

Troubleshooting

In this article

- 400 error for an unsupported API version
- 404 error for an existing repository
- Not all results returned
- Basic authentication errors
- Timeouts
- Insufficient permissions errors

Learn how to resolve the most common problems people encounter in the REST API.

If you're encountering some oddities in the API, here's a list of resolutions to some of the problems you may be experiencing.

400 error for an unsupported API version

You should use the `X-GitHub-Api-Version` header to specify an API version. For example:

```
curl --header "X-GitHub-Api-Version:2022-11-28" https://api.github.com/zen
```

If you specify a version that does not exist, you will receive a `400` error.

For more information, see "[API Versions](#)."

404 error for an existing repository

Typically, we send a `404` error when your client isn't properly authenticated. You might expect to see a `403 Forbidden` in these cases. However, since we don't want to provide *any* information about private repositories, the API returns a `404` error instead.

To troubleshoot, ensure [you're authenticating correctly](#), [your OAuth access token has the required scopes](#), [third-party application restrictions](#) are not blocking access, and that [the token has not expired or been revoked](#).

Not all results returned

Most API calls accessing a list of resources (*e.g.*, users, issues, *etc.*) support pagination. If you're making requests and receiving an incomplete set of results, you're probably only seeing the first page. You'll need to request the remaining pages in order to get more results.

It's important to *not* try and guess the format of the pagination URL. Not every API call uses the same structure. Instead, extract the pagination information from the link header, which is returned with every request. For more information about pagination, see "[Using pagination in the REST API](#)."

Basic authentication errors

On November 13, 2020 username and password authentication to the REST API and the OAuth Authorizations API were deprecated and no longer work.

Using `username/password` for basic authentication

If you're using `username` and `password` for API calls, then they are no longer able to authenticate. For example:

```
curl -u YOUR-USERNAME:YOUR-PASSWORD https://api.github.com/user/repos
```

Instead, use a personal access token or an access token for a GitHub App when testing endpoints or doing local development:

```
curl -H 'Authorization: Bearer YOUR-TOKEN' https://api.github.com/user/repos
```

For more information, see "[Managing your personal access tokens](#)" and "[About authentication with a GitHub App](#)."

For OAuth apps, you should use the [web application flow](#) to generate an OAuth token to use in the API call's header:

```
curl -H 'Authorization: Bearer YOUR-OAUTH-TOKEN' https://api.github.com/user/repos
```

Timeouts

If GitHub Enterprise Cloud takes more than 10 seconds to process an API request, GitHub Enterprise Cloud will terminate the request and you will receive a timeout response.

Insufficient permissions errors

If you are using a GitHub App or fine-grained personal access token and you receive an error due to your token having insufficient permissions, you can use the `X-Accepted-GitHub-Permissions` header to identify the permissions that are required to access the REST API endpoint.

The value of the `X-Accepted-GitHub-Permissions` header is a comma separated list of the permissions that are required to use the endpoint. Occasionally, you can choose from multiple permission sets. In these cases, multiple comma separated lists will be separated by a semicolon.

For example:

- `X-Accepted-GitHub-Permissions: contents=read` means that your GitHub App or fine-grained personal access token needs read access to the contents permission.
- `X-Accepted-GitHub-Permissions: pull_requests=write,contents=read` means that your GitHub App or fine-grained personal access token needs write access to the pull request permission and read access to the contents permission.
- `X-Accepted-GitHub-Permissions: pull_requests=read,contents=read; issues=read,contents=read` means that your GitHub App or fine-grained personal access token needs either read access to the pull request permission and read access to the contents permission, or read access to the issues permission and read access to the contents permission.

Legal