

Pull Request Tutorial

A Visual Guide to Pull Requests

Pull Request Tutorial

What is a Pull Request?

From Github's [Using Pull Requests Page](#)

Pull requests let you tell others about changes you've pushed to a GitHub repository. Once a pull request is sent, interested parties can review the set of changes, discuss potential modifications, and even push follow-up commits if necessary.

Pull Requests are commonly used by teams and organizations collaborating using the [Shared Repository Model](#), where everyone shares a single repository and topic branches are used to develop features and isolate changes. Many open source projects on Github use pull requests to manage changes from contributors as they are useful in providing a way to notify project maintainers about changes one has made and in initiating code review and general discussion about a set of changes before being merged into the main branch.

Here's [an example pull request](#) from jQuery's github repo.

Creating a Pull Request

There are 2 main work flows when dealing with pull requests:

1. Pull Request from a [forked repository](#)
2. Pull Request from a branch within a repository

Here we are going to focus on 2.

Creating a Topical Branch

First, we will need to create a branch from the latest commit on master. Make sure your repository is up to date first using

```
git pull origin master
```

Note: `git pull` does a `git fetch` followed by a `git merge` to update the local repo with the remote repo. For a more detailed explanation, see [this stackoverflow post](#).

To create a branch, use `git checkout -b <new-branch-name> [<base-branch-name>]`, where `base-branch-name` is optional and defaults to `master`. I'm going to create a new branch called `pull-request-demo` from the `master` branch and push it to github.

```
git checkout -b pull-request-demo
git push origin pull-request-demo
```

Creating a Pull Request

To create a pull request, you must have changes committed to the your new branch.

Go to the repository page on github. And click on "Pull Request" button in the repo header.



Pick the branch you wish to have merged using the "Head branch" dropdown. You should leave the rest of the fields as is, unless you are working from a remote branch. In that case, just make sure that the base repo and base branch are set correctly.

branch?

head repo: yangsu/pull-request-tutorial

head branch: master

Choose a Head Branch

Filter branches/tags

✓ master

pull-request-demo

Enter a title and description for your pull request. Remember you can use [Github Flavored Markdown](#) in the description and comments

New Pull Request

Commits 2

Pull request demo

Write Preview

Leave a comment

Finally, click on the green "Send pull request" button to finish creating the pull request.

Send pull request

You should now see an open pull request.

Open

yangsu wants to merge 2 commits into master from pull-request-demo

8

#2

- Discussion
- Commits 2
- Files Changed 1



yangsu opened this pull request 2 minutes ago

Edit

Pull request demo

No one is assigned

No milestone

This is a demo of how to do a pull request!

Open

+ 8 additions

- 0 deletions

1 participant



yangsu added some commits 20 minutes ago

yangsu	started the creating a pull request section	4782b63
yangsu	Added first step in creating a pull request	f356b3c

You can add more commits to this pull request by pushing to the pull-request-demo branch on yangsu/pull-request-tutorial

This pull request can be automatically merged.

Merge pull request

Using a Pull Request

You can write comments related to a pull request,



Write Preview

You can write comments about this change here.

view all the commits by all contained by a pull request under the commits tab,

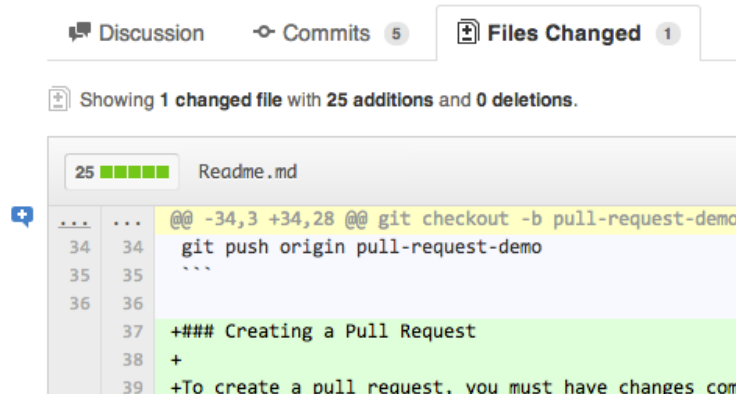
- Discussion
- Commits 2
- Files Changed 1

Showing 2 unique commits by 1 author.

