Rob Allen's DevNotes (/)

Developing software in the Real World

Home (/)	Hire me! (https://19ft.com)	Talks (/talks)

Slim (https://akrabat.com/category/slim-framework/)

rst2pdf (https://akrabat.com/category/rst2pdf/)

OpenWhisk (https://akrabat.com/category/openwhisk/)

Archives (https://akrabat.com/archives/)	About (https://akrabat.com/about/)

The beginner's guide to contributing to a GitHub project

This is a guide to contributing to an open source project that uses GitHub. It’s mostly based on how I’ve seen Zend Framework (http://framework.zend.com), Slim Framework (http://www.slimframework.com) and joind.in (https://joind.in) operate. However, this is a general guide so check your project’s README for specifics.

TL;DR

Skip to the summary.

Step 1: Set up a working copy on your computer

Firstly you need a local fork of the the project, so go ahead and press the “fork” button in GitHub. This will create a copy of the repository in your own GitHub account and you’ll see a note that it’s been forked underneath the project name:

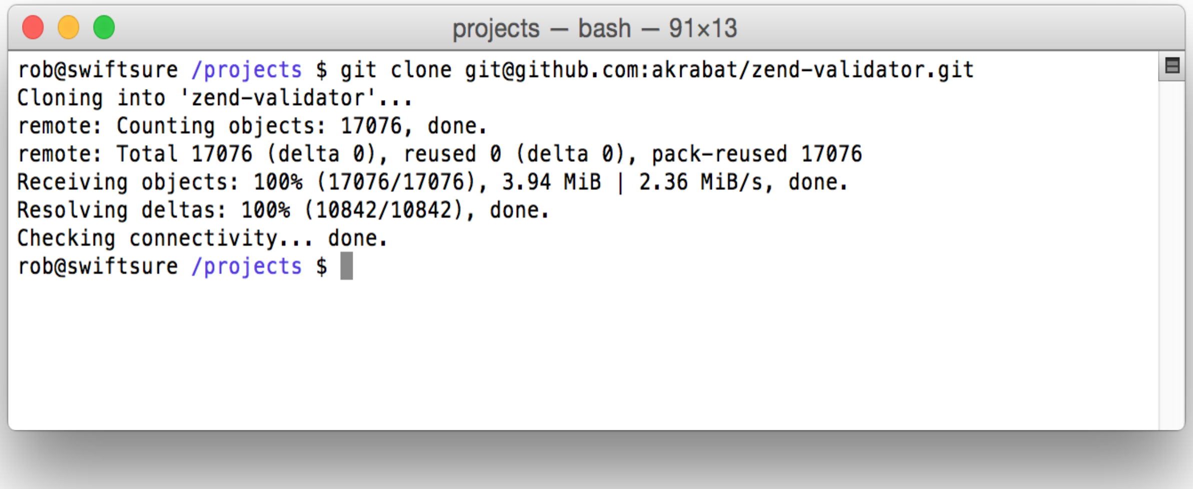
akrabat / zend-validator

forked from zendframework/zend-validator

Now you need a copy locally, so find the “SSH clone URL” in the right hand column and use that to clone locally using a terminal:

```
$ git clone git@github.com:akrabat/zend-validator.git
```

Which will do something like this:



```
rob@swiftsure /projects $ git clone git@github.com:akrabat/zend-validator.git
Cloning into 'zend-validator'...
remote: Counting objects: 17076, done.
remote: Total 17076 (delta 0), reused 0 (delta 0), pack-reused 17076
Receiving objects: 100% (17076/17076), 3.94 MiB | 2.36 MiB/s, done.
Resolving deltas: 100% (10842/10842), done.
Checking connectivity... done.
rob@swiftsure /projects $
```

Change into the new project's directory:

```
$ cd zend-validator
```

Finally, in this stage, you need to set up a new remote that points to the original project so that you can grab any changes and bring them into your local copy. Firstly clock on the link to the original repository – it's labeled "Forked from" at the top of the GitHub page. This takes you back to the projects main GitHub page, so you can find the "SSH clone URL" and use it to create the new remote, which we'll call *upstream*.

```
$ git remote add upstream git@github.com:zendframework/zend-validator.git
```

You now have two remotes for this project on disk:

1. *origin* which points to your GitHub fork of the project. You can read and write to this remote.
2. *upstream* which points to the main project's GitHub repository. You can only read from this remote.

