GIT

LAB ASSIGNMENTS

Pre-Requisitions:

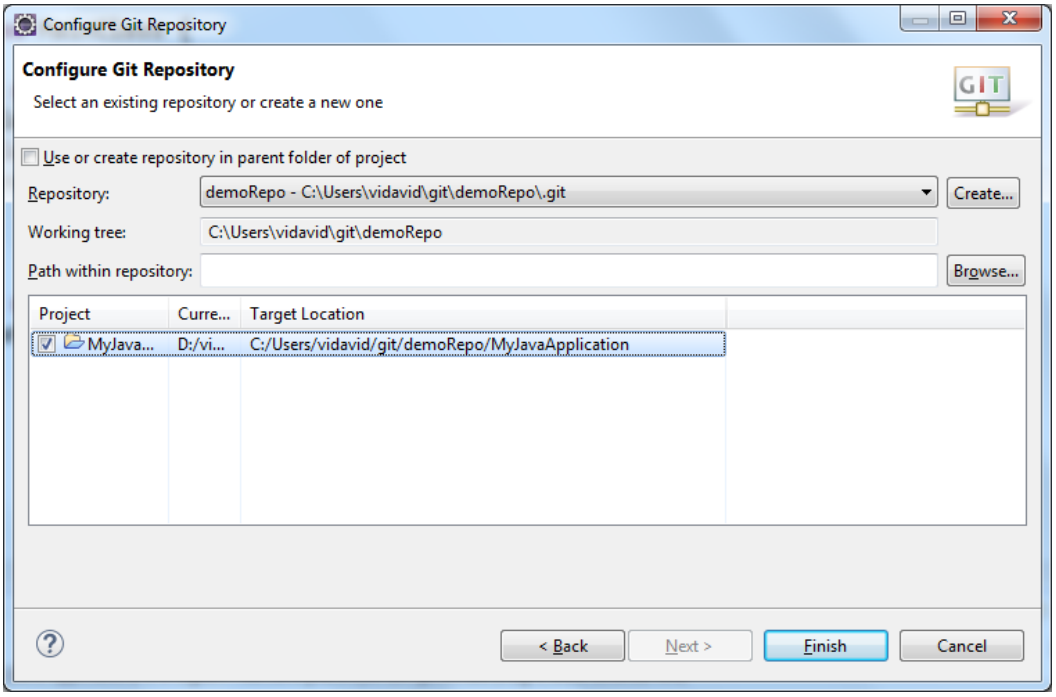
1. Create one account with GitHub (https://github.com/)
    a. Create one repository in the name of "**demoRepo**"
2. JDK 1.6 or higher
3. Git client should be installed. You can download GIT client from the below URL.
    a. https://git-scm.com/download/
4. Install EGIT plug-in in Spring Tool Suite / Eclipse. You can use below link for installing EGIT.
    a. http://www.eclipse.org/egit/download/

Lab 1:

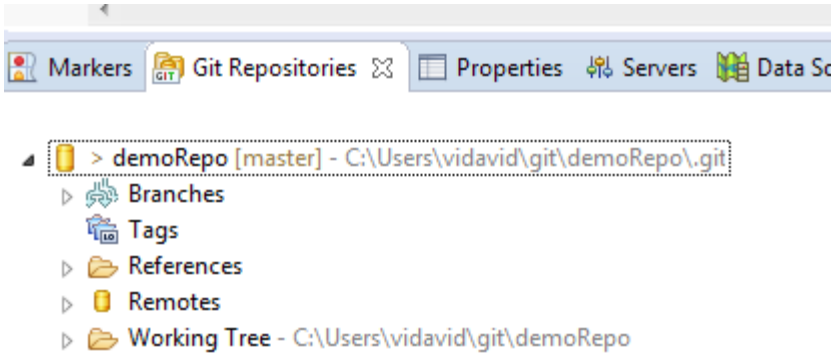
Create one simple Java Project in Eclipse/STS. Configure one remote repository and Check-in your project with local and remote GIT Repository. Before start the assignment create one GitHub account. Use GitHub as your remote repository for this complete lab assignments.

