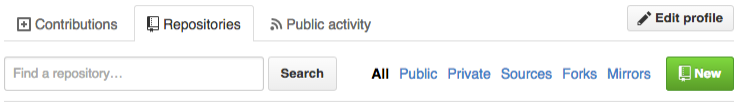
Github Guide

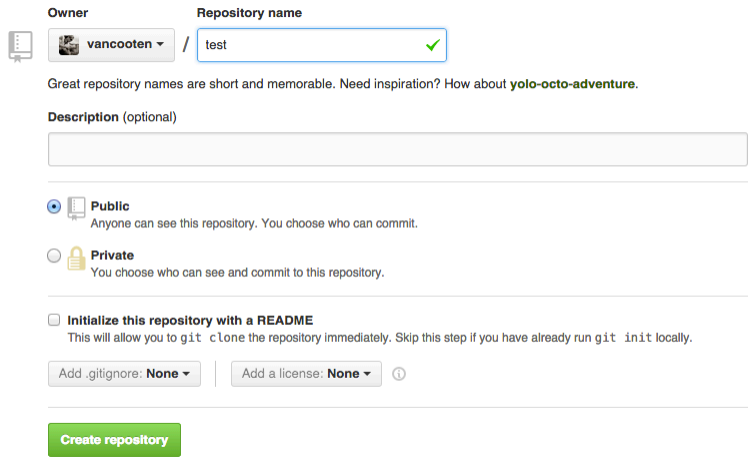
Adding a new repository to Github

This quick guide assumes you've signed up with Github installed the Github app on your computer and signed in with your Github username and password.

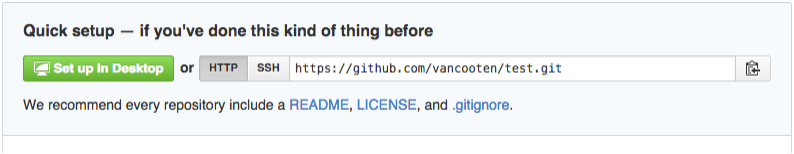
1. First go to Github and and create a new repo.



2. Name your repository.



3. Copy the url



4. On your computer create a folder / directory.

5. Add some file e.g. index.html

6. In Terminal change directory to the folder:

cd /users/username/test

7. To initialise a repository type:

git init

8a. To add all files to the repository:

git add .

8b. Or to add a file to a repository where your file name is called, index.html:

git add index.html

8c. To remove a file, if your file is called index.html you can type:

git rm index.html

9. Comment on the commit:

git commit -m "Added all files"

10. Add the remote origin to you repository where your remote repository url is, https://github.com/username/test.git:

git remote add origin https://github.com/username/test.git

11. Push changes to Github type:

git push origin master

Cloning a repository

Cloning a repository to your computer is useful if you're working from two computers.

1. On Github copy the url of the repository you want to clone.

2. In Terminal navigate to where you want the repository on your local computer.

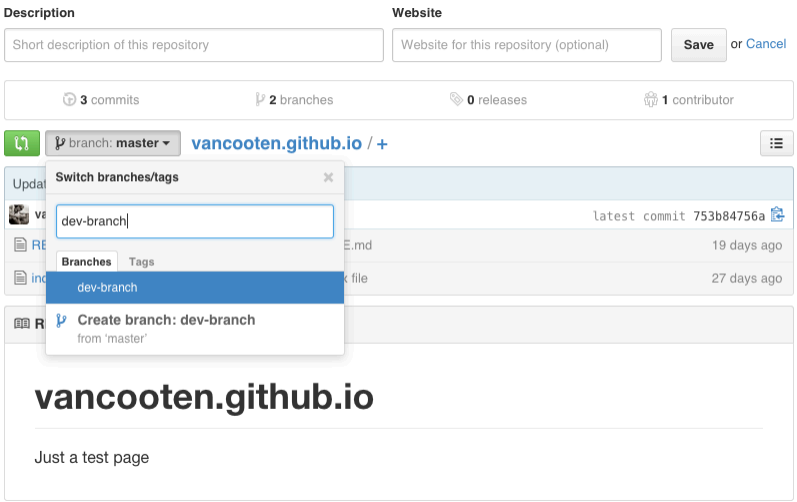
3. In Terminal type:

git clone https://github.com/username/repo-name.git

Working with branches

Assuming you've got a repository on your local computer that's mirrored on Github.

1. On Github create a new branch for that repository.



2. In Terminal navigate to the location of your repository.

3. In Terminal type:

git fetch origin

4. Now change branches, if the name of your new branch is 'dev-branch', type:

git checkout dev-branch

5. Make some changes to your files.

6. Now follow from the previous instructions 'Adding a new repository to Github' starting at step 8. When you are up to step 11, make sure to push the changes to the branch you created and not the master. If the name of your branch is 'dev-branch'Type:

git push origin dev-branch

7. To check which branch you're currently on (identified with an *), type:

git branch

Merging branches

1. Checkout the branch that you're merging into, if you're merging the changes into master checkout master, type:

git checkout master

2. If the development branch is called 'dev-branch' you can merge it, type:

git merge dev-branch

3. To push changes to the repository type:

git push origin master

References

An informative guide to using Github: http://git-scm.com

Trouble Shooting

Basic merge conflicts: http://git-scm.com/book/en/Git-Branching-Basic-Branching-and-Merging#Basic-Merge-Conflicts