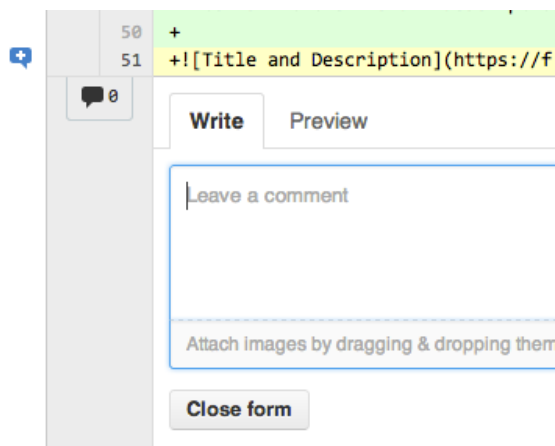
Mar 29, 2013

yangsu	started the creating a pull request section
yangsu	Added first step in creating a pull request

or see all the file changes from the pull request across all the commits under the "Files Changed" tab.



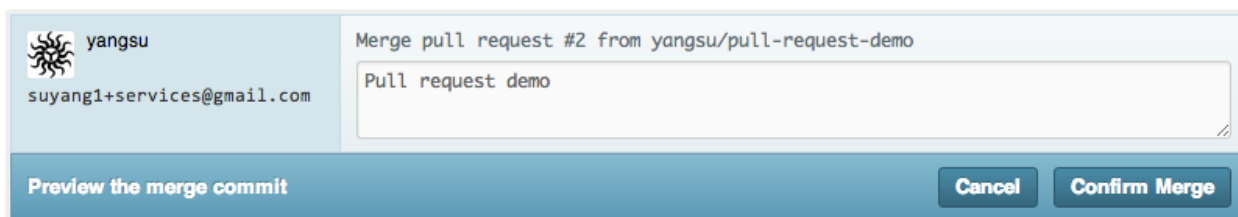
You can even leave a comment on particular lines in the code change simply by hovering to the left of a line and clicking on the blue note icon.



Merging a Pull Request

Once you and your collaborators are happy with the changes, you start to merge the changes back to master. There are a few ways to do this.

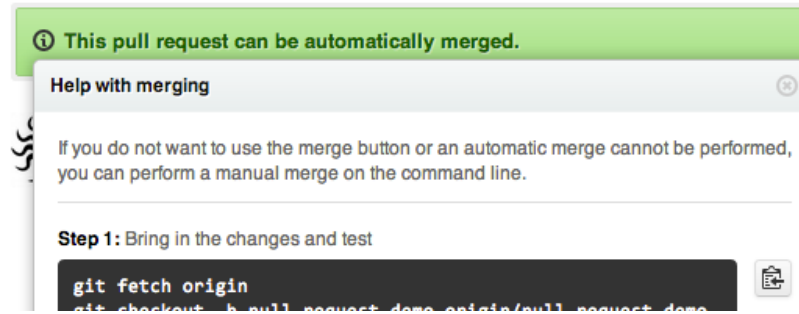
First, you can use github's "Merge pull request" button at the bottom of your pull request to merge your changes. This is only available when github can detect that there will be no merge conflicts with the base branch. If all goes well, you just have to add a commit message and click on "Confirm Merge" to merge the changes.



Merging Locally

If the pull request cannot be merged online due to merge conflicts, or you wish to test things locally before sending the merge to the repo on Github, you can perform the merge locally instead.

You can find the instruction to do so by clicking the (i) icon on the merge bar.



However, there's an alternative that may be better for long standing branches.

