Generating an installation access token for a GitHub App

In this article

Learn how to generate an installation access token for your GitHub App.

About installation access tokens

In order to authenticate as an app installation, you must generate an installation access token. For more information about authenticating as an app installation, see "Authenticating as a GitHub App installation."

Note: Instead of generating an installation access token, you can use GitHub's Octokit SDKs to authenticate as an app. The SDK will take care of generating an installation access token for you and will regenerate the token once it expires. For more information about authenticating as an app installation, see "Authenticating as a GitHub App installation."

You should keep your installation access token secure. For more information, see "Best practices for creating a GitHub App."

Generating an installation access token

1. Generate a JSON web token (JWT) for your app. For more information, see "Generating a JSON Web Token (JWT) for a GitHub App".
2. Get the ID of the installation that you want to authenticate as.

   If you are responding to a webhook event, the webhook payload will include the installation ID.

   You can also use the REST API to find the ID for an installation of your app. For example, you can get an installation ID with the `GET /users/{username}/installation`, `GET /repos/{owner}/{repo}/installation`, `GET /orgs/{org}/installation`, or `GET /app/installations` endpoints. For more information, see "GitHub Apps".
3. Send a REST API `POST` request to `/app/installations/INSTALLATION_ID/access_tokens`. Include your JSON web token in the `Authorization` header of your request. Replace `INSTALLATION_ID` with the ID of the installation that you want to authenticate as.

   For example, send this curl request. Replace `INSTALLATION_ID` with the ID of the installation and `JWT` with your JSON web token:

```
curl --request POST \
--url
"https://api.github.com/app/installations/INSTALLATION_ID/access_tokens" \
--header "Accept: application/vnd.github+json" \
--header "Authorization: Bearer JWT" \
--header "X-GitHub-Api-Version: 2022-11-28"
```

Optionally, you can use the `repositories` or `repository_ids` body parameters to specify individual repositories that the installation access token can access. If you don't use `repositories` or `repository_ids` to grant access to specific repositories, the installation access token will have access to all repositories that the installation was granted access to. The installation access token cannot be granted access to repositories that the installation was not granted access to.

Optionally, use the `permissions` body parameter to specify the permissions that the installation access token should have. If `permissions` is not specified, the installation access token will have all of the permissions that were granted to the app. The installation access token cannot be granted permissions that the app was not granted.

The response will include an installation access token, the time that the token expires, the permissions that the token has, and the repositories that the token can access. The installation access token will expire after 1 hour.

For more information about this endpoint, see "GitHub Apps".

Note: In most cases, you can use `Authorization: Bearer` or `Authorization: token` to pass a token. However, if you are passing a JSON web token (JWT), you must use `Authorization: Bearer`.