The little Git-wiki

The little Git-wiki

Daniel Mejia R. (danielmejia55 at gmail.com)

Feb 2015

Table of contents

About this wiki

This wiki is build upon the hundreds of public knowledge forum post, blogs and web pages among the web Some of them are not result of my own research but I forgot to cite properly. I'm sorry for that. Most of the sources are from forums like stackoverflow and the amazing book *Pro Git* available online. A last resource can be found on this amazing site.

GIT BRANCHES

List branches

List remote and local branches

```
$ git branch -a
```

List local branches

```
$ git branch -l
```

Move between branches

```
$ git checkout local_branch_name
```

Create new branches

Create new branch

```
$ git branch branchname
```

Create new local branch that tracks remote existing branch (the same name of remote branch)

```
$ git checkout --track origin/remote_branch_name
```

Delete branches

Delete remote branch (not the local copy)

As of Git v1.7.0, you can delete a remote branch using

```
$ git push origin --delete <branchName>
```

Which is easier to remember than

```
$ git push origin :<branchName>
```

which was added in Git v1.5.0 “to delete a remote branch or a tag.”

Delete local branch (not the remote copy)

```
$ git brach -d name_of_branch_to_delete
```

This will not affect the remote branch

Renaming branches

Rename a local branch

```
$ git branch -m old_branch_name new_brach_name
```

Rename a remote branch

Before renaming a remote branch, we know we can simply rename a local branch by using following command:

```
$ git branch -m old_branch_name new_brach_name
```

But, for a remote branch on Github, we need to follow these 3 steps:

- Rename the branch in your local repository.
- Remove the branch name you want to change on GitHub.
- Push the new local branch name to GitHub.

For example:

Rename the branch in your local repository.

```
$ git branch -l
  master
 *branch01
$ git branch -m branch01 branch02
$ git branch
  master   /
 *branch02
```

Remove the branch name you want to change on GitHub.

```
$ git push origin :branch01
To git@github.com:your_user_name/your_project.git
- [deleted]          branch01
```

Push the new local branch name to GitHub.

```
$ git push origin branch02
Counting objects: 56, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (44/44), done.
Writing objects: 100% (44/44), 127.97 KiB, done.
Total 44 (delta 18), reused 1 (delta 0)
To git@github.com:your_user_name/your_project.git
 * [new branch]      branch02 -&gt; branch02
Done.
```

Working with branches

Reset local branch to look like a remote brach

Before reseting is worth to keep a copy of the branch you are reseting. Create a new branch from there by

```
$ git branch local_branch_name
```

Then tell git to fetch all the branches and information on remote

```
$ git fetch --all
```

Tell to the branch you are working (local_branch_name in this case) to track a remote branch

```
$ git reset --hard origin/remote_branch_name
```

Finally pull the changes

```
$ git pull
```

You end up with two local branches, the old one you discarded and the new you've just created which tracks the remote branch you choose.

GIT REMOTES

List Remotes

To list the remotes associated with the repository

```
$ git remote -v
```

Add a remote

You can have many "remotes" that store branches of the same repo (no matter if the name is the same) To add a remote:

```
$ git remote add remote_name remote_url(with .git)
```

For example:

```
$ git remote add dev user@server.domain.do:~/MyRepo.git
```

Adds a remote called `dev` on the server `server.domain.do` in the repository `MyRepo`.

when you call `remote -v` you would get the list of actual remotes