

Git: Renaming a master Branch to main Step-by-Step



Dirk Avery

Follow



Jul 20, 2020 · 5 min read



If we, as technologists, have the power to make the world better, more inclusive, and less offensive, we should accept the challenge to act. Here I go through the steps to rename a Git default branch called `master` to `main` for a GitHub repository that I maintain called [scratchrelaxtv](https://github.com/DirkAvery/slashrelaxtv).

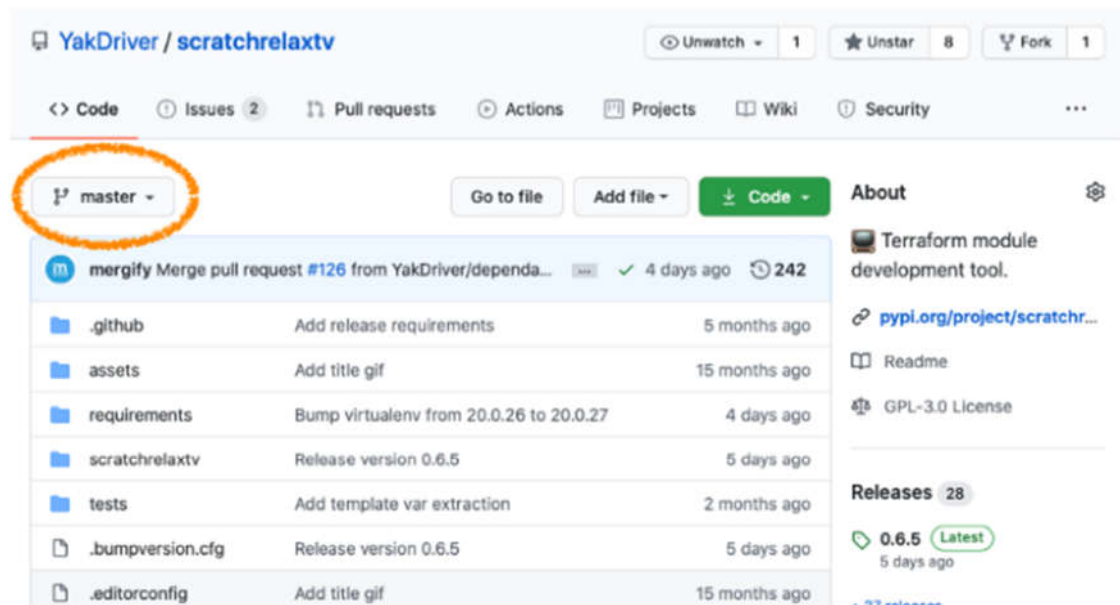


(Photo credit: Andrea Piacquadio)

NOTE: These steps are designed for a non-forked repository that you own and where the `origin` is in your GitHub account. Renaming an organizational repository will involve a similar process but is beyond the scope of these steps.

Step 1. Investigate Your Repository

The first thing you'll want to do is verify that the default branch is indeed called `master`. Browse to the repository's main page. For example, here is the main page for `scratchrelaxtv` (with URL <https://github.com/YakDriver/scratchrelaxtv>).



The default branch is shown prominently on the main page so we know that the repository's default branch is `master`.

Step 2. Update Your Local master Branch

We need to make sure that our local branch is in sync with what's out on GitHub. We'll go to the directory of the repository's local clone and update the local and remote default branches:

```
% cd scratchrelaxtv
% git checkout master
Already on 'master'
Your branch is up to date with 'origin/master'.
% git fetch --all -p
Fetching origin
% git pull
Already up to date.
% git push
Everything up-to-date
```