Step 2: Get it working on your machine

Now that you have the source code, get it working on your computer. Hopefully there will be documentation how to do this in the `README` or `INSTALL` file.

If you get it working, but the documentation is unclear, then improving that bit should be your first PR to the project. This happens to me a lot and I've updated getting-started docs a lot as my first PR to a project!

Step 3: Do some work

This is the fun bit where you get to contribute to the project. It's usually best to start by fixing a bug that is either annoying you or you've found on the project's issue tracker. If you're looking for a place to start, a lot of projects use the "easy pick" label (http://seld.be/notes/encouragingcontributions-with-the-easy-pick-label) (or some variation) to indicate that this issue can be addressed by someone new to the project.

Now that you have picks an issue, reproduce it on your version. Once you have reproduced it, read the code to work out where the problem is. Once you've found the code problem, you can move on to fixing it.

Branch!

The number one rule is to put each piece of work on its own branch. If the project is using git-flow (http://nvie.com/posts/a-successful-git-branching-model/), then it will have both a *master* and a *develop* branch. The general rule is that if you are bug fixing, then branch from master and if you are adding a new feature then branch from develop. If the project only has a *master* branch, the branch from that. Some projects, like Slim use branches named after the version number (*2.x* and *3.x* in Slim's case). In this case, pick the branch that's relevant.

For this example, we'll assume we're fixing a bug in zend-validator, so we branch from *master*:

```
$ git checkout master
$ git pull upstream master && git push origin master
$ git checkout -b hotfix/readme-update
```

Firstly we ensure we're on the master branch. Then the `git pull` command will sync our local copy with the upstream project and the `git push` syncs it to our forked GitHub project. Finally we create our new branch. You can name your branch whatever you like, but it helps for it to be meaningful. Including the issue number is usually helpful. If the project uses git-flow as zend-validator does, then there are specific naming conventions where the branch is prefixed with "hotfix/" or "feature/".

Now you can do the work to fix the issue.

If the project has tests, run them to ensure you haven't broken anything. You may also add a new test to show that your change fixes the original problem.

Ensure that you only fix the thing you're working on. Do not be tempted to fix some other things that you see along the way, including formatting issues, as your PR will probably be rejected.

Make sure that you commit in logical blocks. Each commit message should be sane. Read Tim Pope's A Note About Git Commit Messages (http://tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html).

Step 4: Create the PR

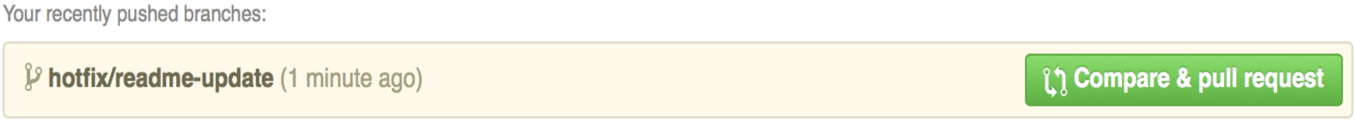
To create a PR you need to push your branch to the *origin* remote and then press some buttons on GitHub.

To push a new branch:

```
$ git push -u origin hotfix/readme-update
```

This will create the branch on your GitHub project. The -u flag links this branch with the remote one, so that in the future, you can simply type `git push origin`.

