The REST API is now versioned. For more information, see "About API versioning."

Repositories

Use the REST API to manage repositories on GitHub.

List organization repositories 🔗

✅ Works with GitHub Apps

Lists repositories for the specified organization.

Note: In order to see the `security_and_analysis` block for a repository you must have admin permissions for the repository or be an owner or security manager for the organization that owns the repository. For more information, see "Managing security managers in your organization."

Parameters for "List organization repositories"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`org` string   Required

The organization name. The name is not case sensitive.

Query parameters

`type` string

Specifies the types of repositories you want returned. `internal` is not yet supported when a GitHub App calls this endpoint with an installation access token.

Can be one of: `all`, `private`, `forks`, `sources`, `member`, `internal`

`sort` string

The property to sort the results by.

Default: `created`
Can be one of: `created`, `updated`, `pushed`, `full_name`

`direction` string

The order to sort by. Default: `asc` when using `full_name`, otherwise `desc`.

Can be one of: `asc`, `desc`

`per page` integer

The number of results per page (max 100).

Default: `30`

---

page integer

Page number of the results to fetch.

Default: `1`

HTTP response status codes for "List organization repositories"

Status code	Description
`200`	OK

Code samples for "List organization repositories"

cURL   JavaScript   GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28"
\ https://api.github.com/orgs/ORG/repos
```

Response

Example response   Response schema

```
  Status: 200
```

```
[ { "id": 1296269, "node_id": "MDEwOlJlcG9zaXRvcnkxMjk2MjY5", "name": "Hello-World", "full_name": "octocat/Hello-World", "owner":
{ "login": "octocat", "id": 1, "node_id": "MDQ6VXNlcjE=", "avatar_url": "https://github.com/images/error/octocat_happy.gif",
"gravatar_id": "", "url": "https://api.github.com/users/octocat", "html_url": "https://github.com/octocat", "followers_url":
"https://api.github.com/users/octocat/followers", "following_url": "https://api.github.com/users/octocat/following{/other_user}",
"gists_url": "https://api.github.com/users/octocat/gists{/gist_id}", "starred_url":
"https://api.github.com/users/octocat/starred{/owner}{/repo}", "subscriptions_url":
```

Create an organization repository 🔗

✅ Works with [GitHub Apps](#)

Creates a new repository in the specified organization. The authenticated user must be a member of the organization.

OAuth scope requirements

When using [OAuth](#), authorizations must include:

- `public_repo` scope or `repo` scope to create a public repository. Note: For GitHub AE, use `repo` scope to create an internal repository.
- `repo` scope to create a private repository

Parameters for "Create an organization repository"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

org string Required

The organization name. The name is not case sensitive.

Body parameters

name string Required

The name of the repository.

description string

A short description of the repository.

homepage string

A URL with more information about the repository.

private boolean

Whether the repository is private.

Default: `false`

visibility string

The visibility of the repository.

Can be one of: `public` , `private` , `internal`

has_issues boolean

Either `true` to enable issues for this repository or `false` to disable them.

Default: `true`

has_projects boolean

Either `true` to enable projects for this repository or `false` to disable them. **Note:** If you're creating a repository in an organization that has disabled repository projects, the default is `false` , and if you pass `true` , the API returns an error.

Default: `true`

has_wiki boolean

Either `true` to enable the wiki for this repository or `false` to disable it.

Default: `true`

has_downloads boolean

Whether downloads are enabled.

Default: `true`

is_template boolean

Either `true` to make this repo available as a template repository or `false` to prevent it.

Default: `false`

team_id integer

The id of the team that will be granted access to this repository. This is only valid when creating a repository in an organization.

---

auto_init boolean

Pass `true` to create an initial commit with empty README.

Default: `false`

---

gitignore_template string

Desired language or platform [.gitignore template](#) to apply. Use the name of the template without the extension. For example, "Haskell".

---

license_template string

Choose an [open source license template](#) that best suits your needs, and then use the [license keyword](#) as the `license_template` string. For example, "mit" or "mpl-2.0".

---

allow_squash_merge boolean

Either `true` to allow squash-merging pull requests, or `false` to prevent squash-merging.

Default: `true`

---

allow_merge_commit boolean

Either `true` to allow merging pull requests with a merge commit, or `false` to prevent merging pull requests with merge commits.

Default: `true`

---

allow_rebase_merge boolean

Either `true` to allow rebase-merging pull requests, or `false` to prevent rebase-merging.

Default: `true`

---

allow_auto_merge boolean

Either `true` to allow auto-merge on pull requests, or `false` to disallow auto-merge.

Default: `false`

---

delete_branch_on_merge boolean

Either `true` to allow automatically deleting head branches when pull requests are merged, or `false` to prevent automatic deletion. **The authenticated user must be an organization owner to set this property to `true` .**

Default: `false`

---

use_squash_pr_title_as_default boolean

Either `true` to allow squash-merge commits to use pull request title, or `false` to use commit message. **This property has been deprecated. Please use `squash_merge_commit_title` instead.

Default: `false`

---

squash_merge_commit_title string

The default value for a squash merge commit title:
- `PR_TITLE` - default to the pull request's title.
- `COMMIT_OR_PR_TITLE` - default to the commit's title (if only one commit) or the pull request's title (when more than one commit).

Can be one of: `PR_TITLE` , `COMMIT_OR_PR_TITLE`

---

squash_merge_commit_message string

The default value for a squash merge commit message:
- `PR_BODY` - default to the pull request's body.
- `COMMIT_MESSAGES` - default to the branch's commit messages.
- `BLANK` - default to a blank commit message.

Can be one of: `PR_BODY` , `COMMIT_MESSAGES` , `BLANK`

---

merge_commit_title   string

The default value for a merge commit title.
- `PR_TITLE` - default to the pull request's title.
- `MERGE_MESSAGE` - default to the classic title for a merge message (e.g., Merge pull request #123 from branch-name).

Can be one of: `PR_TITLE` , `MERGE_MESSAGE`

---

merge_commit_message   string

The default value for a merge commit message.
- `PR_TITLE` - default to the pull request's title.
- `PR_BODY` - default to the pull request's body.
- `BLANK` - default to a blank commit message.

Can be one of: `PR_BODY` , `PR_TITLE` , `BLANK`

HTTP response status codes for "Create an organization repository"

Status code	Description
`201`	Created
`403`	Forbidden
`422`	Validation failed, or the endpoint has been spammed.

Code samples for "Create an organization repository"

POST `/orgs/{org}/repos`

cURL   JavaScript   GitHub CLI

```
curl -L \ -X POST \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version:
2022-11-28" \ https://api.github.com/orgs/ORG/repos \ -d '{"name":"Hello-World","description":"This is your first
repository","homepage":"https://github.com","private":false,"has_issues":true,"has_projects":true,"has_wiki":true}'
```

Response

Example response   Response schema

Status: 201

