Deciding when to build a GitHub App

In this article

When building an integration, you should consider using a GitHub App in the following scenarios, instead of an OAuth app, personal access token, or GitHub Actions.

Using a GitHub App instead of an OAuth app

In general, GitHub Apps are preferred over OAuth apps.

Both OAuth apps and GitHub Apps use OAuth 2.0.

OAuth apps can only act on behalf of a user while GitHub Apps can either act on behalf of a user or independently of a user.

For more information, see "Differences between GitHub Apps and OAuth apps."

For information on how to migrate an existing OAuth app to a GitHub App, see "Migrating OAuth apps to GitHub Apps."

GitHub Apps offer enhanced security

GitHub Apps provide more control over what the app can do. Instead of the broad scopes that OAuth apps use, GitHub Apps use fine-grained permissions. For example, if your app needs to read the contents of a repository, an OAuth app would require the `repo` scope, which would also let the app edit the repository contents and settings. A GitHub App can request read-only access to repository contents, which will not let the app take more privileged actions like editing the repository contents or settings.

GitHub Apps also offer more control over repository access. With a GitHub App, the user or organization owner who installed the app can decide what repositories the app can access. Conversely, an OAuth app can access every repository that the user who authorized the app can access.

GitHub Apps use short lived tokens. If the token is leaked, the token will be valid for a shorter amount of time, which reduces the damage that can be done. Conversely, OAuth app tokens do not expire until the person who authorized the OAuth app revokes the token.

These security features help harden your GitHub App's security by limiting the damage that could be done if your app's credentials were leaked. Additionally, this lets organizations with stricter security policies use your app.

GitHub Apps can act independently of or on behalf of a user

GitHub Apps can act independently of a user. This is beneficial for automations that do not require user input.

Similar to OAuth apps, GitHub Apps can still take actions on behalf of a user. Unlike OAuth apps, which don't indicate that the action was performed by the app, GitHub Apps indicate that the action was performed by the app on behalf of the user.

GitHub Apps are not tied to a user account and do not consume a seat on GitHub Enterprise Server. GitHub Apps remain installed even when the person who initially installed the app leaves the organization. This lets your integrations continue to work even if people leave your team.

GitHub Apps have scalable rate limits

The rate limit for GitHub Apps using an installation access token scales with the number of repositories and number of organization users. Conversely, OAuth apps have lower rate limits and do not scale. For more information, see "Rate limits for GitHub Apps."

GitHub Apps have built in webhooks

GitHub Apps have built-in, centralized webhooks. GitHub Apps can receive webhook events for all repositories and organizations the app can access. Conversely, OAuth apps must configure webhooks individually for each repository and organization.

API access differs slightly

In general, GitHub Apps and OAuth apps can make the same API requests. However, there are some differences:

- The REST API to manage check runs and check suites is only available to GitHub Apps.
- Enterprise-level resources such as the enterprise object itself are not available to GitHub Apps. This means that GitHub Apps cannot call endpoints like `GET /enterprise/settings/license`. However, enterprise-owned organization and repository resources are available.
- Some requests may return incomplete data depending on the permissions and repository access that was granted to an GitHub App. For example, if your app makes a request to get all repositories that a user can access, the response will only include the repositories that the app was also granted access to.

For more information about the REST API endpoints that are available to GitHub Apps, see "Endpoints available for GitHub App installation access tokens."

Choosing between a GitHub App or a personal access token

If you want to access GitHub resources on behalf of a user or in an organization, or you anticipate a long-lived integration, we recommend building a GitHub App.

You can use personal access tokens for API testing or short-lived scripts. Since a personal access token is associated with a user, your automation could break if the user no longer has access to the resources you need. A GitHub App installed in an organization is not dependent on a user. Additionally, unlike a user, a GitHub App does not consume a GitHub seat.

GitHub supports two types of personal access tokens, but recommends that you use fine-grained personal access tokens instead of personal access tokens (classic) whenever possible. For more information about personal access tokens, see "Managing your personal access tokens."

Choosing between a GitHub App or GitHub Actions

GitHub Apps and GitHub Actions both provide ways to build automation and workflow tools.

GitHub Actions provide automation that can perform jobs like continuous integration, deployment tasks, and project management in a repository. They run directly on GitHub-hosted runner machines or self-hosted runners that your administrator sets up. GitHub Actions do not run persistently. GitHub Actions workflows run in response to events that occur in their repository, and only have access to the resources of the repository that they are set up for. However, custom actions can be shared across repositories and organizations, allowing developers to reuse and modify existing actions to meet their needs. GitHub Actions also come with built-in secret management, which you can use to securely interact with third-party services and manage deploy keys safely.

GitHub Apps run persistently on a server or compute infrastructure that you provide or run on a user device. They can react to GitHub webhook events as well as events from outside the GitHub ecosystem. They are a good option for operations that span multiple repositories or organizations, or for providing hosted services to other organizations. A GitHub App is the best choice when building a tool with functions that occur primarily outside of GitHub or require more execution time or permissions than what a GitHub Actions workflow is allotted.

For more information about comparing GitHub Actions to GitHub Apps, see "About custom actions."

You can use a GitHub App to authenticate in a GitHub Actions workflow if the built in `GITHUB_TOKEN` does not have sufficient permissions. For more information, see "Making authenticated API requests with a GitHub App in a GitHub Actions workflow."