4 Digest | IP, JA4 Digest | IP, JA4 Digest, User Agent and |
| Counting algorithm | Fixed window | Fixed window | Fixed window, Token bucket |
| Counting window | Minimum: **10s**, Maximum: **10mins** | Minimum: **10s**, Maximum: **10mins** | Minimum: **10s**, Maximum: **1 |
| Number of rules | 1 per project | 40 per project | 1000 per project |
| Included requests | 1,000,000 Allowed requests | 1,000,000 Allowed requests | |

## Pricing

The pricing is based on the region(s) from which the requests come from.

---
title: "Rate Limiting SDK"
description: "Learn how to configure a custom rule with rate limit in your code."
last_updated: "2026-01-16T02:19:36.143Z"
source: "https://vercel.com/docs/vercel-firewall/vercel-waf/rate-limiting-sdk"
---

# Rate Limiting SDK

You can configure a custom rule with rate limit in your code by using the [`@vercel/firewall`](https://github.com/vercel/vercel/tree/main

- You need to set a rate limit on requests in your backend
- You want to use additional conditions with the rate limit that are not possible in the custom rule configuration of the dashboard

## Using `@vercel/firewall`

- ### Create a `@vercel/firewall` rule
  1. From your [dashboard](https://vercel.com/dashboard/), select the project that you'd like to configure rate limiting for. Then select
  2. Select **Configure** on the top right of the Firewall overview page. Then, select **+ New Rule**
  3. Complete the fields for the rule as follows
     1. Type a name such as "Firewall api rule"
     2. In the **Configure** section, for the first **If** condition, select `@vercel/firewall`
     3. Use `update-object` as the **Rate limit ID**
     4. Use the default values for **Rate Limit** and **Then**
  4. Select **Save Rule**
  5. Apply the changes:
     - When you make any change, you will see a **Review Changes** button appear or update on the top right with the number of changes re
     - Select **Review Changes** and review the changes to be applied
     - Select **Publish** to apply the changes to your production deployment

- ### Configure rate limiting in code
  You can now use the Rate limit ID `update-object` set up above with `@vercel/firewall` to rate limit any request based on your own cond
  ```ts filename="rate-limit.ts"
  import { checkRateLimit } from '@vercel/firewall';

  export async function POST(request: Request) {
    const { rateLimited } = await checkRateLimit('update-object', { request });
    if (rateLimited) {
      return new Response(
        JSON.stringify({
          error: 'Rate limit exceeded',
        }),
        {
          status: 429,
          headers: {
  ```

```
            'Content-Type': 'application/json',
          },
        },
      );
    }
    // Otherwise, continue with other tasks
  }
```

- ### Test in a preview deployment
  For your code to run when deployed in a preview deployment, you need to:
  - Enable [Protection Bypass for Automation](/docs/security/deployment-protection/methods-to-bypass-deployment-protection/protection-byp
  - Ensure [System Environment Variables are automatically exposed](/docs/environment-variables/system-environment-variables#system-envir

## Target a user's organization

For example, you can include an additional filter for a request header and check whether this header matches a key from the user's authen

### Update the custom rule filters

Edit the custom rule in the dashboard and add an **If** condition with the following values, and click **Save Rule**:

- Filter dropdown: **#Request Header**
- Value: `xrr-internal-header`
- Operator: Equals
- Match value: `internal`

### Use the `rateLimitKey` in code

Use the following code to apply the rate limit only to users of the organization.

```ts filename="rate-limit.ts"
import { checkRateLimit } from '@vercel/firewall';
import { authenticateUser } from './auth';

export async function POST(request: Request) {
  const auth = await authenticateUser(request);
  const { rateLimited } = await checkRateLimit('update-object', {
    request,
    rateLimitKey: auth.orgId,
  });
  if (rateLimited) {
    return new Response(
      JSON.stringify({
        error: 'Rate limit exceeded',
      }),
      {
        status: 429,
        headers: {
          'Content-Type': 'application/json',
        },
      },
    );
  }
}
```

---
title: "Rule Configuration Reference"
description: "List of configurable options with the Vercel WAF"
last_updated: "2026-01-16T02:19:36.147Z"
source: "https://vercel.com/docs/vercel-firewall/vercel-waf/rule-configuration"
---

# Rule Configuration Reference

For each custom rule that you create, you can configure one or more conditions with [**parameters**](#parameters) from the incoming traff

You also specify an [**action**](#actions) executed when all the conditions are met.

## Parameters

## Operators

All operators are case insensitive.

## Actions

---
title: "WAF System Bypass Rules"
description: "Learn how to configure IP-based system bypass rules with the Vercel Web Application Firewall (WAF)."
last_updated: "2026-01-16T02:19:36.154Z"
source: "https://vercel.com/docs/vercel-firewall/vercel-waf/system-bypass-rules"
---

# WAF System Bypass Rules

While Vercel's system-level mitigations (such as [DDoS protection](/docs/security/ddos-mitigation)) safeguard your websites and applicati

You can ensure that specific IP addresses or CIDR ranges are never blocked by the Vercel Firewall's system mitigations with System Bypass

> **💡 Note:** If you need to allow requests blocked by your own [WAF Custom
> Rules](/docs/vercel-waf/custom-rules), use another [custom rule with a bypass
> action](/docs/vercel-firewall/vercel-waf/managed-rulesets#bypassing-custom-rules).

## Get started

To add an IP address that should bypass system mitigations, navigate to the **Firewall** tab of your project and follow these steps:

1. Select **Configure** on the top right of the Firewall overview page
2. Scroll down to the **System Bypass Rules** section
3. Select the **+ Add Rule** button
4. Complete the following fields in the **Configure New System Bypass** modal:
   - IP Address Or CIDR (required)
   - Domain (required): The domain connected to the project or use `*` to specify all domains connected to a project
   - Note: For future reference
5. Select the **Create System Bypass** button
6. Apply the changes:
   - When you make any change, you will see a **Review Changes** button appear or update on the top right with the number of changes requ
   - Select **Review Changes** and review the changes to be applied
   - Select **Publish** to apply the changes to your production deployment

## Limits

System Bypass Rules have limits based on your [account plan](/docs/plans).

| Resource | [Hobby](/docs/plans/hobby) | [Pro](/docs/plans/pro-plan) | [Enterprise](/docs/plans/enterpr
| --- | --- | --- | ---
| Number of system bypass rules per project | N/A | 25 | 100

---
title: "Usage & Pricing for Vercel WAF"
description: "Learn how the Vercel WAF can affect your usage and how specific features are priced."
last_updated: "2026-01-16T02:19:36.159Z"
source: "https://vercel.com/docs/vercel-firewall/vercel-waf/usage-and-pricing"
---

# Usage & Pricing for Vercel WAF

Vercel Firewall features that are available under all plans, are free to use. This includes [DDoS mitigation](/docs/security/ddos-mitigat

## Free features usage

Although you are not charged for Firewall features available under all plans, you may incur [Edge Requests (ER)](/docs/manage-cdn-usage#e

| Feature | ER | FDT | Note
| --- | --- | --- | ---
| [WAF custom rule](/docs/security/vercel-waf/custom-rules) | Charged | Charged | When
| [WAF custom rule with persistent actions](/docs/security/vercel-waf/custom-rules#persistent-actions) | Not charged | Not charged | As t
| [DDoS mitigation](/docs/security/ddos-mitigation) | Not charged | Not charged | Revi
| [Attack Challenge Mode](/docs/attack-challenge-mode) | Not charged | Not charged | When
| [Account level IP Blocking](/docs/security/vercel-waf/ip-blocking#account-level-ip-blocking) | Not charged | Not charged | Requ
| [Project level IP Blocking](/docs/security/vercel-waf/ip-blocking#project-level-ip-blocking) | Charged | Charged | This

## Priced features usage

Enterprise only features are priced as described below.

### Rate limiting pricing

### Managed ruleset pricing

---
title: "Sandbox CLI Reference"
description: "Based on the Docker CLI, you can use the Sandbox CLI to manage your Vercel Sandbox from the command line."
last_updated: "2026-01-16T02:19:36.197Z"
source: "https://vercel.com/docs/vercel-sandbox/cli-reference"
---

# Sandbox CLI Reference

The Sandbox CLI, based on the Docker CLI, allows you to manage sandboxes, execute commands, copy files, and more from your terminal. This

Use the CLI for manual testing and debugging, or use the [SDK](/docs/vercel-sandbox/sdk-reference) to automate sandbox workflows in your

## Installation

Install the Sandbox CLI globally to use all commands:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i sandbox
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i sandbox
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i sandbox
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i sandbox
    ```
  </Code>
</CodeBlock>

You can invoke the CLI using the  or  commands in your terminal.

## Authentication

Log in to use Vercel Sandbox:

```bash filename="Terminal"
sandbox login
```

## `sandbox --help`

Get help information for all available sandbox commands:

```bash filename="terminal"
sandbox <subcommand>
```

**Description:** Interfacing with Vercel Sandbox

**Available subcommands:**

- [`list`](#sandbox-list): List all sandboxes for the specified account and project. \[alias: `ls`]
- [`create`](#sandbox-create): Create a sandbox in the specified account and project.
- [`copy`](#sandbox-copy): Copy files between your local filesystem and a remote sandbox \[alias: `cp`]
- [`exec`](#sandbox-exec): Execute a command in an existing sandbox
- [`ssh`](#sandbox-ssh): Start an interactive shell in an existing sandbox
- [`stop`](#sandbox-stop): Stop one or more running sandboxes \[aliases: `rm`, `remove`]
- [`run`](#sandbox-run): Create and run a command in a sandbox
- [`sh`](#sandbox-sh): Create and run an interactive shell in a new sandbox. \[alias: shell]
- [`login`](#sandbox-login): Log in to the Sandbox CLI
- [`logout`](#sandbox-logout): Log out of the Sandbox CLI

For more help, try running 

## `sandbox list`

List all sandboxes for the specified account and project.

```bash filename="terminal"
sandbox list [OPTIONS]
```

### Example

```bash filename="terminal"
# List all running sandboxes
sandbox list

# List all sandboxes (including stopped ones)
sandbox list --all

# List sandboxes for a specific project
sandbox list --project my-nextjs-app
```

### Options

| Option | Alias | Description |
| --- | --- | --- |
| `--token <token>` | - | Your Vercel authentication token. If you don't provide it, we'll use a stored token or prompt you to |
| `--project <project>` | - | The project name or ID you want to use with this command. |
| `--scope <team>` | `--team` | The team you want to use with this command. |

### Flags

| Flag | Short | Description |
| --- | --- | --- |
| `--all` | `-a` | Show all sandboxes, including stopped ones (we only show running ones by default). |
| `--help` | `-h` | Display help information. |

## `sandbox create`

Create a sandbox in the specified account and project.

```bash filename="terminal"
sandbox create [OPTIONS]
```

### Example

```bash filename="terminal"
# Create a basic Node.js sandbox
sandbox create

# Create a Python sandbox with custom timeout
sandbox create --runtime python3.13 --timeout 1h

# Create sandbox with port forwarding
sandbox create --publish-port 8080 --project my-app

# Create sandbox silently (no output)
sandbox create --silent
```

### Options

| Option | Alias | Description |
| --- | --- | --- |
| `--token <token>` | - | Your Vercel authentication token. If you don't provide it, we'll use a stored token or prompt you |
| `--project <project>` | - | The project name or ID you want to use with this command. |
| `--scope <team>` | `--team` | The team you want to use with this command. |
| `--runtime <runtime>` | - | Choose between Node.js ('node24' or 'node22') or Python ('python3.13'). We'll use Node.js 24 by de |
| `--timeout <duration>` | - | How long the sandbox can run before we automatically stop it. Examples: '5m', '1h'. We'll stop it |

| `--publish-port <port>` | `-p`     | Make a port from your sandbox accessible via a public URL.

### Flags

| Flag       | Short | Description                                                |
| ---------- | ----- | ---------------------------------------------------------- |
| `--silent` | -     | Create the sandbox without showing you the sandbox ID. |
| `--help`   | `-h`  | Display help information.                                  |

## `sandbox copy`

Copy files between your local filesystem and a remote sandbox.

```bash filename="terminal"
sandbox copy [OPTIONS] <SANDBOX_ID:PATH> <SANDBOX_ID:PATH>
```

### Example

```bash filename="terminal"
# Copy file from local to sandbox
sandbox copy ./local-file.txt sb_1234567890:/app/remote-file.txt

# Copy file from sandbox to local
sandbox copy sb_1234567890:/app/output.log ./output.log

# Copy directory from sandbox to local
sandbox copy sb_1234567890:/app/dist/ ./build/
```

### Options

| Option                | Alias    | Description
| --------------------- | -------- | ----------------------------------------------------------------------------------------------------------------
| `--token <token>`     | -        | Your Vercel authentication token. If you don't provide it, we'll use a stored token or prompt you to
| `--project <project>` | -        | The project name or ID you want to use with this command.
| `--scope <team>`      | `--team` | The team you want to use with this command.

### Flags

| Flag     | Short | Description              |
| -------- | ----- | ------------------------ |
| `--help` | `-h`  | Display help information. |

### Arguments

| Argument            | Description                                                                                 |
| ------------------- | ------------------------------------------------------------------------------------------- |
| `<SANDBOX_ID:PATH>` | The source file path (either a local file or sandbox_id:path for remote files).      |
| `<SANDBOX_ID:PATH>` | The destination file path (either a local file or sandbox_id:path for remote files). |

## `sandbox exec`

Execute a command in an existing sandbox.

```bash filename="terminal"
sandbox exec [OPTIONS] <sandbox_id> <command> [...args]
```

### Example

```bash filename="terminal"
# Execute a simple command in a sandbox
sandbox exec sb_1234567890 ls -la

# Run with environment variables
sandbox exec --env DEBUG=true sb_1234567890 npm test

# Execute interactively with sudo
sandbox exec --interactive --sudo sb_1234567890 sh

# Run command in specific working directory
sandbox exec --workdir /app sb_1234567890 python script.py
```

### Options

| Option                  | Alias    | Description
| ----------------------- | -------- | ----------------------------------------------------------------------------------------------------------------
| `--token <token>`       | -        | Your Vercel authentication token. If you don't provide it, we'll use a stored token or prompt you
| `--project <project>`   | -        | The project name or ID you want to use with this command.
| `--scope <team>`        | `--team` | The team you want to use with this command.
| `--workdir <directory>` | `-w`     | Set the directory where you want the command to run.
| `--env <key=value>`     | `-e`     | Set environment variables for your command.

### Flags

| Flag            | Short | Description                                        |
| --------------- | ----- | -------------------------------------------------- |
| `--sudo`        | -     | Run the command with admin privileges.             |
| `--interactive` | `-i`  | Run the command in an interactive shell.           |
| `--tty`         | `-t`  | Enable terminal features for interactive commands. |
| `--help`        | `-h`  | Display help information.                          |

### Arguments

| Argument       | Description                                                |
| -------------- | ---------------------------------------------------------- |
| `<sandbox_id>` | The ID of the sandbox where you want to run the command. |
| `<command>`    | The command you want to run.                               |

| `[...args]`    | Additional arguments for your command.                   |

## `sandbox ssh`

Start an interactive shell in an existing sandbox.

```bash filename="terminal"
sandbox ssh [OPTIONS] <sandbox_id>
```

### Example

```bash filename="terminal"
# Connect to an existing sandbox
sandbox ssh sb_1234567890

# Connect with a specific working directory
sandbox ssh --workdir /app sb_1234567890

# Connect with environment variables and sudo
sandbox ssh --env DEBUG=true --sudo sb_1234567890
```

### Options

| Option                  | Alias    | Description
| ----------------------- | -------- | ----------------------------------------------------------------------------------------------------------
| `--token <token>`       | -        | Your Vercel authentication token. If you don't provide it, we'll use a stored token or prompt you
| `--project <project>`   | -        | The project name or ID you want to use with this command.
| `--scope <team>`        | `--team` | The team you want to use with this command.
| `--workdir <directory>` | `-w`     | Set the directory where you want the command to run.
| `--env <key=value>`     | `-e`     | Set environment variables for your command.

### Flags

| Flag                 | Short | Description
| -------------------- | ----- | ----------------------------------------------------------------------------------------------------------
| `--sudo`             | -     | Run the command with admin privileges.
| `--no-extend-timeout` | -     | Do not extend the sandbox timeout while running an interactive command. Only affects interactive execut
| `--help`             | `-h`  | Display help information.

### Arguments

| Argument       | Description                                              |
| -------------- | -------------------------------------------------------- |
| `<sandbox_id>` | The ID of the sandbox where you want to start a shell.   |

## `sandbox stop`

Stop one or more running sandboxes.

```bash filename="terminal"
sandbox stop [OPTIONS] <sandbox_id> [...sandbox_id]
```

### Example

```bash filename="terminal"
# Stop a single sandbox
sandbox stop sb_1234567890

# Stop multiple sandboxes
sandbox stop sb_1234567890 sb_0987654321

# Stop sandbox for a specific project
sandbox stop --project my-app sb_1234567890
```

### Options

| Option                | Alias    | Description
| --------------------- | -------- | ----------------------------------------------------------------------------------------------------------
| `--token <token>`     | -        | Your Vercel authentication token. If you don't provide it, we'll use a stored token or prompt you to
| `--project <project>` | -        | The project name or ID you want to use with this command.
| `--scope <team>`      | `--team` | The team you want to use with this command.