```
{ "id": 1296269, "node_id": "MDEwOlJlcG9zaXRvcnkxMjk2MjY5", "name": "Hello-World", "full_name": "octocat/Hello-World", "owner": {
"login": "octocat", "id": 1, "node_id": "MDQ6VXNlcjE=", "avatar_url": "https://github.com/images/error/octocat_happy.gif",
"gravatar_id": "", "url": "https://api.github.com/users/octocat", "html_url": "https://github.com/octocat", "followers_url":
"https://api.github.com/users/octocat/followers", "following_url": "https://api.github.com/users/octocat/following{/other_user}",
"gists_url": "https://api.github.com/users/octocat/gists{/gist_id}", "starred_url":
"https://api.github.com/users/octocat/starred{/owner}{/repo}", "subscriptions_url":
```

Get a repository 🔗

✓ Works with [GitHub Apps](#)

The `parent` and `source` objects are present when the repository is a fork. `parent` is the repository this repository was forked from, `source` is the ultimate source for the network.

Note: In order to see the `security_and_analysis` block for a repository you must have admin permissions for the repository or be an owner or security manager for the organization that owns the repository. For more information, see "[Managing security managers in your organization](#)."

Parameters for "Get a repository"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`owner` string Required

The account owner of the repository. The name is not case sensitive.

`repo` string Required

The name of the repository without the `.git` extension. The name is not case sensitive.

HTTP response status codes for "Get a repository"

Status code	Description
`200`	OK
`301`	Moved permanently
`403`	Forbidden
`404`	Resource not found

Code samples for "Get a repository"

GET `/repos/{owner}/{repo}`

cURL   JavaScript   GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28"
\ https://api.github.com/repos/OWNER/REPO
```

Default response

Example response   Response schema

```
Status: 200
```

{ "id": 1296269, "node_id": "MDEwOlJlcG9zaXRvcnkxMjk2MjY5", "name": "Hello-World", "full_name": "octocat/Hello-World", "owner": { "login": "octocat", "id": 1, "node_id": "MDQ6VXNlcjE=", "avatar_url": "https://github.com/images/error/octocat_happy.gif", "gravatar_id": "", "url": "https://api.github.com/users/octocat", "html_url": "https://github.com/octocat", "followers_url": "https://api.github.com/users/octocat/followers", "following_url": "https://api.github.com/users/octocat/following{/other_user}", "gists_url": "https://api.github.com/users/octocat/gists{/gist_id}", "starred_url": "https://api.github.com/users/octocat/starred{/owner}{/repo}", "subscriptions_url":

Update a repository 🔗

✔ Works with [GitHub Apps](#)

Note: To edit a repository's topics, use the [Replace all repository topics](#) endpoint.

Parameters for "Update a repository"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`owner` string   Required

The account owner of the repository. The name is not case sensitive.

`repo` string   Required

The name of the repository without the `.git` extension. The name is not case sensitive.

Body parameters

`name` string

The name of the repository.

`description` string

A short description of the repository.

`homepage` string

A URL with more information about the repository.

`private` boolean

Either `true` to make the repository private or `false` to make it public. Default: `false` .
Note: You will get a `422` error if the organization restricts [changing repository visibility](#) to organization owners and a non-owner tries to change the value of private.

Default: `false`

`visibility` string

The visibility of the repository.

Can be one of: `public` , `private` , `internal`

security_and_analysis   object or null

Specify which security and analysis features to enable or disable for the repository.

To use this parameter, you must have admin permissions for the repository or be an owner or security manager for the organization that owns the repository. For more information, see "[Managing security managers in your organization](#)."

For example, to enable GitHub Advanced Security, use this data in the body of the `PATCH` request: `{ "security_and_analysis": {"advanced_security": { "status": "enabled" } } }`.

You can check which security and analysis features are currently enabled by using a `GET /repos/{owner}/{repo}` request.

▶ Properties of `security_and_analysis`

---

has_issues   boolean

Either `true` to enable issues for this repository or `false` to disable them.

Default: `true`

---

has_projects   boolean

Either `true` to enable projects for this repository or `false` to disable them. **Note:** If you're creating a repository in an organization that has disabled repository projects, the default is `false`, and if you pass `true`, the API returns an error.

Default: `true`

---

has_wiki   boolean

Either `true` to enable the wiki for this repository or `false` to disable it.

Default: `true`

---

is_template   boolean

Either `true` to make this repo available as a template repository or `false` to prevent it.

Default: `false`

---

default_branch   string

Updates the default branch for this repository.

---

allow_squash_merge   boolean

Either `true` to allow squash-merging pull requests, or `false` to prevent squash-merging.

Default: `true`

---

allow_merge_commit   boolean

Either `true` to allow merging pull requests with a merge commit, or `false` to prevent merging pull requests with merge commits.

Default: `true`

---

allow_rebase_merge   boolean

Either `true` to allow rebase-merging pull requests, or `false` to prevent rebase-merging.

Default: `true`

---

allow_auto_merge   boolean

Either `true` to allow auto-merge on pull requests, or `false` to disallow auto-merge.

Default: `false`

---

delete_branch_on_merge   boolean

Either `true` to allow automatically deleting head branches when pull requests are merged, or `false` to prevent automatic deletion.

Default: `false`

---

allow_update_branch   boolean

Either `true` to always allow a pull request head branch that is behind its base branch to be updated even if it is not required to be up

Either `true` to always allow a pull request head branch that is behind its base branch to be updated even if it is not required to be up to date before merging, or false otherwise.

Default: `false`

`use_squash_pr_title_as_default` boolean

Either `true` to allow squash-merge commits to use pull request title, or `false` to use commit message. **This property has been deprecated. Please use `squash_merge_commit_title` instead.

Default: `false`

`squash_merge_commit_title` string

The default value for a squash merge commit title:
- `PR_TITLE` - default to the pull request's title.
- `COMMIT_OR_PR_TITLE` - default to the commit's title (if only one commit) or the pull request's title (when more than one commit).

Can be one of: `PR_TITLE` , `COMMIT_OR_PR_TITLE`

`squash_merge_commit_message` string

The default value for a squash merge commit message:
- `PR_BODY` - default to the pull request's body.
- `COMMIT_MESSAGES` - default to the branch's commit messages.
- `BLANK` - default to a blank commit message.

Can be one of: `PR_BODY` , `COMMIT_MESSAGES` , `BLANK`

`merge_commit_title` string

The default value for a merge commit title.
- `PR_TITLE` - default to the pull request's title.
- `MERGE_MESSAGE` - default to the classic title for a merge message (e.g., Merge pull request #123 from branch-name).

Can be one of: `PR_TITLE` , `MERGE_MESSAGE`

`merge_commit_message` string

The default value for a merge commit message.
- `PR_TITLE` - default to the pull request's title.
- `PR_BODY` - default to the pull request's body.
- `BLANK` - default to a blank commit message.

Can be one of: `PR_BODY` , `PR_TITLE` , `BLANK`

`archived` boolean

Whether to archive this repository. `false` will unarchive a previously archived repository.

Default: `false`

`allow_forking` boolean

Either `true` to allow private forks, or `false` to prevent private forks.

Default: `false`

`web_commit_signoff_required` boolean

Either `true` to require contributors to sign off on web-based commits, or `false` to not require contributors to sign off on web-based commits.

