

# Git

## Tags

Action	Command
Create a tag (annotated)	<code>git tag -a -m 'MESSAGE' TAGNAME [COMMIT]</code>
List tags (and NUM lines of annotation)	<code>git tag -l -nNUM</code>
Delete tag	<code>git tag -d TAGNAME</code>

## References

- [Tag naming conventions](#) (Stack Overflow)

## Completely remove a Git submodule

Source: <http://davidwalsh.name/git-remove-submodule>

1. Delete section from `.gitmodules`: 3 lines, starting with  
`[submodule "MODULE"]`
2. Stage `.gitmodules`: `git add .gitmodules`
3. Delete section from `.git/config`: 2 lines, starting with  
`[submodule "MODULE"]`
4. `git rm --cached path/to/submodule`
5. `rm -rf .git/modules/MODULE`
6. `git commit -m 'Removed module MODULE'`
7. `rm -rf path/to/submodule` (removes now untracked submodule files)

## Extract a subproject with Git Subtree

You have a project `app` where one of the modules, `widget`, would better be kept in a separate Git repository. We'll assume `widget` now resides in a directory immediately under the project root `~/projects/app/widget/`. The `app` project is on Github, at `git@github.com:user/app.git`.

The goal is to have `widget` as a separate project `~/projects/widget/`, with its own Github repository `git@github.com:user/widget.git`. We'll assume this repository has been created and is still empty.

1. Go to project root:

```
$ cd ~/projects/app/
```

2. Create a branch that contains only commits for files in `widget`:

```
$ git subtree split --prefix=widget/ --branch=subtree/widget
```

3. Push this new branch to its own repository:

```
$ git push git@github.com:user/widget.git subtree/widget:master
```

4. Create widget project directory:

```
$ mkdir ~/projects/widget/; cd ~/projects/widget/
```

5. Pull the widget project from Github into its own directory:

```
$ git clone git@github.com:user/widget.git
```

6. Maybe do some more development, and release the first version:

```
$ git tag -a -m 'First stable release' v1.0.0  
$ git push --tags
```

At this time, you have two separate repositories. However, app/widget/ is still part of the original repository. However, now it should be maintained in its own repository, and changes should be pulled into app from widget:

1. Remove the widget directory from the app project:

```
$ cd ~/projects/app/  
$ git rm -rf widget/  
$ rm -rf widget/ # delete ignored files that were left behind
```

2. Fetch the widget release from Github (--squash "summarizes" all changes as single commit):

```
$ git subtree add --prefix=widget/ --squash github.com:user/widget.git tags/v1.0.0
```

## References

- <http://ariya.ofilabs.com/2014/07/extracting-parts-of-git-repository-and-keeping-the-history.html>
- <https://git.kernel.org/cgit/git/git.git/tree/contrib/subtree/git-subtree.txt>