GIT

Submodules

Submodules allow you to keep a Git repository as a subdirectory of another Git repository.

This lets you clone another repository into your project and keep your commits separate.

It often happens that while working on one project, you need to use another project from within it. Perhaps it’s a library that a third party developed or that you’re developing separately and using in multiple parent projects. A common issue arises in these scenarios: you want to be able to treat the two projects as separate yet still be able to use one from within the other.

git submodule add https://github.com/chaconinc/DbConnector

git status

On branch master

Your branch is up-to-date with 'origin/master'.

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: .gitmodules

new file: DbConnecto

[submodule "DbConnector"]

path = DbConnector

url = https://github.com/chaconinc/DbConnector

cloning a submodule:

git clone https://github.com/chaconinc/MainProject

cd DbConnector/ -🡪 empty

$ ls

 You must run two commands:git submodule init to initialize your local configuration file, and git submodule update

Method2: git clone --recurse-submodules https://github.com/chaconinc/MainProject

#### Pulling in Upstream Changes from the Submodule Remo

Update submodule for new changes

If you run git submodule update --remote, Git will go into your submodules and fetch and update for you.

**Pulling Upstream Changes from the Project Remote**

git submodule update --init --recursive

git log -n 3 -3 most recent commits

git log --author="John Smith"

git log --committer="John smith"

git log --after="2019-3-2"

git log -- main.py -commits only to a file

git log -S"# Introduction" 🡪word

git log –oneline -- list of commit IDs and their msgs

git log –stat -allows you to display the number of lines of code added to and deleted from a repository in each commit.

Git log –p – details of commits

git shortlog - The output of the git shortlog command is grouped by author