Squash, Rebase, and Cherry Pick

In long standing branches, merging can often cause lots problems when updating if changes in a given branch conflict with changes recently merged into the master branch. If there are many commits to the same file, **git merge** may force you to fix the same merge conflicts over and over again, causing a real headache. While there are many ways to mitigate this issue, such as enabling **git rerere** to reuse recorded resolution of conflict merges, squashing a series of related changes into 1 commit and cherry-picking it into the master is a great solution, especially for topic branches and isolated features.

There are several advantages of performing merges this way. First, you only have to deal with merge conflicts once, since all commits are compressed into 1. Second, each commit represents an entire set of changes required for a feature or task, which makes it easy to pin point bugs and other problems when they arise and to remove a change set when it's no longer necessary.

There are also disadvantages of squashing commits. First, you will lose the details and information for each change, as all changes squashed are compressed together. So the net effect is the same. Second, it can be dangerous and problematic if used incorrectly, such as squashing commits that have been pushed to the remote server and others depend on for their work. Because squashing is changing the git history, you can cause many conflicts that way. However, if you are using this locally or you are the only person working on your branch and you know exactly what you are doing.

To perform this, use the following command

```
git rebase -i HEAD~10
```

-i stands for **interactive mode** and **HEAD~10** means to examine the 10 latest commits.

If you see an **fatal: Needed a single revision** error, this usually means that there are not that many commits left. Try a lower number.

This will open up an editor with git commit messages that looks something like this:

```
pick fc283f2 Section: Creating a Topical Branch
pick 9197a2f Added command to push the branch to github
pick 4782b63 started the creating a pull request section
pick f356b3c Added first step in creating a pull request
pick e5e5802 Creating a pull request: head branch drop
pick 7ab623d Creating a pull request: title and description
pick 95494e2 Creating a pull request: open pull request
pick aa140bc Using a pull request: comments, commits,
pick 607c564 Merging a pull request: merge pull request
pick 778a8f6 Merging a pull request: merging locally
```

There are many options available at this stage. These are detailed in [this github help page](#). Here, I'm going to simply squash all changes in this pull request into one. Save and close the editor.

```
pick 4782b63 started the creating a pull request section
s ea78cfb Creating a pull request
s 8a2ddd5 Using a pull request: comments, commits, diff, and in line co
s 9d8c711 Merging a pull request: merge pull request button
s 48d5755 Merging a pull request: merging locally
s f67cf44 Merging a pull request: rebase discussion
s bb06bab Merging a pull request: rebase commands
s ac03230 Merging a pull request: squashing commits
s 5497c80 Merging a pull request: force push
```

The next screen will pop up asking you to edit your commit messages. You can choose to edit them or simply continue. Save and close the editor.

```
$ git rebase -i HEAD~10
[detached HEAD ea78cfb] Creating a pull request
1 file changed, 21 insertions(+)
Successfully rebased and updated refs/heads/pull-request-demo.
```

Once your squash completes, you can push it to the remote repo. In this case, these squashed commits have been pushed to the server. However, I'm the only user of this branch and can safely force push the commit to update the git repo.

```
git push origin pull-request-demo -f
```


To merge the commit, we will use `git cherry-pick`.


```
$ git cherry-pick d97150d566a7032478d4c0f624f5c9d1363501d1
[master 9a55902] started the creating a pull request section
1 file changed, 98 insertions(+)
```

You are done! Github should detect the changes and update the pull request. You can then mark the pull request as merged and optionally delete the branch.

Closing a Pull Request

You can simply click on the "Close" button on the pull request to close it. Optionally, you can delete the branch directly using the "Delete this branch" button.

Merged  yangsu merged commit 33a3916 into master from pull-request-demo-s_ just now

Closed  yangsu closed the pull request just now

Pull request merged — the `pull-request-demo-squashed` branch can be deleted safely.

 **Delete this branch**

Pull Request Tutorial is maintained by [yangsu](#)

This page was generated by [GitHub Pages](#). Tactile theme by [Jason Long](#).