[Syncing a fork branch from the command line](https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/working-with-forks/syncing-a-fork#syncing-a-fork-branch-from-the-command-line)

Link: <https://www.freecodecamp.org/news/how-to-sync-your-fork-with-the-original-git-repository>

Before you can sync your fork with an upstream repository, you must configure a remote that points to the upstream repository in Git. For more information, see "Configuring a remote repository for a fork."

# Add a new remote upstream repository

git remote add upstream https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY.git

# Sync your fork

git fetch upstream

git checkout master

git merge upstream/master

**Configuring a remote repository for a fork**

You must configure a remote that points to the upstream repository in Git to sync changes you make in a fork with the original repository. This also allows you to sync changes made in the original repository with the fork.

**Open Git Bash.**

List the current configured remote repository for your fork.

$ git remote -v

> origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (fetch)

> origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (push)

Specify a new remote upstream repository that will be synced with the fork.

git remote add upstream https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY.git

Verify the new upstream repository you've specified for your fork.

$ git remote -v

> origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (fetch)

> origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (push)

> upstream https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY.git (fetch)

> upstream https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY.git (push)

GitBash:

Step 1:

Open Git Bash.

Change the current working directory to your local project.

Step 2:

Fetch the branches and their respective commits from the upstream repository. Commits to BRANCHNAME will be stored in the local branch upstream/BRANCHNAME.

Step 3

$ git fetch upstream

> remote: Counting objects: 75, done.

> remote: Compressing objects: 100% (53/53), done.

> remote: Total 62 (delta 27), reused 44 (delta 9)

> Unpacking objects: 100% (62/62), done.

> From https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY

> \* [new branch] main -> upstream/main

Step 4

Check out your fork's local default branch - in this case, we use main.

$ git checkout main

> Switched to branch 'main'

Step 5

Merge the changes from the upstream default branch - in this case, upstream/main - into your local default branch. This brings your fork's default branch into sync with the upstream repository, without losing your local changes.

$ git merge upstream/main

> Updating a422352..5fdff0f

> Fast-forward

> README | 9 -------

> README.md | 7 ++++++

> 2 files changed, 7 insertions(+), 9 deletions(-)

> delete mode 100644 README

> create mode 100644 README.md

If your local branch didn't have any unique commits, Git will perform a fast-forward. For more information, see Basic Branching and Merging in the Git documentation.

$ git merge upstream/main

> Updating 34e91da..16c56ad

> Fast-forward

> README.md | 5 +++--

> 1 file changed, 3 insertions(+), 2 deletions(-)

If your local branch had unique commits, you may need to resolve conflicts. For more information, see "Addressing merge conflicts."