Choosing permissions for a GitHub App

In this article

The permissions of a GitHub App determine what the app can do with GitHub's APIs and what webhooks the app can receive.

About GitHub App permissions

GitHub Apps don't have any permissions by default. When you register a GitHub App, you can select permissions for the app. The permissions that you select determine what the app can do with GitHub's APIs and what webhooks the app can subscribe to. You should select the minimum permissions required for the app.

Although GitHub Apps don't have any permissions by default, they do have implicit permissions to read public resources when acting on behalf of a user. When a user authorizes the app to act on their behalf, the GitHub App can use the resulting user access token to make requests to the REST API and the GraphQL API to read public resources. To learn more about acting on behalf of a user, see "Authenticating with a GitHub App on behalf of a user."

App permissions are classified as repository, organization, or account permissions. Repository permissions allow your app to access resources related to repositories that are owned by the account where the app is installed. Organization permissions allow your app to access resources related to the organization where the app is installed, if it is installed on an organization account. Account permissions allow your app to access resources related to a user if the user has also authorized your app. For more information about user authorization of apps, see "Authenticating with a GitHub App on behalf of a user."

When a user installs an app on their account or organization, they see and grant the repository and organization permissions that the app requested. They will also see a list of account permissions that the app can request for individual users. When a user authorizes an app to act on their behalf, they will see and grant the account permissions that the app requested.

The success of an API request with a user access token depends on the user's permissions as well as the app's permissions. For example, if the app was granted permission to write the contents of a repository, but the user can only read the contents, then the user access token can only read the contents. The success of an API request with an installation access token only depends on the app's permissions.

You can modify the permissions for your app at any time. When you modify the permissions, the owner of each account where the app was installed will be prompted to approve the new permissions. If the account owner does not approve the new

permissions, their installation will continue to use the old permissions.

Some webhooks and API access requires "Administration" permissions. If your app requires "Administration" permissions, consider explaining this requirement on your app's homepage. This will help users understand why your app needs a high level permission.

For more information about specifying permissions during GitHub App registration, see "Registering a GitHub App." For more information about modifying permissions, see "Modifying a GitHub App registration."

Choosing permissions for webhook access

The webhook documentation indicates whether each webhook is available to GitHub Apps. For each webhook that you want to subscribe to, refer to the webhook documentation to see what permissions a GitHub App needs to subscribe to that webhook. For more information, see "Webhook events and payloads."

For example, if you want your app to subscribe to `team` events, your app must have the "Members" organization permission.

On your GitHub App registration page, the available webhook events will change as you change your app's permissions. If you did not select sufficient permissions for your GitHub App to subscribe to an event, the event will not appear as an option on your app registration page.

Choosing permissions for REST API access

For more information about which REST API endpoints you can access with each permission, see "Permissions required for GitHub Apps." Some endpoints may require multiple permissions, and some endpoints may require one of multiple permissions. For more information, see the documentation for the endpoint.

For example, to use the `GET /orgs/{org}/dependabot/secrets` endpoint, your app must have at least read-level permission for the "organization dependabot secrets" permission.

If your app makes a REST API request with insufficient permissions, the API will return a `403` response.

Choosing permissions for GraphQL API access

For GraphQL requests, you should test your app to ensure that it has the required permissions for the GraphQL queries and mutations that you want to make.

If your app makes a GraphQL API query or mutation with insufficient permissions, the API will return a `401` response.

Choosing permissions for Git access

If you want your app to use an installation or user access token to authenticate for HTTP-based Git access, you should request the "Contents" repository permission. If your app specifically needs to access or edit Actions files in the `.github/workflows` directory, request the "Workflows" repository permission.

You can then use the access token as the HTTP password. Replace `TOKEN` with the access token:

```
git clone https://x-access-token:TOKEN@github.com/owner/repo.git
```