

Converting a Subfolder of a Git Repository into a Separate Repository with Preserved History

Aleksandr B. Sahakyan

11 December 2015

While a git-tracked project grows, it is commonplace to separate project parts as individual projects, still controlled by git with additionally preserved prior history related to the content of the separated folder. Such separation is very useful if you, like me, keep a single closed git repository at GitHub with all the draft projects inside, once in a while opening the ones that you consider to be ready for a public release.

Below is the step-by-step guide on how to extract a subfolder into a separate distinct git project. For the examples below, the bigger project will be called PARENT, situated in a folder PARENT that is git-tracked and pushed into GitHub with the following address: <https://github.com/USERNAME/PARENT.git>. The subproject to be separated will be called OFFSPRING, situated in the subfolder PARENT/OFFSPRING, to be pushed into a new GitHub project <https://github.com/USERNAME/OFFSPRING.git>. Obviously, all the capitalised names in the lines below should be altered to suit your particular scenario.

Step 1. Clone the PARENT directory from GitHub into a location of your choice and go to that directory.

```
$ git clone https://github.com/USERNAME/PARENT.git
$ cd PARENT
```

Step 2. Separate the OFFSPRING subfolder into a new branch.

```
$ git subtree split --prefix=OFFSPRING/ --branch=OFFSPRING_br
```

Step 3. Create a directory for the new OFFSPRING repository with the same name, initialise git, and pull the separated branch from there.

```
$ cd ..
$ mkdir OFFSPRING
$ cd OFFSPRING
$ git init
$ git pull ../PARENT/ OFFSPRING_br
```

Step 4. Copy the .gitignore file from the PARENT repository and edit as needed.

```
$ cp ../PARENT/.gitignore ./gitignore
$ # vim .gitignore
```

Step 5. Create an empty remote GitHub repository named OFFSPRING via one of the means recommended by GitHub.

Step 6. From the newly created, git initialised and pulled OFFSPRING directory (not the one left inside PARENT directory), add the new remote location and push the files.

```
$ git remote add origin https://github.com/USERNAME/OFFSPRING.git
$ git push -u origin --all
$ git add --all
$ git commit -m "Gitignore update."
$ git push -u origin master
```

Step 7. Now, go back to the PARENT repository, delete the OFFSPRING_br branch, then delete the OFFSPRING folder and add/commit/push the changes. Please note, that the PARENT repository will still keep the history for the previous OFFSPRING subfolder.

```
$ cd ../PARENT
$ git show-branch --list # for verification/demonstration
$ git branch -D OFFSPRING_br
$ git show-branch --list # for verification/demonstration
$ rm -rf OFFSPRING
$ git add --all
$ git commit -m "Release-removal of a subproject."
$ git push -u origin master
```

Done!