

Github Pull Request

Christian Külker

2024-03-18

Contents

1	Introduction	1
2	Understanding The Remote Origin	4
3	Delete An Unused Branch	5
4	Keeping Up-To-Date	5
5	History	5
6	Disclaimer of Warranty	5
7	Limitation of Liability	5

1 Introduction

A common way to participate in git-managed software projects is to submit a “pull request” to certain services such as <https://github.com> (it’s called a “merge request” on <https://gitlab.com>). This technique is very popular with git because people often have a one-time commitment and do not have write access to the repository. A pull request is therefore a one-time **write** exception to the repository. In traditional VCS, for example, a tarball would be emailed, and the repository maintainer would manually save the tarball to the file-system, extract the contents and add the changes to the repository, review it, make a diff, and decide whether to include the changes in the next commit of the repository to allow for a one-time contribution. This was a labor-intensive, error-prone process. Git and GitHub use the concept of a “pull request” and a “fork” to provide a semi-automated method for a one-time write contribution. This section is about how to make

a pull request (and fork) on GitHub specifically. As this is not a Git feature as such, the web interface of GitHub must be used in conjunction with the command line.

1. Log into <https://github.com>
2. Go to the website projects repository, press the fork button
3. Clone **your** fork of the repository

```
user@host:~$ git clone git@github.com:USER/REPOSITORY.git
```

4. Enter the REPOSITORY

```
user@host:~$ cd REPOSITORY
```

5. Switch to the main or master branch (your branch should be redirected from the master) and check the other branches so that your new name does not conflict. For example

```
user@host:~$ git checkout master
user@host:~$ git branch -a
* master
remotes/origin/HEAD -> origin/master
remotes/origin/SabineLoss-patch-1
remotes/origin/SabineLoss-patch-2
remotes/origin/master
```

6. Create a new self-descriptive branch with a `BRANCHNAME`

```
git branch BRANCHNAME
```

7. Switch to the new branch

```
git checkout BRANCHNAME
```

The short form of the last 2 steps, create a self-describing branch of the intended change.

```
user@host:~$ git checkout -b BRANCHNAME
```

For example

```
user@host:~$ git checkout -b
↳ fix/typo-overview-security.md-identification-against
```

Make sure you are at your branch (example from above)

```
user@host:~$ git branch -a
* fix/typo-overview-security.md-identification-against
```

```
master
remotes/origin/HEAD -> origin/master
remotes/origin/SabineLoss-patch-1
remotes/origin/SabineLoss-patch-2
remotes/origin/master
```

8. Create very few commits with **good** commit messages

```
user@host:~$ sed -i -e 's%A%B%' file.md
user@host:~$ git commit -m 'Misspelling (A -> B)' file.md
```

9. Cleanup the branch before pushing

10. Pushing the branch to `github.com`

When using just `git push` you will get an error message like this

```
user@host:~$ git push
fatal: The current branch BRANCHNAME has no upstream branch. To push the
current branch and set the remote as upstream, use

git push --set-upstream origin BRANCHNAME
```

However you can just push it with a one time statement

```
user@host:~$ git push origin BRANCHNAME
```

Example

```
user@host:~$ git push origin
  ↪ fix/typo-overview-security.md-identification-against
Counting objects: 6, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 604 bytes | 0 bytes/s, done.
Total 6 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), completed with 2 local objects.
remote:
remote: Create a pull request for
      'fix/typo-overview-security.md-identification-against' on GitHub by
      visiting:
remote:   https://github.com/ckuelker/cwa-documentation/pull/new/\
      fix/typo-overview-security.md-identification-against
remote:
```

```
To github.com:ckuelker/cwa-documentation.git
* [new branch]      fix/typo-overview-security.md-identification-against
↩ -> \
                    fix/typo-overview-security.md-identification-against
```

After the push the GitHub Web Interface shows a message:

```
BRANCHMAME had recent pushes X seconds|minutes ago [Compare & pull
request]
```

11. Request A Pull

The last commit already shows the URL to go to <https://github.com/UPSTREAM-USER/UPSTREAM-REPO/pull>
Use the URL and press the green “Compare & Pull Request” button. Add a message.

OR use the button under GitHub Web interface mentioned at the end of the last section.

2 Understanding The Remote Origin

```
user@host:~$ git remote show origin
```

Example

```
user@host:~$ git remote show origin
* remote origin
Fetch URL: git@github.com:ckuelker/cwa-testresult-server
Push URL: git@github.com:ckuelker/cwa-testresult-server
HEAD branch: master
Remote branches:
  fix/no-nullable-result          tracked
  fix/typo-docs-architecture-overview.md-security tracked
  master                          tracked
Local branch configured for 'git pull':
  master merges with remote master
Local refs configured for 'git push':
  fix/typo-docs-architecture-overview.md-security pushes to \
  fix/typo-docs-architecture-overview.md-security (up to date)
  master                          pushes to master \
                                   (up to date)
```

3 Delete An Unused Branch

Sometimes, while working on a pull request, one realizes that the name of the branch is not appropriate, that an addition is needed. Usually the `-d` option is used.

```
user@host:~$ git branch -d BRANCHNAME
```

However, if the branch is not merged, `-D` must be used.

```
user@host:~$ git branch -D BRANCHNAME
```

4 Keeping Up-To-Date

See [managing forks](#) for details.

5 History

Version	Date	Notes
0.1.5	2024-03-18	Fix typos
0.1.4	2024-02-27	Small improvements and changes in formatting
0.1.3	2023-05-08	Improve writing
0.1.2	2022-06-23	Improve introduction
0.1.1	2022-05-31	Fix front matter, shell->bash
0.1.0	2020-06-08	Initial release

6 Disclaimer of Warranty

THERE IS NO WARRANTY FOR THIS INFORMATION, DOCUMENTS AND PROGRAMS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE INFORMATION, DOCUMENT OR THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE INFORMATION, DOCUMENTS AND PROGRAMS IS WITH YOU. SHOULD THE INFORMATION, DOCUMENTS OR PROGRAMS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

7 Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE INFORMATION, DOCUMENTS OR PROGRAMS AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE INFORMATION, DOCUMENTS OR PROGRAMS (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED

INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE INFORMATION, DOCUMENTS OR PROGRAMS TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.