Richard Clarke

A Breakdown of Git

Git is a tool for managing multiple versions of code repositories using distributed version control. In other words, developers can use Git to make changes to files without being on the same network. They host the code on their local machines, make edits, and push their local repository to a remote one where it can be seen and accessed by other developers.

One remote repository, Github, allows users to track the history of their code. With each change, Github take a snapshot of a file. The site then provides users a list of these snapshots for review. Git offers robust features for easing collaboration between developers. One feature in particular, forking, lets developers separate their work from the original code.

A fork is a copy of a code repository. One path leads to the original repo owner's code, and the other is a copy saved to another account. Any changes made in the separate account do not affect the original. Like a fork in the road, both paths have distinct features. This is ideal for working with open source code because developers can start contributing to projects faster. No need to wait for access to a project; it's always available to be copied. After forking, developers can download a local version of the code for offline work. This is known as cloning. Once they start working, developers can create different branches to manage changes.

Branches are are environments created by developers; they contain different versions of the original code. There is a master branch, which contains the main project code. Each branch can be updated at anytime without changing the master branch. This is good for experimenting with new code changes that might cause issues when combined with the master branch.

After changes are made on the local repository, developers can add their files to what is known as a staging area. The staging area collects files that will be used in the snapshots to be stored. Developers add each file to the staging area before committing their changes. A commit is an optional description containing an overview of changes. Once the commit is complete, developers push the code from their local repository to the remote one, updating it with the latest changes and the description attached. From here, developers can submit a pull request to the owner of the original repository asking them to merge the updated code into their project. The owner can view the differences between their code and the updates. This helps them decide whether or not to deny or approve the pull request.

Git has a myriad of features that make development smoother across teams. I’m looking forward to learning as many as I can. This is an excellent resource and I feel it’s needed if you want to interface with a remote repository.

Sources

Media, T. (2017, February 05). Git & GitHub Crash Course For Beginners. Retrieved from https://www.youtube.com/watch?v=SWYqp7iY_Tc

Resources to learn Git. (n.d.). Retrieved from https://try.github.io/