

Merging by rebase

This can only be done by repo collaborators. If you use the github “merge” button, a merge commit will always be created, even if the merge was fast-forward.

```
git checkout master
git pull upstream master
git rebase master feature_branch # will leave you in feature_branch
git checkout master
git merge feature_branch # will not create a merge commit!
git push upstream master
```

How to fix messed history

Sometimes whenever one perform a rebase, the history is messed on the github pull request and includes many unrelated commits. The following procedure should allow to clean the history and keep only the relevant commits.

```
git checkout master
git fetch upstream
git rebase upstream/master
git checkout new-branch
git rebase master
git log # Check that it has worked as expected
git push -f
```

Check out pull requests to your local repo

```
fetch = +refs/pull/*/head:refs/upstream/pr/*
```

Find & replace in the whole repo

```
git ls-files | xargs sed -i -e 's/old-method-name/new-method-name/g'
```