### Flags

| Flag     | Short | Description               |
| -------- | ----- | ------------------------- |
| `--help` | `-h`  | Display help information. |

### Arguments

| Argument          | Description                              |
| ----------------- | ---------------------------------------- |
| `<sandbox_id>`    | The ID of the sandbox you want to stop. |
| `[...sandbox_id]` | Additional sandbox IDs to stop.          |

## `sandbox run`

Create and run a command in a sandbox.

```bash filename="terminal"
sandbox run [OPTIONS] <command> [...args]
```

### Example

```bash filename="terminal"
```

```
# Run a simple Node.js script
sandbox run -- node --version

# Run with custom environment and timeout
sandbox run --env NODE_ENV=production --timeout 10m -- npm start

# Run interactively with port forwarding
sandbox run --interactive --publish-port 3000 --tty npm run dev

# Run with auto-cleanup
sandbox run --rm -- python3 script.py
```

### Options

| Option | Alias | Description |
| --- | --- | --- |
| `--token <token>` | - | Your Vercel authentication token. If you don't provide it, we'll use a stored token or prompt you |
| `--project <project>` | - | The project name or ID you want to use with this command. |
| `--scope <team>` | `--team` | The team you want to use with this command. |
| `--runtime <runtime>` | - | Choose between Node.js ('node24' or 'node22') or Python ('python3.13'). We'll use Node.js 24 by de |
| `--timeout <duration>` | - | How long the sandbox can run before we automatically stop it. Examples: '5m', '1h'. We'll stop it |
| `--publish-port <port>` | `-p` | Make a port from your sandbox accessible via a public URL. |
| `--workdir <directory>` | `-w` | Set the directory where you want the command to run. |
| `--env <key=value>` | `-e` | Set environment variables for your command. |

### Flags

| Flag | Short | Description |
| --- | --- | --- |
| `--sudo` | - | Run the command with admin privileges. |
| `--interactive` | `-i` | Run the command in an interactive shell. |
| `--tty` | `-t` | Enable terminal features for interactive commands. |
| `--rm` | - | Automatically delete the sandbox when the command finishes. |
| `--help` | `-h` | Display help information. |

### Arguments

| Argument | Description |
| --- | --- |
| `<command>` | The command you want to run. |
| `[...args]` | Additional arguments for your command. |

## `sandbox sh`

Create and run an interactive shell in a new sandbox.

```bash filename="terminal"
sandbox sh [OPTIONS] [...args]
```

### Example

```bash filename="terminal"
# Start an interactive shell in a new sandbox
sandbox sh

# Start a Node.js 22 sandbox and run a shell command
sandbox sh --runtime node22 -lc "node --version"
```

### Options

| Option | Alias | Description |
| --- | --- | --- |
| `--token <token>` | - | Your Vercel authentication token. If you don't provide it, we'll use a stored token or prompt you |
| `--project <project>` | - | The project name or ID you want to use with this command. |
| `--scope <team>` | `--team` | The team you want to use with this command. |
| `--runtime <runtime>` | - | Choose between Node.js ('node24' or 'node22') or Python ('python3.13'). We'll use Node.js 24 by de |
| `--timeout <duration>` | - | How long the sandbox can run before we automatically stop it. Examples: '5m', '1h'. We'll stop it |
| `--publish-port <port>` | `-p` | Make a port from your sandbox accessible via a public URL. |
| `--workdir <directory>` | `-w` | Set the directory where you want the command to run. |
| `--env <key=value>` | `-e` | Set environment variables for your command. |

### Flags

| Flag | Short | Description |
| --- | --- | --- |
| `--sudo` | - | Run the command with admin privileges. |
| `--no-extend-timeout` | - | Do not extend the sandbox timeout while running an interactive command. Only affects interactive execut |
| `--rm` | - | Automatically remove the sandbox when the command finishes. |
| `--help` | `-h` | Display help information. |

### Arguments

| Argument | Description |
| --- | --- |
| `[...args]` | Additional arguments for your command. |

## `sandbox login`

Log in to the Sandbox CLI.

```bash filename="terminal"
sandbox login
```

### Example

```bash filename="terminal"
```

```
# Log in to the Sandbox CLI
sandbox login
```

### Flags

| Flag | Short | Description |
| --- | --- | --- |
| `--help` | `-h` | Display help information. |

## `sandbox logout`

Log out of the Sandbox CLI.

```bash filename="terminal"
sandbox logout
```

### Example

```bash filename="terminal"
# Log out of the Sandbox CLI
sandbox logout
```

### Flags

| Flag | Short | Description |
| --- | --- | --- |
| `--help` | `-h` | Display help information. |

---
title: "Vercel Sandbox examples"
description: "Vercel Sandbox allows you to run arbitrary code in isolated, ephemeral Linux VMs."
last_updated: "2026-01-16T02:19:36.207Z"
source: "https://vercel.com/docs/vercel-sandbox/examples"
---

# Vercel Sandbox examples

Learn how to use the Sandbox SDK through real-life examples.

## Using a private repository

In this example, you create an isolated environment from a private Git repository by authenticating with a [GitHub personal access token]

The `Sandbox.create()` method initializes the environment with the provided repository and configuration options, including authenticatio

### GitHub access token options

There are several ways to authenticate with private GitHub repositories.

#### Fine-grained personal access token

Fine-grained tokens provide repository-specific access and enhanced security:

1. Go to [GitHub Settings → Developer settings → Personal access tokens → Fine-grained tokens](https://github.com/settings/personal-acc
2. Click **Generate new token**
3. Configure the token:
   - **Token name**: Give it a descriptive name (e.g., "Vercel Sandbox Access")
   - **Expiration**: Set an appropriate expiration date
   - **Resource owner**: Select your account or organization
   - **Repository access**: Choose "Selected repositories" and select your private repo
   - **Repository permissions**: Grant at minimum:
     - **Contents**: Read (to clone the repository)
     - **Metadata**: Read (for basic repository information)
4. Click **"Generate token"** and copy the token
5. Set it as an environment variable and run your sandbox script:

#### Other Github methods

- [Create a classic personal access token](https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-p
- [Create a GitHub App installation token](https://docs.github.com/en/apps/creating-github-apps/authenticating-with-a-github-app/generati

## Install system packages

You can install system packages using the `dnf` system package manager:

You can find the [list of available packages](https://docs.aws.amazon.com/linux/al2023/release-notes/all-packages-AL2023.7.html) on the A

In the example, `sudo: true` allows the command to run with elevated privileges.

## Extend the timeout of a running sandbox

You can extend the timeout of a running sandbox using the `extendTimeout` method,
which takes a duration in milliseconds:

You can extend the timeout as many times as you'd like, until the [max timeout
for your plan](/docs/vercel-sandbox/pricing#maximum-runtime-duration) has been reached.

## SSH into a sandbox

You can SSH into a running sandbox by using the [`ssh` command of the CLI](/docs/vercel-sandbox/cli-reference#sandbox-ssh). This is parti

```bash filename="terminal"
sandbox ssh <sandbox-id>
```

---
title: "Vercel Sandbox"
description: "Vercel Sandbox allows you to run arbitrary code in isolated, ephemeral Linux VMs."
last_updated: "2026-01-16T02:19:36.226Z"
source: "https://vercel.com/docs/vercel-sandbox"
---

# Vercel Sandbox

Vercel Sandbox is an ephemeral compute primitive designed to safely run untrusted or user-generated code on Vercel. It supports dynamic,

With Vercel Sandbox, you can:

- **Execute untrusted or third-party code**: When you need to run code that has not been reviewed, such as AI agent output or user upload

- **Build dynamic, interactive experiences**: If you are creating tools that generate or modify code on the fly, such as AI-powered UI bu

- **Test backend logic in isolation**: Preview how user-submitted or agent-generated code behaves in a self-contained environment with ac

- Run a development server to test your application.

## Using Vercel Sandbox

The [Sandbox SDK](/docs/vercel-sandbox/sdk-reference) is the recommended way to integrate Vercel Sandbox into your applications. It provi

- **[SDK](/docs/vercel-sandbox/sdk-reference)** (recommended): Use `@vercel/sandbox` for TypeScript to automate sandbox workflows in your
- **[CLI](/docs/vercel-sandbox/cli-reference)**: Use the `sandbox` CLI for manual testing, debugging, and one-off operations

## Authentication

### Vercel OIDC token

The SDK uses Vercel OIDC tokens to authenticate whenever available. This is the most straightforward and recommended way to authenticate.

When developing locally, you can download a development token to `.env.local` using `vercel env pull`. After 12 hours the development tok

In production, Vercel manages token expiration for you.

## System specifications

Sandbox includes a `node24`, `node22` and `python3.13` image. In all these images:

- User code is executed as the `vercel-sandbox` user.
- The default working directory is `/vercel/sandbox`.
- `sudo` access is available.

| | Runtime | Package managers |
| --- | --- | --- |
| `node24` | `/vercel/runtimes/node24` | `npm`, `pnpm` |
| `node22` | `/vercel/runtimes/node22` | `npm`, `pnpm` |
| `python3.13` | `/vercel/runtimes/python` | `pip`, `uv` |

`node24` is the default runtime if the `runtime` property is not specified.

### Available packages

The base system is Amazon Linux 2023 with the following additional packages:

```
bind-utils
bzip2
findutils
git
gzip
iputils
libicu
libjpeg
libpng
ncurses-libs
openssl
openssl-libs
procps
tar
unzip
which
whois
zstd
```

Users can install additional packages using the `dnf` package manager:

```typescript filename="install-packages.ts"
import { Sandbox } from '@vercel/sandbox';

const sandbox = await Sandbox.create();
await sandbox.runCommand({
  cmd: 'dnf',
  args: ['install', '-y', 'golang'],
  sudo: true,
});
```

You can find the [list of available packages](https://docs.aws.amazon.com/linux/al2023/release-notes/all-packages-AL2023.7.html) on the A

### Sudo config

The sandbox sudo configuration is designed to be easy to use:

- `HOME` is set to `/root`. Commands executed with sudo will source root's configuration files (e.g. `.gitconfig`, `.bashrc`, etc).
- `PATH` is left unchanged. Local or project-specific binaries will still be available when running with elevated privileges.

- The executed command inherits all other environment variables that were set.

## Observability

To view sandboxes that were started per project, inspect the command history and view the sandbox URLs, access the **Sandboxes** [insight

- From the Vercel dashboard, go to the project where you created the sandbox
- Click the **Observability** tab
- Click **Sandboxes** on the left side of the **Observability** page

To track compute usage for your sandboxes across projects, go to the [Usage](/docs/pricing/manage-and-optimize-usage#viewing-usage) tab o

---
title: "Vercel Sandbox pricing and limits"
description: "Vercel Sandbox allows you to run arbitrary code in isolated, ephemeral Linux VMs."
last_updated: "2026-01-16T02:19:36.233Z"
source: "https://vercel.com/docs/vercel-sandbox/pricing"
---

# Vercel Sandbox pricing and limits

## Resource limits

- Each sandbox can use a maximum of 8 vCPUs with 2 GB of memory allocated per vCPU
- Sandboxes have a maximum runtime duration of 5 hours for Pro/Enterprise and 45 minutes for Hobby, with a default of 5 minutes. You can
- You can run Node.js or Python runtimes. Review the [system specifications](/docs/vercel-sandbox#system-specifications).
- Sandboxes can have up to 4 open ports.

## Pricing

Vercel tracks sandbox usage by:

- **Active CPU**: The amount of CPU time your code consumes, measured in milliseconds. Waiting for I/O (e.g. calling AI models, database
- **Provisioned memory**: The memory size of your sandbox instances (in GB), multiplied by the time they are running (measured in hours).
- **Network bandwidth**: The incoming and outgoing network traffic in and out of your sandbox for tasks such as installing packages and s
- **Sandbox creations**: The number of times you started a sandbox.

### Included allotment

| Metric | Monthly amount included for Hobby |
| --- | --- |
| CPU (hour) | 5 |
| Provisioned Memory (GB-hr) | 420 |
| Network (GB) | 20 |
| Sandbox creations | 5000 |

You can use sandboxes under Pro and Enterprise plans based on the following regional pricing:

| Active CPU time (per hour) | Provisioned Memory (per GB-hr) | Network (per GB) | Sandbox creations (per 1M) |
| --- | --- | --- | --- |
| $0.128 | $0.0106 | $0.15 | $0.60 |

> **⚠️ Warning:** Currently, Vercel Sandbox is only available in the `iad1` region.

### Maximum runtime duration

Sandboxes can run for up to several hours based on your plan. The default is 5 minutes.

| Plan | Duration limit |
| --- | --- |
| Hobby | 45 minutes |
| Pro | 5 hours |
| Enterprise | 5 hours |

You can configure the maximum runtime duration using the `timeout` option of `Sandbox.create()` and extend it later using `sandbox.extend`

You can extend the timeout as many times as you need, until the maximum timeout
has been reached.

### Concurrent sandboxes limit

At any time, based on your plan, you can run up to a maximum number of sandboxes at the same time. You can [upgrade](/docs/plans/hobby#up

| Plan | Concurrent sandboxes limit |
| --- | --- |
| Hobby | 10 |
| Pro | 2000 |
| Enterprise | 2000 |

Please [get in touch with our sales team](/contact/sales) if you need more concurrent sandboxes.

### Sandboxes creation rate limit

The number of vCPUs you can allocate to active sandboxes depends on your plan. If you need more, you can [upgrade](/docs/plans/hobby#upgr

For example, with the Pro plan's 200 vCPUs per minute limit, you can create:

- 25 large sandboxes (8 vCPUs each) every minute
- Or 100 small sandboxes (2 vCPUs each) every minute

| Plan | Sandboxes vCPUs allocation limit |
| --- | --- |
| Hobby | 40 vCPUs/10 minute |
| Pro | 200 vCPUs/minute |
| Enterprise | 400 vCPUs/minute |

Please [get in touch with our sales team](/contact/sales) if you need a higher rate of sandboxes vCPUs allocations.

---
title: "Quickstart"
description: "Learn how to run your first code in a Vercel Sandbox."
last_updated: "2026-01-16T02:19:36.240Z"
source: "https://vercel.com/docs/vercel-sandbox/quickstart"
---

# Quickstart

This guide shows you how to run your first code in a Vercel Sandbox.

## Prerequisites

- A [Vercel account](https://vercel.com/signup)
- [Vercel CLI](/docs/cli) installed (`npm i -g vercel`)
- Node.js 22+

## Quickstart

### 1. Install the SDK

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i @vercel/sandbox
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @vercel/sandbox
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @vercel/sandbox
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i @vercel/sandbox
    ```
  </Code>
</CodeBlock>

### 2. Connect to a Vercel project

Before you can create sandboxes, you need to connect your local directory to a Vercel project. This project handles authentication for yo

**Don't have a project yet?** No problem. You can create an empty one just for sandbox development:

```bash filename="terminal"
mkdir my-sandbox-app && cd my-sandbox-app
npm init -y
vercel link
```

When prompted, select **Create a new project**. The project doesn't need any code deployed. It just needs to exist so Vercel can generate

**Already have a project?** Link to it from your working directory:

```bash filename="terminal"
vercel link
```

### 3. Pull your authentication token

Once linked, pull your environment variables to get an authentication token:

```bash filename="terminal"
vercel env pull
```

This creates a `.env.local` file containing a token that the SDK uses to authenticate your requests. The token expires after 12 hours dur

When you deploy to Vercel, token management happens automatically.

### 4. Write your code

Create a file that creates a sandbox and runs a command:

```ts filename="index.ts"
import { Sandbox } from '@vercel/sandbox';

const sandbox = await Sandbox.create();

const result = await sandbox.runCommand('echo', ['Hello from Vercel Sandbox!']);
console.log(await result.stdout());

await sandbox.stop();
```

### 5. Run it

```bash filename="terminal"
node --env-file .env.local index.ts
```

You should see:

```
```

Hello from Vercel Sandbox!
```

## What you just did

1. **Installed the SDK**: Added the Vercel Sandbox package to your project.
2. **Connected to a project**: Linked to a Vercel project that handles authentication.
3. **Pulled a token**: Downloaded credentials that let the SDK create sandboxes on your behalf.
4. **Created a sandbox**: Spun up an isolated Linux microVM.
5. **Ran a command**: Executed code inside the secure environment.
6. **Cleaned up**: Stopped the sandbox to free resources.

## Next steps

Now that you have the basics working, explore more:

- [Run a development server](/docs/vercel-sandbox/examples): Clone a repo and run a Next.js app in a sandbox.
- [Use private repositories](/docs/vercel-sandbox/examples#using-a-private-repository): Authenticate with GitHub to clone private repos.
- [Install system packages](/docs/vercel-sandbox/examples#install-system-packages): Use `dnf` to install additional tools.
- [SDK Reference](/docs/vercel-sandbox/sdk-reference): Full API documentation.
- [CLI Reference](/docs/vercel-sandbox/cli-reference): Manage sandboxes from the terminal.


---
title: "Sandbox SDK Reference"
description: "A comprehensive reference for the Vercel Sandbox SDK, which allows you to run code in a secure, isolated environment."
last_updated: "2026-01-16T02:19:36.276Z"
source: "https://vercel.com/docs/vercel-sandbox/sdk-reference"
---

# Sandbox SDK Reference

The Vercel Sandbox Software Development Kit (SDK) lets you create ephemeral Linux microVMs on demand. Use it to evaluate user-generated c

## Prerequisites

Install the SDK:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i @vercel/sandbox
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i @vercel/sandbox
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i @vercel/sandbox
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i @vercel/sandbox
    ```
  </Code>
</CodeBlock>

After installation:

- Link your project and pull environment variables with `vercel link` and `vercel env pull` so the SDK can read a Vercel OpenID Connect (
- Choose a runtime: `node24`, `node22`, or `python3.13`.

## Core classes

| Class | What it does | Example |
| --- | --- | --- |
| `Sandbox` | Creates and manages isolated microVM environments | `const sandbox = await Sandbox.create()` |
| `Command` | Handles running commands inside the sandbox | `const cmd = await sandbox.runCommand()` |
| `CommandFinished` | Contains the result after a command completes | Access `cmd.exitCode` and `cmd.stdout()` |

### Basic workflow

```ts
// 1. Create a sandbox
const sandbox = await Sandbox.create({ runtime: 'node24' });

// 2. Run a command - it waits for completion and returns the result
const result = await sandbox.runCommand('node', ['--version']);

// 3. Check the result
console.log(result.exitCode); // 0
console.log(await result.stdout()); // v22.x.x
```

## Sandbox class

The `Sandbox` class gives you full control over isolated Linux microVMs. Use it to create new sandboxes, inspect active ones, stream comm

### Accessors

#### `sandboxId`

Use `sandboxId` to identify the current microVM so you can reconnect to it later with `Sandbox.get()` or trace command history. Store thi

**Returns:** `string`.
```

```ts
console.log(sandbox.sandboxId);
```

#### `status`

The `status` accessor reports the lifecycle state of the sandbox so you can decide when to queue new work or perform cleanup. Poll this v

**Returns:** `"pending" | "running" | "stopping" | "stopped" | "failed"`.

```ts
console.log(sandbox.status);
```

#### `timeout`

`timeout` shows how many milliseconds remain before the sandbox stops automatically. Compare the remaining time against upcoming commands

**Returns:** `number`.

```ts
console.log(sandbox.timeout);
```

#### `createdAt`

The `createdAt` accessor returns the date and time when the sandbox was created. Use this to track sandbox age or calculate how long a sa

**Returns:** `Date`.

```ts
console.log(sandbox.createdAt);
```

### Static methods

#### `Sandbox.list()`

Use `Sandbox.list()` to enumerate sandboxes for a project, optionally filtering by time range or page size. Combine `since` and `until` w