Default: `false`

HTTP response status codes for "Update a repository"

Status code	Description
200	OK
307	Temporary Redirect
403	Forbidden
404	Resource not found
422	Validation failed, or the endpoint has been spammed.

Code samples for "Update a repository"

`PATCH` /repos/{owner}/{repo}

cURL   JavaScript   GitHub CLI

```
curl -L \ -X PATCH \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version:
2022-11-28" \ https://api.github.com/repos/OWNER/REPO \ -d '{"name":"Hello-World","description":"This is your first
repository","homepage":"https://github.com","private":true,"has_issues":true,"has_projects":true,"has_wiki":true}'
```

Response

Example response   Response schema

Status: 200

```
{ "id": 1296269, "node_id": "MDEwOlJlcG9zaXRvcnkxMjk2MjY5", "name": "Hello-World", "full_name": "octocat/Hello-World", "owner": {
"login": "octocat", "id": 1, "node_id": "MDQ6VXNlcjE=", "avatar_url": "https://github.com/images/error/octocat_happy.gif",
"gravatar_id": "", "url": "https://api.github.com/users/octocat", "html_url": "https://github.com/octocat", "followers_url":
"https://api.github.com/users/octocat/followers", "following_url": "https://api.github.com/users/octocat/following{/other_user}",
"gists_url": "https://api.github.com/users/octocat/gists{/gist_id}", "starred_url":
"https://api.github.com/users/octocat/starred{/owner}{/repo}", "subscriptions_url":
```

Delete a repository 🔗

✅ Works with [GitHub Apps](#)

Deleting a repository requires admin access. If OAuth is used, the `delete_repo` scope is required.

If an organization owner has configured the organization to prevent members from deleting organization-owned repositories, you will get a `403 Forbidden` response.

Parameters for "Delete a repository"

Headers

accept   string

Setting to `application/vnd.github+json` is recommended.

Path parameters

owner string Required

The account owner of the repository. The name is not case sensitive.

---

repo string Required

The name of the repository without the `.git` extension. The name is not case sensitive.

HTTP response status codes for "Delete a repository"

Status code	Description
`204`	No Content
`307`	Temporary Redirect
`403`	If an organization owner has configured the organization to prevent members from deleting organization-owned repositories, a member will get this response:
`404`	Resource not found

Code samples for "Delete a repository"

> **DELETE** /repos/{owner}/{repo}

cURL   JavaScript   GitHub CLI

```
curl -L \ -X DELETE \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version:
2022-11-28" \ https://api.github.com/repos/OWNER/REPO
```

Response

```
Status: 204
```

List repository activities 🔗

✔ Works with [GitHub Apps](#)

Lists a detailed history of changes to a repository, such as pushes, merges, force pushes, and branch changes, and associates these changes with commits and users.

For more information about viewing repository activity, see "[Viewing activity and data for your repository](#)."

Parameters for "List repository activities"

Headers

---

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

owner string   Required

The account owner of the repository. The name is not case sensitive.

repo string   Required

The name of the repository without the `.git` extension. The name is not case sensitive.

Query parameters

direction string

The direction to sort the results by.

Default: `desc`
Can be one of: `asc` , `desc`

per_page integer

The number of results per page (max 100).

Default: `30`

before string

A cursor, as given in the [Link header](#). If specified, the query only searches for results before this cursor.

after string

A cursor, as given in the [Link header](#). If specified, the query only searches for results after this cursor.

ref string

The Git reference for the activities you want to list.
The `ref` for a branch can be formatted either as `refs/heads/BRANCH_NAME` or `BRANCH_NAME` , where `BRANCH_NAME` is the name of your branch.

actor string

The GitHub username to use to filter by the actor who performed the activity.

time_period string

The time period to filter by.
For example, `day` will filter for activity that occurred in the past 24 hours, and `week` will filter for activity that occurred in the past 7 days (168 hours).

Can be one of: `day` , `week` , `month` , `quarter` , `year`

activity_type string

The activity type to filter by.
For example, you can choose to filter by "force_push", to see all force pushes to the repository.

Can be one of: `push` , `force_push` , `branch_creation` , `branch_deletion` , `pr_merge` , `merge_queue_merge`

HTTP response status codes for "List repository activities"

Status code	Description
200	OK

200	OK
422	Validation failed, or the endpoint has been spammed.

Code samples for "List repository activities"

GET `/repos/{owner}/{repo}/activity`

cURL   JavaScript   GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28"
\ https://api.github.com/repos/OWNER/REPO/activity
```

Response

Example response   Response schema

Status: 200

```
[ { "id": 1296269, "node_id": "MDEwOlJlcG9zaXRvcnkxMjk2MjY5", "before": "6dcb09b5b57875f334f61aebed695e2e4193db5e", "after":
"827efc6d56897b048c772eb4087f854f46256132", "ref": "refs/heads/main", "pushed_at": "2011-01-26T19:06:43Z", "push_type": "normal",
"pusher": { "login": "octocat", "id": 1, "node_id": "MDQ6VXNlcjE=", "avatar_url":
"https://github.com/images/error/octocat_happy.gif", "gravatar_id": "", "url": "https://api.github.com/users/octocat", "html_url":
"https://github.com/octocat", "followers_url": "https://api.github.com/users/octocat/followers", "following_url":
"https://api.github.com/users/octocat/following{/other_user}", "gists_url": "https://api.github.com/users/octocat/gists{/gist_id}",
```

Check if automated security fixes are enabled for a repository 🔗

✓ Works with [GitHub Apps](#)

Shows whether automated security fixes are enabled, disabled or paused for a repository. The authenticated user must have admin read access to the repository. For more information, see "[Configuring automated security fixes](#)".

Parameters for "Check if automated security fixes are enabled for a repository"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

owner string   Required

The account owner of the repository. The name is not case sensitive.

repo string   Required

The name of the repository without the `.git` extension. The name is not case sensitive.

HTTP response status codes for "Check if automated security fixes are enabled for a repository"

Status code	Description
200	Response if dependabot is enabled
404	Not Found if dependabot is not enabled for the repository

Code samples for "Check if automated security fixes are enabled for a repository"

GET `/repos/{owner}/{repo}/automated-security-fixes`

cURL | JavaScript | GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28"
\ https://api.github.com/repos/OWNER/REPO/automated-security-fixes
```

Response if dependabot is enabled

Example response | Response schema

Status: 200

```
{ "enabled": true, "paused": false }
```

Enable automated security fixes 🔗

✅ Works with [GitHub Apps](#)

Enables automated security fixes for a repository. The authenticated user must have admin access to the repository. For more information, see "[Configuring automated security fixes](#)".

Parameters for "Enable automated security fixes"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

owner string Required

The account owner of the repository. The name is not case sensitive.

repo string Required

The name of the repository without the `.git` extension. The name is not case sensitive.