Steps:

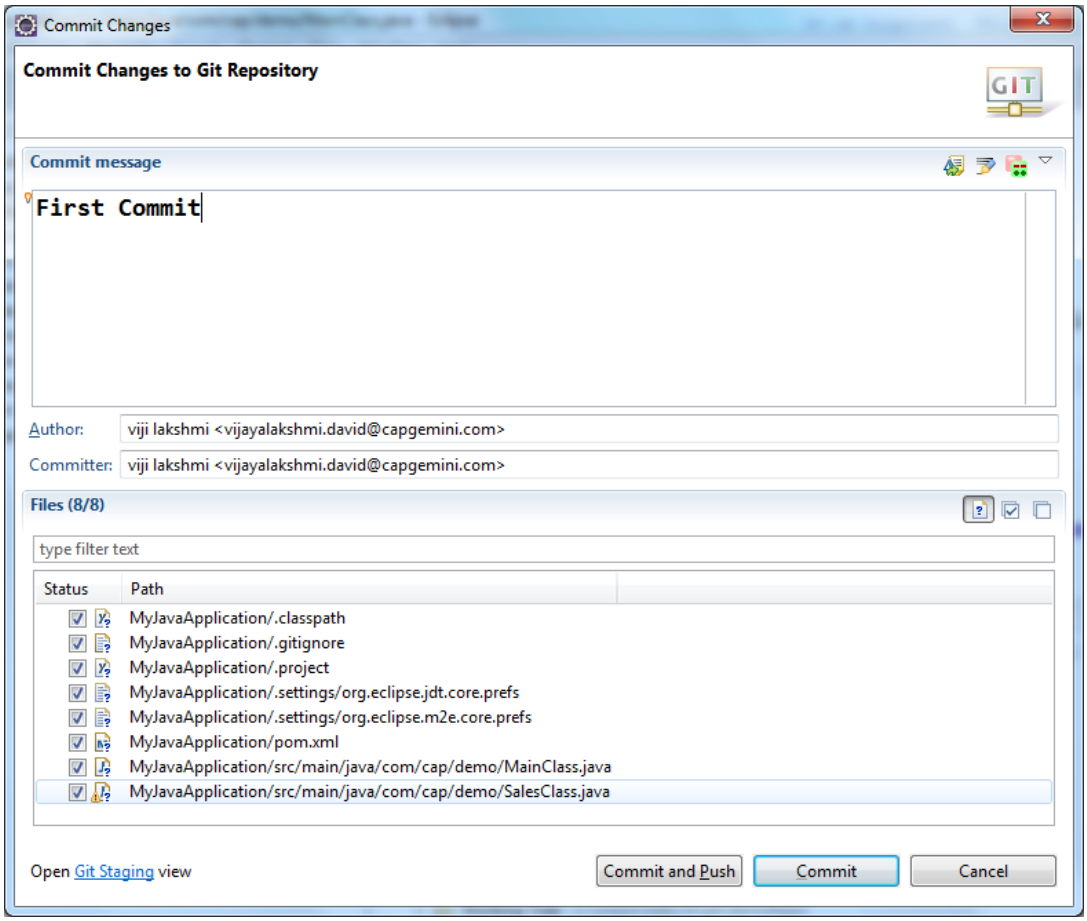
1. Open GitHub URL (https://github.com/caprepo/softwareRepo)
2. Click **MyJavaApplication.zip folder,** next page click download link. The zip file will be downloaded.
3. Unzip the folder
4. Go to Eclipse IDE, choose import ->General -> Existing Projects into Workspace
5. Under select root directory click browse button.
6. Go to the unzip folder location and choose the folder in the name of MyJavaApplication.
7. Now the Product Application project will be imported in your local machine.
8. Share the project with GIT local repository by using the followings:
    a. Right Click the project name (MyJavaApplication) → Team → Share Project → Choose GIT in the screen.
    b. Click create button to create new local repository in the name of demoRepo.
    c. Click Finish



