**Working with forks**

upstream- repository that you fork from

origin – your forked repository

**Option 1: Web UI**

A screenshot of a computer

Description automatically generated

If I don’t have change I will press “Sync fork”:

A screenshot of a computer

Description automatically generated

If I have changes I will get:

A screenshot of a computer

Description automatically generated

I press “Open pull request”.And then I open Pull request to the upstream if I want:

A screenshot of a computer

Description automatically generated

I need to resolve conflict first so I press “Resolve confict” via github or resolve it via IDE(prefer IDE):

Resolving it in github I have A screenshot of a computer

Description automatically generated

This part in yellow I should resolve and only my change will remain

A screenshot of a computer

Description automatically generated

I press “Mark as resolved”:

A close-up of a computer screen

Description automatically generated

And commit merge:

A screenshot of a computer

Description automatically generated

A screenshot of a chat

Description automatically generated

And I am ready with my pull request to the upstream

A screenshot of a computer

Description automatically generated

**Option2 :Command line**

**Syncing a fork**

The Setup

Before you can sync, you need to add a remote that points to the upstream repository. You may have done this when you originally forked.

*Tip: Syncing your fork only updates your local copy of the repository; it does not update your repository on GitHub.*

**$ git remote -v**

# List the current remotes

origin https://github.com/user/repo.git (fetch)

origin https://github.com/user/repo.git (push)

$ **git remote add upstream https://github.com/otheruser/repo.git**

# Set a new remote

$ **git remote -v**

# Verify new remote

origin https://github.com/user/repo.git (fetch)

origin https://github.com/user/repo.git (push)

upstream https://github.com/otheruser/repo.git (fetch)

upstream https://github.com/otheruser/repo.git (push)

Syncing

There are two steps required to sync your repository with the upstream: first you must fetch from the remote, then you must merge the desired branch into your local branch.

Fetching

Fetching from the remote repository will bring in its branches and their respective commits. These are stored in your local repository under special branches.

$ **git fetch upstream**

**#** Grab the upstream remote's branches

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/otheruser/repo

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

We now have the upstream's master branch stored in a local branch, upstream/master

$ git branch -va

# List all local and remote-tracking branches

\* master a422352 My local commit

remotes/origin/HEAD -> origin/master

remotes/origin/master a422352 My local commit

remotes/upstream/master 5fdff0f Some upstream commit

Merging

Now that we have fetched the upstream repository, we want to merge its changes into our local branch. This will bring that branch into sync with the upstream, without losing our local changes.

$ git checkout master

# Check out our local master branch

Switched to branch 'master'

$ **git merge upstream/master**

# Merge upstream's master into our own

Updating a422352..5fdff0f

Issue git push origin main

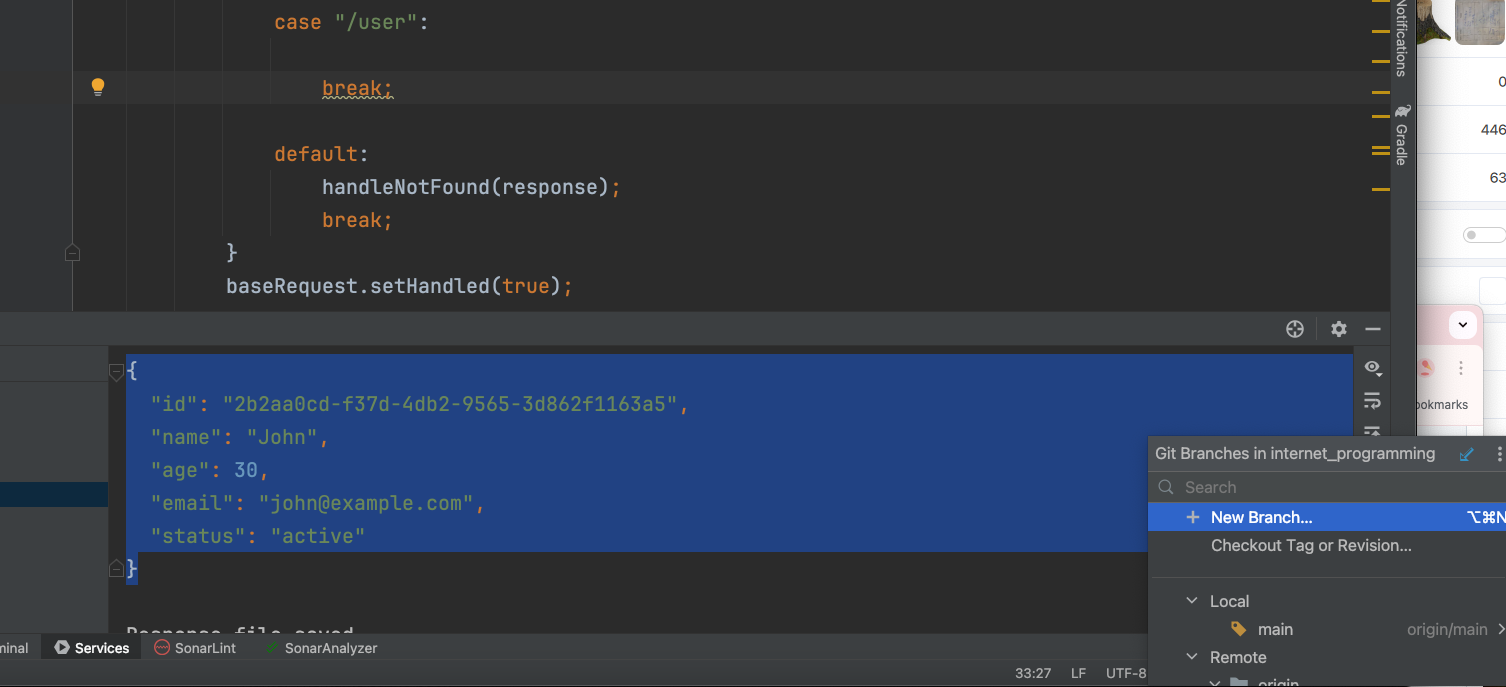
**Option 3:GitHub CLI**

To update the remote fork from its parent, use the gh repo sync subcommand and supply your fork name as argument.

$ gh repo sync owner/cli-fork

**Working with branches of cloned repo**

1. I clone the repo
2. I create new branch:



**A screen shot of a computer

Description automatically generated**

I do the changes then commit of changed files and push my branch

In Github I am ready to make Pull Request for my branch “test” to “main” branch

If you have local changes you can stash them.

A screenshot of a computer

Description automatically generated