HTTP response status codes for "Enable automated security fixes"

Status code	Description
204	No Content

Code samples for "Enable automated security fixes"

`PUT` /repos/{owner}/{repo}/automated-security-fixes

cURL   JavaScript   GitHub CLI

```
curl -L \ -X PUT \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version:
2022-11-28" \ https://api.github.com/repos/OWNER/REPO/automated-security-fixes
```

Response

```
Status: 204
```

Disable automated security fixes ∅

✔ Works with [GitHub Apps](#)

Disables automated security fixes for a repository. The authenticated user must have admin access to the repository. For more information, see "[Configuring automated security fixes](#)".

Parameters for "Disable automated security fixes"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

owner string   Required

The account owner of the repository. The name is not case sensitive.

repo string   Required

The name of the repository without the `.git` extension. The name is not case sensitive.

HTTP response status codes for "Disable automated security fixes"

Status code	Description
204	No Content

Code samples for "Disable automated security fixes"

DELETE `/repos/{owner}/{repo}/automated-security-fixes`

---

cURL   JavaScript   GitHub CLI   ⧉

```
curl -L \ -X DELETE \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version:
2022-11-28" \ https://api.github.com/repos/OWNER/REPO/automated-security-fixes
```

Response

```
Status: 204
```

List CODEOWNERS errors 🔗

✓ Works with [GitHub Apps](#)

List any syntax errors that are detected in the CODEOWNERS file.

For more information about the correct CODEOWNERS syntax, see "[About code owners](#)."

Parameters for "List CODEOWNERS errors"

Headers

---

accept   string

Setting to `application/vnd.github+json` is recommended.

Path parameters

---

owner   string   Required

The account owner of the repository. The name is not case sensitive.

---

repo   string   Required

The name of the repository without the `.git` extension. The name is not case sensitive.
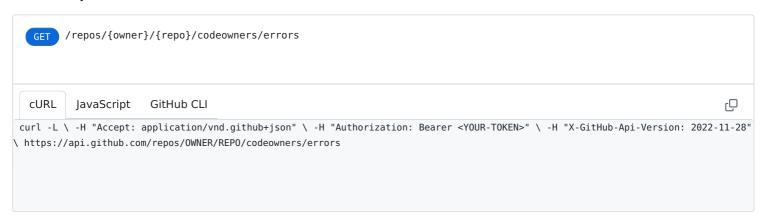

Query parameters

---

ref   string

A branch, tag or commit name used to determine which version of the CODEOWNERS file to use. Default: the repository's default branch (e.g. `main`)

HTTP response status codes for "List CODEOWNERS errors"

Status code	Description
200	OK
404	Resource not found

Code samples for "List CODEOWNERS errors"

`GET` /repos/{owner}/{repo}/codeowners/errors

cURL   JavaScript   GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28"
\ https://api.github.com/repos/OWNER/REPO/codeowners/errors
```

Response

Example response   Response schema

Status: 200

```
{ "errors": [ { "line": 3, "column": 1, "kind": "Invalid pattern", "source": "***/*.rb @monalisa", "suggestion": "Did you mean
`**/*.rb`?", "message": "Invalid pattern on line 3: Did you mean `**/*.rb`?\n\n ***/*.rb @monalisa\n ^", "path": ".github/CODEOWNERS"
}, { "line": 7, "column": 7, "kind": "Invalid owner", "source": "*.txt docs@", "suggestion": null, "message": "Invalid owner on line
7:\n\n *.txt docs@\n ^", "path": ".github/CODEOWNERS" } ] }
```

List repository contributors 🔗

✓ Works with [GitHub Apps](#)

Lists contributors to the specified repository and sorts them by the number of commits per contributor in descending order. This endpoint may return information that is a few hours old because the GitHub REST API caches contributor data to improve performance.

GitHub identifies contributors by author email address. This endpoint groups contribution counts by GitHub user, which includes all associated email addresses. To improve performance, only the first 500 author email addresses in the repository link to GitHub users. The rest will appear as anonymous contributors without associated GitHub user information.

Parameters for "List repository contributors"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

owner string Required

The account owner of the repository. The name is not case sensitive.

repo string Required

The name of the repository without the `.git` extension. The name is not case sensitive.

Query parameters

anon string

Set to `1` or `true` to include anonymous contributors in results.

per_page integer

The number of results per page (max 100).

Default: `30`

page integer

Page number of the results to fetch.

Default: `1`

HTTP response status codes for "List repository contributors"

Status code	Description
200	If repository contains content
204	Response if repository is empty
403	Forbidden
404	Resource not found

Code samples for "List repository contributors"

GET `/repos/{owner}/{repo}/contributors`

cURL   JavaScript   GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \ https://api.github.com/repos/OWNER/REPO/contributors
```

If repository contains content

Example response   Response schema

Status: 200

```
[ { "login": "octocat", "id": 1, "node_id": "MDQ6VXNlcjE=", "avatar_url": "https://github.com/images/error/octocat_happy.gif", "gravatar_id": "", "url": "https://api.github.com/users/octocat", "html_url": "https://github.com/octocat", "followers_url": "https://api.github.com/users/octocat/followers", "following_url": "https://api.github.com/users/octocat/following{/other_user}", "gists_url": "https://api.github.com/users/octocat/gists{/gist_id}", "starred_url":
```

"https://api.github.com/users/octocat/starred{/owner}{/repo}", "subscriptions_url":
"https://api.github.com/users/octocat/subscriptions", "organizations_url": "https://api.github.com/users/octocat/orgs",

Create a repository dispatch event 🔗

✅ Works with [GitHub Apps](#)

You can use this endpoint to trigger a webhook event called `repository_dispatch` when you want activity that happens outside of GitHub Enterprise Cloud to trigger a GitHub Actions workflow or GitHub App webhook. You must configure your GitHub Actions workflow or GitHub App to run when the `repository_dispatch` event occurs. For an example `repository_dispatch` webhook payload, see "[RepositoryDispatchEvent](#)."

The `client_payload` parameter is available for any extra information that your workflow might need. This parameter is a JSON payload that will be passed on when the webhook event is dispatched. For example, the `client_payload` can include a message that a user would like to send using a GitHub Actions workflow. Or the `client_payload` can be used as a test to debug your workflow.

This endpoint requires write access to the repository by providing either:

- Personal access tokens with `repo` scope. For more information, see "[Creating a personal access token for the command line](#)" in the GitHub Help documentation.
- GitHub Apps with both `metadata:read` and `contents:read&write` permissions.

This input example shows how you can use the `client_payload` as a test to debug your workflow.

Parameters for "Create a repository dispatch event"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

owner string Required

The account owner of the repository. The name is not case sensitive.

repo string Required

The name of the repository without the `.git` extension. The name is not case sensitive.

Body parameters

event_type string Required

A custom webhook event name. Must be 100 characters or fewer.

client_payload object

JSON payload with extra information about the webhook event that your action or workflow may use. The maximum number of top-level properties is 10.

HTTP response status codes for "Create a repository dispatch event"

Status code	Description
204	No Content
422	Validation failed, or the endpoint has been spammed.

