

Registering a GitHub App using URL parameters

In this article

- About URL parameters for registering GitHub Apps
- Creating a custom configuration URL with query parameters
- GitHub App configuration parameters
- GitHub App permissions
- GitHub App webhook events

You can use URL query parameters to help other people quickly set up a new GitHub App with a specific configuration you have preselected.

About URL parameters for registering GitHub Apps

You can use URL parameters to preselect the configuration settings of a new GitHub App registration and share a custom link with other people. The link will take people to a GitHub App registration page, where the app settings will be pre-filled according to the URL parameters you included in the URL.

This approach is useful for integrators who want customers to set up an app on their personal account or organization with certain specifications, or for customers using GitHub Enterprise Server who aren't able to install apps from the GitHub Marketplace.

Alternatively, you can create a GitHub App manifest. For more information, see "[Registering a GitHub App from a manifest](#)."

Creating a custom configuration URL with query parameters

To create a custom configuration URL for a GitHub App on a personal or organization account, add query parameters after the following base URLs.

- To register an app on a personal account, add URL parameters to:
`https://github.com/settings/apps/new`
- To register an app on an organization account, add URL parameters to:
`https://github.com/organizations/ORGANIZATION/settings/apps/new`. Replace `ORGANIZATION` with the name of the organization where you'd like the customer to register the app.

On the app registration page, the person registering the app can edit the preselected values before submitting the app. If you do not include parameters for required values (like `name`) in the URL query string, the person registering the app will need to input a value before they can register the app.

For example, the following URL registers a new public app named `octocat-github-app` on a personal account. Using query parameters, the URL preconfigures a description and a

callback URL. It also selects read and write permissions for `checks` , activates webhooks using the `webhook_active` parameter, subscribes to the `check_run` and `check_suite` webhook events, and selects the option to request user authorization (OAuth) during installation:

```
https://github.com/settings/apps/new?name=octocat-github-app&description=An%20octocat%20App&callback_urls[]=https://example.com&request_oauth_on_install=true
```

GitHub App configuration parameters

You can use the following query parameters to select a specific configuration for the GitHub App registration. For example, to name the app "octocat-github-app", your query string would include `name=octocat-github-app` .

| Parameter name | Type | Description |
|---------------------------------------|-------------------------------|---|
| <code>name</code> | <code>string</code> | The name of the GitHub App. Give your app a clear and succinct name. Your app cannot have the same name as an existing GitHub user, unless it is your own user or organization name. A slugged version of your app's name will be shown in the user interface when your integration takes an action. |
| <code>description</code> | <code>string</code> | A description of the GitHub App. |
| <code>url</code> | <code>string</code> | The full URL of your GitHub App's website homepage. |
| <code>callback_urls</code> | <code>array of strings</code> | A full URL to redirect to after someone authorizes an installation. You can provide up to 10 callback URLs. These URLs are used if your app needs to generate a user access token. For example, <code>callback_urls[]=https://example.com&callback_urls[]=https://example-2.com</code> . For more information, see " About the user authorization callback URL ." |
| <code>request_oauth_on_install</code> | <code>boolean</code> | If your app authorizes users using the OAuth flow, you can set this option to <code>true</code> to allow people to authorize the app when they install it, saving a step. If you select this option, the <code>setup_url</code> becomes unavailable and users will be redirected to your <code>callback_url</code> after installing the app. |
| <code>setup_url</code> | <code>string</code> | The full URL to redirect to after |

someone installs the GitHub App if the app requires additional setup after installation. For more information, see "[About the setup URL](#)."

| | | |
|-------------------|------------------|---|
| setup_on_update | boolean | Set to <code>true</code> to redirect people to the setup URL when installations have been updated, for example, after repositories are added or removed. |
| public | boolean | Set to <code>true</code> when your GitHub App is available to the public or <code>false</code> when it is only accessible to the owner of the app. |
| webhook_active | boolean | Set to <code>true</code> to enable webhook. Webhook is disabled by default. |
| webhook_url | string | The full URL that you would like to send webhook event payloads to. |
| events | array of strings | Webhook events. Some webhook events require <code>read</code> or <code>write</code> permissions for a resource before you can select the event when registering a new GitHub App. For more information, see the " GitHub App webhook events " section. You can select multiple events in a query string. For example, <code>events[]=public&events[]=label</code> . |
| single_file_name | string | This is a narrowly-scoped permission that allows the app to access a single file in any repository. When you set the <code>single_file</code> permission to <code>read</code> or <code>write</code> , this field provides the path to the single file your GitHub App will manage. If you need to manage multiple files, see <code>single_file_paths</code> below. |
| single_file_paths | array of strings | This allows the app to access up to ten specified files in a repository. When you set the <code>single_file</code> permission to <code>read</code> or <code>write</code> , this array can store the paths for up to ten files that your GitHub App will manage. These files all receive the same permission set by <code>single_file</code> , and do not have separate individual permissions. When two or more |

permissions. When two or more files are configured, the API returns

`multiple_single_files=true`, otherwise it returns `multiple_single_files=false`.

GitHub App permissions

You can use query parameters to select the permissions for the GitHub App registration. For more information about the possible permissions you can select for the app, and the endpoints that the app can access with each permission, see "[Permissions required for GitHub Apps](#)." For the URL query parameter, use the permission name as the query parameter name, and set the query value to one of the possible values for that permission set.

For example, to select "Read & write" permissions in the user interface for `contents`, your query string would include `contents=write`. To select "Read-only" permissions in the user interface for `blocking`, your query string would include `blocking=read`. To select "No access" in the user interface for `checks`, your query string would not include the `checks` permission.

For more information about permissions and GitHub Apps, see "[Choosing permissions for a GitHub App](#)."

GitHub App webhook events

You can use query parameters to enable the GitHub App webhook, designate a webhook URL, and subscribe the app to receive webhook payloads for specific events.

To enable the GitHub App webhook, use `webhook_active=true` in your query string. To designate a full URL that you would like to send webhook event payloads to, use `webhook_url` in your query string. To subscribe the app to specific webhook payload events, use `events[]` as the query parameter name, and set the query value to the name of the webhook event. For more information about the possible webhook events and the GitHub App permissions required to subscribe to each event, see "[Webhook events and payloads](#)."

For example, to subscribe a GitHub App to receive webhook payloads for activity relating to commit comments, the query string would include

`&webhook_active=true&webhook_url=https://example.com&events[]=commit_comment`. Note that the `commit_comment` webhook event requires the GitHub App to have at least read-level access for the "Contents" repository permission. So your query string should also include a parameter to set the `contents` permission to `read` or `write`. For more information, see "[GitHub app permissions](#)."

You cannot use query parameters to set the value of a webhook secret. If an app requires a secret to secure its webhook, the value of the secret must be set in the GitHub UI by the person registering the app.

For more information about webhooks and GitHub Apps, see "[Using webhooks with GitHub Apps](#)."

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)