Git Tutorial v1.0

I. What is Git, Github, and Bitbucket?

Git is a type of version control software. It keeps tracks of all changes made to the project by multiple users. It does this through the use of a “master” branch and many sub-branches. Sub-branches are essentially lines of changes that users are working on. They can create and “check out” a branch to work on, and then commit those changes to the master. Git will not only keep track of all changes, but also do its best to merge the changes. The master branch plus all of its sub-branches is referred to as a repository.

Github is the online hosting service for projects that are using Git. Users can store all their projects online for others to see. It is great for open-source projects and many users contribute code. Unfortunately, to create a private repository on Github you need to pay, which is why we use…

Bitbucket is another online hosting service that allows free, unlimited, private repositories. The one downside is repositories are limited to 5 users. It also allows team accounts.

II. Using Git and Bitbucket together

1. The first thing you need to do is set up Git on your computer.

If you are using Windows:

<https://confluence.atlassian.com/display/BITBUCKET/Set+up+Git+and+Mercurial>

Skip Step 2, we aren’t using Mercurial, another type of version control software.

If you are using Mac OSX:

<https://confluence.atlassian.com/pages/viewpage.action?pageId=269981802>

Skip Step 3, we aren’t using Mercurial.

After this point, everything will be the same for Mac and Windows, with the exception that Mac will be sending commands using Terminal, while Windows will be using the downloaded Git Bash.

2. Sign up for a Bitbucket account.

If you don’t yet have a BitBucket account, you will need to sign up here:

<https://confluence.atlassian.com/display/BITBUCKET/Create+an+Account+and+a+Git+Repo>

Don’t worry about creating a repository for now. I have created a repository that we can use.

3. Access the online repository.

Once you are invited to the repository, you should be able to see all the files using this link:

<https://bitbucket.org/oliverfang/asat>

“Overview” will list all the changes made.

“Source” will list all the files in this project/repository.

Feel free to have a look around to get used to the layout.

4. Create your local repository.

Now you are ready to create the local repository on your computer. First, open Terminal (Mac OSX) or the Git bash (Windows). This is our command line tool, and we will type commands to interact with the repository on Bitbucket.

5. Git walkthrough.

[1] First, go to your home directory. Cd means change directory.

|  |
| --- |
| cd ~ |

[2] Next, go to the directory that contains your MATLAB files. This will become the repository file.

|  |
| --- |
| cd ~/filepath/to/directory |

[3] Now, create a repository in the current directory. You only need to do this once. Note that any command

|  |
| --- |
| git init |

[4] Try viewing all the files in the current directory. You will notice there a newly created .git directory. It’s hidden normally, but the -a reveals hidden files and directories.

|  |
| --- |
| ls -a |

Right now, the .git directory is empty – your MATLAB files in the current directory have not yet been added to the repository. This is the next step.

[5] Now simply add all the files in the current directory to the repository. The period means “all files.”

|  |
| --- |
| git add . |

[6] Now that the files are all inside the repository, Git will keep track of changes made. Let’s say we made a change to a file and we want to commit that change to the repository. After making your change, you will want to save your file in the text editor first, then type this, which will commit the changes to the repository.

|  |
| --- |
| Git commit –m “Name of your commit” |

The –m means that you want to add a comment. Note that it is good practice to name your commit with something descriptive so others will know what changes have been made.

You will see the following output. I made a test to the file called noise\_test.m. Note that the first line says I am on the master branch. More on that in a second. The name of my commit was “Testing a Change.”

|  |
| --- |
| [master 04d2bf5] Testing a Change  1 file changed, 1 insertion(+) |

Every time you make a change and want to commit a change made to a file, you will repeat steps [5] AND [6].

[7] So far, everything has been done locally on your computer. We want to take advantage of the fact that Git is a distributed version control software and upload them online so other users (the team) can access them too. You will only have to do this once.

|  |
| --- |
| git remote add origin <https://oliverfang@bitbucket.org/oliverfang/asat.git> |

I provided the url to the asat repository I created. If you go to the “Overview” section, you can see on the right side there is a box labeled HTTPS. You can copy the link from there if you want to create your own repositories.

[8] Now that the online repository is linked to our local repository, we can attempt to push our local changes to the server.

|  |
| --- |
| git push –u origin master |

You will see an output that looks like this.

|  |
| --- |
| Counting objects: 5, done.  Delta compression using up to 2 threads.  Compressing objects: 100% (3/3), done.  Writing objects: 100% (3/3), 299 bytes, done.  Total 3 (delta 2), reused 0 (delta 0)  remote: bb/acl: oliverfang is allowed. accepted payload.  To https://oliverfang@bitbucket.org/oliverfang/asat.git  12fcb6a..04d2bf5 master -> master  Branch master set up to track remote branch master from origin. |

Now you can check the online repository at <https://bitbucket.org/oliverfang/asat> and you will see the changes have been made. Note that if you try to commit changes such that in conflicts with someone

[9] You can also pull down changes from the online repository to your local repository. This will fetch all the latest files and then merge them with your own.

|  |
| --- |
| git pull |

This is the simple explanation. Alternatively, you can follow the instructions here (you should read this):

<https://confluence.atlassian.com/display/BITBUCKET/Working+with+pull+requests>

[10] Okay, that should be everything you need to know to get started. Here are some other useful command you may need to use.

Branches: Branches are essentially new versions of the original code. Let’s say you are trying to test out a new feature, but don’t want to accidentally ruin your current master copy. You can create a branch called oliverfang/test and Git will save a complete version of your code at the new branch. While you are working in this branch, edits you make will only affect oliverfang/test files. The master files are unchanged. When you are done testing, you can merge it back with the master copy to finalize your changes.

If you are in the master directory, this will create a branch called oliverfang/test.

|  |
| --- |
| git branch oliverfang/test |

If you are in the master directory, this will switch you into the new branch. Note that when you do this, all the files you have open will switch from the “master” version to the “branch” version.

|  |
| --- |
| git checkout oliverfang/test |

Likewise, you can switch back using.

|  |
| --- |
| git checkout master |

If you are in the master directory, this will merge the branch with the master.

|  |
| --- |
| git merge oliverfang/test |

If you want to remove the git repository all together, you will need to go into the original directory, and remove the .git directory.

|  |
| --- |
| rm –rf .git |

Lastly, I will direct you towards this link:

<https://confluence.atlassian.com/display/BITBUCKET/Bitbucket+101>

This will explain how and why we use BitBucket/Git as well as how to deal with any problems you encounter.

Thanks!

Oliver