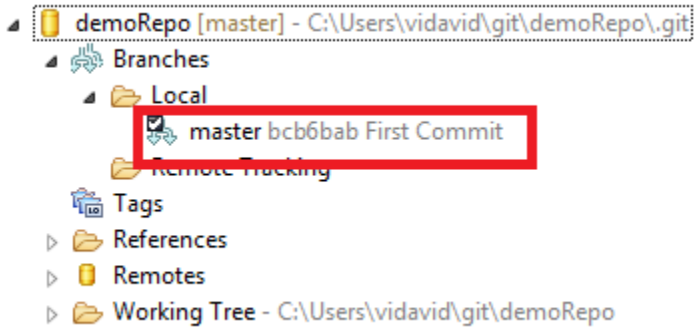
9. You can view the local repository under Git Repositories view:



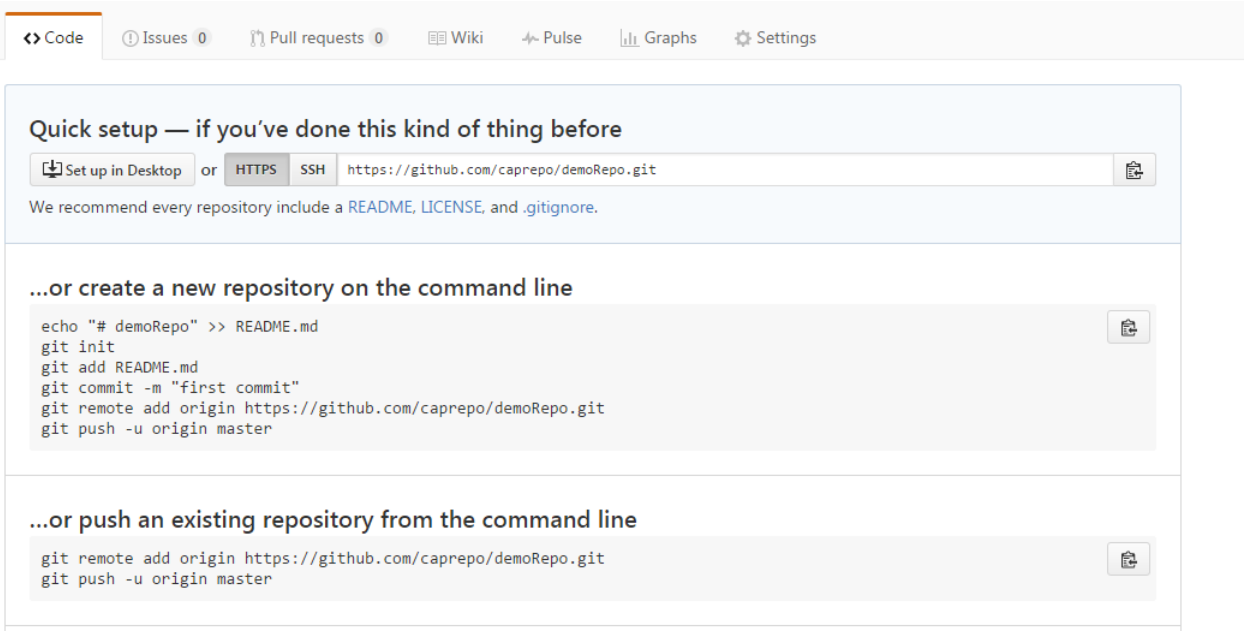
10. Right Click your project select Team → Commit, it will bring below form, in this enter your comments before you commit. After given the suitable comments, select all the file which you want to store in the local repository. Then click commit.



