

CS312 - Setting up Git for the project

David Kauchak

We're going to use GitHub for now to host our projects. It's free, it's robust, it's easy to use and has a nice user interface. However, to use it for free GitHub requires that you make your project publicly available. If any team has a problem with this please come talk to me and we can figure an alternative out.

Signing up for GitHub

Everyone will need to sign up for an account on GitHub. Go to <https://github.com/>

1. Click on the button that says "Plans, Pricing and Signup"
2. Click on the button that says "Create a free account"
3. Enter the relevant information and click continue.
4. GitHub (along with many git servers) uses ssh to connect to it. Because of this, you'll need to setup an ssh key. Go to:
<http://help.github.com/linux-set-up-git/>
and follow the instructions in the section "Next: Set Up SSH Keys"
Stop after the "Test everything out" section.

If you plan on working from multiple different places (i.e. the lab vs. your laptop), you'll need to make sure that you either use the same ssh key or that you upload any additional ssh keys to your account. Come talk to me if you have problems with working from multiple locations.

Setting Up Your Project

One person on your team needs to setup the project on GitHub. To do this, first have everyone on the team sign-up for GitHub accounts. Get all of the GitHub usernames (you don't technically need them all up front since you can add them later on if you want). Then have that **one** person take the following steps:

1. Login to GitHub online

2. Click on “New Repository” (in the bottom right)
3. Fill in the form. Make sure you give the project a good name since it’s going to be with you for 6 weeks :)
4. Follow the directions on the next page for creating the initial repository. It should be something like (make sure to follow the directions on the GitHub page since the ones below are not specific to your project):

```
- mkdir ProjectName
- cd ProjectName
- git init
- touch README
- edit the README file and put an initial project description
- git add README
- git commit -m 'adding the README'
- git remote add origin git@github.com:username/ProjectName.git
- git push -u origin master
```

5. Click continue on the GitHub page once you’ve done this.
You should now see the README file online.
6. Now you need to add all of the members on the team as collaborators. From this page, click on the “Admin” button near the top of the page.
7. Then, click on “Collaborators” on the left.
8. Add each of the team’s usernames as collaborators. This will allow other members of the team to push commits to the GitHub repository.

After this, your repository should be setup and everyone should be able to read/write to the repository.

Getting a local copy of the project

If you weren’t the person to create the repository, then you’ll need to get your own local copy once GitHub is all setup.

1. Go online to GitHub and login
2. In the bottom right are your repositories that you have access to. If everything got setup right adding you as a collaborator, you should see your new project there. Click on the repository.
3. Now you can get your own local copy of the project setup. In the GitHub project page, there is a field in the middle of the screen where an “SSH” button should be selected. To the right of that is the location of this project. It should be something like `git@github.com:username/ProjectName.git`. Copy this (I’ll refer to it below as the `git_address`).

4. Now go to where you want to house your data locally and type:

```
$ git clone git_address
```

where “git_address” is whatever you copied from the previous step. This should create the project directory and copy any files in the repository.

Interacting with GitHub

You should be all set now to start making changes. GitHub will contain the shared version of your repository. When a repository is cloned, it has a default remote called *origin* that points to the version on GitHub, which you can now use to push and pull to/from.

pull

If you ever want to get the latest version of the project, you can *pull* from the GitHub repository. To do this, you type:

```
git pull origin
```

in your git directory that you cloned. Remember *origin* refers to the remote repository on GitHub, so you’re telling git to pull the data from the remote repository and merge it with your local data.

pull automatically tries to merge so you may get conflicts that you’ll need to resolve. If you don’t want to do an immediate merge, you can do:

```
git fetch origin
```

Which will grab the latest version, but not try and do a merge. If you want to see the difference (or other operations) you interact with *origin/master*. For example:

```
git diff master origin/master
```

which gets the difference between your master branch and the *origin/master* branch which you fetched.

You can then do the merge:

```
git merge master origin/master
```

Sometimes you may just want to do the *pull* or sometimes you might want to do the *fetch* and then *merge*. It’s up to you.

push

Eventually, you're going to want to push changes you've made to the GitHub (i.e. team) repository so that other people can see your changes.

1. First, make sure that you've committed all of your changes to your local repository.
2. Second, you need to make sure that you're up to date with the latest version of the repository. Do a `pull` (or `fetch` and `merge`) to get yourself up to date.
3. Finally, you can push your changes to the remote GitHub repository:

```
git push origin
```

which tells git to push your version to origin (i.e. the remote repository).

It takes a little while to get used to using git, but once you get the hang of it it's a great tool. There are some tutorials linked on the course web page and you can find lots of information by searching online as well. If you get stuck, come talk to me.