**Returns:** `Promise<Parsed<{ sandboxes: SandboxSummary[]; pagination: Pagination; }>>`.

| Parameter | Type | Required | Details |
| --- | --- | --- | --- |
| `projectId` | `string` | Yes | Project whose sandboxes you want to list. |
| `limit` | `number` | No | Maximum number of sandboxes to return. |
| `since` | `number \| Date` | No | List sandboxes created after this time. |
| `until` | `number \| Date` | No | List sandboxes created before this time. |
| `signal` | `AbortSignal` | No | Cancel the request if necessary. |

```ts
const { sandboxes } = await Sandbox.list({ projectId });
```

#### `Sandbox.create()`

`Sandbox.create()` launches a new microVM with your chosen runtime, source, and resource settings. Defaults to an empty workspace when no

**Returns:** `Promise<Sandbox>`.

| Parameter | Type | Required | Details / Values |
| --- | --- | --- | --- |
| `source` | `git` | No | Clone a Git repository. `url`: string `username`: string `password`: string `depth`?: n |
| `source` | `tarball` | No | Mount a tarball. `url`: string |
| `resources.vcpus` | `number` | No | Override CPU count (defaults to plan baseline). |
| `runtime` | `string` | No | Runtime image such as `"node24"`, `"node22"`, or `"python3.13"`. |
| `ports` | `number[]` | No | Ports to expose for `sandbox.domain()`. |
| `timeout` | `number` | No | Initial timeout in milliseconds. |
| `signal` | `AbortSignal` | No | Cancel sandbox creation if needed. |

```ts
const sandbox = await Sandbox.create({ runtime: 'node24' });
```

#### `Sandbox.get()`

`Sandbox.get()` rehydrates an active sandbox by ID so you can resume work or inspect logs. It throws if the sandbox no longer exists, so

**Returns:** `Promise<Sandbox>`.

| Parameter | Type | Required | Details |
| --- | --- | --- | --- |
| `sandboxId` | `string` | Yes | Identifier of the sandbox to retrieve. |
| `signal` | `AbortSignal` | No | Cancel the request if necessary. |

```ts
const sandbox = await Sandbox.get({ sandboxId });
```

### Instance methods

#### `sandbox.getCommand()`

Call `sandbox.getCommand()` to retrieve a previously executed command by its ID, which is especially helpful after detached executions wh

**Returns:** `Promise<Command>`.

| Parameter | Type | Required | Details |
| --- | --- | --- | --- |
| `cmdId` | `string` | Yes | Identifier of the command to fetch. |

| `opts.signal` | `AbortSignal` | No | Cancel the lookup if it takes too long. |

```ts
const command = await sandbox.getCommand(cmdId);
```

#### `sandbox.runCommand()`

`sandbox.runCommand()` executes commands inside the microVM, either blocking until completion or returning immediately in detached mode. I

**Returns:** `Promise<CommandFinished>` when `detached` is falsy; `Promise<Command>` when `detached` is `true`.

| Parameter | Type | Required | Details |
| --- | --- | --- | --- |
| `command` | `string` | Yes | Command to execute (string overload). |
| `args` | `string[]` | No | Arguments for the string overload. |
| `opts.signal` | `AbortSignal` | No | Cancel the command (string overload). |
| `params.cmd` | `string` | Yes | Command to execute when using the object overload. |
| `params.args` | `string[]` | No | Arguments for the object overload. |
| `params.cwd` | `string` | No | Working directory for execution. |
| `params.env` | `Record<string, string>` | No | Additional environment variables. |
| `params.sudo` | `boolean` | No | Run the command with sudo. |
| `params.detached` | `boolean` | No | Return immediately with a live `Command` object. |
| `params.stdout` | `Writable` | No | Stream standard output to a writable. |
| `params.stderr` | `Writable` | No | Stream standard error to a writable. |
| `params.signal` | `AbortSignal` | No | Cancel the command when using the object overload. |

```ts
const result = await sandbox.runCommand('node', ['--version']);
```

#### `sandbox.mkDir()`

`sandbox.mkDir()` creates directories in the sandbox filesystem before you write files or clone repositories. Paths are relative to `/ver

```ts
await sandbox.mkDir('tmp/assets');
```

| Parameter | Type | Required | Details |
| --- | --- | --- | --- |
| `path` | `string` | Yes | Directory to create. |
| `opts.signal` | `AbortSignal` | No | Cancel the operation. |

**Returns:** `Promise<void>`.

#### `sandbox.readFile()`

Use `sandbox.readFile()` to pull file contents from the sandbox to a `ReadableStream`. The promise resolves to `null` when the file does

```ts
const stream = await sandbox.readFile({ path: 'package.json' });
```

| Parameter | Type | Required | Details |
| --- | --- | --- | --- |
| `file.path` | `string` | Yes | Path to the file inside the sandbox. |
| `file.cwd` | `string` | No | Base directory for resolving `file.path`. |
| `opts.signal` | `AbortSignal` | No | Cancel the read operation. |

**Returns:** `Promise<null | ReadableStream>`.

#### `sandbox.readFileToBuffer()`

Use `sandbox.readFileToBuffer()` to pull entire file contents from the sandbox to an in-memory buffer. The promise resolves to `null` whe

```ts
const buffer = await sandbox.readFileToBuffer({ path: 'package.json' });
```

| Parameter | Type | Required | Details |
| --- | --- | --- | --- |
| `file.path` | `string` | Yes | Path to the file inside the sandbox. |
| `file.cwd` | `string` | No | Base directory for resolving `file.path`. |
| `opts.signal` | `AbortSignal` | No | Cancel the read operation. |

**Returns:** `Promise<null | Buffer>`.

#### `sandbox.downloadFile()`

Use `sandbox.downloadFile()` to pull file contents from the sandbox to a local destination. The promise resolves to the absolute destinat

```ts
const dstPath = await sandbox.downloadFile(
  { path: 'package.json', cwd: '/vercel/sandbox' },
  { path: 'local-package.json', cwd: '/tmp' }
);
```

| Parameter | Type | Required | Details |
| --- | --- | --- | --- |
| `src.path` | `string` | Yes | Path to the file inside the sandbox. |
| `src.cwd` | `string` | No | Base directory for resolving `src.path`. |
| `dst.path` | `string` | Yes | Path to local destination. |
| `dst.cwd` | `string` | No | Base directory for resolving `dst.path`. |
| `opts.signal` | `AbortSignal` | No | Cancel the download operation. |
| `opts.mkdirRecursive` | `boolean` | No | Create destination directories recursively if they do not exist. |

**Returns:** `Promise<null | string>`.

#### `sandbox.writeFiles()`

`sandbox.writeFiles()` uploads one or more files into the sandbox filesystem. Paths default to `/vercel/sandbox`; use absolute paths for

```ts
await sandbox.writeFiles([{ path: 'hello.txt', content: Buffer.from('hi') }]);
```

| Parameter | Type | Required | Details |
| --- | --- | --- | --- |
| `files` | `{ path: string; content: Buffer; }[]` | Yes | File descriptors to write. |
| `opts.signal` | `AbortSignal` | No | Cancel the write operation. |

**Returns:** `Promise<void>`.

#### `sandbox.domain()`

`sandbox.domain()` resolves a publicly accessible URL for a port you exposed during creation. It throws if the port is not registered to

```ts
const previewUrl = sandbox.domain(3000);
```

| Parameter | Type | Required | Details |
| --- | --- | --- | --- |
| `p` | `number` | Yes | Port number declared in `ports`. |

**Returns:** `string`.

#### `sandbox.stop()`

Call `sandbox.stop()` to terminate the microVM and free resources immediately. It's safe to call multiple times; subsequent calls resolve

```ts
await sandbox.stop();
```

| Parameter | Type | Required | Details |
| --- | --- | --- | --- |
| `opts.signal` | `AbortSignal` | No | Cancel the stop operation. |

**Returns:** `Promise<void>`.

#### `sandbox.extendTimeout()`

Use `sandbox.extendTimeout()` to extend the sandbox lifetime by the specified duration. This lets you keep the sandbox running up to the

```ts
await sandbox.extendTimeout(60000); // Extend by 60 seconds
```

| Parameter | Type | Required | Details |
| --- | --- | --- | --- |
| `duration` | `number` | Yes | Duration in milliseconds to extend the timeout by. |
| `opts.signal` | `AbortSignal` | No | Cancel the operation. |

**Returns:** `Promise<void>`.

## Command class

`Command` instances represent processes that run inside a sandbox. Detached executions created through `sandbox.runCommand({ detached: tr

### Properties

#### `exitCode`

The `exitCode` property holds the process exit status once the command finishes. For detached commands, this value starts as `null` and g

```ts
if (command.exitCode !== null) {
  console.log(`Command exited with code: ${command.exitCode}`);
}
```

**Returns:** `number | null`.

### Accessors

#### `cmdId`

Use `cmdId` to identify the specific command execution so you can look it up later with `sandbox.getCommand()`. Store this value whenever

```ts
console.log(command.cmdId);
```

**Returns:** `string`.

#### `cwd`

The `cwd` accessor shows the working directory where the command is executing. Compare this value against expected paths when debugging f

```ts
console.log(command.cwd);
```

**Returns:** `string`.

#### `startedAt`

`startedAt` returns the Unix timestamp (in milliseconds) when the command started executing. Subtract this from the current time to monit

```ts
const duration = Date.now() - command.startedAt;
console.log(`Command has been running for ${duration}ms`);
```

**Returns:** `number`.

### Methods

#### `logs()`

Call `logs()` to stream structured log entries in real time so you can watch command output as it happens. Each entry includes the stream

```ts
for await (const log of command.logs()) {
  if (log.stream === 'stdout') {
    process.stdout.write(log.data);
  } else {
    process.stderr.write(log.data);
  }
}
```

| Parameter | Type | Required | Details |
| --- | --- | --- | --- |
| `opts.signal` | `AbortSignal` | No | Cancel log streaming if needed. |

**Returns:** `AsyncGenerator<{ stream: "stdout" | "stderr"; data: string; }, void, void>`.

**Note:** May throw `StreamError` if the sandbox stops while streaming logs.

#### `wait()`

Use `wait()` to block until a detached command finishes and get the resulting `CommandFinished` object with the populated exit code. This

```ts
const detachedCmd = await sandbox.runCommand({
  cmd: 'sleep',
  args: ['5'],
  detached: true,
});
const result = await detachedCmd.wait();
if (result.exitCode !== 0) {
  console.error('Something went wrong...');
}
```

| Parameter | Type | Required | Details |
| --- | --- | --- | --- |
| `params.signal` | `AbortSignal` | No | Cancel waiting if you need to abort early. |

**Returns:** `Promise<CommandFinished>`.

#### `output()`

Use `output()` to retrieve stdout, stderr, or both as a single string. Choose `"both"` when you want combined output for logging, or spec

```ts
const combined = await command.output('both');
const stdoutOnly = await command.output('stdout');
```

| Parameter | Type | Required | Details |
| --- | --- | --- | --- |
| `stream` | `"stdout" \| "stderr" \| "both"` | Yes | The output stream to read. |
| `opts.signal` | `AbortSignal` | No | Cancel output streaming. |

**Returns:** `Promise<string>`.

**Note:** This may throw string conversion errors if the command output contains invalid Unicode.

#### `stdout()`

`stdout()` collects the entire standard output stream as a string, which is handy when commands print JSON or other structured data that

```ts
const output = await command.stdout();
const data = JSON.parse(output);
```

| Parameter | Type | Required | Details |
| --- | --- | --- | --- |
| `opts.signal` | `AbortSignal` | No | Cancel the read while the command runs. |

**Returns:** `Promise<string>`.

**Note:** This may throw string conversion errors if the command output contains invalid Unicode.

#### `stderr()`

`stderr()` gathers all error output produced by the command. Combine this with `exitCode` to build user-friendly error messages or forwar

```ts
const errors = await command.stderr();
if (errors) {
  console.error('Command errors:', errors);
}
```

| Parameter | Type | Required | Details |
| --- | --- | --- | --- |
| `opts.signal` | `AbortSignal` | No | Cancel the read while collecting error output. |

**Returns:** `Promise<string>`.

**Note:** This may throw string conversion errors if the command output contains invalid Unicode.

#### `kill()`

Call `kill()` to terminate a running command using the specified signal. This lets you stop long-running processes without destroying the

```ts
await command.kill('SIGKILL');
```

| Parameter | Type | Required | Details |
| --- | --- | --- | --- |
| `signal` | `Signal` | No | The signal to send to the process. Defaults to `SIGTERM`. |
| `opts.abortSignal` | `AbortSignal` | No | Cancel the kill operation. |

**Returns:** `Promise<void>`.

## CommandFinished class

`CommandFinished` is the result you receive after a sandbox command exits. It extends the `Command` class, so you keep access to streamin

### Properties

#### `exitCode`

The `exitCode` property reports the numeric status returned by the command. A value of `0` indicates success; any other value means the p

```ts
if (result.exitCode === 0) {
  console.log('Command succeeded');
}
```

**Returns:** `number`.

### Accessors

#### `cmdId`

Use `cmdId` to identify the specific command execution so you can reference it in logs or retrieve it later with `sandbox.getCommand()`.

```ts
console.log(result.cmdId);
```

**Returns:** `string`.

#### `cwd`

The `cwd` accessor shows the working directory where the command executed. Compare this value against expected paths when debugging file-

```ts
console.log(result.cwd);
```

**Returns:** `string`.

#### `startedAt`

`startedAt` returns the Unix timestamp (in milliseconds) when the command started executing. Subtract this from the current time or from

```ts
const duration = Date.now() - result.startedAt;
console.log(`Command took ${duration}ms`);
```

**Returns:** `number`.

### Methods

`CommandFinished` inherits all methods from `Command` including `logs()`, `output()`, `stdout()`, `stderr()`, and `kill()`. See the [Comm

## Example workflows

- [Clone and run a private Git repository](/docs/vercel-sandbox/examples#using-a-private-repository) to validate builds before merging pu
- [Install system packages](/docs/vercel-sandbox/examples#install-system-packages) while keeping sudo-enabled commands isolated.
- [Extend sandbox timeouts](/docs/vercel-sandbox/examples#extend-the-timeout-of-a-running-sandbox) to support longer-running workloads li
- Browse more scenarios in the [Sandbox examples](/docs/vercel-sandbox/examples) catalog.

## Authentication

The SDK prioritizes Vercel OIDC tokens that the CLI stores in `.env.local`. Link your project and pull environment variables with `vercel

> **💡 Note:** Sandbox credentials expire. Rotate downloaded tokens at least every 12 hours
> during development.

For more background, read the [authentication guide](/docs/rest-api#authentication) or the [Vercel Sandbox authentication](/docs/vercel-s

## Environment defaults

- **Operating system:** Amazon Linux 2023 with common build tools such as `git`, `tar`, `openssl`, and `dnf`.
- **Available runtimes:** `node24`, `node22`, and `python3.13` images with their respective package managers.
- **Resources:** Choose the number of virtual CPUs (`vcpus`) per sandbox. Pricing and plan limits appear in the [Sandbox pricing table](/

- **Timeouts:** The default timeout is 5 minutes. You can extend it programmatically up to 45 minutes on the Hobby plan and up to 5 hours
- **Sudo:** `sudo` commands run as `vercel-sandbox` with the root home directory set to `/root`.

> **💡 Note:** The filesystem is ephemeral. You must export artifacts to durable storage if
> you need to keep them after the sandbox stops.

---
title: "Accessibility Audit Tool"
description: "Learn how to use the Accessibility Audit Tool to automatically check the Web Content Accessibility Guidelines 2.0 level A a
last_updated: "2026-01-16T02:19:36.325Z"
source: "https://vercel.com/docs/vercel-toolbar/accessibility-audit-tool"
---

# Accessibility Audit Tool

The accessibility audit tool automatically checks the [Web Content Accessibility Guidelines 2.0](https://www.w3.org/TR/WCAG20/) level A a

## Accessing the accessibility audit tool

To access the accessibility audit tool:

1. [Open the Toolbar Menu](/docs/vercel-toolbar#using-the-toolbar-menu)
2. Select the **Accessibility Audit** option. If there are accessibility issues detected on the page, a badge will display next to the op
3. The **Accessibility** panel will open on the right side of the screen. Here you can filter by **All**, **Critical**, **Serious**, **Mo

## Enabling or disabling the accessibility audit tool

The accessibility audit tool is enabled by default. To disable it:

1. Open the **Preferences** panel by selecting the toolbar menu icon, then scrolling down to the **Preferences** section
2. Toggle the **Accessibility Audit** option to enable or disable the tool

## Inspecting accessibility issues

To inspect an accessibility issue select the filter option you want to inspect. A list of issues will are displayed as dropdowns. You can

## Recording accessibility issues

By default the accessibility audit tool will log issues on page load. To test ephemeral states, such as hover or focus, you can record is

## More resources

- [Interaction Timing Tool](/docs/vercel-toolbar/interaction-timing-tool)
- [Layout Shift Tool](/docs/vercel-toolbar/layout-shift-tool)

---
title: "Toolbar Browser Extensions"
description: "The browser extensions enable you to use the toolbar in production environments, take screenshots and attach them to commen
last_updated: "2026-01-16T02:19:36.281Z"
source: "https://vercel.com/docs/vercel-toolbar/browser-extension"
---

# Toolbar Browser Extensions

The browser extension is supported in Chrome, Firefox, Opera, Microsoft Edge, in addition to other Chromium-based browsers that support e

- Enables the toolbar to detect when you are logged in to Vercel.
- Operates faster and with fewer network requests.
- Remembers your [personal preferences](#setting-user-preferences) for when the toolbar hides and activates.
- Allows you to [take screenshots](#taking-screenshots-with-the-extension) and attach them to comments.
- Click the extension to hide and show the toolbar, and pin it to your browser bar for quick access.

## Installing the browser extension

Install the browser extension from your browser's extension page:

-
-

You can also install the Chrome extension using the link above in Opera and Microsoft Edge.

## Setting user preferences

With the browser extension you are able to toggle on the following preferences that affect how the toolbar behaves for you without alteri

| Setting | Description |
| --- | --- |
| Always Activate | Sets the toolbar to activate anytime you are authenticated as your Vercel user instead of waiting to be clicked. |
| Start Hidden | Sets the toolbar to start hidden. Read more about [hiding and showing the toolbar](/docs/vercel-toolbar/managing-tool |

## Taking screenshots with the extension

The extension enables you to leave comments with screenshots attached by clicking, dragging, and releasing to select the area of the page

1. Select **Comment** in the toolbar menu.
2. Click, drag, and release to select the area of the page you'd like to screenshot.
3. Compose your comment and click the send icon.