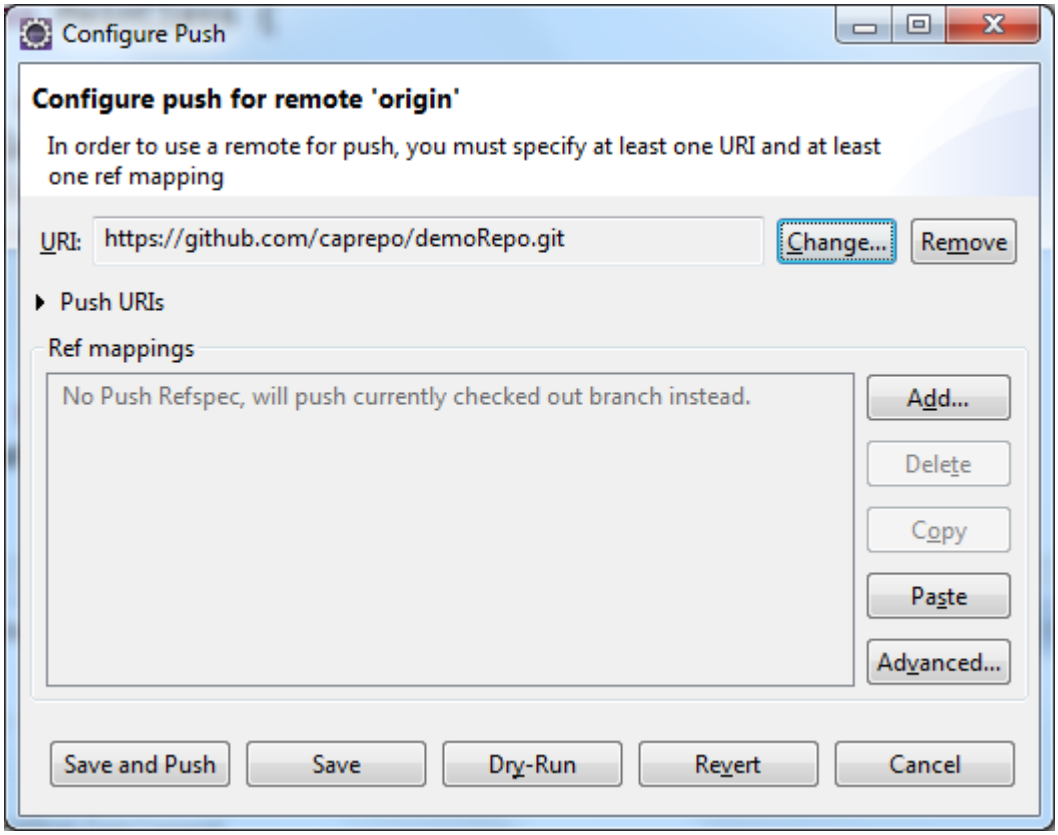
11. Under the branches, you could see the master branch with the comments:



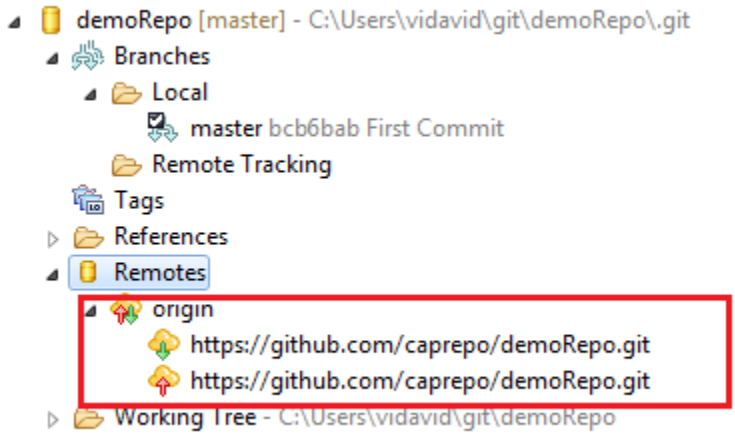
12. Right Click on master branch which has been highlighted in the last snapshot.
13. Now check in this project with GitHub repository.
14. Right Click **Remotes** →Create Remote → RemoteName : Orgin.
15. Select configure push → Ok
16. Click Change button and paste your remote repository URL, you can copy the URL from GitHub Url



