

[devroom.io](#) - [blog](#) - [projects](#) - [about](#) - [gpg](#) - [contact](#)

# The git submodule cheat sheet

Posted: 2020-03-09

~400wrds (~2min)

Tagged [git](#)

A git submodule, in its essence, is a reference to another git repository. It's a great way to include vendor code (like plugins or themes) into your own code base. This post contains some examples on how to use git submodules effectively.

## Add a submodule

You need to know the remote git repository url and where you want to place that it in your repository.

```
git submodule add https://example.com/submodule-repo.git path/to/submodule
git add .
git commit -m "adds submodule path/to/submodule"
```

## Cloning a project with submodules

When you clone a repository that contains submodules there are a few extra steps to be taken.

```
git clone http://example.com/repo.git repo
cd repo
git submodule init
git submodule update
```

If you're sure you want to fetch all submodules (and their submodules), you can also use this fancy one-liner:

```
git clone --recurse-submodules http://example.com/repo.git
```

## Update your submodule

If you're simply tracking the `master` branch for the submodule, you can suffice with a simple `fetch` and `merge`.

```
cd path/to/submodule
git fetch
git merge origin/master
```

If you're in a hurry, you can streamline this for all submodules in your repo with:

```
git submodule update --remote --recursive
```

Don't forget to commit this change to your own repo, so others are locked to this new version of the submodule as well.

## Track a specific branch of version

The repo for your submodule may have a specific branch (e.g. **stable**) or tag you want to track, instead of **master**.

```
git config -f .gitmodules submodule.path/to/submodule.branch stable
git submodule update --remote
```

Again, don't forget to commit your changes to **.gitmodules** to send this change to other contributors to your repository.

## Remove a submodule

Removing a git submodule consists of two steps: removing the reference and removing the locally cached version.

```
git submodule deinit path/to/submodule
git rm path/to/submodule
git commit -m "removes submodule path/to/submodule"

rm -rf .git/modules/path/to/submodule
```

## Bonus: see submodule status in **git status**

You can configure git to show a submodule summary when you do a **git status**. There is a small performance trade-off here, but it might be useful to you.

```
git config status.submodulesummary 1
```

My site is free of ads and trackers. I record privacy-respecting usage statistics with [Fathom](#).

Was this post helpful to you? Why not ☕ [Buy me a coffee](#)

Copyright © 1999-2020 Ariejan de Vroom <[ariejan@devroom.io](mailto:ariejan@devroom.io)>

Live now; make *now* always the most precious time. *Now* will never come again.  
— Jean-Luc Picard