Code samples for "Create a repository dispatch event"

POST /repos/{owner}/{repo}/dispatches

cURL   JavaScript

```
curl -L \ -X POST \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version:
2022-11-28" \ https://api.github.com/repos/OWNER/REPO/dispatches \ -d '{"event_type":"on-demand-test","client_payload":
{"unit":false,"integration":true}}'
```

Response

```
Status: 204
```

List repository languages 🔗

✅ Works with [GitHub Apps](#)

Lists languages for the specified repository. The value shown for each language is the number of bytes of code written in that language.

Parameters for "List repository languages"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

owner string   Required

The account owner of the repository. The name is not case sensitive.

repo string   Required

The name of the repository without the `.git` extension. The name is not case sensitive.

HTTP response status codes for "List repository languages"

Status code	Description
200	OK

Code samples for "List repository languages"

GET `/repos/{owner}/{repo}/languages`

cURL   JavaScript   GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \ https://api.github.com/repos/OWNER/REPO/languages
```

Response

Example response   Response schema

Status: 200

```
{ "C": 78769, "Python": 7769 }
```

Enable private vulnerability reporting for a repository 🔗

✅ Works with [GitHub Apps](#)

Enables private vulnerability reporting for a repository. The authenticated user must have admin access to the repository. For more information, see "[Privately reporting a security vulnerability](#)."

Parameters for "Enable private vulnerability reporting for a repository"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

owner string   Required

The account owner of the repository. The name is not case sensitive.

repo string   Required

The name of the repository without the `.git` extension. The name is not case sensitive.

HTTP response status codes for "Enable private vulnerability reporting for a repository"

Status code	Description

204	A header with no content is returned.
422	Bad Request

Code samples for "Enable private vulnerability reporting for a repository"

PUT `/repos/{owner}/{repo}/private-vulnerability-reporting`

---

cURL   JavaScript   GitHub CLI

```
curl -L \ -X PUT \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version:
2022-11-28" \ https://api.github.com/repos/OWNER/REPO/private-vulnerability-reporting
```

A header with no content is returned.

```
Status: 204
```

Disable private vulnerability reporting for a repository 🔗

✅ Works with [GitHub Apps](#)

Disables private vulnerability reporting for a repository. The authenticated user must have admin access to the repository. For more information, see "[Privately reporting a security vulnerability](#)".

Parameters for "Disable private vulnerability reporting for a repository"

Headers

`accept`   string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`owner`   string   Required

The account owner of the repository. The name is not case sensitive.

`repo`   string   Required

The name of the repository without the `.git` extension. The name is not case sensitive.

HTTP response status codes for "Disable private vulnerability reporting for a repository"

Status code	Description
204	A header with no content is returned.
422	Bad Request

Code samples for "Disable private vulnerability reporting for a repository"

DELETE `/repos/{owner}/{repo}/private-vulnerability-reporting`

cURL   JavaScript   GitHub CLI   ⧉

```
curl -L \ -X DELETE \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version:
2022-11-28" \ https://api.github.com/repos/OWNER/REPO/private-vulnerability-reporting
```

A header with no content is returned.

```
Status: 204
```

List repository tags ⧉

✔ Works with GitHub Apps

Parameters for "List repository tags"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`owner` string   Required

The account owner of the repository. The name is not case sensitive.

`repo` string   Required

The name of the repository without the `.git` extension. The name is not case sensitive.

Query parameters

`per_page` integer

The number of results per page (max 100).

Default: `30`

`page` integer

Page number of the results to fetch.

Default: `1`

HTTP response status codes for "List repository tags"

Status code	Description
`200`	OK

Code samples for "List repository tags"

GET `/repos/{owner}/{repo}/tags`

cURL   JavaScript   GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \ https://api.github.com/repos/OWNER/REPO/tags
```

Response

Example response   Response schema

Status: 200

```
[ { "name": "v0.1", "commit": { "sha": "c5b97d5ae6c19d5c5df71a34c7fbeeda2479ccbc", "url": "https://api.github.com/repos/octocat/Hello-World/commits/c5b97d5ae6c19d5c5df71a34c7fbeeda2479ccbc" }, "zipball_url": "https://github.com/octocat/Hello-World/zipball/v0.1", "tarball_url": "https://github.com/octocat/Hello-World/tarball/v0.1", "node_id": "MDQ6VXNlcjE=" } ]
```

List repository teams 🔗

✓ Works with [GitHub Apps](#)

Lists the teams that have access to the specified repository and that are also visible to the authenticated user.

For a public repository, a team is listed only if that team added the public repository explicitly.

Personal access tokens require the following scopes:

- `public_repo` to call this endpoint on a public repository
- `repo` to call this endpoint on a private repository (this scope also includes public repositories)

This endpoint is not compatible with fine-grained personal access tokens.

Parameters for "List repository teams"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

owner string   Required

The account owner of the repository. The name is not case sensitive.

repo string   Required

The name of the repository without the `.git` extension. The name is not case sensitive.

Query parameters

per_page integer

The number of results per page (max 100).

Default: `30`

page integer

Page number of the results to fetch.

Default: `1`

HTTP response status codes for "List repository teams"

Status code	Description
`200`	OK
`404`	Resource not found

Code samples for "List repository teams"

> `GET` /repos/{owner}/{repo}/teams

cURL   JavaScript   GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28"
\ https://api.github.com/repos/OWNER/REPO/teams
```

Response

Example response   Response schema

Status: 200

```
[ { "id": 1, "node_id": "MDQ6VGVhbTE=", "url": "https://api.github.com/teams/1", "html_url":
"https://github.com/orgs/github/teams/justice-league", "name": "Justice League", "slug": "justice-league", "description": "A great
team.", "privacy": "closed", "notification_setting": "notifications_enabled", "permission": "admin", "members_url":
"https://api.github.com/teams/1/members{/member}", "repositories_url": "https://api.github.com/teams/1/repos", "parent": null } ]
```

Get all repository topics 🔗

✔ Works with GitHub Apps

Parameters for "Get all repository topics"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`owner` string Required

The account owner of the repository. The name is not case sensitive.

`repo` string Required

The name of the repository without the `.git` extension. The name is not case sensitive.

Query parameters

`page` integer

Page number of the results to fetch.

Default: `1`

`per_page` integer

The number of results per page (max 100).

Default: `30`

HTTP response status codes for "Get all repository topics"

Status code	Description
`200`	OK
`404`	Resource not found

Code samples for "Get all repository topics"

GET `/repos/{owner}/{repo}/topics`

cURL   JavaScript   GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28"
\ https://api.github.com/repos/OWNER/REPO/topics
```

Response

Example response   Response schema

Status: 200

```
{ "names": [ "octocat", "atom", "electron", "api" ] }
```

Replace all repository topics 🔗

✓ Works with [GitHub Apps](#)

Parameters for "Replace all repository topics"

Headers

`accept` string	

Setting to `application/vnd.github+json` is recommended.

Path parameters

`owner` string Required	

The account owner of the repository. The name is not case sensitive.

`repo` string Required	

The name of the repository without the `.git` extension. The name is not case sensitive.