Swap back to the browser and navigate to your fork of the project (https://github.com/akrabat/zend-validator in my case) and you'll see that your new branch is listed at the top with a handy "Compare & pull request" button:

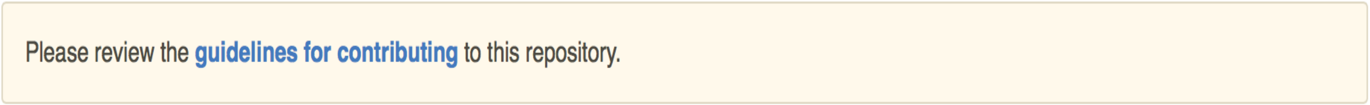


Your recently pushed branches:

hotfix/readme-update (1 minute ago)

Go ahead and press the button!

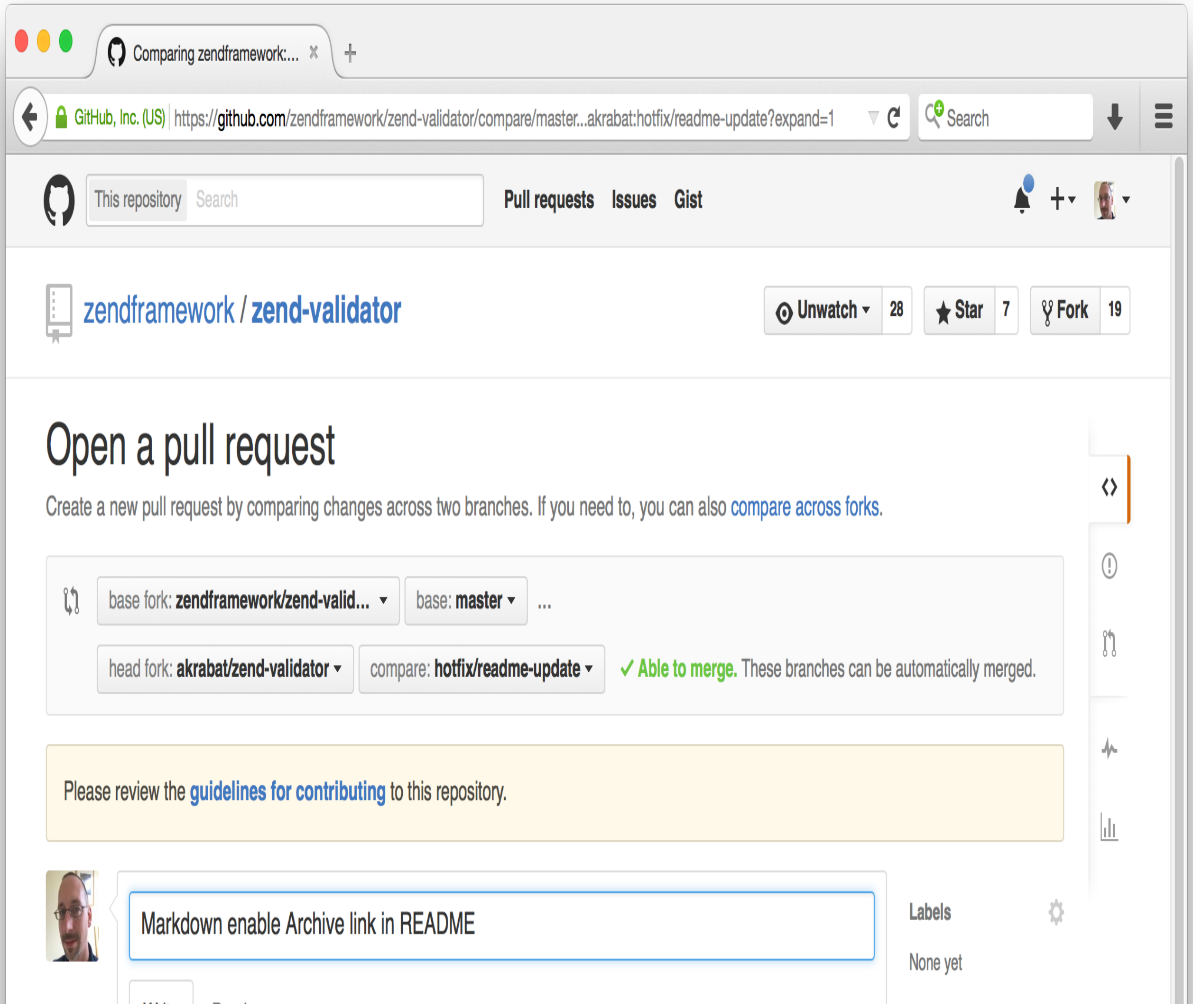
If you see a yellow box like this:



Please review the guidelines for contributing to this repository.

