

The REST API is now versioned. For more information, see "[About API versioning](#)."

OAuth Authorizations

Use the REST API to interact with OAuth apps and OAuth authorizations of GitHub Apps

About OAuth apps and OAuth authorizations of GitHub Apps

You can use these endpoints to manage the OAuth tokens that OAuth apps or GitHub Apps use to access people's accounts on your GitHub Enterprise Server instance.

Tokens for OAuth apps have the prefix `gho_`, while OAuth tokens for GitHub Apps, used for authenticating on behalf of the user, have the prefix `ghu_`. You can use the following endpoints for both types of OAuth tokens.

Delete an app authorization

OAuth and GitHub application owners can revoke a grant for their application and a specific user. You must use [Basic Authentication](#) when accessing this endpoint, using the OAuth application's `client_id` and `client_secret` as the username and password. You must also provide a valid OAuth `access_token` as an input parameter and the grant for the token's owner will be deleted. Deleting an application's grant will also delete all OAuth tokens associated with the application for the user. Once deleted, the application will have no access to the user's account and will no longer be listed on [the application authorizations settings screen within GitHub](#).

Parameters for "Delete an app authorization"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`client_id` string **Required**

The client ID of the GitHub app.

Body parameters

`access_token` string **Required**

The OAuth access token used to authenticate to the GitHub API.

HTTP response status codes for "Delete an app authorization"

Status code	Description
-------------	-------------

204	No Content
422	Validation failed, or the endpoint has been spammed.

Code samples for "Delete an app authorization"

DELETE

/applications/{client_id}/grant

cURL

JavaScript

GitHub CLI

```
curl -L \ -X DELETE \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/applications/Iv1.8a61f9b3a7aba766/grant \ -d '{"access_token":"e72e16c7e42f292c6912e7710c838347ae178b4a"}'
```

Response

Status: 204

Check a token

OAuth applications and GitHub applications with OAuth authorizations can use this API method for checking OAuth token validity without exceeding the normal rate limits for failed login attempts. Authentication works differently with this particular endpoint. You must use [Basic Authentication](#) to use this endpoint, where the username is the application `client_id` and the password is its `client_secret` . Invalid tokens will return `404 NOT FOUND` .

Parameters for "Check a token"

Headers

accept string
Setting to `application/vnd.github+json` is recommended.

Path parameters

client_id string **Required**
The client ID of the GitHub app.

Body parameters

access_token string **Required**
The access_token of the OAuth or GitHub application.

HTTP response status codes for "Check a token"

Status code	Description
-------------	-------------

200	OK
404	Resource not found
422	Validation failed, or the endpoint has been spammed.

Code samples for "Check a token"

POST

/applications/{client_id}/token

cURL

JavaScript

GitHub CLI

```
curl -L \ -X POST \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/applications/Iv1.8a61f9b3a7aba766/token \ -d '{"access_token":"e72e16c7e42f292c6912e7710c838347ae178b4a"}'
```

Response

Example response

Response schema

Status: 200

```
{ "id": 1, "url": "https://HOSTNAME/authorizations/1", "scopes": [ "public_repo", "user" ], "token": "ghu_16C7e42F292c6912E7710c838347Ae178B4a", "token_last_eight": "Ae178B4a", "hashed_token": "25f94a2a5c7fbaf499c665bc73d67c1c87e496da8985131633ee0a95819db2e8", "app": { "url": "http://my-github-app.com", "name": "my github app", "client_id": "Iv1.8a61f9b3a7aba766" }, "note": "optional note", "note_url": "http://optional/note/url", "updated_at": "2011-09-06T20:39:23Z", "created_at": "2011-09-06T17:26:27Z", "fingerprint": "jklmnop12345678", "expires_at": "2011-09-08T17:26:27Z", "user": { "login": "octocat", "id": 1, "node_id": "MDQ6VXNlcjE=", "avatar_url":
```

Reset a token

OAuth applications and GitHub applications with OAuth authorizations can use this API method to reset a valid OAuth token without end-user involvement. Applications must save the "token" property in the response because changes take effect immediately. You must use [Basic Authentication](#) when accessing this endpoint, using the application's `client_id` and `client_secret` as the username and password. Invalid tokens will return `404 NOT FOUND` .

Parameters for "Reset a token"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`client_id` string **Required**

The client ID of the GitHub app.

Body parameters

`access token` string **Required**

The `access_token` of the OAuth or GitHub application.

HTTP response status codes for "Reset a token"

Status code	Description
200	OK
422	Validation failed, or the endpoint has been spammed.

Code samples for "Reset a token"

PATCH

`/applications/{client_id}/token`

cURL

JavaScript

GitHub CLI



```
curl -L \ -X PATCH \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/applications/Iv1.8a61f9b3a7aba766/token \ -d '{ "access_token": "e72e16c7e42f292c6912e7710c838347ae178b4a" }'
```

Response

Example response

Response schema

Status: 200

```
{ "id": 1, "url": "https://HOSTNAME/authorizations/1", "scopes": [ "public_repo", "user" ], "token": "ghu_16C7e42F292c6912E7710c838347Ae178B4a", "token_last_eight": "Ae178B4a", "hashed_token": "25f94a2a5c7fbaf499c665bc73d67c1c87e496da8985131633ee0a95819db2e8", "app": { "url": "http://my-github-app.com", "name": "my github app", "client_id": "Iv1.8a61f9b3a7aba766" }, "note": "optional note", "note_url": "http://optional/note/url", "updated_at": "2011-09-06T20:39:23Z", "created_at": "2011-09-06T17:26:27Z", "fingerprint": "jklmnop12345678", "expires_at": "2011-09-08T17:26:27Z", "user": { "login": "octocat", "id": 1, "node_id": "MDQ6VXNlcjE=", "avatar_url":
```

Delete an app token

OAuth or GitHub application owners can revoke a single token for an OAuth application or a GitHub application with an OAuth authorization. You must use [Basic Authentication](#) when accessing this endpoint, using the application's `client_id` and `client_secret` as the username and password.

Parameters for "Delete an app token"

Headers

accept

string

Setting to `application/vnd.github+json` is recommended.

Path parameters

client_id

string

Required

The client ID of the GitHub app.

The client ID of the GitHub app.

Body parameters

`access_token` string **Required**

The OAuth access token used to authenticate to the GitHub API.

HTTP response status codes for "Delete an app token"

Status code	Description
204	No Content
422	Validation failed, or the endpoint has been spammed.

Code samples for "Delete an app token"

DELETE

/applications/{client_id}/token

cURL

JavaScript

GitHub CLI



```
curl -L \ -X DELETE \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/applications/Iv1.8a61f9b3a7aba766/token \ -d '{"access_token":"e72e16c7e42f292c6912e7710c838347ae178b4a"}'
```

Response

Status: 204

Legal