---
title: "Add the Vercel Toolbar to your local environment"
description: "Learn how to use the Vercel Toolbar in your local environment."
last_updated: "2026-01-16T02:19:36.308Z"
source: "https://vercel.com/docs/vercel-toolbar/in-production-and-localhost/add-to-localhost"
---

# Add the Vercel Toolbar to your local environment

To enable the toolbar in your local environment, add it to your project using the [`@vercel/toolbar`](https://www.npmjs.com/package/@verc

- ### Install the `@vercel/toolbar` package and link your project
  Install the package using the following command:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i @vercel/toolbar
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i @vercel/toolbar
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i @vercel/toolbar
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i @vercel/toolbar
      ```
    </Code>
  </CodeBlock>
  Then link your local project to your Vercel project with the [`vercel link`](/docs/cli/link) command using [Vercel CLI](/docs/cli).
  ```bash filename="terminal"
  vercel link [path-to-directory]
  ```

- ### Add the toolbar to your project
  > For \['nextjs', 'nextjs-app']:
  To use the Vercel Toolbar locally in a Next.js project, define `withVercelToolbar` in your `next.config.js` file and export it, as show
  > For \['sveltekit']:
  To use the Vercel Toolbar locally in a SvelteKit project, add the `vercelToolbar` plugin to your `vite.config.js` file, as shown below:
  > For \['nuxt']:
  To use the Vercel Toolbar locally in a Nuxt project, install the Nuxt module:
  > For \['other']:
  The toolbar works locally out of the box with Next.js. To use it with a framework other than Next.js, you can add the following script
  ```js filename="next.config.js" framework=nextjs-app
  /** @type {import('next').NextConfig} */
  const createWithVercelToolbar = require('@vercel/toolbar/plugins/next');
  const nextConfig = {
    // Config options here
  };

  const withVercelToolbar = createWithVercelToolbar();
  // Instead of module.exports = nextConfig, do this:
  module.exports = withVercelToolbar(nextConfig);
  ```
  ```ts filename="next.config.js" framework=nextjs-app
  /** @type {import('next').NextConfig} */
  const createWithVercelToolbar = require('@vercel/toolbar/plugins/next');
  const nextConfig = {
    // Config options here
  };

  const withVercelToolbar = createWithVercelToolbar();
  // Instead of module.exports = nextConfig, do this:
  module.exports = withVercelToolbar(nextConfig);
  ```
  ```js filename="next.config.js" framework=nextjs
  /** @type {import('next').NextConfig} */
  const createWithVercelToolbar = require('@vercel/toolbar/plugins/next');
  const nextConfig = {
    // Config options here
  };

  const withVercelToolbar = createWithVercelToolbar();
  // Instead of module.exports = nextConfig, do this:
  module.exports = withVercelToolbar(nextConfig);
  ```
  ```ts filename="next.config.js" framework=nextjs
  /** @type {import('next').NextConfig} */
  const createWithVercelToolbar = require('@vercel/toolbar/plugins/next');
  const nextConfig = {
    // Config options here
  };

  const withVercelToolbar = createWithVercelToolbar();
  // Instead of module.exports = nextConfig, do this:
  module.exports = withVercelToolbar(nextConfig);
  ```
  ```js filename="vite.config.js" framework=sveltekit
  import { sveltekit } from '@sveltejs/kit/vite';
  import { vercelToolbar } from '@vercel/toolbar/plugins/vite';
  import { defineConfig } from 'vite';

  export default defineConfig({
    plugins: [sveltekit(), vercelToolbar()],
  });
  ```
  ```ts filename="vite.config.ts" framework=sveltekit
  import { sveltekit } from '@sveltejs/kit/vite';
  import { vercelToolbar } from '@vercel/toolbar/plugins/vite';
  import { defineConfig } from 'vite';

  export default defineConfig({
    plugins: [sveltekit(), vercelToolbar()],
  ```

```
});
```
```tsx filename="index.ts" framework=other
<script
  src="https://vercel.live/_next-live/feedback/feedback.js"
  data-explicit-opt-in="true"
  data-owner-id="user-id-or-team-id-here"
  data-project-id="project-id-here"
  data-branch="branch-name-here"
></script>
```
```jsx filename="index.js" framework=other
<script
  src="https://vercel.live/_next-live/feedback/feedback.js"
  data-explicit-opt-in="true"
  data-owner-id="user-id-or-team-id-here"
  data-project-id="project-id-here"
  data-branch="branch-name-here"
></script>
```
> For \['other']:
To find your project ID, see [project ID](/docs/projects/overview#project-id). To find your user or team ID, see [Find your Team ID](/d
> For \['nextjs-app']:
Then add the following code to your `layout.tsx` or `layout.jsx` file:
```tsx filename="app/layout.tsx" framework=nextjs-app
import { VercelToolbar } from '@vercel/toolbar/next';

export default function RootLayout({
  children,
}: {
  children: React.ReactNode;
}) {
  const shouldInjectToolbar = process.env.NODE_ENV === 'development';
  return (
    <html lang="en">
      <body>
        {children}
        {shouldInjectToolbar && <VercelToolbar />}
      </body>
    </html>
  );
}
```
```jsx filename="app/layout.jsx" framework=nextjs-app
import { VercelToolbar } from '@vercel/toolbar/next';

export default function RootLayout(children) {
  const shouldInjectToolbar = process.env.NODE_ENV === 'development';
  return (
    <html lang="en">
      <body>
        {children}
        {shouldInjectToolbar && <VercelToolbar />}
      </body>
    </html>
  );
}
```
> For \['nextjs']:
Then add the following code to your `_app.tsx` or `_app.jsx` file:
```ts filename="pages/_app.tsx" framework=nextjs
import { VercelToolbar } from '@vercel/toolbar/next';
import type { AppProps } from 'next/app';

export default function MyApp({ Component, pageProps }: AppProps) {
  const shouldInjectToolbar = process.env.NODE_ENV === 'development'
  return (
    <>
      <Component {...pageProps} />
      {shouldInjectToolbar && <VercelToolbar />}
    </>
  );
}
```
```js filename="pages/_app.jsx" framework=nextjs
import { VercelToolbar } from '@vercel/toolbar/next';

export default function MyApp({ Component, pageProps }) {
  const shouldInjectToolbar = process.env.NODE_ENV === 'development';
  return (
    <>
      <Component {...pageProps} />
      {shouldInjectToolbar && <VercelToolbar />}
    </>
  );
}
```
> For \['sveltekit']:
Then add the following code to your root `+layout.svelte` file:
```ts filename="src/routes/+layout.svelte" framework=sveltekit
<script lang="ts">
  import { mountVercelToolbar } from '@vercel/toolbar/vite';
  import { onMount } from 'svelte';
  import { dev } from '$app/environment';

  if (dev) {
    onMount(() => mountVercelToolbar());
  }
</script>
```

  ```js filename="src/routes/+layout.svelte" framework=sveltekit
  <script>
    import { mountVercelToolbar } from '@vercel/toolbar/vite';
    import { onMount } from 'svelte';
    import { dev } from '$app/environment';

    if (dev) {
      onMount(() => mountVercelToolbar());
    }
  </script>
  ```
  > For \['nuxt']:
  This will automatically add the `@vercel/toolbar` module to your Nuxt configuration file.
  ```js filename="nuxt.config.js" framework=nuxt
  export default defineNuxtConfig({
    modules: ['@vercel/toolbar'],
  });
  ```
  ```ts filename="nuxt.config.ts" framework=nuxt
  export default defineNuxtConfig({
    modules: ['@vercel/toolbar'],
  });
  ```
  You do not need to configure anything else.


---
title: "Add the Vercel Toolbar to your production environment"
description: "Learn how to add the Vercel Toolbar to your production environment and how your team members can use tooling to access the
last_updated: "2026-01-16T02:19:36.297Z"
source: "https://vercel.com/docs/vercel-toolbar/in-production-and-localhost/add-to-production"
---

# Add the Vercel Toolbar to your production environment

As a [team owner](/docs/rbac/access-roles#owner-role) or [member](/docs/rbac/access-roles#member-role), you can enable the toolbar in you

## Adding the toolbar using the browser extension

For team members that use supported browsers and want the most straightforward experience, we recommend using the [Vercel Browser Extensi

For team members that use browsers for which a Vercel extension is not available, to allow toolbar access for everyone that accesses your

## Adding the toolbar using the `@vercel/toolbar` package

For team members that do not use the browser extension or if you have more complex rules for when the toolbar shows in production, you ca

- ### Install the `@vercel/toolbar` package and link your project
  Install the package in your project using the following command:
  <CodeBlock>
    <Code tab="pnpm">
      ```bash
      pnpm i @vercel/toolbar
      ```
    </Code>
    <Code tab="yarn">
      ```bash
      yarn i @vercel/toolbar
      ```
    </Code>
    <Code tab="npm">
      ```bash
      npm i @vercel/toolbar
      ```
    </Code>
    <Code tab="bun">
      ```bash
      bun i @vercel/toolbar
      ```
    </Code>
  </CodeBlock>
  Then link your local project to your Vercel project with the [`vercel link`](/docs/cli/link) command using [Vercel CLI](/docs/cli).
  ```bash filename="terminal"
  vercel link [path-to-directory]
  ```

- ### Add the toolbar to your project
  Before using the Vercel Toolbar in a production deployment **Vercel recommends conditionally injecting the toolbar**. Otherwise, all vi

  The following example demonstrates code that will show the Vercel Toolbar to a team member on a production deployment.
  ```ts filename="vanilla-example.ts" framework=other
  import { mountVercelToolbar } from '@vercel/toolbar';

  // You should inject the toolbar conditionally
  // to avoid showing it to all visitors
  mountVercelToolbar();
  ```
  ```js filename="vanilla-example.js" framework=other
  import { mountVercelToolbar } from '@vercel/toolbar';

  // You should inject the toolbar conditionally
  // to avoid showing it to all visitors
  mountVercelToolbar();
  ```
  ```ts filename="pages/_app.tsx" framework=nextjs
  import { VercelToolbar } from '@vercel/toolbar/next';
  import type { AppProps } from 'next/app';

  function useIsEmployee() {
    // Replace this stub with your auth library implementation
  ```

```
  return false;
}

export default function MyApp({ Component, pageProps }: AppProps) {
  const isEmployee = useIsEmployee();

  return (
    <>
      <Component {...pageProps} />
      {isEmployee ? <VercelToolbar /> : null}
    </>
  );
}
```
```js filename="pages/_app.jsx" framework=nextjs
import { VercelToolbar } from '@vercel/toolbar/next';

function useIsEmployee() {
  // Replace this stub with your auth library implementation
  return false;
}

export default function MyApp({ Component, pageProps }) {
  const isEmployee = useIsEmployee();

  return (
    <>
      <Component {...pageProps} />
      {isEmployee ? <VercelToolbar /> : null}
    </>
  );
}
```
```tsx filename="components/staff-toolbar.tsx" framework=nextjs-app
'use client';

import { VercelToolbar } from '@vercel/toolbar/next';

function useIsEmployee() {
  // Replace this stub with your auth library hook
  return false;
}

export function StaffToolbar() {
  const isEmployee = useIsEmployee();
  return isEmployee ? <VercelToolbar /> : null;
}
```
```tsx filename="app/layout.tsx" framework=nextjs-app
import { Suspense, type ReactNode } from 'react';
import { StaffToolbar } from '../components/staff-toolbar';

export default function RootLayout({ children }: { children: ReactNode }) {
  return (
    <html lang="en">
      <body>
        {children}
        <Suspense fallback={null}>
          <StaffToolbar />
        </Suspense>
      </body>
    </html>
  );
}
```
```jsx filename="@components/staff-toolbar" framework=nextjs-app
'use client';

import { VercelToolbar } from '@vercel/toolbar/next';

function useIsEmployee() {
  // Replace this stub with your auth library hook
  return false;
}

export function StaffToolbar() {
  const isEmployee = useIsEmployee();
  return isEmployee ? <VercelToolbar /> : null;
}
```
```jsx filename="app/layout.jsx" framework=nextjs-app
import { Suspense } from 'react';
import { StaffToolbar } from '../components/staff-toolbar';

export default function RootLayout({ children }) {
  return (
    <html lang="en">
      <body>
        {children}
        <Suspense fallback={null}>
          <StaffToolbar />
        </Suspense>
      </body>
    </html>
  );
}
```
```js filename="nuxt.config.js" framework=nuxt
export default defineNuxtConfig({
  modules: ['@vercel/toolbar'],
```

```
    vercelToolbar: {
      mode: 'manual',
    },
  });
  ```
  ```ts filename="nuxt.config.ts" framework=nuxt
  export default defineNuxtConfig({
    modules: ['@vercel/toolbar'],
    vercelToolbar: {
      mode: 'manual',
    },
  });
  ```
  ```js filename="app/plugins/toolbar.client.js" framework=nuxt
  import { useAuth } from 'lib/auth'; // Your auth library
  export default defineNuxtPlugin(() => {
    const auth = useAuth();

    onNuxtReady(async () => {
      if (!auth.isEmployee()) return;

      const { mountVercelToolbar } = await import('@vercel/toolbar/vite');
      mountVercelToolbar();
    });
  });
  ```
  ```ts filename="app/plugins/toolbar.client.ts" framework=nuxt
  import { useAuth } from 'lib/auth'; // Your auth library
  export default defineNuxtPlugin(() => {
    const auth = useAuth();

    onNuxtReady(async () => {
      if (!auth.isEmployee()) return;

      const { mountVercelToolbar } = await import('@vercel/toolbar/vite');
      mountVercelToolbar();
    });
  });
  ```
  ```js filename="vite.config.js" framework=sveltekit
  import { sveltekit } from '@sveltejs/kit/vite';
  import { vercelToolbar } from '@vercel/toolbar/plugins/vite';
  import { defineConfig } from 'vite';

  export default defineConfig({
    plugins: [sveltekit(), vercelToolbar()],
  });
  ```
  ```ts filename="vite.config.ts" framework=sveltekit
  import { sveltekit } from '@sveltejs/kit/vite';
  import { vercelToolbar } from '@vercel/toolbar/plugins/vite';
  import { defineConfig } from 'vite';

  export default defineConfig({
    plugins: [sveltekit(), vercelToolbar()],
  });
  ```
  ```ts filename="src/routes/+layout.svelte" framework=sveltekit
  <script lang="ts">
    import { mountVercelToolbar } from '@vercel/toolbar/vite';
    import { onMount } from 'svelte';

    // You should inject the toolbar conditionally
    // to avoid showing it to all visitors
    onMount(() => mountVercelToolbar());
  </script>
  ```
  ```js filename="src/routes/+layout.svelte" framework=sveltekit
  <script>
    import {mountVercelToolbar} from '@vercel/toolbar/vite';
    import {onMount} from 'svelte';
    // You should inject the toolbar conditionally
    // to avoid showing it to all visitors
    onMount(() => mountVercelToolbar());
  </script>
  ```
  > For \['other']:

- ### Managing notifications and integrations for Comments on production
  Unlike comments on preview deployments, alerts for new comments won't be sent to a specific user by default. Vercel recommends [linking

## Enabling the Vercel Toolbar

Alternatively to using the package, you can enable access to the Vercel Toolbar for your production environment at the team or project le

1. Navigate to [your Vercel dashboard](/dashboard) and make sure that you have selected your team from the [scope selector](/docs/dashboa
2. From your [dashboard](/dashboard), select the **Settings** tab.
3. In the **General** section, find **Vercel Toolbar**.
4. Under each environment (**Preview** and **Production**), select either **On** or **Off** from the dropdown to determine the visibility
5. Once set at the team level, you can optionally choose to allow the setting to be overridden at the project level.

### Disabling the toolbar

If you have noticed that the toolbar is showing up for team members on your production sites, you can disable it at either the team or pr

1. Navigate to [your Vercel dashboard](/dashboard) and make sure that you have selected your team from the [scope selector](/docs/dashboa
2. From your [dashboard](/dashboard), select the **Settings** tab.
3. In the **General** section, find **Vercel Toolbar**.
4. Under **Production** select **Off** from the dropdown.

## Acessing the toolbar using the Vercel dashboard

You can send team members and users a production deployment with the Vercel Toolbar included from the dashboard. To do so:

1. From your dashboard, go to your project and select the **Projects** tab. Alternatively, you can also use the deployment overview page.
2. Click the dropdown on the **Visit** button and select **Visit with Toolbar**. This will take you to your production deployment with th

This will not show for users who have the browser extension installed, as the extension will already show the toolbar whenever you visit

## Accessing the toolbar using the Browser extension

Provided [the Vercel toolbar is enabled](/docs/vercel-toolbar/managing-toolbar#enable-or-disable-the-toolbar-project-wide) for your proje

1. Install the [Vercel Browser Extension](/docs/vercel-toolbar/browser-extension).
2. Ensure that you are logged in to your Vercel account on vercel.com. You must be signed in for the extension to know which domains you
3. Ensure that you have deployed to production. Older deployments do not support injection through the browser extension.
4. Ensure that any team members that need access to the toolbar in production follow these steps to install the domain.

## Accessing the toolbar using the toolbar menu

Provided [the Vercel toolbar is enabled](/docs/vercel-toolbar/managing-toolbar#enable-or-disable-the-toolbar-project-wide) for your proje

1. Open a preview deployment of your project.
2. Select the menu icon in the toolbar.
3. Scroll down to **Enable Vercel Toolbar in Production** and select it.
4. Choose the domain you want to enable the toolbar on.