Click the link which will take you to the project's `CONTRIBUTING` file and read it! It contains valuable information on how to work with the project's code base and will help you get your contribution accepted.

On this page, ensure that the "base fork" points to the correct repository and branch. Then ensure that you provide a good, succinct title for your pull request and explain why you have created it in the description box. Add any relevant issue numbers if you have them.



Comparing zendframework:...
GitHub, Inc. (US) https://github.com/zendframework/zend-validator/compare/master...akrabat:hotfix/readme-update?expand=1
Search
This repository
Search
Pull requests
Issues
Gist
zendframework / zend-validator
Unwatch 28
Star 7
Fork 19

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base fork: zendframework/zend-valid... base: master ...

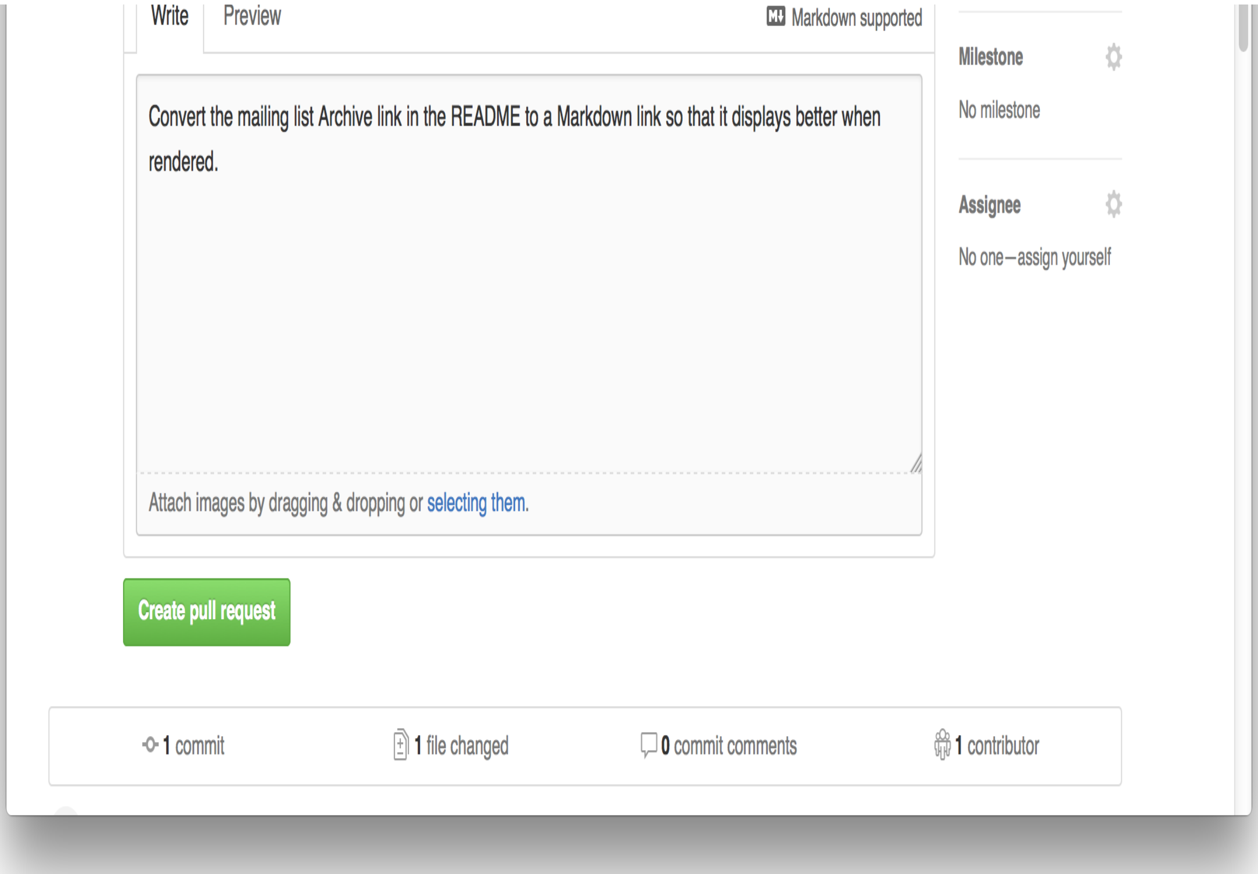
head fork: akrabat/zend-validator compare: hotfix/readme-update ✓ Able to merge. These branches can be automatically merged.