Body parameters

`names` array of strings Required	

An array of topics to add to the repository. Pass one or more topics to *replace* the set of existing topics. Send an empty array (`[]`) to clear all topics from the repository. **Note:** Topic `names` cannot contain uppercase letters.

HTTP response status codes for "Replace all repository topics"

Status code	Description
`200`	OK
`404`	Resource not found
`422`	Validation failed, or the endpoint has been spammed.

Code samples for "Replace all repository topics"

PUT `/repos/{owner}/{repo}/topics`

cURL   JavaScript   GitHub CLI

```
curl -L \ -X PUT \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version:
2022-11-28" \ https://api.github.com/repos/OWNER/REPO/topics \ -d '{"names":["octocat","atom","electron","api"]}'
```

Response

Example response   Response schema

Status: 200

```
{ "names": [ "octocat", "atom", "electron", "api" ] }
```

Transfer a repository 🔗

✓ Works with [GitHub Apps](#)

A transfer request will need to be accepted by the new owner when transferring a personal repository to another user. The response will contain the original `owner`, and the transfer will continue asynchronously. For more details on the requirements to transfer personal and organization-owned repositories, see [about repository transfers](#). You must use a personal access token (classic) or an OAuth token for this endpoint. An installation access token or a fine-grained personal access token cannot be used because they are only granted access to a single account.

Parameters for "Transfer a repository"

Headers

`accept`   string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`owner`   string   Required

The account owner of the repository. The name is not case sensitive.

`repo`   string   Required

The name of the repository without the `.git` extension. The name is not case sensitive.

Body parameters

`new_owner`   string   Required

The username or organization name the repository will be transferred to.

`new_name`   string

The new name to be given to the repository.

`team_ids`   array of integers

ID of the team or teams to add to the repository. Teams can only be added to organization-owned repositories.

HTTP response status codes for "Transfer a repository"

Status code	Description
202	Accepted

Code samples for "Transfer a repository"

POST `/repos/{owner}/{repo}/transfer`

cURL   JavaScript   GitHub CLI

```
curl -L \ -X POST \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version:
2022-11-28" \ https://api.github.com/repos/OWNER/REPO/transfer \ -d '{"new_owner":"github","team_ids":
[12,345],"new_name":"octorepo"}'
```

Response

Example response   Response schema

Status: 202

```
{ "id": 1296269, "node_id": "MDEwOlJlcG9zaXRvcnkxMjk2MjY5", "name": "Hello-World", "full_name": "octocat/Hello-World", "owner": {
"login": "octocat", "id": 1, "node_id": "MDQ6VXNlcjE=", "avatar_url": "https://github.com/images/error/octocat_happy.gif",
"gravatar_id": "", "url": "https://api.github.com/users/octocat", "html_url": "https://github.com/octocat", "followers_url":
"https://api.github.com/users/octocat/followers", "following_url": "https://api.github.com/users/octocat/following{/other_user}",
"gists_url": "https://api.github.com/users/octocat/gists{/gist_id}", "starred_url":
"https://api.github.com/users/octocat/starred{/owner}{/repo}", "subscriptions_url":
```

Check if vulnerability alerts are enabled for a repository 🔗

Shows whether dependency alerts are enabled or disabled for a repository. The authenticated user must have admin read access to the repository. For more information, see "About security alerts for vulnerable dependencies".

Parameters for "Check if vulnerability alerts are enabled for a repository"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

owner string   Required

The account owner of the repository. The name is not case sensitive.
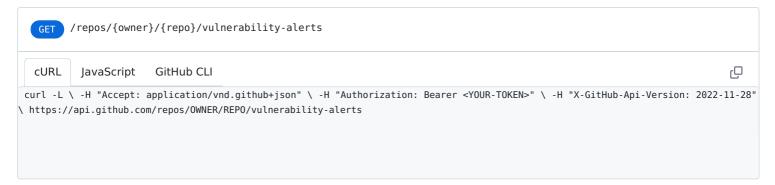
repo string   Required

The name of the repository without the `.git` extension. The name is not case sensitive.

The name of the repository without the `.git` extension. The name is not case sensitive.

HTTP response status codes for "Check if vulnerability alerts are enabled for a repository"

Status code	Description
204	Response if repository is enabled with vulnerability alerts
404	Not Found if repository is not enabled with vulnerability alerts

Code samples for "Check if vulnerability alerts are enabled for a repository"

GET	/repos/{owner}/{repo}/vulnerability-alerts

cURL   JavaScript   GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28"
\ https://api.github.com/repos/OWNER/REPO/vulnerability-alerts
```

Response if repository is enabled with vulnerability alerts

```
Status: 204
```

Enable vulnerability alerts 🔗

✓ Works with [GitHub Apps](#)

Enables dependency alerts and the dependency graph for a repository. The authenticated user must have admin access to the repository. For more information, see "[About security alerts for vulnerable dependencies](#)".

Parameters for "Enable vulnerability alerts"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

owner string Required

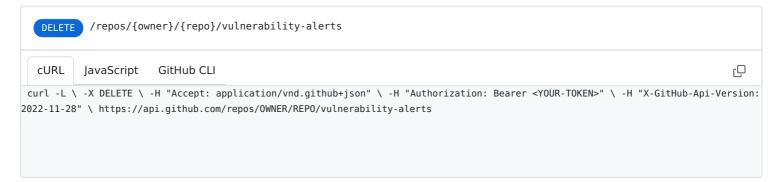The account owner of the repository. The name is not case sensitive.

repo string Required

The name of the repository without the `.git` extension. The name is not case sensitive.

HTTP response status codes for "Enable vulnerability alerts"

Status code	Description
204	No Content

Code samples for "Enable vulnerability alerts"

PUT /repos/{owner}/{repo}/vulnerability-alerts

cURL   JavaScript   GitHub CLI

```
curl -L \ -X PUT \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version:
2022-11-28" \ https://api.github.com/repos/OWNER/REPO/vulnerability-alerts
```

Response

```
Status: 204
```

Disable vulnerability alerts 🔗

✔ Works with [GitHub Apps](#)

Disables dependency alerts and the dependency graph for a repository. The authenticated user must have admin access to the repository. For more information, see "[About security alerts for vulnerable dependencies](#)".

Parameters for "Disable vulnerability alerts"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`owner` string   Required

The account owner of the repository. The name is not case sensitive.

`repo` string   Required

The name of the repository without the `.git` extension. The name is not case sensitive.

HTTP response status codes for "Disable vulnerability alerts"

Status code	Description
204	No Content

Code samples for "Disable vulnerability alerts"

DELETE	/repos/{owner}/{repo}/vulnerability-alerts

cURL	JavaScript	GitHub CLI	

```
curl -L \ -X DELETE \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version:
2022-11-28" \ https://api.github.com/repos/OWNER/REPO/vulnerability-alerts
```

Response