Step 3. Rename Your Local master Branch

In this step, we are simply changing, locally, the name of `master` to `main`. Notice that remotely, our local `main` branch is still tracking the remote `master`.

```
% git branch -m master main
% git status
On branch main
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

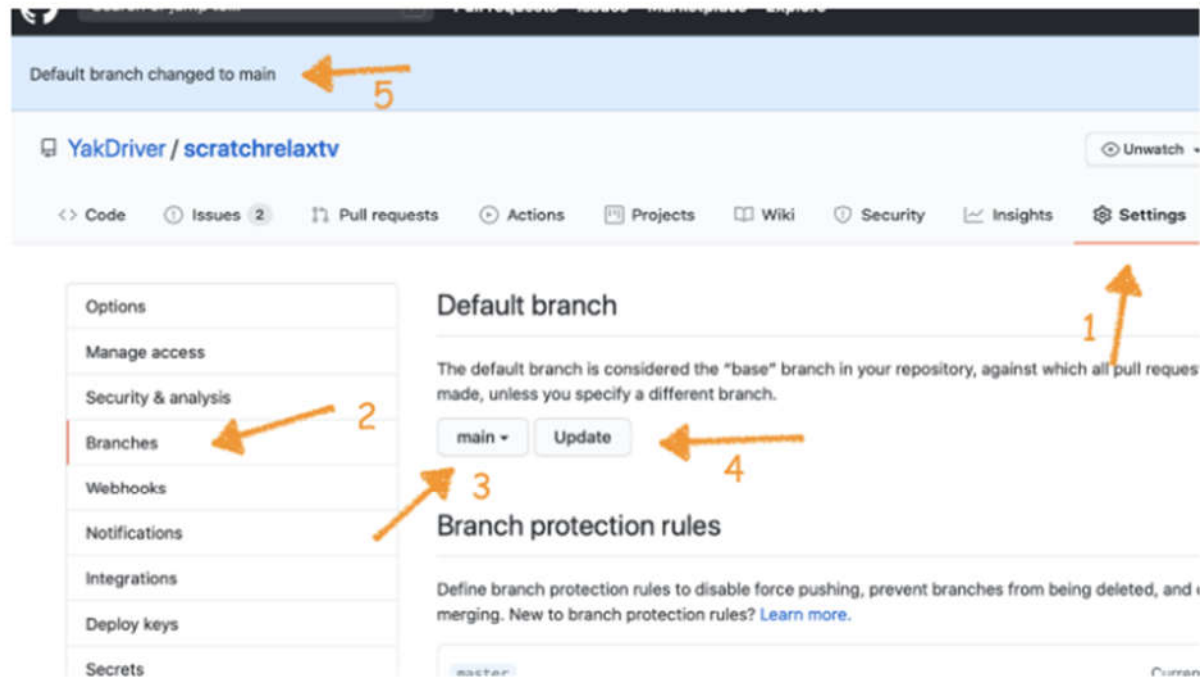
Step 4. Push the Renamed Main Branch

Now, we're going to send our renamed branch off to the interwebs.

```
% git push -u origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'main' on GitHub by visiting:
remote:   https://github.com/YakDriver/scratchrelaxtv/pull/new/main
remote:
To https://github.com/YakDriver/scratchrelaxtv.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Step 5. Update Default Branch on GitHub

Let's go back to GitHub in the browser. Choose **Settings** and **Branches**. Under the **Default branch** heading, select the `main` branch and choose **Update**.



Step 6. Update Branch Protection Rules

Still under **Settings** and **Branches** on GitHub, if any branch protection rules apply to `master`, choose **Edit** next to the rule.

Default branch

The default branch is considered the "base" branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

main Update

Branch protection rules

Add rule

Define branch protection rules to disable force pushing, prevent branches from being deleted, and optionally require status checks before merging. New to branch protection rules? [Learn more.](#)

master

Currently applies to 1 branch

Edit

Delete

Previous

Next

Under **Branch name pattern**, enter “main,” and choose **Save changes**.

Branch protection rule

Branch name pattern

main

Applies to 1 branch

master

Protect matching branches

☒ **Require pull request reviews before merging**
When enabled, all commits must be made to a non-protected branch and submitted via a pull request with the required number of approving reviews and no changes requested.

Rules applied to everyone including administrators

☐ **Allow force pushes**
Permit force pushes for all users with push access.

☐ **Allow deletions**
Allow users with push access to delete matching branches.