Please review the guidelines for contributing to this repository.

Markdown enable Archive link in README

Labels

None yet



Convert the mailing list Archive link in the README to a Markdown link so that it displays better when rendered.

If you scroll down a bit, you'll see a diff of your changes. **Double check that it contains what you expect.**

Once you are happy, press the "Create pull request" button and you're done.

Step 5: Review by the maintainers

For your work to be integrated into the project, the maintainers will review your work and either request changes or merge it.

Lorna Mitchell's article Code Reviews: Before You Even Run The Code (http://www.lornajane.net/posts/2015/code-reviews-before-you-even-run-the-code) covers the things that the maintainers will look for, so read it and ensure you've made their lives as easy as possible.

To sum up

That's all there is to it. The fundamentals are:

1. Fork the project & clone locally.
2. Create an *upstream* remote and sync your local copy before you branch.
3. Branch for each separate piece of work.
4. Do the work, write good commit messages (https://blogs.gnome.org/danni/2011/10/25/a-guide-to-writing-git-commit-messages/), and read the CONTRIBUTING file if there is one.
5. Push to your *origin* repository.

6. Create a new PR in GitHub.
7. Respond to any code review (http://www.lornajane.net/posts/2015/code-reviews-before-you-even-run-the-code) feedback.

If you want to contribute to an open source project, the best one to pick is one that you are using yourself. The maintainers will appreciate it!

This article was posted on 22 September 2015 in PHP (https://akrabat.com/category/php/)

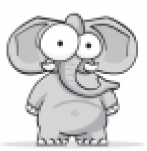
50 thoughts on “The beginner's guide to contributing to a GitHub project”

Jaime Olmo (http://www.jaimeolmo.com)

Excellent article. I have a question, how CI / CD would fit in this process? It would be before of the Review or as an activity once the review is completed?

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=32292#respond)

24 September 2015 at 3:05 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-32292)



Vlad

I don't know how this work on github. Actually, I think it is not github. It is some CI platform like travis.

It can be set UP on different ways:
1) build will trigger automatically on creating pull request. Pull request can be merged only if build was success
2) build is triggered by changes on your work branch
3) any other way…

So, the answer depends for each project, I think.

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=38245#respond)

20 May 2016 at 7:28 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-38245)

Felipe Martins

WOWW, nice nice nice! Many thanks, Mr. Rob!

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=32346#respond)

26 September 2015 at 1:59 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-32346)

Pingback: O guia para iniciantes sobre como contribuir com um projeto no GitHub | ROGERIO PRADO J | rogeriopradoj.com (http://rogeriopradoj.com/2015/09/28/o-guia-para-iniciantes-sobre-como-contribuir-com-um-projeto-no-github/)

Pingback: How do we hire ? | 42MATE Blog (https://blog.42mate.com/?p=139)

Pingback: Como contribuir com um projeto no GitHub - (http://imasters.com.br/desenvolvimento/como-contribuir-com-um-projeto-no-github/?

trace=1519021197&source)