```
Status: 204
```

Create a repository using a template 🔗

Creates a new repository using a repository template. Use the `template_owner` and `template_repo` route parameters to specify the repository to use as the template. If the repository is not public, the authenticated user must own or be a member of an organization that owns the repository. To check if a repository is available to use as a template, get the repository's information using the [Get a repository](#) endpoint and check that the `is_template` key is `true`.

OAuth scope requirements

When using [OAuth](#), authorizations must include:

- `public_repo` scope or `repo` scope to create a public repository. Note: For GitHub AE, use `repo` scope to create an internal repository.
- `repo` scope to create a private repository

Parameters for "Create a repository using a template"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

template_owner string   Required

The account owner of the template repository. The name is not case sensitive.

template_repo string   Required

The name of the template repository without the `.git` extension. The name is not case sensitive.

Body parameters

owner string

The organization or person who will own the new repository. To create a new repository in an organization, the authenticated user must be a member of the specified organization.

name string Required

The name of the new repository.

description string

A short description of the new repository.

include_all_branches boolean

Set to `true` to include the directory structure and files from all branches in the template repository, and not just the default branch. Default: `false`.

Default: `false`

private boolean

Either `true` to create a new private repository or `false` to create a new public one.

Default: `false`

HTTP response status codes for "Create a repository using a template"

Status code	Description
`201`	Created

Code samples for "Create a repository using a template"

`POST` `/repos/{template_owner}/{template_repo}/generate`

cURL   JavaScript   GitHub CLI

```
curl -L \ -X POST \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version:
2022-11-28" \ https://api.github.com/repos/TEMPLATE_OWNER/TEMPLATE_REPO/generate \ -d '{"owner":"octocat","name":"Hello-
World","description":"This is your first repository","include_all_branches":false,"private":false}'
```

Response

Example response   Response schema

Status: 201

```
{ "id": 1296269, "node_id": "MDEwOlJlcG9zaXRvcnkxMjk2MjY5", "name": "Hello-World", "full_name": "octocat/Hello-World", "owner": {
"login": "octocat", "id": 1, "node_id": "MDQ6VXNlcjE=", "avatar_url": "https://github.com/images/error/octocat_happy.gif",
"gravatar_id": "", "url": "https://api.github.com/users/octocat", "html_url": "https://github.com/octocat", "followers_url":
"https://api.github.com/users/octocat/followers", "following_url": "https://api.github.com/users/octocat/following{/other_user}",
"gists_url": "https://api.github.com/users/octocat/gists{/gist_id}", "starred_url":
"https://api.github.com/users/octocat/starred{/owner}{/repo}", "subscriptions_url":
```

List public repositories 🔗

✅ Works with GitHub Apps

Lists all public repositories in the order that they were created.

Note:

- For GitHub Enterprise Server, this endpoint will only list repositories available to all users on the enterprise.
- Pagination is powered exclusively by the `since` parameter. Use the [Link header](#) to get the URL for the next page of repositories.

Parameters for "List public repositories"

Headers

`accept`	string

Setting to `application/vnd.github+json` is recommended.

Query parameters

`since`	integer

A repository ID. Only return repositories with an ID greater than this ID.

HTTP response status codes for "List public repositories"

Status code	Description
`200`	OK
`304`	Not modified
`422`	Validation failed, or the endpoint has been spammed.

Code samples for "List public repositories"

> **GET** `/repositories`

cURL   JavaScript   GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28"
\ https://api.github.com/repositories
```

Response

Example response   Response schema

Status: 200

```
[ { "id": 1296269, "node_id": "MDEwOlJlcG9zaXRvcnkxMjk2MjY5", "name": "Hello-World", "full_name": "octocat/Hello-World", "owner":
{ "login": "octocat", "id": 1, "node_id": "MDQ6VXNlcjE=", "avatar_url": "https://github.com/images/error/octocat_happy.gif",
"gravatar_id": "", "url": "https://api.github.com/users/octocat", "html_url": "https://github.com/octocat", "followers_url":
"https://api.github.com/users/octocat/followers", "following_url": "https://api.github.com/users/octocat/following{/other_user}",
"gists_url": "https://api.github.com/users/octocat/gists{/gist_id}", "starred_url":
"https://api.github.com/users/octocat/starred{/owner}{/repo}", "subscriptions_url":
```

List repositories for the authenticated user 🔗

Lists repositories that the authenticated user has explicit permission (`:read` , `:write` , or `:admin`) to access.

The authenticated user has explicit permission to access repositories they own, repositories where they are a collaborator, and repositories that they can access through an organization membership.

Parameters for "List repositories for the authenticated user"

Headers

---

accept   string

Setting to `application/vnd.github+json` is recommended.

Query parameters

---

visibility   string

Limit results to repositories with the specified visibility.

Default: `all`
Can be one of: `all` , `public` , `private`

---

affiliation   string

Comma-separated list of values. Can include:
- `owner` : Repositories that are owned by the authenticated user.
- `collaborator` : Repositories that the user has been added to as a collaborator.
- `organization_member` : Repositories that the user has access to through being a member of an organization. This includes every repository on every team that the user is on.

Default: `owner,collaborator,organization_member`

---

type   string

Limit results to repositories of the specified type. Will cause a `422` error if used in the same request as **visibility** or **affiliation**.

Default: `all`
Can be one of: `all` , `owner` , `public` , `private` , `member`

---

sort   string

The property to sort the results by.

Default: `full_name`
Can be one of: `created` , `updated` , `pushed` , `full_name`

---

direction   string

The order to sort by. Default: `asc` when using `full_name` , otherwise `desc` .

Can be one of: `asc` , `desc`

---

per_page   integer

The number of results per page (max 100).

Default: `30`

---

page   integer

Page number of the results to fetch.

Default: `1`

---

since string

Only show repositories updated after the given time. This is a timestamp in [ISO 8601](#) format: `YYYY-MM-DDTHH:MM:SSZ` .
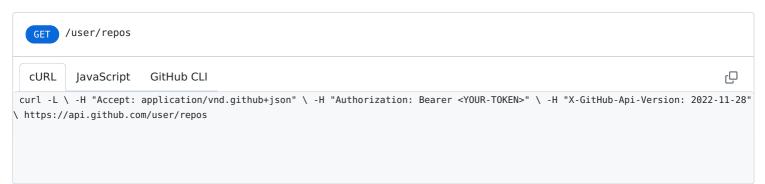
---

before string

Only show repositories updated before the given time. This is a timestamp in [ISO 8601](#) format: `YYYY-MM-DDTHH:MM:SSZ` .

HTTP response status codes for "List repositories for the authenticated user"

Status code	Description
200	OK
304	Not modified
401	Requires authentication
403	Forbidden
422	Validation failed, or the endpoint has been spammed.

