

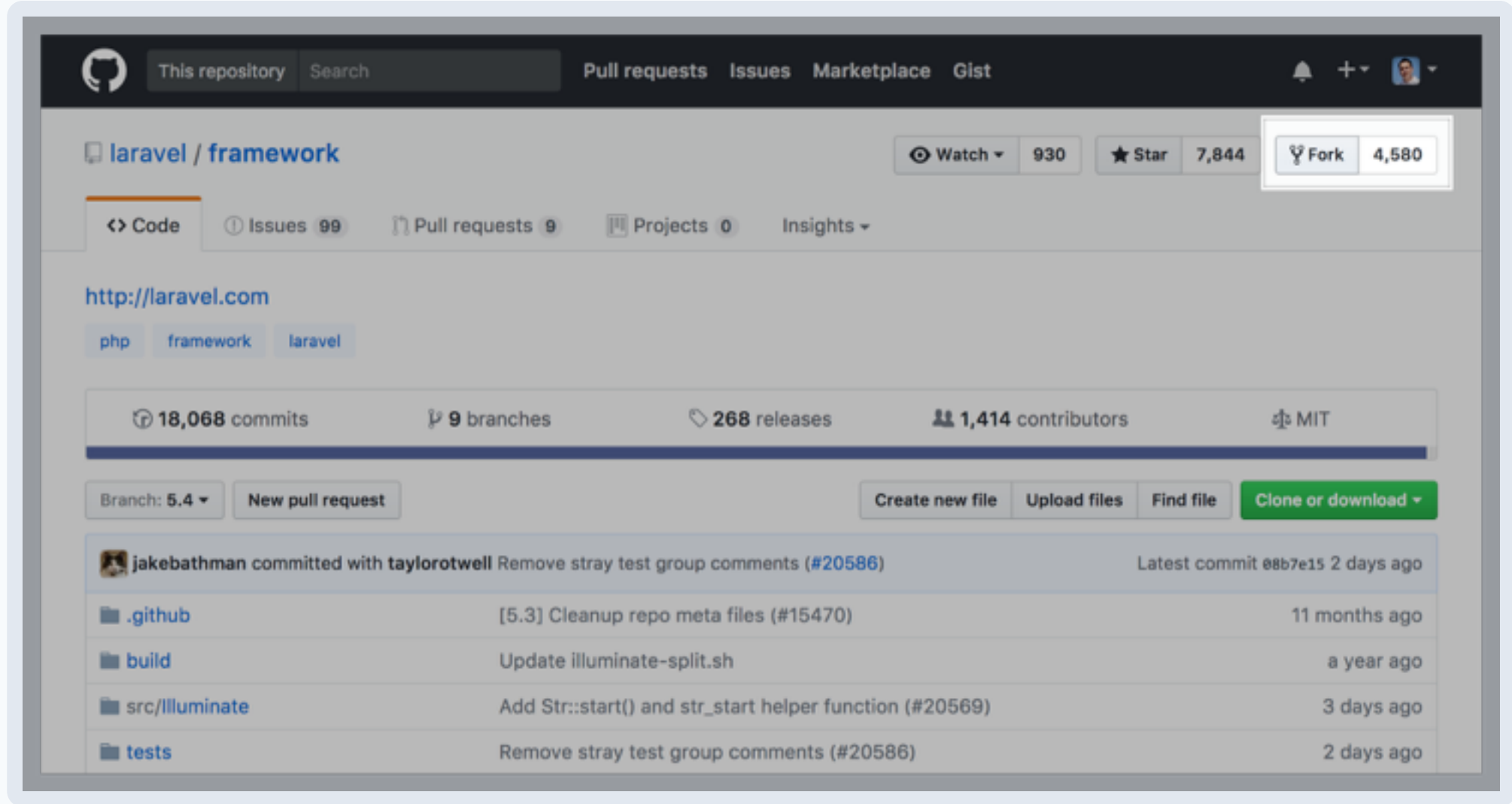
How to keep your Git-Fork up to date

Published on August 25th, 2017

When it comes to the situation that you fork a repository and you contribute to it, then it could happen that your fork and the upstream are not in sync anymore. So the goal is, that you get a current version of the upstream repository and then you can merge the new changes into your fork, right? Okay! Let's get started.

1. Create a fork

Before you can keep your fork updated, you need a fork. Wuhhh! For the ones who don't know where you can create a fork, see the screenshot below. This is GitHub but works also for any other git hosted platform, like Bitbucket or GitLab. Hint: A fork is a copy of someone others repository in your account, which can be an independent development project.



2. Clone the fork

Clone the fork as a regular repository, as you always do.

```
$ git clone git@github.com:stefanbauer/framework.git
```

3. Add the upstream

Now we should add the **upstream** . The original repository is mostly called "upstream". In our case, we use Laravel for instance. **cd** into your fork repository and add the upstream. You can call it however you want. Upstream is just best practice.

```
$ git remote add upstream git://github.com/laravel/framework.git
```

If you now have a look at your remote URLs, you should see the following:

```
$ framework (master) git rv

origin https://github.com/stefanbauer/framework (fetch)
origin https://github.com/stefanbauer/framework (push)
upstream https://github.com/laravel/framework (fetch)
upstream https://github.com/laravel/framewor (push)
```

By the way... I suggest you to take a look at my "These Git-Aliases make my work faster and easier" post, that the command above will work. It's worth, promised!

4. Keep the upstream updated

Now as we have both URLs get tracked, we can update the two sources independently. With

```
$ git fetch upstream
```

you can fetch all the stuff from the upstream.

5. Merge the upstream with your fork

Then you can just merge the changes.

```
$ git merge upstream/master master
```

With that, you merge the latest changes from the master branch of the upstream into your local master branch. If you like, you can also use **git pull** , which is nothing else than fetching and merging in one step.

Pro Tip: The best way in my eyes is, to rebase because that fetches the latest changes of the upstream branch and replay your work on top of that. Here is, how it works:

```
$ git rebase upstream/master
```

Bobby Bouwmann and I are writing a book called "Laravel Secrets".

You will learn everything about the undocumented secrets of the popular Laravel framework. You can sign up using the form below to get early updates.

Your email address

SUBSCRIBE

Visit laravelsecrets.com to get more information.