---
title: "Add the Vercel Toolbar to local and production environments"
description: "Learn how to use the Vercel Toolbar in production and local environments."
last_updated: "2026-01-16T02:19:36.316Z"
source: "https://vercel.com/docs/vercel-toolbar/in-production-and-localhost"
---

# Add the Vercel Toolbar to local and production environments

The Vercel Toolbar is available by default on all [preview environments](/docs/deployments/environments#preview-environment-pre-productio

All toolbar features such as [Comments](/docs/comments/using-comments), [Feature Flags](/docs/feature-flags), [Draft Mode](/docs/draft-mo

- [Add the toolbar to your local or production environment](/docs/vercel-toolbar/in-production-and-localhost/add-to-localhost)

---
title: "Interaction Timing Tool"
description: "The interaction timing tool allows you to inspect in detail each interaction"
last_updated: "2026-01-16T02:19:36.313Z"
source: "https://vercel.com/docs/vercel-toolbar/interaction-timing-tool"
---

# Interaction Timing Tool

As you navigate your site, the interaction timing tool allows you to inspect in detail each interaction's latency and get notified with t

## Accessing the Interaction Timing Tool

To access the interaction timing tool:

1. [Open the Toolbar Menu](/docs/vercel-toolbar#using-the-toolbar-menu)
2. Select the **Interaction Timing** option. If any interaction has been detected on the page, a badge will display next to the option. T
3. The **Interaction Timing** popover will open on the right side of the screen. As you navigate your site, each interaction will appear

## Interaction Timing Tool Preferences

To change preferences for the interaction timing tool:

1. [Open the Toolbar Menu](/docs/vercel-toolbar#using-the-toolbar-menu)
2. Select the **Preferences** option
3. Select your desired setting for **Measure Interaction Timing**
   - **On** will show the toasts for interactions taking >200ms
   - **On (Silent)** will not show toasts, but will still track interaction timing and display it in the interaction timing side panel wh
   - **Off** will turn off tracking for interaction timing

## More resources

- [Preview deployments overview](/docs/deployments/environments#preview-environment-pre-production)
- [Using comments with preview deployments](/docs/comments/using-comments)
- [Draft mode](/docs/draft-mode)

---
title: "Layout Shift Tool"
description: "The layout shift tool gives you insight into any elements that may cause layout shifts on the page."
last_updated: "2026-01-16T02:19:36.332Z"
source: "https://vercel.com/docs/vercel-toolbar/layout-shift-tool"
---

# Layout Shift Tool

The layout shift tool gives you insight into any elements that may cause layout shifts on the page. The cause for a layout shift could be

- Elements that change in height or width
- Custom font loading
- Media embeds (images, iframes, videos, etc.) that do not have set dimensions
- Dynamic content that's injected at runtime
- Animations that affect layout

Layout shifts play a part in [Core Web Vitals](/docs/speed-insights/metrics#core-web-vitals-explained) and contribute to [Speed Insights]

## Accessing the layout shift tool

To access the layout shift tool:

1. [Open the toolbar menu](/docs/vercel-toolbar#using-the-toolbar-menu)
2. Select the **Layout Shifts** option. If there are layout shifts detected on the page, a badge will display next to the option. The num
3. The **Layout Shifts** popover will open on the right side of the screen. Here you can filter, inspect, and replay any detected layout

Each shift details its impact, the responsible element, and a description of the shift if available. For example, "became taller when its

## Inspecting layout shifts

You can replay a layout shift by either:

- Double-clicking it
- Selecting it and using the **Replay selected shift** button

You can also select more than one shift and play them at the same time. You may want to do this to see the combined effect of element shi

When you replay layout shifts, the Vercel Toolbar will become your stop button. Press this to stop replaying layout shifts. Alternatively

You can also disable layout shift detection on a per element basis. You can do this by adding a `data-allow-shifts` attribute to an eleme

## Disabling the layout shift tool

To disable the layout shift tool completely:

1. [Open the Toolbar Menu](/docs/vercel-toolbar#using-the-toolbar-menu)
2. Select **Preferences**
3. Toggle the setting for **Layout Shift Detection**

## More resources

- [Preview deployments overview](/docs/deployments/environments#preview-environment-pre-production)
- [Using comments with preview deployments](/docs/comments/using-comments)
- [Draft mode](/docs/draft-mode)

---
title: "Managing the visibility of the Vercel Toolbar"
description: "Learn how to enable or disable the Vercel Toolbar for your team, project, and session."
last_updated: "2026-01-16T02:19:36.534Z"
source: "https://vercel.com/docs/vercel-toolbar/managing-toolbar"
---

# Managing the visibility of the Vercel Toolbar

## Viewing the toolbar

When the toolbar is enabled, you'll be able to view it on any preview or enabled environment. By default, the toolbar will appear as a ci

Once a tool is used, the toolbar will show a second icon next to the menu, so you can access your most recently used tool.

## Enable or disable the toolbar team-wide

To disable the toolbar by default for all projects in your team:

1. Navigate to [your Vercel dashboard](/dashboard) and make sure that you have selected your team from the [scope selector](/docs/dashboa
2. From your [dashboard](/dashboard), select the **Settings** tab.
3. In the **General** section, find **Vercel Toolbar**.
4. Under each environment (**Preview** and **Production**), select either **On** or **Off** from the dropdown to determine the visibility
5. You can optionally choose to allow the setting to be overridden at the project level.

## Enable or disable the toolbar project-wide

To disable the toolbar project-wide:

1. From your [dashboard](/dashboard), select the project you want to enable or disable Vercel Toolbar for.
2. Navigate to **Settings** tab.
3. In the **General** section, find **Vercel Toolbar**.
4. Under each environment (**Preview** and **Production**), select either an option from the dropdown to determine the visibility of Verc
   - **Default**: Respect team-level visibility settings.
   - **On**: Enable the toolbar for the environment.
   - **Off**: Disable the toolbar for the environment.

## Disable toolbar for session

To disable the toolbar in the current browser tab:

1. Activate the Vercel Toolbar by clicking on it
2. In the toolbar menu, scroll down the list and select **Disable for Session**.

To show the toolbar again, open a new browser session.

Alternatively, you can also hide the toolbar in any of the following ways:

- Select the toolbar icon and drag it to the X that appears at the bottom of the screen.
- Click the [browser extension](/docs/vercel-toolbar/browser-extension) icon if you have it pinned to your browser bar.
- Use .

To show the toolbar when it is hidden you can use that same key command or click the browser extension.

Users with the browser extension can set the toolbar to start hidden by toggling on **Start Hidden** in **Preferences** from the Toolbar m

## Disable toolbar for automation

You can use the `x-vercel-skip-toolbar` header to prevent interference with automated end-to-end tests:

1. Add the `x-vercel-skip-toolbar` header to the request sent to [the preview deployment URL](/docs/deployments/environments#preview-envi
2. Optionally, you can assign the value `1` to the header. However, presence of the header itself triggers Vercel to disable the toolbar

## Enable or disable the toolbar for a specific branch

You can use Vercel's [preview environment variables](/docs/environment-variables#preview-environment-variables) to manage the toolbar for

To enable the toolbar for an individual branch, add the following to the environment variables for the desired preview branch:

```txt filename=".env"
VERCEL_PREVIEW_FEEDBACK_ENABLED=1
```

To disable the toolbar for an individual branch, set the above environment variable's value to `0`:

```txt filename=".env"
VERCEL_PREVIEW_FEEDBACK_ENABLED=0
```

## Using the toolbar with a custom alias domain

To use the toolbar with preview deployments that have [custom alias domains](/docs/domains/add-a-domain), you must opt into the toolbar e

## Using a Content Security Policy

If you have a [Content Security Policy (CSP)](https://developer.mozilla.org/docs/Web/HTTP/CSP) configured, you **may** need to adjust the

You can make the following adjustments to the `Content-Security-Policy` [response header](/docs/headers/cache-control-headers#custom-resp

- Add the following to `script-src` (Most commonly used):
  ```bash
    script-src https://vercel.live
  ```
- Add the following to `connect-src`:
  ```bash
    connect-src https://vercel.live wss://ws-us3.pusher.com
  ```
- Add the following to `img-src`:
  ```bash
    img-src https://vercel.live https://vercel.com data: blob:
  ```
- Add the following to `frame-src`:
  ```bash
    frame-src https://vercel.live
  ```
- Add the following to `style-src`:
  ```bash
    style-src https://vercel.live 'unsafe-inline'
  ```
- Add the following to `font-src`:
  ```bash
    font-src https://vercel.live https://assets.vercel.com
  ```

---
title: "Vercel Toolbar"
description: "Learn how to use the Vercel Toolbar to leave feedback, navigate through important dashboard pages, share deployments, use D
last_updated: "2026-01-16T02:19:36.351Z"
source: "https://vercel.com/docs/vercel-toolbar"
---

# Vercel Toolbar

The Vercel Toolbar is a tool that assists in the iteration and development process. Through the toolbar, you can:

- Leave feedback on deployments with [Comments](/docs/comments)
- Navigate [through dashboard pages](/docs/vercel-toolbar#using-the-toolbar-menu), and [share deployments](/docs/vercel-toolbar#sharing-d
- Read and set [Feature Flags](/docs/feature-flags)
- Use [Draft Mode](/docs/draft-mode) for previewing unpublished content
- Edit content in real-time using [Edit Mode](/docs/edit-mode)
- Inspect for [Layout Shifts](/docs/vercel-toolbar/layout-shift-tool) and [Interaction Timing](/docs/vercel-toolbar/interaction-timing-to
- Check for accessibility issues with the [Accessibility Audit Tool](/docs/vercel-toolbar/accessibility-audit-tool)

## Activating the Toolbar

By default, when the toolbar first shows up on your deployments it is sleeping. This means it will not run any tools in the background or

Users who have installed the browser extension can toggle on **Always Activate** in **Preferences** from the Toolbar menu.

## Enabling or Disabling the toolbar

The Vercel Toolbar is enabled by default for all preview deployments. You can disable the toolbar at the [team](/docs/vercel-toolbar/manag

You can also manage its visibility for [automation](/docs/vercel-toolbar/managing-toolbar#disable-toolbar-for-automation) with HTTP heade

To enable the toolbar for your local or production environments, see [Adding the toolbar to your environment](/docs/vercel-toolbar/in-pro

## Using the Toolbar Menu

You can access the Toolbar Menu by pressing  on your keyboard.

Alternatively, you can also access the Toolbar Menu through the Vercel Toolbar by clicking the menu icon. If you haven't activated the to

| Feature | Description |
| --- | --- |
| **Search** | Quickly search the toolbar and access dashboard pages. |
| **Quick branch access** | View the current branch and commit hash. |
| **Switch branches** | Quickly switch between branches (on preview and product |
| [**Layout shifts**](/docs/vercel-toolbar/layout-shift-tool) | Open the Layout Shift Tool to identify elements causing |
| [**Interaction timing**](/docs/vercel-toolbar/interaction-timing-tool) | Inspect in detail each interaction's latency and view y |
| [**Accessibility audit tool**](/docs/vercel-toolbar/accessibility-audit-tool) | Automatically check the Web Content Accessibility Guide |

| **Open Graph** | View [open graph](https://ogp.me/#metadata) properties
| [**Comments**](/docs/comments) | Access the Comments panel to leave or view feedback.
| [**View inbox**](/docs/comments/using-comments#comment-threads) | View all open comments.
| **Navigate to your team** | Navigate to your team's dashboard.
| **Navigate to your project** | Navigate to your project's dashboard.
| **Navigate to your deployment** | Navigate to your deployment's dashboard.
| [**Hide Toolbar**](#enabling-or-disabling-the-toolbar) | Hide the toolbar.
| [**Disable for session**](#enabling-or-disabling-the-toolbar) | Disable the toolbar for the current session.
| [**Set preferences**](#toolbar-menu-preferences) | Set personal preferences for the toolbar.
| **Logout** | Logout of the toolbar.

## Setting Custom Keyboard Shortcuts

You can set your own keyboard shortcuts to quickly access specific tools. Additionally, you can change the default keyboard shortcuts for

1. Select Preferences in the Toolbar Menu
2. Select Configure next to Keyboard Shortcuts
3. Select Record shortcut… (or click the X if you have an existing keyboard shortcut set) next to the tool you'd like to set it for
4. Press the keys you'd like to use as the shortcut for that tool
5. To change the keyboard shortcuts for opening the Toolbar Menu and for showing and hiding the toolbar, you must have the [Browser Exten

## Sharing deployments

You can use the Share button in deployments with the Vercel Toolbar enabled, as well as in all preview deployments, to share your deploym

To share a deployment:

1. Go to the deployment you want to share and ensure you're logged into the Vercel Toolbar.
2. Find the **Share**  button in the Toolbar Menu and select it.
3. From the **Share** dialog, ensure you're allowing the right permissions and click **Copy Link** to copy the deployment URL to your cli

If you're on an [Enterprise](/docs/plans/enterprise) team, you will be able to see who shared deployment URLs in your [audit logs](/docs/

## Reposition toolbar

You can reposition the toolbar by dragging it to either side of your screen. It will snap into place and appear there across deployments

## Toolbar Menu preferences

When logged into the Vercel Toolbar, you'll find a **Preferences** button in the Toolbar Menu. In this menu, you can update the following

| Setting | Description
| --- | ---
| **[Notifications](/docs/comments/managing-comments#notifications)** | Set when you will receive notifications for comments in the deplo
| **Theme** | Select your color theme
| **Layout Shift Detection** | Enable or disable the [Layout Shift Tool](/docs/vercel-toolbar/la
| **[Keyboard Shortcuts](#setting-custom-keyboard-shortcuts)** | Set custom keyboard shortcuts for tools and change the default ke
| **Accessibility Audit** | Enable or disable the [Accessibility Audit Tool](/docs/vercel-too
| **Measure Interaction Timing** | Enable or disable the [Interaction Timing Tool](/docs/vercel-tool
| **[Browser Extension](/docs/vercel-toolbar/browser-extension)** | Add Vercel's extension to your browser to take screenshots, enabl
| **Always Activate** | Sets the toolbar to activate anytime you are authenticated as you
| **Start Hidden** | Sets the toolbar to start hidden. Read more about [hiding and sho

## More resources

- [Preview deployments](/docs/deployments/environments#preview-environment-pre-production)
- [Comments](/docs/comments)
- [Draft Mode](/docs/draft-mode)
- [Edit Mode](/docs/edit-mode)

---
title: "Setting Up Webhooks"
description: "Learn how to set up webhooks and use them with Vercel Integrations."
last_updated: "2026-01-16T02:19:36.363Z"
source: "https://vercel.com/docs/webhooks"
---

# Setting Up Webhooks

A webhook is a trigger-based HTTP endpoint configured to receive HTTP POST requests through events. When an event happens, a webhook is s

Webhooks configured with Vercel can trigger a deployment when a specific event occurs. Vercel integrations receive platform events throug

## Account Webhooks

Vercel allows you to add a [generic](# "What is a generic webhook?") endpoint for events from your dashboard. [Pro](/docs/plans/pro-plan)

### Configure a webhook

- ### Go to your team settings
  Choose your team scope on the dashboard, and go to **Settings → Webhooks**.

- ### Select the events to listen to
  The configured webhook listens to one or more events before it triggers the function request. Vercel supports event selections from the
  #### Deployment Events
  Configurable webhooks listen to the following deployment-based events:
  - **Deployment Created**: Listens for when any new deployment is initiated
  - **Deployment Succeeded**: Listens for a successful deployment
  - **Deployment Promoted**: Listens for when a deployment is successfully promoted, either manually or automatically, does not include r
  - **Deployment Error**: Listens for any failed deployment
  - **Deployment Cancelled**: Listens for a canceled deployment due to any failure
  #### Project Events
  > **💡 Note:** Project events are only available when "All Team Projects" is selected as the
  > [project scope](#choose-your-target-projects).
  Configurable webhooks listen to the following project-based events:
  - **Project Created**: Listens whenever a new project is created
  - **Project Removed**: Listens whenever any project is deleted from the team account
  - **Project Renamed**: Listens whenever a project is renamed
  #### Firewall events

  Configurable webhooks listen to the following firewall-based events:
  - **Attack Detected**: Listens for when the [Vercel Firewall](/docs/vercel-firewall) detects and mitigates a [DDoS attack](/docs/securi
  The events you select should depend on your use case and the workflow you want to implement.

- ### Choose your target projects
  After selecting the event types, choose the scope of team projects for which webhooks will listen for events.

- ### Enter your endpoint URL
  The endpoint URL is the destination that triggers the events. All events are forwarded to this URL as a POST request. In case of an eve

  Once you have configured your webhook, click the **Create Webhook** button.

  The **Webhook Created** dialog will display a secret key, which won't be shown again. You should secure your webhooks by comparing the

  Once complete, click **Done**.

  To view all your new and existing webhooks, go to the **Webhooks** section of your team's dashboard. To remove any webhook, click the c

## Integration Webhooks

Webhooks can also be created through [Integrations](/docs/integrations). When [creating a new integration](/docs/integrations/create-inte

For native integrations, you can also receive billing-related webhook events such as invoice creation, payment, and refunds. Learn more a

## Events

The webhook URL receives an HTTP POST request with a JSON payload for each event. All the events have the following format:

```json filename="webhook-payload"
  "id": <eventId>,
  "type": <event-type>,
  "createdAt": <javascript-timestamp>,
  "payload": <payload for the event>,
  "region": <RegionId>,
```

Here's a [list of supported event types](/docs/webhooks/webhooks-api#supported-event-types) and their [`payload`](/docs/webhooks/webhooks

---
title: "Webhooks API Reference"
description: "Vercel Integrations allow you to subscribe to certain trigger-based events through webhooks. Learn about the supported webh
last_updated: "2026-01-16T02:19:36.524Z"
source: "https://vercel.com/docs/webhooks/webhooks-api"
---

# Webhooks API Reference

Vercel Integrations allow you to subscribe to certain trigger-based events through webhooks. An example use-cases for webhooks might be c

## Payload

The webhook payload is a JSON object with the following keys.

| Key | Description |
| --- | --- |
| **type** | The [event type](#supported-event-types). |
| **id** | The ID of the webhook delivery. |
| **createdAt** | The date and time the webhook event was generated. |
| **region** | The region the event occurred in (possibly null). |
| **payload** | The payload of the webhook. See [Supported Event Types](#supported-event-types) for more information. |

## Supported Event Types

### deployment.canceled

Occurs whenever a deployment is canceled.

| Key | Description |
| --- | --- |
| **payload.team.id** | The ID of the event's team (possibly null). |
| **payload.user.id** | The ID of the event's user. |
| **payload.deployment.id** | The ID of the deployment. |
| **payload.deployment.meta** | A Map of deployment metadata. |
| **payload.deployment.url** | The URL of the deployment. |
| **payload.deployment.name** | The project name used in the deployment URL. |
| **payload.links.deployment** | The URL on the Vercel Dashboard to inspect the deployment. |
| **payload.links.project** | The URL on the Vercel Dashboard to the project. |
| **payload.target** | A String that indicates the target. Possible values are `production`, `staging` or `null`. |
| **payload.project.id** | The ID of the project. |
| **payload.plan** | The plan type of the deployment. |
| **payload.regions** | An array of the supported regions for the deployment. |

### deployment.check-rerequested

Occurs when a user has requested for a [check](/docs/integrations/checks-overview) to be rerun after it failed.

| Key | Description |
| --- | --- |
| **payload.team.id** | The ID of the event's team (possibly null). |
| **payload.user.id** | The ID of the event's user. |
| **payload.deployment.id** | The ID of the deployment. |
| **payload.check.id** | The ID of the check. |

### deployment.cleanup

Occurs whenever a deployment is cleaned up after it has been fully removed either due to explicit removal or retention rules.

| Key | Description |
| --- | --- |

| **payload.team.id** | | The ID of the event's team (possibly null).
| **payload.user.id** | | The ID of the event's user.
| **payload.deployment.id** | | The ID of the deployment.
| **payload.deployment.meta** | | A Map of deployment metadata.
| **payload.deployment.url** | | The URL of the deployment.
| **payload.deployment.name** | | The project name used in the deployment URL.
| **payload.deployment.alias** | | An array of aliases that will get assigned when the deployment is ready.
| **payload.deployment.target** | | A String that indicates the target. Possible values are `production`, `staging` o
| **payload.deployment.customEnvironmentId** | | The ID of the custom environment, if the custom environment is used.
| **payload.deployment.regions** | | An array of the supported regions for the deployment.
| **payload.project.id** | | The ID of the project.