Pingback: PHP Annotated Monthly – October 2015 | JetBrains PhpStorm Blog (http://blog.jetbrains.com/phpstorm/2015/10/php-annotated-monthly-october-2015/)

Sean Behan (http://www.boxitoff.com)

Thanks for this article it was very clear and easy to follow.
I've just my first Pull Request and I'm now dealing with the a Code Review.
Just a note for anyone else – Updating your pull request is as simple as making the changes and committing them, the Pull Request is automatically updated.
Thanks again

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=37145#respond)

15 March 2016 at 10:50 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-37145)

Noureddine LOUAHEDJ (http://noureddine.me/)

Thanks for your article, it is very clear and simple.
Best regards

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=37413#respond)

31 March 2016 at 11:20 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-37413)

Sirolad

This is an eye-opener

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=37731#respond)

24 April 2016 at 9:13 am (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-37731)

Ukor Jidechi E. (http://ukorjidechi.com)

Thank you very much, the article was very helpful and well explained.

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=37993#respond)

11 May 2016 at 9:16 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-37993)

Sebastjan

Just what I was looking for, thank you!

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=38317#respond)

24 May 2016 at 6:54 am (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-38317)

Vignesh (http://vigneshrajkumar.com)

Very explanatory! Thanks!

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=38701#respond)

19 June 2016 at 4:26 am (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-38701)

really enlightening article on git…thanks a million

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=45872#respond)

clever

14 May 2017 at 6:57 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-45872)

Great tutorial to get you started on contributing to open source. Recommend it !

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=38717#respond)

Usman Haider Raja

19 June 2016 at 8:52 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-38717)

Thanks Rob. If I'm submitting a PR for test coverage and badge, how do I add the author's credentials to, say coveralls to be able to generate the coverage and badge.

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=39415#respond)

Rowland

4 August 2016 at 1:30 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-39415)

very well written, thank you

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=39439#respond)

Hadi

9 August 2016 at 6:56 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-39439)

Amazing Article :)

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=40139#respond)

Shibasis Patel (http://shibasisp.github.io)

27 September 2016 at 7:14 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-40139)

Thanks. This article was useful today :)

Clear and simple.

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=41268#respond)

Kenji (http://yargies.com)

10 November 2016 at 8:36 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-41268)

Very helpful.

Thank you very much.

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=42265#respond)

Akash (http://appsparkler.com)

17 December 2016 at 2:54 am (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-42265)

I've started on a project to help beginners get started with contributing to open source.
https://github.com/Roshanjossey/first-contributions

Roshan Jossey (https://github.com/Roshanjossey/first-contributions)

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=42655#respond)

30 December 2016 at 1:35 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-42655)

You don't know how much you helped me with my first contribution to my team's project. Very grateful.

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=74566#respond)

ginjy23 29 August 2019 at 1:45 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-74566)

Perfect guide – thanks for sharing!

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=45415#respond)

James D 17 April 2017 at 5:04 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-45415)

Nice post. Thanks

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=45749#respond)

ali 3 May 2017 at 4:39 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-45749)

Excellenet article, thanks!

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=45786#respond)

hd84355 6 May 2017 at 9:56 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-45786)

nice article

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=45983#respond)

Arjun Sisodia 22 May 2017 at 5:43 am (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-45983)

This is awesome. I just realized now how easy it is to contribute to existing repository projects online. Thanks!

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=46090#respond)

Papoj Thamjaroenporn 30 May 2017 at 3:58 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-46090)

Good tutorial.

You wrote: "Swap back to the browser and navigate to your fork of the project". That is true, we also can see the same thing if we navigate to the original repository.

begueradj (http://www.begueradj.com)

Thank you
BEGUERADJ Billal

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=58015#respond)

9 December 2018 at 10:05 am (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-58015)

Nice post. If the upstream ssh address doesn't work then could use http address.

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=58250#respond)

dajiang zhong

25 December 2018 at 10:45 am (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-58250)

Thanks this was mostly simple and succinct.
Ultimately valuable.

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=62002#respond)

Ken Ingram

9 February 2019 at 11:57 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-62002)

Thank you for such an awesome and simple tutorial.

