About webhooks

In this article

Webhooks provide a way for notifications to be delivered to an external web server whenever certain events occur on GitHub.

About webhooks

Webhooks let you subscribe to events happening in a software system and automatically receive a delivery of data to your server whenever those events occur.

Webhooks are used to receive data as it happens, as opposed to polling an API (calling an API intermittently) to see if data is available. With webhooks, you only need to express interest in an event once, when you create the webhook.

Webhooks are used in a wide range of scenarios, including:

- Triggering CI (continuous integration) pipelines on an external CI server. For example, to trigger CI in Jenkins or CircleCI when code is pushed to a branch.
- Sending notifications about events on GitHub to collaboration platforms. For example, sending a notification to Discord or Slack when there's a review on a pull request.
- Updating an external issue tracker like Jira.
- Deploying to a production server.
- Logging events as they happen on GitHub, for audit purposes.

About webhooks on GitHub

When you create a webhook, you specify a URL and subscribe to events that occur on GitHub. When an event that your webhook is subscribed to occurs, GitHub will send an HTTP request with data about the event to the URL that you specified. If your server is set up to listen for webhook deliveries at that URL, it can take action when it receives one.

For example, you could subscribe your webhook to events that occur when code is pushed to a repository, a pull request is opened, a GitHub Pages site is built, or a new member is added to a team. You server could respond by deploying code to production, triggering a CI pipeline, sending a notification, or creating a GitHub project for the new team member.

You must create a webhook within a specific repository, organization, GitHub Enterprise, GitHub Marketplace account, GitHub Sponsors account, or GitHub App. The webhook can only access resources that are available in the repository, organization, GitHub Enterprise, GitHub Marketplace account, GitHub Sponsors account, or GitHub App where

it is installed. For more information, see "Types of webhooks."

For more information about creating webhooks, see "Creating webhooks." For more information about the types of events you can subscribe to, see "Webhook events and payloads." For more information about configuring your server to take an action in response to a payload delivery, see "Handling webhook deliveries."

> **Note**: GitHub webhooks do not currently support IPv6 but will in the future. The `/meta` REST API endpoint returns IPv6 ranges to enable that transition.

Choosing webhooks or the REST API

Using webhooks has the following advantages over using the API:

- Webhooks require less effort and less resources than polling an API.
- Webhooks scale better than API calls. If you need to monitor many resources, calling the API for each resource may cause you to hit your API rate limit quota quickly. Instead, you can subscribe to multiple webhook events and receive information only when an event happens.
- Webhooks allow near real-time updates, since webhooks are triggered when an event happens.

If you only need information once or intermittently, or only want to get information from a small set of resources with no plans to scale up, you can call the API when you need the relevant information.

For information on best practices to follow when using webhooks, see "Best practices for using webhooks."

> **Note:** GitHub Services (sometimes referred to as Service Hooks) was deprecated in 2019, in favor of integrating with webhooks. For more information about migrating your integration from using GitHub Services to using webhooks, see the blog post.

Further reading

- "Types of webhooks"