Code samples for "List repositories for the authenticated user"

GET `/user/repos`

cURL   JavaScript   GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28"
\ https://api.github.com/user/repos
```

Default response

Example response   Response schema

Status: 200

```
[ { "id": 1296269, "node_id": "MDEwOlJlcG9zaXRvcnkxMjk2MjY5", "name": "Hello-World", "full_name": "octocat/Hello-World", "owner":
{ "login": "octocat", "id": 1, "node_id": "MDQ6VXNlcjE=", "avatar_url": "https://github.com/images/error/octocat_happy.gif",
"gravatar_id": "", "url": "https://api.github.com/users/octocat", "html_url": "https://github.com/octocat", "followers_url":
"https://api.github.com/users/octocat/followers", "following_url": "https://api.github.com/users/octocat/following{/other_user}",
"gists_url": "https://api.github.com/users/octocat/gists{/gist_id}", "starred_url":
"https://api.github.com/users/octocat/starred{/owner}{/repo}", "subscriptions_url":
```

Create a repository for the authenticated user 🔗

Creates a new repository for the authenticated user.

OAuth scope requirements

When using [OAuth](#), authorizations must include:

- `public_repo` scope or `repo` scope to create a public repository. Note: For GitHub AE, use `repo` scope to create an

internal repository.

- `repo` scope to create a private repository.

Parameters for "Create a repository for the authenticated user"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Body parameters

`name` string   Required

The name of the repository.

`description` string

A short description of the repository.

`homepage` string

A URL with more information about the repository.

`private` boolean

Whether the repository is private.

Default: `false`

`has_issues` boolean

Whether issues are enabled.

Default: `true`

`has_projects` boolean

Whether projects are enabled.

Default: `true`

`has_wiki` boolean

Whether the wiki is enabled.

Default: `true`

`has_discussions` boolean

Whether discussions are enabled.

Default: `false`

`team_id` integer

The id of the team that will be granted access to this repository. This is only valid when creating a repository in an organization.

`auto_init` boolean

Whether the repository is initialized with a minimal README.

Default: `false`

`gitignore_template` string

The desired language or platform to apply to the .gitignore.

`license_template`   string

The license keyword of the open source license for this repository.

`allow_squash_merge`   boolean

Whether to allow squash merges for pull requests.

Default: `true`

`allow_merge_commit`   boolean

Whether to allow merge commits for pull requests.

Default: `true`

`allow_rebase_merge`   boolean

Whether to allow rebase merges for pull requests.

Default: `true`

`allow_auto_merge`   boolean

Whether to allow Auto-merge to be used on pull requests.

Default: `false`

`delete_branch_on_merge`   boolean

Whether to delete head branches when pull requests are merged

Default: `false`

`squash_merge_commit_title`   string

The default value for a squash merge commit title:
- `PR_TITLE` - default to the pull request's title.
- `COMMIT_OR_PR_TITLE` - default to the commit's title (if only one commit) or the pull request's title (when more than one commit).

Can be one of: `PR_TITLE` , `COMMIT_OR_PR_TITLE`

`squash_merge_commit_message`   string

The default value for a squash merge commit message:
- `PR_BODY` - default to the pull request's body.
- `COMMIT_MESSAGES` - default to the branch's commit messages.
- `BLANK` - default to a blank commit message.

Can be one of: `PR_BODY` , `COMMIT_MESSAGES` , `BLANK`

`merge_commit_title`   string

The default value for a merge commit title.
- `PR_TITLE` - default to the pull request's title.
- `MERGE_MESSAGE` - default to the classic title for a merge message (e.g., Merge pull request #123 from branch-name).

Can be one of: `PR_TITLE` , `MERGE_MESSAGE`

`merge_commit_message`   string

The default value for a merge commit message.
- `PR_TITLE` - default to the pull request's title.
- `PR_BODY` - default to the pull request's body.
- `BLANK` - default to a blank commit message.

Can be one of: `PR_BODY` , `PR_TITLE` , `BLANK`

`has downloads`   boolean

Whether downloads are enabled.

Default: `true`

`is_template` boolean

Whether this repository acts as a template that can be used to generate new repositories.

Default: `false`

HTTP response status codes for "Create a repository for the authenticated user"

Status code	Description
`201`	Created
`304`	Not modified
`400`	Bad Request
`401`	Requires authentication
`403`	Forbidden
`404`	Resource not found
`422`	Validation failed, or the endpoint has been spammed.

Code samples for "Create a repository for the authenticated user"

POST `/user/repos`

cURL   JavaScript   GitHub CLI

```
curl -L \ -X POST \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version:
2022-11-28" \ https://api.github.com/user/repos \ -d '{"name":"Hello-World","description":"This is your first
repo!","homepage":"https://github.com","private":false,"is_template":true}'
```

Response

Example response   Response schema

Status: 201

```
{ "id": 1296269, "node_id": "MDEwOlJlcG9zaXRvcnkxMjk2MjY5", "name": "Hello-World", "full_name": "octocat/Hello-World", "owner": {
"login": "octocat", "id": 1, "node_id": "MDQ6VXNlcjE=", "avatar_url": "https://github.com/images/error/octocat_happy.gif",
"gravatar_id": "", "url": "https://api.github.com/users/octocat", "html_url": "https://github.com/octocat", "followers_url":
"https://api.github.com/users/octocat/followers", "following_url": "https://api.github.com/users/octocat/following{/other_user}",
"gists_url": "https://api.github.com/users/octocat/gists{/gist_id}", "starred_url":
"https://api.github.com/users/octocat/starred{/owner}{/repo}", "subscriptions_url":
```

List repositories for a user 🔗

✅ Works with GitHub Apps

Lists public repositories for the specified user. Note: For GitHub AE, this endpoint will list internal repositories for the

Lists public repositories for the specified user. Note: For GitHub AE, this endpoint will list internal repositories for the specified user.

Parameters for "List repositories for a user"

Headers

`accept` string   ⎘

Setting to `application/vnd.github+json` is recommended.

Path parameters

`username` string   Required

The handle for the GitHub user account.

Query parameters

`type` string

Limit results to repositories of the specified type.

Default: `owner`
Can be one of: `all` , `owner` , `member`

`sort` string

The property to sort the results by.

Default: `full_name`
Can be one of: `created` , `updated` , `pushed` , `full_name`

`direction` string

The order to sort by. Default: `asc` when using `full_name` , otherwise `desc` .

Can be one of: `asc` , `desc`

`per_page` integer

The number of results per page (max 100).

Default: `30`

`page` integer

Page number of the results to fetch.

Default: `1`

HTTP response status codes for "List repositories for a user"

Status code	Description
`200`	OK

Code samples for "List repositories for a user"

GET `/users/{username}/repos`

cURL   JavaScript   GitHub CLI   ⎘

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28"
```

```
\ https://api.github.com/users/USERNAME/repos
```

Response

Example response   Response schema

Status: 200

```
[ { "id": 1296269, "node_id": "MDEwOlJlcG9zaXRvcnkxMjk2MjY5", "name": "Hello-World", "full_name": "octocat/Hello-World", "owner":
{ "login": "octocat", "id": 1, "node_id": "MDQ6VXNlcjE=", "avatar_url": "https://github.com/images/error/octocat_happy.gif",
"gravatar_id": "", "url": "https://api.github.com/users/octocat", "html_url": "https://github.com/octocat", "followers_url":
"https://api.github.com/users/octocat/followers", "following_url": "https://api.github.com/users/octocat/following{/other_user}",
"gists_url": "https://api.github.com/users/octocat/gists{/gist_id}", "starred_url":
"https://api.github.com/users/octocat/starred{/owner}{/repo}", "subscriptions_url":
```

Legal