Best regards.

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=64469#respond)

Fabrizio

23 February 2019 at 8:10 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-64469)

Nice one

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=68551#respond)

Sulenchy

22 March 2019 at 6:47 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-68551)

Great stuff, this was very instructional and useful. Thanks!

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=68780#respond)

Rosario Gueli (http://roxcode.com)

24 March 2019 at 5:14 am (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-68780)

Hey Buddy, Thank You for this wonderful write up on git.

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=72023#respond)

Pankaj

13 April 2019 at 10:42 am (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-72023)

Great write up with easy to follow instructions, thanks.

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=72820#respond)

Peter Witham (https://peterwitham.com)

24 April 2019 at 8:58 am (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-72820)

Alex

Hello,
I am sorry, I feel stupid reading that it was clear for everybody here. Being a real dummy probably, I was not able to follow. When do you commit? After this, right ?
$ git checkout master
$ git pull upstream master && git push origin master
$ git checkout -b hotfix/readme-update

So why do you push?

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=73080#respond)

10 May 2019 at 8:12 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-73080)

Rob
(http://akrabat.com)

Alex,

After you have created the new branch with `git checkout -b hotfix/readme-update`, you then make changes and commit them.

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=73147#respond)

15 May 2019 at 8:28 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-73147)

Ales_blaze

he first pulled the changes from original project and then pushed the changes to the forked version of the project.

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=73198#respond)

20 May 2019 at 7:28 am (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-73198)

Ales_blaze

Thankyou so much for the guide , i was not being able to contribute on github repos , but after this guide i made my commit and successfully contributed to the open source project.

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=73197#respond)

20 May 2019 at 7:27 am (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-73197)

Himanshi

Please make sure you have the correct access rights
and the repository exists.
When I cloned the repository this message came up

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=73217#respond)

21 May 2019 at 11:37 am (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-73217)

paul
(http://paulayo.github.io)

I grateful for the article, you made me understand the workflow properly. Thank a lot

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=73219#respond)

21 May 2019 at 1:08 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-73219)

Carl Graff

This is great but all tutorials seem tps top after the PR is cpmplete – step 7 in the summary

At this point I believe you are left with both a local and origin remote brnach named "hotfix/readme-update"

So now I want to make more changes do i:
1. merge the local hotfix/readme-update with the local master and push to remote origin master
2. change to local master and create a new branch named for example hotfix/readme-update-more
3. delete older local and origin branch named hotfix/readme-update
4. use new branch hotfix/readme-update-more through to PR acceptance as you descibed above.
5. repeat steps 1 – 4 above for all new development
— this first set of steps assumes we want to get rid of branches that are completed

1. change to local master
2. pull from upstrem master in case some new changes have crept in
3. create new local branch for example hotfix/readme-update-more
4 . continue as you explained above only usethe new branch through to PR acceptance
5. repeat steps 1 – 4 above for all new development
— this second set of steps assume the old local and remote origin branchs are left as history

I think it matters to a certain degree what order these steps are done in though it may vary silighty.

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=73234#respond)

22 May 2019 at 7:53 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-73234)

Rob (http://akrabat.com)

Carl,

Good question! I should write a follow up for the next steps. Until then…

You don't merge your work to master as the project maintainers will do so. Also, you never re-use a branch for new work – create a new one.

Once your PR has been merged, you can delete your branch. Firstly update your master with the changes from the maintainers:

```
$ git checkout master
$ git pull upstream master
$ git push origin master
```

You can now delete your branch:

```
$ git branch -d hotfix/readme-update
$ git push origin --delete hotfix/readme-update
```

You now have the latest version of the code, so for your next change you can start again at step 3 with:

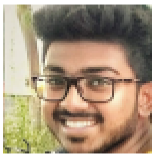
```
$ git checkout master
$ git pull upstream master && git push origin master
$ git checkout -b hotfix/my-next-fix
```

and away you go making the changes required.

Hopefully that makes sense!