### deployment.created

Occurs whenever a deployment is created.

| Key | Description |
| --- | --- |
| **payload.team.id** | The ID of the event's team (possibly null). |
| **payload.user.id** | The ID of the event's user. |
| **payload.alias** | An array of aliases that will get assigned when the deployment is ready. |
| **payload.deployment.id** | The ID of the deployment. |
| **payload.deployment.meta** | A Map of deployment metadata. |
| **payload.deployment.url** | The URL of the deployment. |
| **payload.deployment.name** | The project name used in the deployment URL. |
| **payload.links.deployment** | The URL on the Vercel Dashboard to inspect the deployment. |
| **payload.links.project** | The URL on the Vercel Dashboard to the project. |
| **payload.target** | A String that indicates the target. Possible values are `production`, `staging` or `null`. |
| **payload.project.id** | The ID of the project. |
| **payload.plan** | The plan type of the deployment. |
| **payload.regions** | An array of the supported regions for the deployment. |

### deployment.error

Occurs whenever a deployment has failed.

| Key | Description |
| --- | --- |
| **payload.team.id** | The ID of the event's team (possibly null). |
| **payload.user.id** | The ID of the event's user. |
| **payload.deployment.id** | The ID of the deployment. |
| **payload.deployment.meta** | A Map of deployment metadata. |
| **payload.deployment.url** | The URL of the deployment. |
| **payload.deployment.name** | The project name used in the deployment URL. |
| **payload.links.deployment** | The URL on the Vercel Dashboard to inspect the deployment. |
| **payload.links.project** | The URL on the Vercel Dashboard to the project. |
| **payload.target** | A String that indicates the target. Possible values are `production`, `staging` or `null`. |
| **payload.project.id** | The ID of the project. |
| **payload.plan** | The plan type of the deployment. |
| **payload.regions** | An array of the supported regions for the deployment. |

### deployment.integration.action.cancel

Occurs when an integration deployment action or the deployment itself is canceled.

| Key | Description |
| --- | --- |
| **payload.configuration.id** | The ID of the integration installation. |
| **payload.installationId** | The ID of the integration installation (same as `configuration.id`). |
| **payload.resourceId** | The ID of the integration resource for which the action is canceled. |
| **payload.action** | The action slug, declared by the integration |
| **payload.deployment.id** | The ID of the deployment. |

### deployment.integration.action.cleanup

Occurs when a deployment that executed an integration deployment action is cleaned up, such as due to the deployment retention policy.

| Key | Description |
| --- | --- |
| **payload.configuration.id** | The ID of the integration installation. |
| **payload.installationId** | The ID of the integration installation (same as `configuration.id`). |
| **payload.resourceId** | The ID of the integration resource for which the action is cleaned up. |
| **payload.action** | The action slug, declared by the integration |
| **payload.deployment.id** | The ID of the deployment. |

### deployment.integration.action.start

Occurs when a deployment starts an integration deployment action.

| Key | Description |
| --- | --- |
| **payload.configuration.id** | The ID of the integration installation. |
| **payload.installationId** | The ID of the integration installation (same as `configuration.id`). |
| **payload.resourceId** | The ID of the integration resource for which the action is started. |
| **payload.action** | The action slug, declared by the integration |
| **payload.deployment.id** | The ID of the deployment. |

### deployment.promoted

Occurs whenever a deployment is promoted.

> **💡 Note:** This event gets fired after a production deployment is
> [promoted](/docs/deployments/promoting-a-deployment#staging-and-promoting-a-production-deployment)
> to start serving production traffic. This can happen automatically after a
> successful build, or after running the [promote](/docs/cli/promote) command.

| Key | Description |
| --- | --- |
| **payload.team.id** | The ID of the event's team (possibly null). |
| **payload.user.id** | The ID of the event's user. |

| **payload.deployment.id** | | The ID of the deployment. |
| **payload.deployment.meta** | | A Map of deployment metadata. |
| **payload.deployment.url** | | The URL of the deployment. |
| **payload.deployment.name** | | The project name used in the deployment URL. |
| **payload.links.deployment** | | The URL on the Vercel Dashboard to inspect the deployment. |
| **payload.links.project** | | The URL on the Vercel Dashboard to the project. |
| **payload.project.id** | | The ID of the project. |
| **payload.plan** | | The plan type of the deployment. |
| **payload.regions** | | An array of the supported regions for the deployment. |

### deployment.ready

Occurs whenever a deployment is ready.

| Key | | Description |
| --- | --- | --- |
| **payload.team.id** | | The ID of the event's team (possibly null). |
| **payload.user.id** | | The ID of the event's user. |
| **payload.deployment.id** | | The ID of the deployment. |
| **payload.deployment.meta** | | A Map of deployment metadata. |
| **payload.deployment.url** | | The URL of the deployment. |
| **payload.deployment.name** | | The project name used in the deployment URL. |
| **payload.links.deployment** | | The URL on the Vercel Dashboard to inspect the deployment. |
| **payload.links.project** | | The URL on the Vercel Dashboard to the project. |
| **payload.target** | | A String that indicates the target. Possible values are `production`, `staging` or `null`. |
| **payload.project.id** | | The ID of the project. |
| **payload.plan** | | The plan type of the deployment. |
| **payload.regions** | | An array of the supported regions for the deployment. |

### deployment.succeeded

Occurs whenever a deployment is successfully built and your integration has registered at least one [check](/docs/integrations/checks-ove

> **💡 Note:** This event gets fired after all blocking Checks have passed. See [
>
> ](/docs/integrations#webhooks/events/deployment-prepared) if you registered
> Checks.

| Key | | Description |
| --- | --- | --- |
| **payload.team.id** | | The ID of the event's team (possibly null). |
| **payload.user.id** | | The ID of the event's user. |
| **payload.deployment.id** | | The ID of the deployment. |
| **payload.deployment.meta** | | A Map of deployment metadata. |
| **payload.deployment.url** | | The URL of the deployment. |
| **payload.deployment.name** | | The project name used in the deployment URL. |
| **payload.links.deployment** | | The URL on the Vercel Dashboard to inspect the deployment. |
| **payload.links.project** | | The URL on the Vercel Dashboard to the project. |
| **payload.target** | | A String that indicates the target. Possible values are `production`, `staging` or `null`. |
| **payload.project.id** | | The ID of the project. |
| **payload.plan** | | The plan type of the deployment. |
| **payload.regions** | | An array of the supported regions for the deployment. |

### domain.created

Occurs whenever a domain has been created.

| Key | | Description |
| --- | --- | --- |
| **payload.team.id** | | The ID of the event's team (possibly null). |
| **payload.user.id** | | The ID of the event's user. |
| **payload.domain.name** | | The Domain name created. |
| **payload.domain.delegated** | | Whether or not the domain was delegated/shared. |

### domain.auto-renew-changed

Occurs whenever a domain's auto-renewal setting is changed.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.team.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's team (possibly null). |
| **payload.user.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's user. |
| **payload.domain.name** | [String](/docs/rest-api/reference/welcome#types) | The name of the domain. |
| **payload.previous** | [Boolean](/docs/rest-api/reference/welcome#types) | The previous auto-renewal setting. |
| **payload.next** | [Boolean](/docs/rest-api/reference/welcome#types) | The new auto-renewal setting. |

### domain.certificate-add

Occurs whenever a new SSL certificate is added for a domain.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.team.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's team (possibly null). |
| **payload.user.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's user. |
| **payload.cert** | [Object](/docs/rest-api/reference/welcome#types) | The certificate object containing certificate details. |

### domain.certificate-add-failed

Occurs whenever adding a new SSL certificate for a domain fails.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.team.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's team (possibly null). |
| **payload.user.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's user. |
| **payload.dnsNames** | [List](/docs/rest-api/reference/welcome#types) | An array of DNS names for which the certificate addition failed |

### domain.certificate-deleted

Occurs whenever an SSL certificate is deleted for a domain.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.team.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's team (possibly null). |
| **payload.user.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's user. |
| **payload.cert** | [Object](/docs/rest-api/reference/welcome#types) | The certificate object containing certificate details. |

### domain.certificate-renew

Occurs whenever an SSL certificate is renewed for a domain.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.team.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's team (possibly null). |
| **payload.user.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's user. |
| **payload.cert** | [Object](/docs/rest-api/reference/welcome#types) | The certificate object containing certificate details. |

### domain.certificate-renew-failed

Occurs whenever renewing an SSL certificate for a domain fails.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.team.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's team (possibly null). |
| **payload.user.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's user. |
| **payload.dnsNames** | [List](/docs/rest-api/reference/welcome#types) | An array of DNS names for which the certificate renewal failed. |

### domain.dns-records-changed

Occurs whenever DNS records for a domain are modified.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.team.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's team (possibly null). |
| **payload.user.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's user. |
| **payload.zone** | [String](/docs/rest-api/reference/welcome#types) | The DNS zone that was modified. |
| **payload.changes** | [List](/docs/rest-api/reference/welcome#types) | An array of changes made to the DNS records. |

### domain.renewal

Occurs whenever a domain is renewed.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.team.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's team (possibly null). |
| **payload.user.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's user. |
| **payload.domain.name** | [String](/docs/rest-api/reference/welcome#types) | The name of the domain that was renewed. |
| **payload.price** | [String](/docs/rest-api/reference/welcome#types) | The renewal price as a decimal number. |
| **payload.expirationDate** | [Date](/docs/rest-api/reference/welcome#types) | The new expiration date of the domain. |
| **payload.renewedAt** | [Date](/docs/rest-api/reference/welcome#types) | The timestamp when the domain was renewed. |

### domain.renewal-failed

Occurs whenever a domain renewal fails.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.team.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's team (possibly null). |
| **payload.user.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's user. |
| **payload.domain.name** | [String](/docs/rest-api/reference/welcome#types) | The name of the domain for which renewal failed. |
| **payload.errorReason** | [String](/docs/rest-api/reference/welcome#types) | The reason why the renewal failed. |
| **payload.failedAt** | [Date](/docs/rest-api/reference/welcome#types) | The timestamp when the renewal failed. |

### domain.transfer-in-completed

Occurs whenever a domain transfer into Vercel is completed.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.team.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's team (possibly null). |
| **payload.user.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's user. |
| **payload.domain.name** | [String](/docs/rest-api/reference/welcome#types) | The name of the domain that was transferred. |

### domain.transfer-in-failed

Occurs whenever a domain transfer into Vercel fails.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.team.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's team (possibly null). |
| **payload.user.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's user. |
| **payload.domain.name** | [String](/docs/rest-api/reference/welcome#types) | The name of the domain for which the transfer failed. |

### domain.transfer-in-started

Occurs whenever a domain transfer into Vercel is initiated.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.team.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's team (possibly null). |
| **payload.user.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's user. |
| **payload.domain.name** | [String](/docs/rest-api/reference/welcome#types) | The name of the domain for which the transfer was started. |

### project.domain-created

Occurs whenever a domain is added to a project.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.team.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's team (possibly null). |
| **payload.user.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's user. |

| **payload.project.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the project. |
| **payload.domain.name** | [String](/docs/rest-api/reference/welcome#types) | The name of the domain that was added to the project. |

### project.domain-deleted

Occurs whenever a domain is removed from a project.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.team.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's team (possibly null). |
| **payload.user.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's user. |
| **payload.project.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the project. |
| **payload.domain.name** | [String](/docs/rest-api/reference/welcome#types) | The name of the domain that was removed from the project. |

### project.domain-moved

Occurs whenever a domain is moved from one project to another.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.team.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's team (possibly null). |
| **payload.user.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's user. |
| **payload.domain.name** | [String](/docs/rest-api/reference/welcome#types) | The name of the domain that was moved. |
| **payload.from.projectId** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the project the domain was moved from. |
| **payload.to.projectId** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the project the domain was moved to. |
| **payload.isRedirect** | [Boolean](/docs/rest-api/reference/welcome#types) | Whether the move created a redirect. |

### project.domain-unverified

Occurs whenever a project domain becomes unverified.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.team.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's team (possibly null). |
| **payload.user.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's user. |
| **payload.project.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the project. |
| **payload.domain.name** | [String](/docs/rest-api/reference/welcome#types) | The name of the domain that became unverified. |

### project.domain-updated

Occurs whenever a project domain is updated.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.team.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's team (possibly null) |
| **payload.user.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's user. |
| **payload.project.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the project. |
| **payload.previous.domain** | [String](/docs/rest-api/reference/welcome#types) | The previous domain name. |
| **payload.previous.redirect** | [String](/docs/rest-api/reference/welcome#types) | The previous redirect URL (possibly null). |
| **payload.previous.redirectStatusCode** | [Number](/docs/rest-api/reference/welcome#types) | The previous redirect status code (possibly |
| **payload.previous.gitBranch** | [String](/docs/rest-api/reference/welcome#types) | The previous git branch (possibly null). |
| **payload.next.domain** | [String](/docs/rest-api/reference/welcome#types) | The new domain name. |
| **payload.next.redirect** | [String](/docs/rest-api/reference/welcome#types) | The new redirect URL (possibly null). |
| **payload.next.redirectStatusCode** | [Number](/docs/rest-api/reference/welcome#types) | The new redirect status code (possibly nul |
| **payload.next.gitBranch** | [String](/docs/rest-api/reference/welcome#types) | The new git branch (possibly null). |

### project.domain-verified

Occurs whenever a project domain is verified.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.team.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's team (possibly null). |
| **payload.user.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the event's user. |
| **payload.project.id** | [ID](/docs/rest-api/reference/welcome#types) | The ID of the project. |
| **payload.domain.name** | [String](/docs/rest-api/reference/welcome#types) | The name of the domain that was verified. |

### integration-configuration.permission-upgraded

Occurs whenever the user changes the project permission for an integration.

| Key | | Description |
| --- | --- | --- |
| **payload.team.id** | | The ID of the event's team (possibly null). |
| **payload.user.id** | | The ID of the event's user. |
| **payload.configuration.id** | | The ID of the configuration. |
| **payload.configuration.projectSelection** | | A String representing the permission for projects. Possible values are `all` or ` |
| **payload.configuration.projects** | | An array of project IDs. |
| **payload.projects.added** | | An array of added project IDs. |
| **payload.projects.removed** | | An array of removed project IDs. |

### integration-configuration.removed

Occurs whenever an integration has been removed.

| Key | | Description |
| --- | --- | --- |
| **payload.team.id** | | The ID of the event's team (possibly null). |
| **payload.user.id** | | The ID of the event's user. |
| **payload.configuration.id** | | The ID of the configuration. |
| **payload.configuration.projectSelection** | | A String representing the permission for projects. Possible values are `all` or ` |
| **payload.configuration.projects** | | An array of project IDs. |

### integration-configuration.scope-change-confirmed

Occurs whenever the user confirms pending scope changes.

| Key | | Description |
| --- | --- | --- |
| **payload.team.id** | | The ID of the event's team (possibly null). |

| **payload.user.id**             |           | The ID of the event's user.               |
| **payload.configuration.id**    |           | The ID of the configuration.              |
| **payload.configuration.scopes** |         | List of all scopes (after confirmation).  |

### integration-configuration.transferred

Occurs whenever the integration installation has been transferred to another team.

| Key                           |  | Description                                                                                 |
| ----------------------------- | ----------------------------------------------------------------------------------------- | ----------------------
| **payload.configuration.id**  |           | The ID of the integration installation.                                          |
| **payload.installationId**    |           | The ID of the integration installation (same as `configuration.id`).             |
| **payload.previousTeamId**    |           | The ID of the previous installation owner team.                                  |
| **payload.previousAccountId** |           | The ID of the previous installation account (for marketplace installations).     |
| **payload.newTeamId**         |           | The ID of the new installation owner team.                                       |
| **payload.newAccountId**      |           | The ID of the new installation account (for marketplace installations).          |

### integration-resource.project-connected

Occurs whenever the user connects the integration resource to a project.

| Key                           |  | Description                                                                                 |
| ----------------------------- | ----------------------------------------------------------------------------------------- | ----------------------
| **payload.configuration.id** |           | The ID of the integration installation.                                          |
| **payload.installationId**   |           | The ID of the integration installation (same as `configuration.id`).             |
| **payload.resourceId**       |           | The ID of the resource.                                                          |
| **payload.project.id**       |           | The ID of the project.                                                           |
| **payload.project.name**     |        | The name of the project.                                                    |
| **payload.projectId**        |           | The ID of the project (same as project.id).                                      |
| **payload.targets**          |         | The list of the deployment targets.                                              |

### integration-resource.project-disconnected

Occurs whenever the user disconnects the integration resource to a project.

| Key                           |  | Description                                                                                 |
| ----------------------------- | ----------------------------------------------------------------------------------------- | ----------------------
| **payload.configuration.id** |           | The ID of the integration installation.                                          |
| **payload.installationId**   |           | The ID of the integration installation (same as `configuration.id`).             |
| **payload.resourceId**       |           | The ID of the resource.                                                          |
| **payload.project.id**       |           | The ID of the project.                                                           |
| **payload.projectId**        |           | The ID of the project (same as project.id).                                      |
| **payload.targets**          |         | The list of the deployment targets.                                              |

### marketplace.invoice.created

Occurs when an invoice was created and sent to the customer.

| Key                           |  | Description                                                                                 |
| ----------------------------- | ----------------------------------------------------------------------------------------- | ----------------------
| **payload.configuration.id**  |           | The ID of the integration installation.                                          |
| **payload.installationId**    |           | The ID of the integration installation (same as `configuration.id`).             |
| **payload.invoiceId**         |           | The ID of the Marketplace invoice.                                               |
| **payload.externalInvoiceId** |           | The ID of the Marketplace invoice, provided by integrator. Possibly `null`.      |
| **payload.period.start**      |       | The invoice's period start date.                                                 |
| **payload.period.end**        |       | The invoice's period end date.                                                   |
| **payload.invoiceDate**       |       | The invoice's date.                                                              |
| **payload.invoiceTotal**      |        | The invoice's total as a decimal number.                                         |

