

Setting up a second remote

This section describes how to set up a second remote. This is so that you can bring in changes that I do to my basecode, and incorporate them into your repository. The steps roughly would be:

1. (Do once) Set up my remote repository as a remote repository to your local clone of your remote repository. We will call this mainSource.
2. When you want to work on a new lab that may have been “updated” on my end:
 - a. Set up a new local branch to track the branch for this lab on your remote (origin). For example for Lab2 you will set up a local branch Lab2 that “tracks” origin/Lab2.
 - b. Fetch the latest version of the lab repository from my remote (mainSource), into a local branch Lab2.
 - c. Merge these changes from mainSource/Lab2 into your local Lab2.
 - d. Push your Lab2 to origin/Lab2.
 - e. Now your remote has the same changes that mine had.

We will now implement these steps:

```
git remote add mainSource git@gitlab.com:skiadas/WebAppsLabs.git
git checkout -b Lab2 --track origin/Lab2
git fetch mainSource
git merge --ff-only mainSource/Lab2
git push origin Lab2
```

The first line sets up a new remote. You only need to do it once. In future runs you will just need to run the remaining four lines.

The second line sets up a local branch, called Lab2, to track your remote’s Lab2.

The third line fetches in all updates from my remote branch (mainSource).

The fourth line merges those updates into your Lab2, but *only* if those are introducing new commits and nothing else (this is known as fast-forward).

The fifth line pushes these changes to your remote (origin). You can do this one from GitKraken as well.