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=73236#respond)

22 May 2019 at 8:37 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-73236)

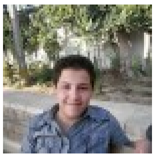


Very very useful.Thank you man.

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=73349#respond)

Sailesh Patra

3 June 2019 at 3:04 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-73349)



Greate Article, Thanks.

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=74081#respond)

Kamal Eddin

21 July 2019 at 11:51 am (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-74081)

That is a cool article, thank you for sharing :)

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=74108#respond)

Paulius

26 July 2019 at 6:14 am (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-74108)

A very nice step by step article. Thanks bro…

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=75643#respond)

shahzeb

18 September 2019 at 3:27 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-75643)

This helped. just wanted to say thanks!

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=76388#respond)

Mourph (http://nope)

24 October 2019 at 7:59 pm (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-76388)

Pingback: How to land freelance projects with no experience - TapChief Blog (https://www.tapchief.com/blog/land-freelance-projects-no-experience/)

Thank you very much for this Sir

Reply (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/?replytocom=80072#respond)

Aishat 13 May 2020 at 10:54 am (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/#comment-80072)

Thoughts? Leave a reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Post Comment

← On blocking ads (https://akrabat.com/on-blocking-ads/)

The beginner's guide to rebasing your PR → (https://akrabat.com/the-beginners-guide-to-rebasing-your-pr/)

Rob Allen (https://akrabat.com) is a sofware consultant and developer concentrating on HTTP APIs. He develops in PHP, Python and other interesting languages, contributing to rst2pdf (https://rst2pdf.org), Slim Framework (https://www.slimframework.org), Apache OpenWhisk (https://openwhisk.apache.org) amongst other OSS projects. Based in Worcester, UK, he runs Nineteen Feet (http://19ft.com/) and is the author of Zend Framework in Action (http://www.zendframeworkinaction.com).

Around the web:
Framing Focus blog (https://framingfocus.net)
Twitter: @akrabat (https://twitter.com/akrabat)
Instagram: akrabatic (https://www.instagram.com/akrabatic)
Flickr: photos (https://www.flickr.com/photos/akrabat)
GitHub: Rob Allen (https://github.com/akrabat)
IRC: akrabat on freenode (http://freenode.net)

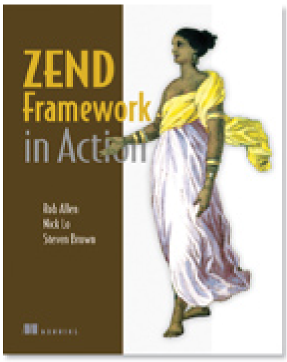
Search

Search

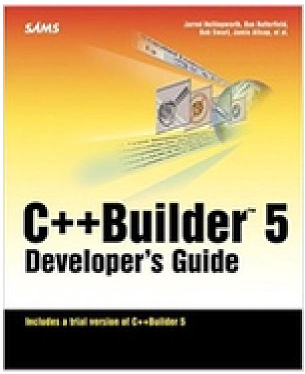
Other pages

The beginner's guide to contributing to a GitHub project (https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/)
The beginner's guide to rebasing your PR (https://akrabat.com/the-beginners-guide-to-rebasing-your-pr/)
Zend Framework Training (https://akrabat.com/zend-framework-training/)
ZF1 Tutorial (https://akrabat.com/zend-framework-tutorial/)
ZF2 Tutorial (https://akrabat.com/zend-framework-2-tutorial/)
A Kitura tutorial (https://akrabat.com/a-kitura-tutorial/)
Shorter Links WP plugin (https://akrabat.com/shorter-links/)
Akra's Diary (https://akrabat.com/akras-diary/)
License for code on this site (https://akrabat.com/license-for-code-published-on-this-site/)
Disclosure policy (https://akrabat.com/disclosure-policy/)
Privacy policy (https://akrabat.com/privacy-policy/)

Books I've written



(http://amzn.to/2wLBrQg)



(http://amzn.to/2vCrkbp)