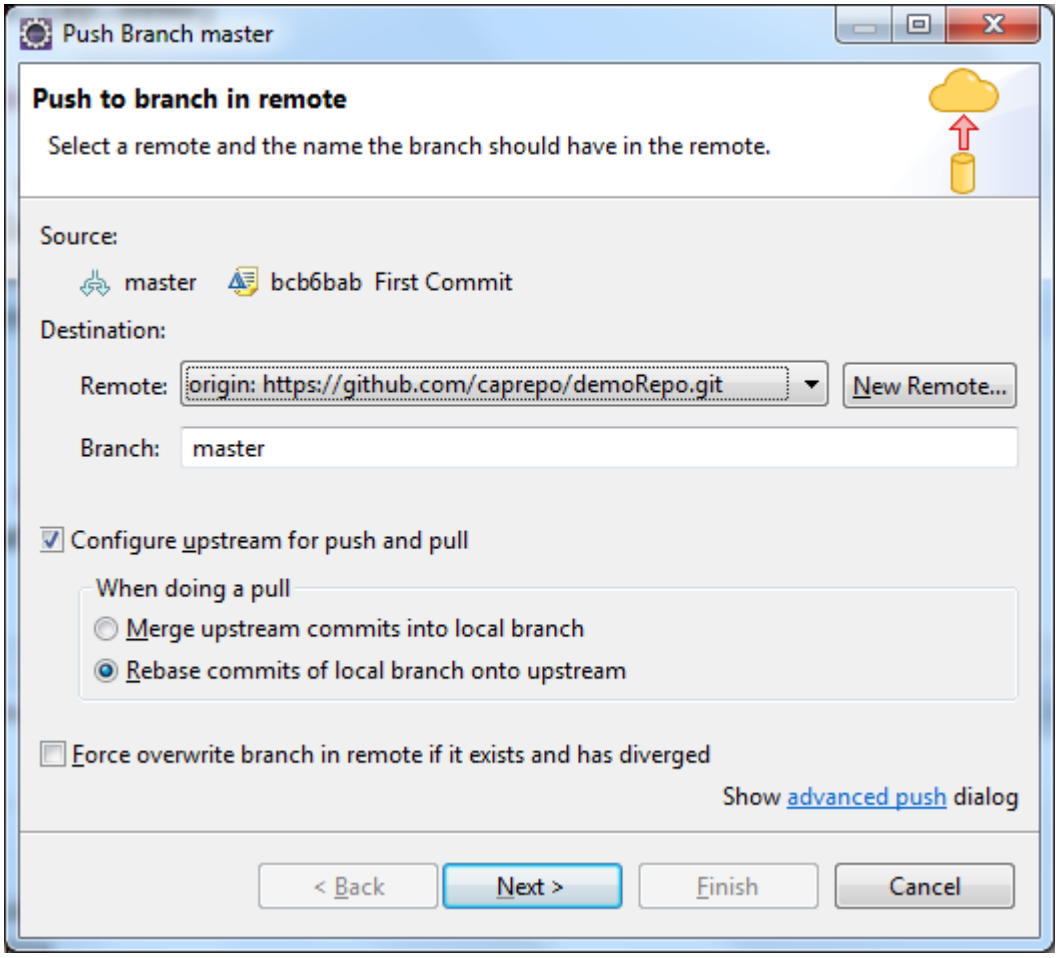
17. Once you pasted the URL, enter your GitHub username and password to access GitHub repository.



