

Splitting a subfolder out into a new repository

You can turn a folder within a Git repository into a brand new repository.

Mac Windows Linux

If you create a new clone of the repository, you won't lose any of your Git history or changes when you split a folder into a separate repository. However, note that the new repository won't have the branches and tags of the original repository.

- 1 Open TerminalTerminalGit Bash.
- 2 Change the current working directory to the location where you want to create your new repository.
- 3 Clone the repository that contains the subfolder.

```
git clone https://HOSTNAME/USERNAME/REPOSITORY-NAME
```

- 4 Change the current working directory to your cloned repository.

```
cd REPOSITORY-NAME
```


- 5 To filter out the subfolder from the rest of the files in the repository, install [git-filter-repo](#), then run `git filter-repo` with the following arguments.

- `FOLDER-NAME`: The folder within your project where you'd like to create a separate repository.



Tip: Windows users should use `/` to delimit folders.

```
$ git filter-repo --path FOLDER-NAME/
# Filter the specified branch in your directory and remove empty commits
> Rewrite 48dc599c80e20527ed902928085e7861e6b3cbe6 (89/89)
> Ref 'refs/heads/BRANCH-NAME' was rewritten
```

The repository should now only contain the files that were in your subfolder(s).

- 6 [Create a new repository](#) on GitHub Enterprise Server.
- 7 At the top of your new repository on your GitHub Enterprise Server instance's Quick Setup page, click  to copy the remote repository URL.

Quick setup — if you've done this kind of thing before

 Set up in Desktop or **HTTPS** SSH <https://github.com/octocat/quick-setup.git> 

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

Tip: For information on the difference between HTTPS and SSH URLs, see "[About remote repositories](#)."

- 8 Add a new remote name with the URL you copied for your repository. For example, `origin` or `upstream` are two common choices.

```
git remote add origin https://HOSTNAME/USERNAME/REPOSITORY-NAME.git
```

- 9 Verify that the remote URL was added with your new repository name.

```
$ git remote -v
# Verify new remote URL
> origin https://HOSTNAME/USERNAME/NEW-REPOSITORY-NAME.git (fetch)
> origin https://HOSTNAME/USERNAME/NEW-REPOSITORY-NAME.git (push)
```

- 10 Push your changes to the new repository on GitHub Enterprise Server.

```
git push -u origin BRANCH-NAME
```

Legal

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