Using webhooks with GitHub Apps

In this article

Your GitHub App can subscribe to webhook events to receive notifications whenever certain activity occurs.

About webhooks and GitHub Apps

Webhooks enable your GitHub App to receive real-time notifications when events happen on GitHub, such as when someone pushes a commit or opens a pull request in a repository that your app can access. For more information about webhooks, see "About webhooks." For a tutorial that demonstrates how to use webhooks with a GitHub App, see "Building a GitHub App that responds to webhook events."

You can configure your GitHub App to receive webhooks for specific events on GitHub and automatically take action on them. For more information about the types of webhooks you can receive, see "Webhook events and payloads."

To receive webhook events in your GitHub App, you must enable webhooks for your GitHub App registration and specify a webhook URL where GitHub will send the webhook payloads.

If your GitHub App does not need to respond to webhooks or will only be used for authentication, you can turn off the webhook function for your GitHub App registration. You do not need to specify a webhook URL.

For more information about registering a GitHub App, see "Registering a GitHub App." For more information about changing the webhooks that a GitHub App registration subscribes to, see "Modifying a GitHub App registration."

Choosing a webhook URL

When you activate webhooks for your GitHub App registration, you will need to specify a webhook URL. The webhook URL is the address of a web server that will receive the webhook event payloads sent to your GitHub App. The server can then take action based on the content of the payload. You should choose a web server that's appropriate for the volume of webhook traffic that your GitHub App will encounter.

Choosing a webhook URL for development and testing

While you develop and test your app, you can use a webhook payload delivery service like Smee to capture and forward webhook payloads to your local development environment. Never use Smee for an application in production, because Smee channels are not authenticated or secure. Alternatively, you can use a tool like ngrok, localtunnel, or the Hookdeck Console that exposes your local machine to the internet to receive the payloads.

Creating a webhook URL with Smee

You can use Smee to create a unique domain where GitHub can send webhook payloads, without exposing your local development to the internet. Smee calls this unique domain a "Webhook Proxy URL." You can use Smee's Webhook Proxy URL as the webhook URL for your GitHub App.

1. To use Smee to create a unique domain, go to https://smee.io and click **Start a new channel**.

2. On the Smee channel page, follow the instructions under "Use the CLI" to install and run the Smee client.

3. To connect your Smee webhook URL to your GitHub App, enter your unique Smee domain in the "Webhook URL" field on your GitHub App registration page. For more information, see "Registering a GitHub App" and "Modifying a GitHub App registration."

Choosing a webhook URL for production

For an application in production that receives a low volume of webhook traffic, you can host it on any dynamic application server. The server-side code for handling the webhook can receive the event, deserialize its JSON payload, and decide what action to take, such as storing the data in a database or calling the GitHub API.

To handle a higher volume of webhook traffic for a large app in production, consider using asynchronous webhook handling on a dedicated server. You can achieve this by employing a queue, where the webhook handler pushes data to the queue, and separate processes perform subsequent actions based on the events. Additionally, you can use cloud functions such as Azure Functions, AWS Lambda, or Hookdeck to help scale the app for handling large volumes of webhook events.

Securing your webhooks with a webhook secret

Once you've configured your server to receive payloads, it will listen for any payload sent to the server. For security reasons, you should limit incoming requests to only those originating from GitHub. You can do that by creating a webhook secret for your app.

To create a webhook secret for your GitHub App, type a secret token under "Webhook secret" on your GitHub App registration page. You should choose a random string of text with high entropy. For more information, see "Registering a GitHub App" and "Modifying a GitHub App registration."

After creating a webhook secret for your app, you will need to configure your server to securely store and validate the webhook secret token. For more information, see "Validating webhook deliveries."

Subscribing to webhook events

You can subscribe your GitHub App to receive webhook payloads for specific events. The specific webhook events that you can select for your GitHub App registration are determined by the type of permissions you selected for your app. You will first need to select the permissions you would like your app to have, and then you can subscribe your app to webhook events that are related to that set of permissions. For more information, see "Choosing permissions for a GitHub App."

For example, if you would like your app to receive a webhook event payload whenever a new issue is opened in your repository, you would first need to give your app permission to access "Issues" under "Repository permissions." Then under "Subscribe to events" you can select "Issues."

For more information about the permissions that are required for each webhook event, see "Webhook events and payloads."