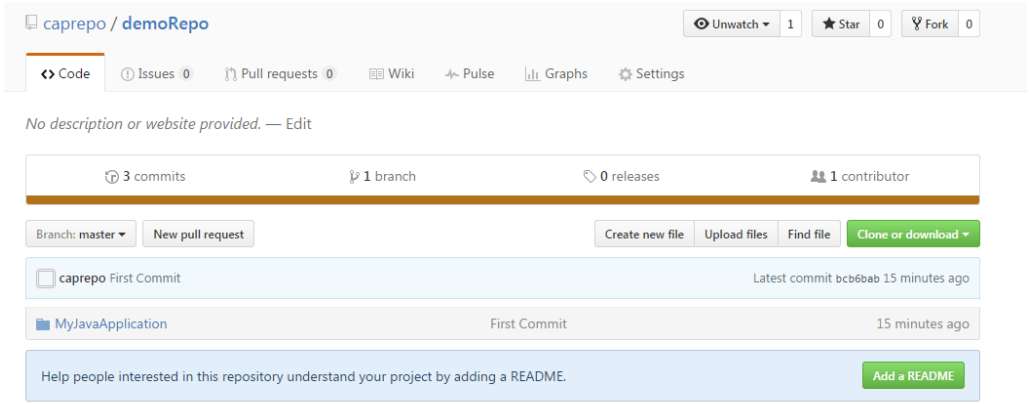
18. Click Save
19. Now you can see remote repository under remotes as below:



20. Again Right Click Master Branch → select **push Branch**
21. You could see below screen:



22. Select Next → Finish → OK
23. Once you give finish, Eclipse will push your code to remote GitHub Repository.
24. Now you can see your project with master details in GitHub Repository as shown below:



25. Successfully you have checked in the project with GitHub repository.

Conclusion:

From the above example, we learnt how to configure GitHub remote repository with Eclipse IDE and also we understand how to check in the existing project with Repository which we configured here.

Note:

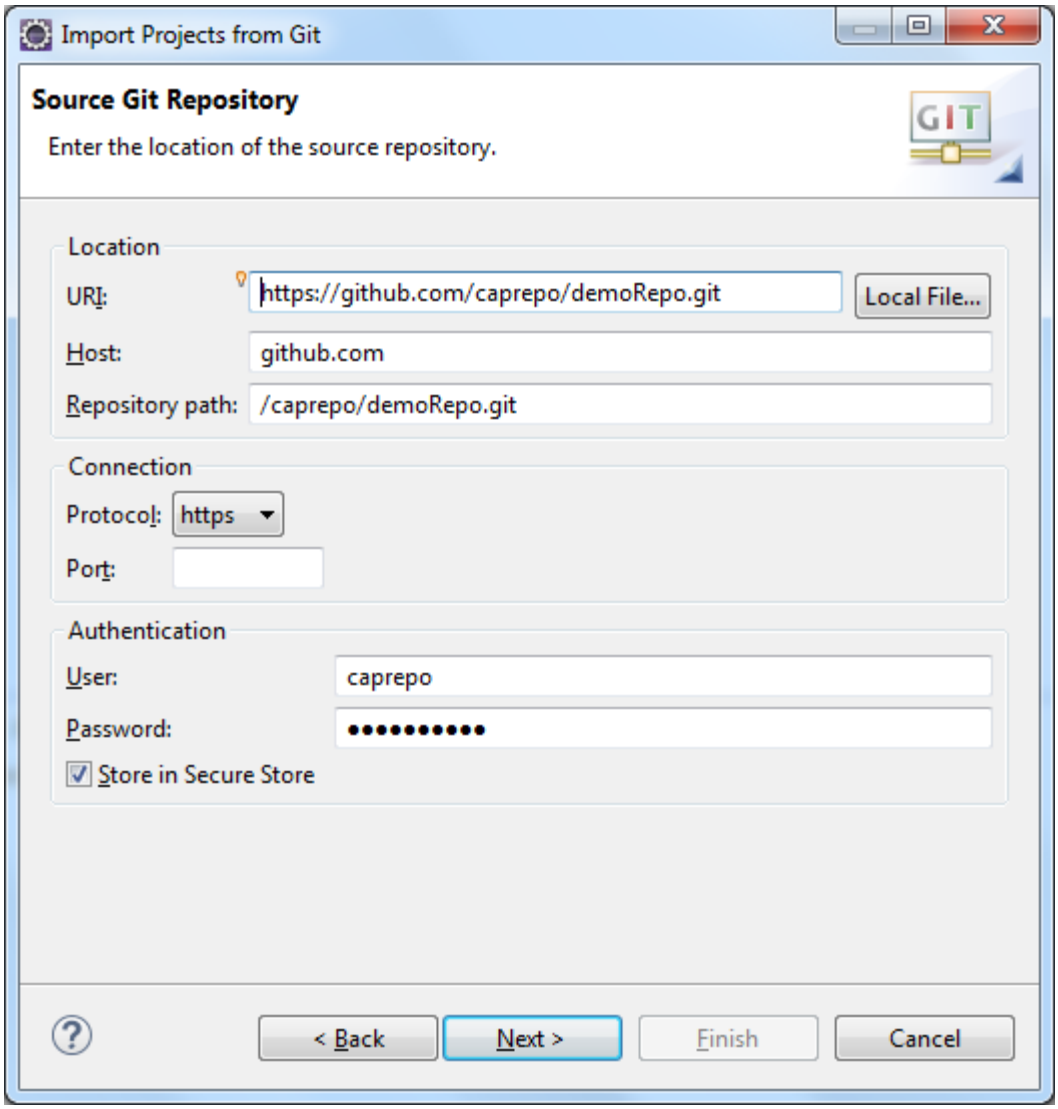
*In case in the current workspace if you have **MyJavaApplication** already, please remove the project. Then import the project in the workspace. If the same name of the project exists in the workspace, eclipse will not allow you to import the project.*

