Troubleshooting cloning errors

In this article

If you're having trouble cloning a repository, check these common errors.

HTTPS cloning errors

There are a few common errors when using HTTPS with Git. These errors usually indicate you have an old version of Git, or you don't have access to the repository.

Here's an example of an HTTPS error you might receive:

```
> error: The requested URL returned error: 401 while accessing
> https://github.com/USER/REPO.git/info/refs?service=git-receive-pack
> fatal: HTTP request failed
```

```
> Error: The requested URL returned error: 403 while accessing
> https://github.com/USER/REPO.git/info/refs
> fatal: HTTP request failed
```

```
> Error: https://github.com/USER/REPO.git/info/refs not found: did you run git
> update-server-info on the server?
```

Check your Git version

There's no minimum Git version necessary to interact with GitHub Enterprise Cloud, but we've found version 1.7.10 to be a comfortable stable version that's available on many platforms. You can always download the latest version on the Git website.

Ensure the remote is correct

The repository you're trying to fetch must exist on GitHub.com, and the URL is case-sensitive.

You can find the URL of the local repository by opening the command line and typing `git remote -v`:

```
$ git remote -v
# View existing remotes
> origin  https://github.com/ghost/reactivecocoa.git (fetch)
> origin  https://github.com/ghost/reactivecocoa.git (push)

$ git remote set-url origin https://github.com/ghost/ReactiveCocoa.git
```

```
# Change the 'origin' remote's URL

$ git remote -v
# Verify new remote URL
> origin  https://github.com/ghost/ReactiveCocoa.git (fetch)
> origin  https://github.com/ghost/ReactiveCocoa.git (push)
```

Alternatively, you can change the URL through our GitHub Desktop application.

Provide an access token

To access GitHub, you must authenticate with a personal access token instead of your password. For more information, see "Managing your personal access tokens."

If you are accessing an organization that uses SAML SSO and you are using a personal access token (classic), you must also authorize your personal access token to access the organization before you authenticate. For more information, see "About authentication with SAML single sign-on" and "Authorizing a personal access token for use with SAML single sign-on."

Check your permissions

When prompted for a username and password, make sure you use an account that has access to the repository.

> **Tip**: If you don't want to enter your credentials every time you interact with the remote repository, you can turn on credential caching. If you are already using credential caching, please make sure that your computer has the correct credentials cached. Incorrect or out of date credentials will cause authentication to fail.

Use SSH instead

If you've previously set up SSH keys, you can use the SSH clone URL instead of HTTPS. For more information, see "About remote repositories."

Error: Repository not found

If you see this error when cloning a repository, it means that the repository does not exist or you do not have permission to access it. There are a few solutions to this error, depending on the cause.

Check your spelling

Typos happen, and repository names are case-sensitive. If you try to clone `git@github.com:user/repo.git`, but the repository is really named `User/Repo` you will receive this error.

To avoid this error, when cloning, always copy and paste the clone URL from the repository's page. For more information, see "Cloning a repository."

To update the remote on an existing repository, see "Managing remote repositories".

Checking your permissions

If you are trying to clone a private repository but do not have permission to view the repository, you will receive this error.

Make sure that you have access to the repository in one of these ways:

- The owner of the repository
- A collaborator on the repository
- A member of a team that has access to the repository (if the repository belongs to an organization)

Check your SSH access

In rare circumstances, you may not have the proper SSH access to a repository.

You should ensure that the SSH key you are using is attached to your personal account on GitHub Enterprise Cloud. You can check this by typing the following into the command line:

```
$ ssh -T git@github.com
> Hi USERNAME! You've successfully authenticated, but GitHub does not
> provide shell access.
```

If the repository belongs to an organization and you're using an SSH key generated by an OAuth app, OAuth app access may have been restricted by an organization owner. For more information, see "About OAuth app access restrictions."

For more information, see Adding a new SSH key to your GitHub account.

Check that the repository really exists

If all else fails, make sure that the repository really exists on GitHub.com! If you're trying to push to a repository that doesn't exist, you'll get this error.

Error: Remote HEAD refers to nonexistent ref, unable to checkout

This error occurs if the default branch of a repository has been deleted on GitHub.com.

Detecting this error is simple; Git will warn you when you try to clone the repository:

```
$ git clone https://github.com/USER/REPO.git
# Clone a repo
> Cloning into 'repo'...
> remote: Counting objects: 66179, done.
> remote: Compressing objects: 100% (15587/15587), done.
> remote: Total 66179 (delta 46985), reused 65596 (delta 46402)
> Receiving objects: 100% (66179/66179), 51.66 MiB | 667 KiB/s, done.
> Resolving deltas: 100% (46985/46985), done.
> warning: remote HEAD refers to nonexistent ref, unable to checkout.
```

To fix the error, you'll need to be an administrator of the repository on GitHub.com. You'll want to change the default branch of the repository.

After that, you can get a list of all the available branches from the command line:

```
$ git branch -a
# Lists ALL the branches
>   remotes/origin/awesome
>   remotes/origin/more-work
>   remotes/origin/new-main
```

Then, you can just switch to your new branch:

```
$ git checkout new-main
# Create and checkout a tracking branch
> Branch new-main set up to track remote branch new-main from origin.
> Switched to a new branch 'new-main'
```