

# Making your GitHub App available for GitHub Enterprise Server

## In this article

About developing GitHub Apps for GitHub Enterprise Server

Each GitHub Enterprise Server instance must register their own GitHub App

The app code must be able to access the GitHub App credentials for the instance

The app code must use the correct URLs

The app code must be aware of feature differences

Each GitHub Enterprise Server instance can configure rate limits

In order for GitHub Enterprise Server instances to use your GitHub App, you must take some additional steps.

## About developing GitHub Apps for GitHub Enterprise Server

If you want your GitHub App to be available to organizations in a GitHub Enterprise Server instance that you are not part of, you must take the following steps.

If GitHub Enterprise Server access is important, consider whether a custom action for GitHub Actions will suit your needs instead. Public actions are available on GitHub Enterprise Server instances with GitHub Connect. For more information, see "[Enabling automatic access to GitHub.com actions using GitHub Connect](#)" in the GitHub Enterprise Server documentation.

## Each GitHub Enterprise Server instance must register their own GitHub App

Organizations owned by a GitHub Enterprise Server instance cannot install GitHub Apps registered on GitHub.com or on another GitHub Enterprise Server instance. Instead, they must register and install their own GitHub App for use on that instance.

- 1 The app developer creates a manifest or URL parameters. For more information, see "[Registering a GitHub App from a manifest](#)" and "[Registering a GitHub App using URL parameters](#)."
- 2 The app developer shares the manifest or URL parameters with the GitHub Enterprise Server administrator that wants to use the app. The same manifest or URL parameters can be shared with multiple GitHub Enterprise Server instances.
- 3 An organization administrator in the instance uses the manifest or URL parameters to register a GitHub App.
- 4 The organization installs the GitHub App that they registered.

Optionally, if the organization made the GitHub App public, other organizations

within the instance can install the GitHub App as well. There is not a way to install a GitHub App on an entire instance, only on organizations within an instance.

## The app code must be able to access the GitHub App credentials for the instance [↗](#)

---

You app's code will need the credentials of the GitHub App that the GitHub Enterprise Server instance registered. It will also need the hostname of the instance. You have two options: get the credentials and hostname from the instance, or have the GitHub Enterprise Server customer host and manage a self-hostable version of the app.

### Get the credentials from the GitHub Enterprise Server instance [↗](#)

The instance can share their GitHub App credentials and hostname with the app developer. The site administrator should only do this if they trust the app developer. Then, the app code can use the appropriate credentials depending on what actions it is taking. The app developer must take precautions to use the appropriate set of credentials and to not leak data.

Advantages:

- The app developer controls the infrastructure that the app runs on.
- The app developer has more control over app updates.
- The app developer may have more insight into app performance.

Disadvantages:

- The app developer must take precautions to avoid leaking data from the instance.
- The site administrator may need to open firewall exceptions for your application to reach the instance, and they may be reluctant to do so.

### Have the GitHub Enterprise Server customer host and manage a self-hostable version of the app [↗](#)

The app developer can provide a self-hostable version of their app. Then, the site administrator can host the app according to app developer's setup and installation instructions.

The method by which the self-hostable version of the app is created and shared is up to the app developer and depends on technology that the app uses.

Advantages:

- The instance remains more secure because they aren't sharing their app credentials.
- The app developer doesn't need to worry about leaking data from the instance.

Disadvantages:

- The app developer relies on the site administrator to provide infrastructure for the app and set things up correctly.
- Releasing updates to the app code may be more complex.
- The app developer may lose visibility about app performance.

## The app code must use the correct URLs [↗](#)

---

GitHub Enterprise Server uses different URLs than GitHub Free, GitHub Pro, GitHub

Team, and GitHub Enterprise Cloud. You should update your app code to use the appropriate URL depending on whether it is working with a GitHub Enterprise Server instance. Replace `HOSTNAME` with the hostname of the GitHub Enterprise Server instance.

GitHub Free GitHub Pro GitHub Team GitHub Enterprise Cloud	GitHub Enterprise Server
<code>https://api.github.com</code>	<code>https://HOSTNAME/api/v3</code>
<code>https://api.github.com/graphql</code>	<code>https://HOSTNAME/api/v3/graphql</code>
<code>https://github.com/login/oauth/authorize</code>	<code>https://HOSTNAME/login/oauth/authorize</code>
<code>https://github.com/login/oauth/access_token</code>	<code>https://HOSTNAME/login/oauth/access_token</code>

## The app code must be aware of feature differences

New REST API endpoints, GraphQL objects, and webhooks are released to GitHub Enterprise Server at a later date than GitHub Free, GitHub Pro, GitHub Team, and GitHub Enterprise Cloud. Additionally, there are multiple versions of GitHub Enterprise Server, and older versions may have different REST API endpoints, GraphQL objects, and webhooks.

Therefore, the app code needs to be aware of these differences. API responses and webhook payloads include a `x-github-enterprise-version` header for GitHub Enterprise Server payloads to help you determine what version you are handling.

## Each GitHub Enterprise Server instance can configure rate limits

Each GitHub Enterprise Server instance can configure its own rate limits. If your app is hitting a rate limit and is already taking precautions to stay under the rate limit, you should talk to the admin of the GitHub Enterprise Server instance.

### Legal