### marketplace.invoice.notpaid

Occurs when an invoice was not paid after a grace period.

| Key                           |  | Description                                                                                 |
| ----------------------------- | ----------------------------------------------------------------------------------------- | ----------------------
| **payload.configuration.id**  |           | The ID of the integration installation.                                          |
| **payload.installationId**    |           | The ID of the integration installation (same as `configuration.id`).             |
| **payload.invoiceId**         |           | The ID of the Marketplace invoice.                                               |
| **payload.externalInvoiceId** |           | The ID of the Marketplace invoice, provided by integrator. Possibly `null`.      |
| **payload.period.start**      |       | The invoice's period start date.                                                 |
| **payload.period.end**        |       | The invoice's period end date.                                                   |
| **payload.invoiceDate**       |       | The invoice's date.                                                              |
| **payload.invoiceTotal**      |        | The invoice's total as a decimal number.                                         |

### marketplace.invoice.paid

Occurs when an invoice was paid.

| Key                           |  | Description                                                                                 |
| ----------------------------- | ----------------------------------------------------------------------------------------- | ----------------------
| **payload.configuration.id**  |           | The ID of the integration installation.                                          |
| **payload.installationId**    |           | The ID of the integration installation (same as `configuration.id`).             |
| **payload.invoiceId**         |           | The ID of the Marketplace invoice.                                               |
| **payload.externalInvoiceId** |           | The ID of the Marketplace invoice, provided by integrator. Possibly `null`.      |
| **payload.period.start**      |       | The invoice's period start date.                                                 |
| **payload.period.end**        |       | The invoice's period end date.                                                   |
| **payload.invoiceDate**       |       | The invoice's date.                                                              |
| **payload.invoiceTotal**      |        | The invoice's total as a decimal number.                                         |

### marketplace.invoice.refunded

Occurs when an invoice is refunded.

| Key                           |  | Description                                                                                 |
| ----------------------------- | ----------------------------------------------------------------------------------------- | ----------------------
| **payload.configuration.id**  |           | The ID of the integration installation.                                          |
| **payload.installationId**    |           | The ID of the integration installation (same as `configuration.id`).             |
| **payload.invoiceId**         |           | The ID of the Marketplace invoice.                                               |
| **payload.externalInvoiceId** |           | The ID of the Marketplace invoice, provided by integrator. Possibly `null`.      |
| **payload.period.start**      |       | The invoice's period start date.                                                 |

| **payload.period.end** | The invoice's period end date. |
| **payload.amount** | The amount being refunded as a decimal number. |
| **payload.reason** | The reason for why the refund has been issued. |

### marketplace.member.changed

Occurs whenever a member is added, removed, or their role changed for an installation.

| Key | Description |
| --- | --- |
| **payload.configuration.id** | The ID of the integration installation. |
| **payload.installationId** | The ID of the integration installation (same as `configuration.id`). |
| **payload.memberId** | The ID of the member. |
| **payload.role** | The member's role: "ADMIN", "USER" or "NONE". "NONE" indicates the member has been removed. |
| **payload.globalUserId** | The ID of the user. Requires separate permission. |
| **payload.userEmail** | The email of the user. Requires separate permission. |

### alerts.triggered

Occurs whenever an alert is triggered.

| Key | [Type](/docs/rest-api/reference/welcome#types) | Description |
| --- | --- | --- |
| **payload.teamId** | [String](/docs/rest-api/reference/welcome#types) | The ID of the team. |
| **payload.projectId** | [String](/docs/rest-api/reference/welcome#types) | The ID of the project. |
| **payload.startedAt** | [Number](/docs/rest-api/reference/welcome#types) | Timestamp when the anomaly started (milliseco |
| **payload.links.observability** | [String](/docs/rest-api/reference/welcome#types) | URL to the observability dashboard for this a |
| **payload.projectSlug** | [String](/docs/rest-api/reference/welcome#types) | The project slug. |
| **payload.teamSlug** | [String](/docs/rest-api/reference/welcome#types) | The team slug. |
| **payload.groupId** | [String](/docs/rest-api/reference/welcome#types) | Optional group identifier for related alerts. |
| **payload.alerts\[].startedAt** | [String](/docs/rest-api/reference/welcome#types) | ISO 8601 timestamp when this specific alert |
| **payload.alerts\[].title** | [String](/docs/rest-api/reference/welcome#types) | Human-readable title for the alert. |
| **payload.alerts\[].unit** | [String](/docs/rest-api/reference/welcome#types) | Unit of measurement (e.g., `requests`). |
| **payload.alerts\[].formattedValues** | [Object](/docs/rest-api/reference/welcome#types) | Formatted values for display purposes. |
| **payload.alerts\[].count** | [Number](/docs/rest-api/reference/welcome#types) | Total count of events during the anomaly per |
| **payload.alerts\[].average** | [Number](/docs/rest-api/reference/welcome#types) | Average value during the anomaly period. |
| **payload.alerts\[].stddev** | [Number](/docs/rest-api/reference/welcome#types) | Standard deviation of the metric. |
| **payload.alerts\[].zscore** | [Number](/docs/rest-api/reference/welcome#types) | Z-score indicating how many standard deviati |
| **payload.alerts\[].zscoreThreshold** | [Number](/docs/rest-api/reference/welcome#types) | Z-score threshold that triggered the alert. |
| **payload.alerts\[].alertId** | [String](/docs/rest-api/reference/welcome#types) | Unique identifier for this alert. |
| **payload.alerts\[].type** | [String](/docs/rest-api/reference/welcome#types) | Alert type |
| **payload.alerts\[].metric** | [String](/docs/rest-api/reference/welcome#types) | Metric identifier, for example, `edge_reques |

See the [Alerts documentation](/docs/alerts) for more details and examples.

### project.created

Occurs whenever a project has been created.

> **💡 Note:** This event is sent only when the Integration has access to all projects in a
> Vercel scope.

| Key | Description |
| --- | --- |
| **payload.team.id** | The ID of the event's team (possibly null). |
| **payload.user.id** | The ID of the event's user. |
| **payload.project.id** | The ID of the project. |
| **payload.project.name** | Name of the project. |

### project.removed

Occurs whenever a project has been removed.

> **💡 Note:** This event is sent only when the integration has access to all projects in a
> Vercel scope.

| Key | Description |
| --- | --- |
| **payload.team.id** | The ID of the event's team (possibly null). |
| **payload.user.id** | The ID of the event's user. |
| **payload.project.id** | The ID of the project. |
| **payload.project.name** | Name of the project. |

### project.renamed

Occurs whenever a project has been renamed.

> **💡 Note:** This event is sent only when the integration has access to all projects in a
> Vercel scope.

| Key | Description |
| --- | --- |
| **payload.team.id** | The ID of the event's team (possibly null). |
| **payload.user.id** | The ID of the event's user (possibly null). |
| **payload.project.id** | The ID of the project. |
| **payload.project.name** | The new name of the project. |
| **payload.previousName** | The previous name of the project. |

### project.rolling-release.approved

Occurs whenever a rolling release stage is approved and progresses to the next stage.

| Key | Description |
| --- | --- |
| **payload.team.id** | The ID of the event's team (possibly null). |
| **payload.user.id** | The ID of the event's user. |
| **payload.project.id** | The ID of the project. |
| **payload.project.name** | Name of the project. |
| **payload.rollingRelease** | The current rolling release configuration. |
| **payload.rollingRelease.projectId** | The ID of the project. |

| **payload.rollingRelease.ownerId** | | The ID of the team or user that owns the rolling release. |
| **payload.rollingRelease.deploymentIds** | | Array of deployment IDs involved in the rolling release. |
| **payload.rollingRelease.state** | | The current state of the rolling release. Possible values are `ACTIVE` |
| **payload.rollingRelease.activeStageIndex** | | The index of the currently active stage. |
| **payload.rollingRelease.default** | | The default deployment configuration. |
| **payload.rollingRelease.default.baseDeploymentId** | | The ID of the base deployment. |
| **payload.rollingRelease.default.targetDeploymentId** | | The ID of the target deployment. |
| **payload.rollingRelease.default.targetPercentage** | | The target percentage of traffic to route to the target deployment. |
| **payload.rollingRelease.default.targetStartAt** | | The timestamp when the target deployment started. |
| **payload.rollingRelease.default.targetUpdatedAt** | | The timestamp when the target deployment was last updated. |
| **payload.rollingRelease.config** | | The rolling release configuration. |
| **payload.rollingRelease.config.target** | | The target environment for the rolling release. |
| **payload.rollingRelease.config.stages** | | Array of stage configurations. |
| **payload.rollingRelease.writtenBy** | | The source that triggered the rolling release update. |
| **payload.prevRollingRelease** | | The previous rolling release configuration before the approval. |

### project.rolling-release.completed

Occurs whenever a rolling release is completed successfully.

| Key | | Description |
| --- | --- | --- |
| **payload.team.id** | | The ID of the event's team (possibly null). |
| **payload.user.id** | | The ID of the event's user. |
| **payload.project.id** | | The ID of the project. |
| **payload.project.name** | | Name of the project. |
| **payload.rollingRelease** | | The completed rolling release configuration. |
| **payload.rollingRelease.projectId** | | The ID of the project. |
| **payload.rollingRelease.ownerId** | | The ID of the team or user that owns the rolling release. |
| **payload.rollingRelease.deploymentIds** | | Array of deployment IDs involved in the rolling release. |
| **payload.rollingRelease.state** | | The state of the rolling release (will be `COMPLETE`). |
| **payload.rollingRelease.activeStageIndex** | | The index of the final stage. |
| **payload.rollingRelease.default** | | The final deployment configuration. |
| **payload.rollingRelease.default.baseDeploymentId** | | The ID of the base deployment. |
| **payload.rollingRelease.default.targetDeploymentId** | | The ID of the target deployment. |
| **payload.rollingRelease.default.targetPercentage** | | The final target percentage (will be 100). |
| **payload.rollingRelease.default.targetStartAt** | | The timestamp when the target deployment started. |
| **payload.rollingRelease.default.targetUpdatedAt** | | The timestamp when the target deployment was last updated. |
| **payload.rollingRelease.config** | | The rolling release configuration. |
| **payload.rollingRelease.config.target** | | The target environment for the rolling release. |
| **payload.rollingRelease.config.stages** | | Array of stage configurations. |
| **payload.rollingRelease.writtenBy** | | The source that completed the rolling release. |
| **payload.prevRollingRelease** | | The previous rolling release configuration before completion. |

### project.rolling-release.aborted

Occurs whenever a rolling release is aborted.

| Key | | Description |
| --- | --- | --- |
| **payload.team.id** | | The ID of the event's team (possibly null). |
| **payload.user.id** | | The ID of the event's user. |
| **payload.project.id** | | The ID of the project. |
| **payload.project.name** | | Name of the project. |
| **payload.rollingRelease** | | The aborted rolling release configuration. |
| **payload.rollingRelease.projectId** | | The ID of the project. |
| **payload.rollingRelease.ownerId** | | The ID of the team or user that owns the rolling release. |
| **payload.rollingRelease.deploymentIds** | | Array of deployment IDs involved in the rolling release. |
| **payload.rollingRelease.state** | | The state of the rolling release (will be `ABORTED`). |
| **payload.rollingRelease.activeStageIndex** | | The index of the stage when aborted. |
| **payload.rollingRelease.default** | | The deployment configuration at the time of abortion. |
| **payload.rollingRelease.default.baseDeploymentId** | | The ID of the base deployment. |
| **payload.rollingRelease.default.targetDeploymentId** | | The ID of the target deployment. |
| **payload.rollingRelease.default.targetStartAt** | | The timestamp when the target deployment started. |
| **payload.rollingRelease.default.targetUpdatedAt** | | The timestamp when the rolling release was aborted. |
| **payload.rollingRelease.config** | | The rolling release configuration. |
| **payload.rollingRelease.config.target** | | The target environment for the rolling release. |
| **payload.rollingRelease.config.stages** | | Array of stage configurations. |
| **payload.rollingRelease.writtenBy** | | The source that aborted the rolling release. |
| **payload.prevRollingRelease** | | The previous rolling release configuration before abortion. |

### project.rolling-release.started

Occurs whenever a rolling release is started.

| Key | | Description |
| --- | --- | --- |
| **payload.team.id** | | The ID of the event's team (possibly null). |
| **payload.user.id** | | The ID of the event's user. |
| **payload.project.id** | | The ID of the project. |
| **payload.project.name** | | Name of the project. |
| **payload.rollingRelease** | | The started rolling release configuration. |
| **payload.rollingRelease.projectId** | | The ID of the project. |
| **payload.rollingRelease.ownerId** | | The ID of the team or user that owns the rolling release. |
| **payload.rollingRelease.deploymentIds** | | Array of deployment IDs involved in the rolling release. |
| **payload.rollingRelease.state** | | The state of the rolling release (will be `ACTIVE`). |
| **payload.rollingRelease.activeStageIndex** | | The index of the initial stage (usually 0). |
| **payload.rollingRelease.default** | | The initial deployment configuration. |
| **payload.rollingRelease.default.baseDeploymentId** | | The ID of the base deployment. |
| **payload.rollingRelease.default.targetDeploymentId** | | The ID of the target deployment. |
| **payload.rollingRelease.default.targetPercentage** | | The initial target percentage for the first stage. |
| **payload.rollingRelease.default.targetStartAt** | | The timestamp when the rolling release started. |
| **payload.rollingRelease.default.targetUpdatedAt** | | The timestamp when the rolling release was last updated. |
| **payload.rollingRelease.config** | | The rolling release configuration. |
| **payload.rollingRelease.config.target** | | The target environment for the rolling release. |
| **payload.rollingRelease.config.stages** | | Array of stage configurations. |
| **payload.rollingRelease.writtenBy** | | The source that started the rolling release. |
| **payload.prevRollingRelease** | | The previous rolling release configuration (if any) before starting th |

## Legacy Payload

The legacy webhook payload is a JSON object with the following keys.

| Key | Description |
| --- | --- |
| **type** | The [legacy event type](#legacy-event-types). |
| **id** | The ID of the webhook delivery. |
| **createdAt** | The date and time the webhook event was generated. |
| **region** | The region the event occurred in (possibly null). |
| **clientId** | The ID of integration's client. |
| **ownerId** | The ID of the event owner (user or team). |
| **teamId** | The ID of the event's team (possibly null). |
| **userId** | The ID of the event's users. |
| **webhookId** | The ID of the webhook. |
| **payload** | The payload of the webhook. See [Legacy Event Types](#legacy-event-types) for more information. |

## Legacy Event Types

The following event types have been deprecated and webhooks that listen for them can no longer be created. Vercel will continue to delive

### deployment

> **💡 Note:** This event is replaced by [deployment.created](#deployment.created).

Occurs whenever a deployment is created.

| Key | Description |
| --- | --- |
| **payload.alias** | An array of aliases that will get assigned when the deployment is ready. |
| **payload.deployment.id** | The ID of the deployment. |
| **payload.deployment.meta** | A Map of deployment metadata. |
| **payload.deployment.url** | The URL of the deployment. |
| **payload.deployment.name** | The project name used in the deployment URL. |
| **payload.links.deployment** | The URL on the Vercel Dashboard to inspect the deployment. |
| **payload.links.project** | The URL on the Vercel Dashboard to the project. |
| **payload.target** | A String that indicates the target. Possible values are `production`, `staging` or `null`. |
| **payload.projectId** | The ID of the project. |
| **payload.plan** | The plan type of the deployment. |
| **payload.regions** | An array of the supported regions for the deployment. |

### deployment-ready

> **💡 Note:** This event is replaced by [deployment.succeeded](#deployment.succeeded).

Occurs whenever a deployment is ready.

> **💡 Note:** This event gets fired after all blocking checks have passed. See [
>
> ](/docs/integrations#webhooks/events-types/deployment-prepared) if you
> registered Checks.

| Key | Description |
| --- | --- |
| **payload.deployment.id** | The ID of the deployment. |
| **payload.deployment.meta** | A Map of deployment metadata. |
| **payload.deployment.url** | The URL of the deployment. |
| **payload.deployment.name** | The project name used in the deployment URL. |
| **payload.links.deployment** | The URL on the Vercel Dashboard to inspect the deployment. |
| **payload.links.project** | The URL on the Vercel Dashboard to the project. |
| **payload.target** | A String that indicates the target. Possible values are `production`, `staging` or `null`. |
| **payload.projectId** | The ID of the project. |
| **payload.plan** | The plan type of the deployment. |
| **payload.regions** | An array of the supported regions for the deployment. |

### deployment-prepared

> **💡 Note:** This event is replaced by [deployment.ready](#deployment.ready).

Occurs whenever a deployment is successfully built and your integration has registered at least one [check](/docs/integrations/checks-ove

| Key | Description |
| --- | --- |
| **payload.deployment.id** | The ID of the deployment. |
| **payload.deployment.meta** | A Map of deployment metadata. |
| **payload.deployment.url** | The URL of the deployment. |
| **payload.deployment.name** | The project name used in the deployment URL. |
| **payload.links.deployment** | The URL on the Vercel Dashboard to inspect the deployment. |
| **payload.links.project** | The URL on the Vercel Dashboard to the project. |
| **payload.target** | A String that indicates the target. Possible values are `production`, `staging` or `null`. |
| **payload.projectId** | The ID of the project. |
| **payload.plan** | The plan type of the deployment. |
| **payload.regions** | An array of the supported regions for the deployment. |

### deployment-canceled

> **💡 Note:** This event is replaced by [deployment.canceled](#deployment.canceled).

Occurs whenever a deployment is canceled.

| Key | Description |
| --- | --- |
| **payload.deployment.id** | The ID of the deployment. |
| **payload.deployment.meta** | A Map of deployment metadata. |
| **payload.deployment.url** | The URL of the deployment. |
| **payload.deployment.name** | The project name used in the deployment URL. |
| **payload.links.deployment** | The URL on the Vercel Dashboard to inspect the deployment. |
| **payload.links.project** | The URL on the Vercel Dashboard to the project. |
| **payload.target** | A String that indicates the target. Possible values are `production`, `staging` or `null`. |
| **payload.projectId** | The ID of the project. |
| **payload.plan** | The plan type of the deployment. |

| **payload.regions** | | An array of the supported regions for the deployment. |

### deployment-error

> **💡 Note:** This event is replaced by [deployment.error](#deployment.error).

