

# Moving Files

Alan Robledo

May 27, 2019

Steps to push files from an old directory which we'll call "repo-A" to a new directory which we'll call "repo-B" while preserving the commit history.

Note: So far, this method will pull in the entire history of your "repo-A" into the new "repo-B". Not just the history of the files you transferred. Update will come soon to fix this issue if you only want to keep specific parts of the history.

From the command line, go to your home directory associated with git. Then type the following:

- (1) `$ git clone <repo-A url> repo-B`
- (2) `$ cd repo-B`
- (3) `$ git remote rm origin`
- (4) `$ git rm -rf .`
- (5) `$ git checkout HEAD -- <filename>`
- (6) `$ git commit -m "<commit message>"`

Note: You can repeat line (5) for every filename in repo-A that you want to move to repo-B. Now everything is set for your local git repository. To give this repo a home on github, do the following:

- (6) Go to github: <https://github.com>
- (7) Login to your account.
- (8) Go to your repositories and click the new repository button.
- (9) Make sure the new repository you make is the same as the one you made locally, i.e. "repo-B". You also don't need to initialize the directory with a README.

Going back to the command line, type:

- (10) `$ git remote add origin <repo-B url>`
- (11) `$ git push -u origin master`

Note:

For line (10), if you're cloning with HTTPS, your url would be: <https://github.com/username/repo-B>

If you're cloning with SSH, your url would be: `git@github.com:username/repo-B`

Do NOT add the ".git" at the end.