Save changes

Step 7. Delete master

At this point, we still have one branch called `master` on GitHub. We can delete the branch two ways, either from GitHub or locally.

From GitHub, choose **branches**.

YakDriver / scratchrelaxtv

Code Issues 2 Pull requests Actions Projects Wiki Security Insights Settings

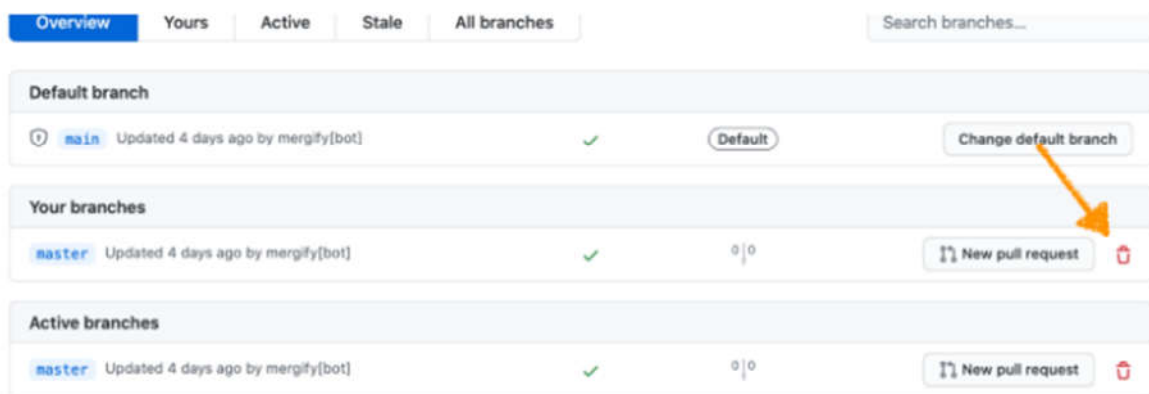
main 2 branches 28 tags

Go to file Add file Code

mergify Merge pull request #126 from YakDriver/dependabot/pip/virtualenv-2... ✓ ef9cde7 4 days ago 242 commits

.github	Add release requirements	5 months ago
assets	Add title gif	15 months ago
requirements	Bump virtualenv from 20.0.26 to 20.0.27	4 days ago
scratchrelaxtv	Release version 0.0.1	5 days ago

Then, choose the trash icon next to the `master` branch.



Alternatively, you can push a remote branch delete request from the command line.

```
% git push origin --delete master
To https://github.com/YakDriver/scratchrelaxtv.git
- [deleted]          master
```

Step 8. Set the Remote HEAD

If you look at the Git log now, you'll see that two references point to the latest `main` commit: the local `HEAD` and the remote branch `main`.

```
* 0f9cde7 - (HEAD -> main, origin/main) Merge pull request #126...
```

The trouble is that `origin/HEAD` is not pointing there and doesn't exist since we just deleted the commit and branch where it pointed. If we attempt to use the symbolic-ref now (e.g., `git pull origin` is an implicit use of this symbolic-ref), we would get an error.

To fix this, we can ask Git to set the remote `HEAD` (i.e., `origin/HEAD`) automatically, which will be the latest commit on the default branch:

```
% git remote set-head origin --auto
```

Now we'll see that three references point to the latest `main` commit:

```
* 0f9cde7 - (HEAD -> main, origin/main, origin/HEAD) Merge pull  
request #126...
```

Step 9. Check Your Code and Config

Updating Git and GitHub is just part of the fix. Since `master` has been considered a well-known term, it's likely we've hardcoded it in CI configuration, build-badge URLs, and automation code. Rather than go through all the possibilities, here is a checklist of places to look. After renaming your default branch to `main`, search your repository for any mentions of `master`.

Here are some specific areas to look:

- `.travis.yml`
- `azure-pipelines.yml`
- `.circleci/config.yaml`
- Badge URLs such as <https://travis-ci.org/YakDriver/scratchrelaxtv.svg?branch=master>
- The `.github/workflows` directory