Occurs whenever a deployment has failed.

| Key | | Description |
| --- | --- | --- |
| **payload.deployment.id** | | The ID of the deployment. |
| **payload.deployment.meta** | | A Map of deployment metadata. |
| **payload.deployment.url** | | The URL of the deployment. |
| **payload.deployment.name** | | The project name used in the deployment URL. |
| **payload.links.deployment** | | The URL on the Vercel Dashboard to inspect the deployment. |
| **payload.links.project** | | The URL on the Vercel Dashboard to the project. |
| **payload.target** | | A String that indicates the target. Possible values are `production`, `staging` or `null`. |
| **payload.projectId** | | The ID of the project. |
| **payload.plan** | | The plan type of the deployment. |
| **payload.regions** | | An array of the supported regions for the deployment. |

### deployment-check-rerequested

> **💡 Note:** This event is replaced by [
> deployment.check-rerequested](#deployment.check-rerequested).

Occurs when a user has requested for a [check](/docs/integrations/checks-overview) to be rerun after it failed.

| Key | | Description |
| --- | --- | --- |
| **payload.deployment.id** | | The ID of the deployment. |
| **payload.check.id** | | The ID of the check. |

### deployment-checks-completed

> **💡 Note:** This event has been removed. [deployment.succeeded](#deployment.succeeded) can
> be used for the same purpose.

Occurs when all checks for a deployment have completed. This does not indicate that they have all passed, only that they are no longer ru

| Key | | Description |
| --- | --- | --- |
| **payload.deployment.id** | | The ID of the deployment. |
| **payload.checks** | | Information about the Checks. |

Each item in `checks` has the following properties:

| Key | | Description |
| --- | --- | --- |
| **payload.id** | | The unique identifier of the check. Always prepended with `check_`. |
| **payload.name** | | The name of the check. |
| **payload.status** | | The status of the check. One of `registered`, `running` or `completed` |
| **payload.conclusion** | | The conclusion of the check. One of `cancelled`, `failed`, `neutral`, `succeeded` or `skipped`. |
| **payload.blocking** | | Whether a deployment should be blocked or not. |
| **payload.integrationId** | | The unique identifier of the integration. |

### project-created

> **💡 Note:** This event is replaced by [project.created](#project.created).

Occurs whenever a project has been created.

> **💡 Note:** This event is sent only when the Integration has access to all projects in a
> Vercel scope.

| Key | | Description |
| --- | --- | --- |
| **payload.project.id** | | The ID of the project. |
| **payload.project.name** | | Name of the project. |

### project-removed

> **💡 Note:** This event is replaced by [project.removed](#project.removed).

Occurs whenever a Project has been removed.

> **💡 Note:** This event is sent only when the Integration has access to all Projects in a
> Vercel scope.

| Key | | Description |
| --- | --- | --- |
| **payload.project.id** | | The ID of the project. |
| **payload.project.name** | | Name of the project. |

### integration-configuration-removed

> **💡 Note:** This event is replaced by [
> integration-configuration.removed](#integration-configuration.removed).

Occurs whenever an integration has been removed.

| Key | | Description |
| --- | --- | --- |
| **payload.configuration.id** | | The ID of the configuration. |
| **payload.configuration.projects** | | An array of project IDs. |

### integration-configuration-permission-updated

> **💡 Note:** This event is replaced by [
> integration-configuration.permission-upgraded](#integration-configuration.permission-upgraded)

> .

Occurs whenever the user changes the project permission for an integration.

| Key | | Description
| --- | --- | ---
| **payload.configuration.id** | | The ID of the configuration.
| **payload.configuration.projectSelection** | | A String representing the permission for projects. Possible values are `all` or `
| **payload.configuration.projects** | | An array of project IDs.
| **payload.projects.added** | | An array of added project IDs.
| **payload.projects.removed** | | An array of removed project IDs.

### integration-configuration-scope-change-confirmed

> **💡 Note:** This event is replaced by [
> integration-configuration.scope-change-confirmed](#integration-configuration.scope-change-confirmed)
> .

Occurs whenever the user confirms pending scope changes.

| Key | | Description |
| --- | --- | --- |
| **payload.configuration.id** | | The ID of the configuration. |
| **payload.configuration.scopes** | | List of all scopes (after confirmation). |

### domain-created

> **💡 Note:** This event is replaced by [domain.created](#domain.created).

Occurs whenever a domain has been created.

| Key | | Description |
| --- | --- | --- |
| **payload.domain.name** | | The Domain name created. |
| **payload.domain.delegated** | | Whether or not the domain was delegated/shared. |

## Securing webhooks

Once your server is configured to receive payloads, it will listen for any payload sent to the endpoint you configured. By knowing the UR

The recommended method to check is to use the [`x-vercel-signature`](/docs/headers/request-headers#x-vercel-signature) security header yo

- For account webhooks, this is the [secret displayed when creating the webhook](/docs/webhooks#enter-your-endpoint-url).
- For integration webhooks, use your Integration Secret (also called Client Secret) from the [Integration Console](https://vercel.com/d?t

For example, you can validate a webhook request as follows:

```ts filename="pages/api/webhook-validator-example.ts" framework="nextjs"
import type { NextApiRequest, NextApiResponse } from 'next';
import crypto from 'crypto';
import getRawBody from 'raw-body';

export default async function handler(
  request: NextApiRequest,
  response: NextApiResponse,
) {
  const { INTEGRATION_SECRET } = process.env;

  if (typeof INTEGRATION_SECRET != 'string') {
    throw new Error('No integration secret found');
  }

  const rawBody = await getRawBody(request);
  const bodySignature = sha1(rawBody, INTEGRATION_SECRET);

  if (bodySignature !== request.headers['x-vercel-signature']) {
    return response.status(403).json({
      code: 'invalid_signature',
      error: "signature didn't match",
    });
  }

  const json = JSON.parse(rawBody.toString('utf-8'));

  switch (json.type) {
    case 'project.created':
    // ...
  }

  return response.status(200).end('OK');
}

function sha1(data: Buffer, secret: string): string {
  return crypto.createHmac('sha1', secret).update(data).digest('hex');
}

export const config = {
  api: {
    bodyParser: false,
  },
};
```

```js filename="pages/api/webhook-validator-example.js" framework="nextjs"
import type { NextApiRequest, NextApiResponse } from 'next';
import crypto from 'crypto';
import getRawBody from 'raw-body';

export default async function handler(
  request: NextApiRequest,
```

```
  response: NextApiResponse,
) {
  const { INTEGRATION_SECRET } = process.env;

  if (typeof INTEGRATION_SECRET != 'string') {
    throw new Error('No integration secret found');
  }

  const rawBody = await getRawBody(request);
  const bodySignature = sha1(rawBody, INTEGRATION_SECRET);

  if (bodySignature !== request.headers['x-vercel-signature']) {
    return response.status(403).json({
      code: 'invalid_signature',
      error: "signature didn't match",
    });
  }

  const json = JSON.parse(rawBody.toString('utf-8'));

  switch (json.type) {
    case 'project.created':
    // ...
  }

  return response.status(200).end('OK');
}

function sha1(data: Buffer, secret: string): string {
  return crypto.createHmac('sha1', secret).update(data).digest('hex');
}

export const config = {
  api: {
    bodyParser: false,
  },
};
```

```ts filename="api/webhook-validator-example.ts" framework="other"
import type { VercelRequest, VercelResponse } from '@vercel/node';
import crypto from 'crypto';
import getRawBody from 'raw-body';

export default async function handler(
  request: VercelRequest,
  response: VercelResponse,
) {
  const { INTEGRATION_SECRET } = process.env;

  if (typeof INTEGRATION_SECRET != 'string') {
    throw new Error('No integration secret found');
  }

  const rawBody = await getRawBody(request);
  const bodySignature = sha1(rawBody, INTEGRATION_SECRET);

  if (bodySignature !== request.headers['x-vercel-signature']) {
    return response.status(403).json({
      code: 'invalid_signature',
      error: "signature didn't match",
    });
  }

  const json = JSON.parse(rawBody.toString('utf-8'));

  switch (json.type) {
    case 'project.created':
    // ...
  }

  return response.status(200).end('OK');
}

function sha1(data: Buffer, secret: string): string {
  return crypto.createHmac('sha1', secret).update(data).digest('hex');
}

export const config = {
  api: {
    bodyParser: false,
  },
};
```

```js filename="api/webhook-validator-example.js" framework="other"
import type { VercelRequest, VercelResponse } from '@vercel/node';
import crypto from 'crypto';
import getRawBody from 'raw-body';

export default async function handler(
  request: VercelRequest,
  response: VercelResponse,
) {
  const { INTEGRATION_SECRET } = process.env;

  if (typeof INTEGRATION_SECRET != 'string') {
    throw new Error('No integration secret found');
  }
```

```
  const rawBody = await getRawBody(request);
  const bodySignature = sha1(rawBody, INTEGRATION_SECRET);

  if (bodySignature !== request.headers['x-vercel-signature']) {
    return response.status(403).json({
      code: 'invalid_signature',
      error: "signature didn't match",
    });
  }

  const json = JSON.parse(rawBody.toString('utf-8'));

  switch (json.type) {
    case 'project.created':
    // ...
  }

  return response.status(200).end('OK');
}

function sha1(data: Buffer, secret: string): string {
  return crypto.createHmac('sha1', secret).update(data).digest('hex');
}

export const config = {
  api: {
    bodyParser: false,
  },
};
```

```ts filename="app/api/webhook-validator-example/route.ts" framework="nextjs-app"
import crypto from 'crypto';

export async function GET(request: Request) {
  const { INTEGRATION_SECRET } = process.env;

  if (typeof INTEGRATION_SECRET != 'string') {
    throw new Error('No integration secret found');
  }

  const rawBody = await request.text();
  const rawBodyBuffer = Buffer.from(rawBody, 'utf-8');
  const bodySignature = sha1(rawBodyBuffer, INTEGRATION_SECRET);

  if (bodySignature !== request.headers.get('x-vercel-signature')) {
    return Response.json({
      code: 'invalid_signature',
      error: "signature didn't match",
    });
  }

  const json = JSON.parse(rawBodyBuffer.toString('utf-8'));

  switch (json.type) {
    case 'project.created':
    // ...
  }

  return new Response('Webhook request validated', {
    status: 200,
  });
}

function sha1(data: Buffer, secret: string): string {
  return crypto.createHmac('sha1', secret).update(data).digest('hex');
}
```

```js filename="app/api/webhook-validator-example/route.js" framework="nextjs-app"
import crypto from 'crypto';

export async function GET(request) {
  const { INTEGRATION_SECRET } = process.env;

  if (typeof INTEGRATION_SECRET != 'string') {
    throw new Error('No integration secret found');
  }

  const rawBody = await request.text();
  const rawBodyBuffer = Buffer.from(rawBody, 'utf-8');
  const bodySignature = sha1(rawBodyBuffer, INTEGRATION_SECRET);

  if (bodySignature !== request.headers.get('x-vercel-signature')) {
    return Response.json({
      code: 'invalid_signature',
      error: "signature didn't match",
    });
  }

  const json = JSON.parse(rawBodyBuffer.toString('utf-8'));

  switch (json.type) {
    case 'project.created':
    // ...
  }

  return new Response('Webhook request validated', {
    status: 200,
  });
```

}

function sha1(data: Buffer, secret: string): string {
  return crypto.createHmac('sha1', secret).update(data).digest('hex');
}
```



> **💡 Note:** For enhanced security against timing attacks, use constant-time comparison
> when verifying the `x-vercel-signature` header. See [x-vercel-signature in
> Request Headers](/docs/headers/request-headers#x-vercel-signature).

You can compute the signature using an HMAC hexdigest from the secret token of OAuth2 and request body, then compare it with the value of

## HTTP Response

You should consider this HTTP request to be an event. Once you receive the request, you should schedule a task for your action.

This request has a timeout of 30 seconds. That means if a `2XX` HTTP response is not received within 30 seconds, the request will be abor

## Delivery Attempts and Retries

If your HTTP endpoint does not respond with a `2XX` HTTP status code, we attempt to deliver the webhook event up to 24 hours with an expo


---
title: "Vercel Workflow"
description: "Build durable, reliable, and observable applications and AI agents with the Workflow Development Kit (WDK)."
last_updated: "2026-01-16T02:19:36.374Z"
source: "https://vercel.com/docs/workflow"
---

# Vercel Workflow

Vercel Workflow is a fully managed platform built on top of the
open-source [Workflow Development Kit (WDK)](https://useworkflow.dev),
a TypeScript framework for building apps and AI agents that can pause,
resume, and maintain state.

With Workflow, Vercel manages the infrastructure for you so you can focus on writing business logic. **Vercel Functions** execute your wo

This means your functions are:

- **Resumable**: Pause for minutes or months, then resume from the exact point.
- **Durable**: Survive deployments and crashes with deterministic replays.
- **Observable**: Use built-in logs, metrics, and tracing and view them in your [Vercel dashboard](https://vercel.com/d?to=%2F%5Bteam%5D%
- **Idiomatic**: Write async/await JavaScript with two directives. No YAML or state machines.

## Getting started

Install the WDK package:

<CodeBlock>
  <Code tab="pnpm">
    ```bash
    pnpm i workflow
    ```
  </Code>
  <Code tab="yarn">
    ```bash
    yarn i workflow
    ```
  </Code>
  <Code tab="npm">
    ```bash
    npm i workflow
    ```
  </Code>
  <Code tab="bun">
    ```bash
    bun i workflow
    ```
  </Code>
</CodeBlock>

Start writing your own workflows by following the [Workflow DevKit getting started guide](https://useworkflow.dev/docs/getting-started).

## Concepts

Workflow introduces two directives that turn ordinary async functions into durable workflows.
You write async/await code as usual, and the framework handles queues, retry logic, and state persistence automatically.

### Workflow

A workflow is a stateful function that coordinates multi-step
logic over time. The ``'use workflow'`` directive marks a function as durable,
which means it remembers its progress and can resume exactly where it left off,
even after pausing, restarting, or deploying new code.

Use a workflow when your logic needs to pause, resume, or span minutes to months:

```typescript filename="app/workflows/ai-content-workflow.ts" {2}
export async function aiContentWorkflow(topic: string) {
  'use workflow';

  const draft = await generateDraft(topic);

  const summary = await summarizeDraft(draft);

````
  return { draft, summary };
}
```

Under the hood, the workflow function compiles into a route that orchestrates execution.
All inputs and outputs are recorded in an event log. If a deploy or crash happens,
the system replays execution deterministically from where it stopped.

### Step

A step is a stateless function that runs a unit of durable work inside a workflow.
The `'use step'` directive marks a function as a step, which gives
it built-in retries and makes it survive failures like network errors or process crashes.

Use a step when calling external APIs or performing isolated operations:

```typescript filename="app/steps/generate-draft.ts" {2,12}
async function generateDraft(topic: string) {
  'use step';

  const draft = await aiGenerate({
    prompt: `Write a blog post about ${topic}`,
  });

  return draft;
}

async function summarizeDraft(draft: string) {
  'use step';

  const summary = await aiSummarize({ text: draft });

  if (Math.random() < 0.3) {
    throw new Error('Transient AI provider error');
  }

  return summary;
}
```

Each step compiles into an isolated API route. While the step executes,
the workflow suspends without consuming resources. When the step
completes, the workflow resumes automatically right where it left off.

### Sleep

Sleep pauses a workflow for a specified duration without consuming compute resources.
This is useful when you need to wait for hours or days before continuing,
like delaying a follow-up email or waiting to issue a reward.

Use sleep to delay execution without keeping any infrastructure running:

```typescript filename="app/workflows/ai-refine.ts" {8}
import { sleep } from 'workflow';

export async function aiRefineWorkflow(draftId: string) {
  'use workflow';

  const draft = await fetchDraft(draftId);

  await sleep('7 days'); // Wait 7 days to gather more signals; no resources consumed

  const refined = await refineDraft(draft);

  return { draftId, refined };
}
```

During sleep, no resources are consumed. The workflow simply pauses and resumes when the time expires.

### Hook

A hook lets a workflow wait for external events such as user actions, webhooks,
or third-party API responses. This is useful for human-in-the-loop workflows
where you need to pause until someone approves, confirms, or provides input.

Use hooks to pause execution until external data arrives:

```typescript filename="app/workflows/approval.ts" {4,15-17}
import { defineHook } from 'workflow';

// Human approval for AI-generated drafts
const approvalHook = defineHook<{
  decision: 'approved' | 'changes';
  notes?: string;
}>();

export async function aiApprovalWorkflow(topic: string) {
  'use workflow';

  const draft = await generateDraft(topic);

  // Wait for human approval events
  const events = approvalHook.create({
    token: 'draft-123',
  });

  for await (const event of events) {
    if (event.decision === 'approved') {
      await publishDraft(draft);
````

```
      break;
    } else {
      const revised = await refineDraft(draft, event.notes);
      await publishDraft(revised);
      break;
    }
  }
}
```

```typescript filename="app/api/resume/route.ts" {5}
// Resume the workflow when an approval is received
export async function POST(req: Request) {
  const data = await req.json();

  await approvalHook.resume('draft-123', {
    decision: data.decision,
    notes: data.notes,
  });

  return new Response('OK');
}
```

When a hook receives data, the workflow resumes automatically. No polling, message queues, or manual state management required.

## Observability

Every step, input, output, sleep, and error inside a workflow is recorded automatically.

You can track runs in real time, trace failures, and analyze performance without writing extra code.

To inspect your runs, go to your [Vercel dashboard](https://vercel.com/d?to=%2F%5Bteam%5D%2F%5Bproject%5D%2Fai%2Fworkflows\&title=Vercel+W
, select your project and navigate to **AI**, then **Workflows**.

> **💡 Note:** During the Beta period, Workflow Observability is free for all plans. Workflow Steps and Storage are billed at the [rates

## Pricing

Workflow pricing is divided into two resources:

- **Workflow Steps**: Individual units of durable work executed inside a workflow.
- **Workflow Storage**: The amount of data stored in the managed persistence layer for workflow state.

All resources are billed based on usage with each plan having an [included allotment](/docs/pricing).

Functions invoked by Workflows continue to be charged at the [existing compute
rates](/docs/functions/usage-and-pricing). We encourage you to use [Fluid compute](/docs/fluid-compute) with Workflow.

## More resources

- [Workflow Development Kit (WDK)](https://useworkflow.dev)
- [Stateful Slack bots with Vercel Workflow Guide](/kb/guide/stateful-slack-bots-with-vercel-workflow)