How to Access Git Via Your Desktop Computer

Markus A. Garbiso
The University of Alabama

July 25, 2020

1 Accessing Git

Marco covered how to use GitHub with the terminal, but you can also use GitHub 's repository features with **GitHub Desktop** (https://desktop.github.com/). Once you have downloaded the program and installed it you should be able to access all the features of the terminal and a nice GUI. Once you have a GitHub account you can access all of our private repositories. By default you will not be able to see any repositories. The first thing you want to do is to "Clone repository". You can select one of your own repositories from a list or any other repository you have access to via it's URL. Now you will be able to select a branch! I will go over some of the GitHub lingo. You have several tabs.

- File
  - *New repository:* You can add a new repository to publish onto GitHub .
  - *Clone repository:* Use this for repositories that are shared and on GitHub .
- Edit
  This is mostly for the cases were you make a mistake and wanted to undo some fatal error.
- Repository
  This the where you will make contributions to the repository, so this menu will see most of the use.[1]
  - *Push:* Used to push current uncommited commits onto GitHub .
  - *Pull:* Used to pull/Fetch from the repository on GitHub .
  - *Remove...:* this will delete the repository on the local machine (and on GitHub).
  - *View on Github:* This will open you browser to GitHub .

[1] Note the shortcuts which can save a siginificant amount of time.

- Branch
  This is used to make new branches and other commands like to remove all recent changes from that branch.

2 Making a Contribution

GitHub shines when used to collaborate on code development. I will got through the *workflow* of making a contribution.

2.1 Branches

The directory of GitHub is the branch. By default every repository has a *master branch*. Usually if you are in collaboration, you would rarely make minor changes to this branch. When in GitHub Desktop you can create a new branch via the *Branch*. Be sure you are on the correct *Current Branch* before you start to make changes (Commits). You will also have to Sync by *Fetching* the data from the GitHub version. You will be able to see a button to fetch.

2.2 Commits

The unit of change in GitHub are the *commits*. Once you make a change to **any** file, your GitHub Desktop will show you a list of changes along with the list of files. If GitHub Desktop can read the file (like a code file) it will attempt to show the changes. Once you have given your commits a name (*Summary*), you can then upload the changes via a *push*.

The Git Philosophy: "...Being able to work offline is yet another aspect which differentiates it from other tools out there. Just think of the advantage of not having to go over the internet for each and every operation! This makes diff's, commits etc seamless and fast. The only time you need to have the git server available is when you decide to push your code or make your code available to others."

It helps to keep commits **small** and such that any commits don't leave the code "broken".

2.3 Pull Request

Once you have complete the commits enough on a branch, you can then perform a *Pull Request* where the changes on the branch you made commits to will be merged into the master branch[2]. I reccomend you request someone on the GitHub site so that you have a second opinion before the branches are merged.

[2]Our whatever you branch you branched out of

2.4 Merging Branches

When merging branches, sometimes you will have merge conflicts. This is when GitHub doesn't know how to merge the two versions of a file. You will have to then go in manually and tell GitHub which *branch* to keep.

3 Conclusion

This is the end of using GitHub Desktop though I would like to mention that GitHub online offers sever features to orginze workflow. I will be using *Projects* and *Issues*, so it's clear where the group is at on a certain code for a project.