

[Article version: GitHub.com](#) [GitHub.com](#) / [Collaborating with issues and pull requests](#) / [Working with forks](#) / [Syncing a fork](#)

Syncing a fork

Sync a fork of a repository to keep it up-to-date with the upstream repository.

Before you can sync your fork with an upstream repository, you must [configure a remote that points to the upstream repository](#) in Git.

- 1 Open Git Bash.
- 2 Change the current working directory to your local project.
- 3 Fetch the branches and their respective commits from the upstream repository. Commits to `BRANCHNAME` will be stored in the local branch `upstream/BRANCHNAME`.

```
$ 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
```

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

```
$ git checkout main
> Switched to branch 'main'
```

- 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 instead perform a "fast-forward":

```
$ git merge upstream/main
> Updating 34e91da..16c56ad
> Fast-forward
>  README.md                   | 5 +++--
>  1 file changed, 3 insertions(+), 2 deletions(-)
```

Tip: Syncing your fork only updates your local copy of the repository. To update your fork on GitHub, you must [push your changes](#).

