Refreshing user access tokens

In this article

To enforce regular token rotation and reduce the impact of a compromised token, you can configure your GitHub App to use user access tokens that expire.

About user access tokens that expire

Note: User access tokens that expire are currently an optional feature and are subject to change. For more information, see "Expiring user-to-server access tokens for GitHub Apps."

To enforce regular token rotation and reduce the impact of a compromised token, you can configure your GitHub App to use user access tokens that expire. If your app uses user access tokens that expire, then you will receive a refresh token when you generate a user access token. The user access token expires after eight hours, and the refresh token expires after six months. For more information, see "Generating a user access token for a GitHub App."

You can use the refresh token to generate a new user access token and a new refresh token. Once you use a refresh token, that refresh token and the old user access token will no longer work.

If your refresh token expires before you use it, you can regenerate a user access token and refresh token by sending users through the web application flow or device flow. For more information, see "Generating a user access token for a GitHub App."

Configuring your app to use user access tokens that expire

When you create your app, expiration of user access tokens is enabled unless you opt out. For more information, see "Registering a GitHub App." You can also configure this setting after your app has been created.

1. Navigate to your account settings.
   - For a GitHub App owned by a personal account, in the upper-right corner of any page, click your profile photo, then click **Settings**.
   - For a GitHub App owned by an organization, in the upper-right corner of any

page, click your profile photo, then click **Your organizations**. Then, to the right of the organization, click **Settings**.

2. In the left sidebar, click **Developer settings**.
3. In the left sidebar, click **GitHub Apps**.
4. Next to the GitHub App that you want to modify, click **Edit**.
5. In the GitHub Apps settings sidebar, click **Optional Features**.
6. Next to "User-to-server token expiration", click **Opt-in** or **Opt-out**. This setting may take a couple of seconds to apply.

   GitHub recommends that you opt in to this feature for improved security.

If you opt into user access tokens that expire after you have already generated user access tokens, the previously generated user access tokens will not expire. You can delete these tokens by using the `DELETE /applications/CLIENT_ID/token` endpoint. For more information, see "OAuth Authorizations."

Refreshing a user access token with a refresh token

1. Make a `POST` request to this URL, along with the following query parameters: `http(s)://HOSTNAME/login/oauth/access_token`

Query parameter	Type	Description
`client_id`	`string`	**Required.** The client ID for your GitHub App. The client ID is different from the app ID. You can find the client ID on the settings page for your app.
`client_secret`	`string`	**Required.** The client secret for your GitHub App. You can generate a client secret on the settings page for your app.
`grant_type`	`string`	**Required.** The value must be "refresh_token".
`refresh_token`	`string`	**Required.** The refresh token that you received when you generated a user access token.

2. GitHub will give a response that includes the following parameters:

Response parameter	Type	Description
`access_token`	`string`	The user access token. The token starts with `ghu_`.
`expires_in`	`integer`	The number of seconds until `access_token` expires. If you disabled expiration of user access tokens, this

		parameter will be omitted. The value will always be `28800` (8 hours).
`refresh_token`	`string`	The refresh token. If you disabled expiration of user access tokens, this parameter will be omitted. The token starts with `ghr_`.
`refresh_token_expires_in`	`integer`	The number of seconds until `refresh_token` expires. If you disabled expiration of user access tokens, this parameter will be omitted. The value will always be `15811200` (6 months).
`scope`	`string`	The scopes that the token has. This value will always be an empty string. Unlike a traditional OAuth token, the user access token is limited to the permissions that both your app and the user have.
`token_type`	`string`	The type of token. The value will always be `bearer`.