Lab 2:

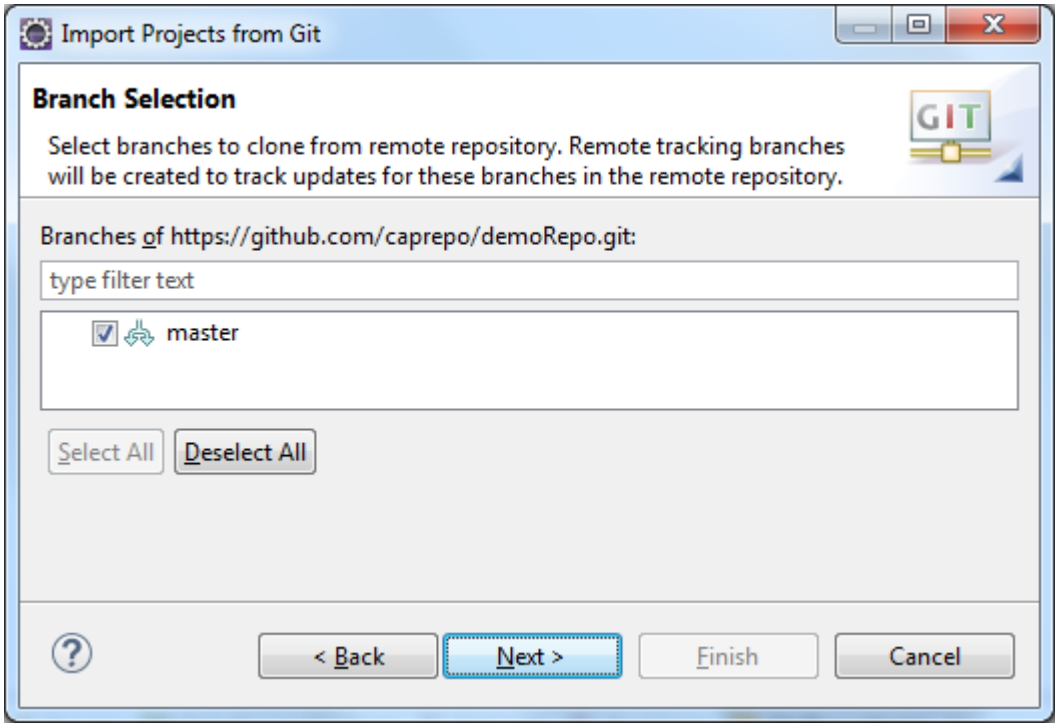
Check out the existing project in GitHub and import the project into Eclipse/STS IDE.

Steps:

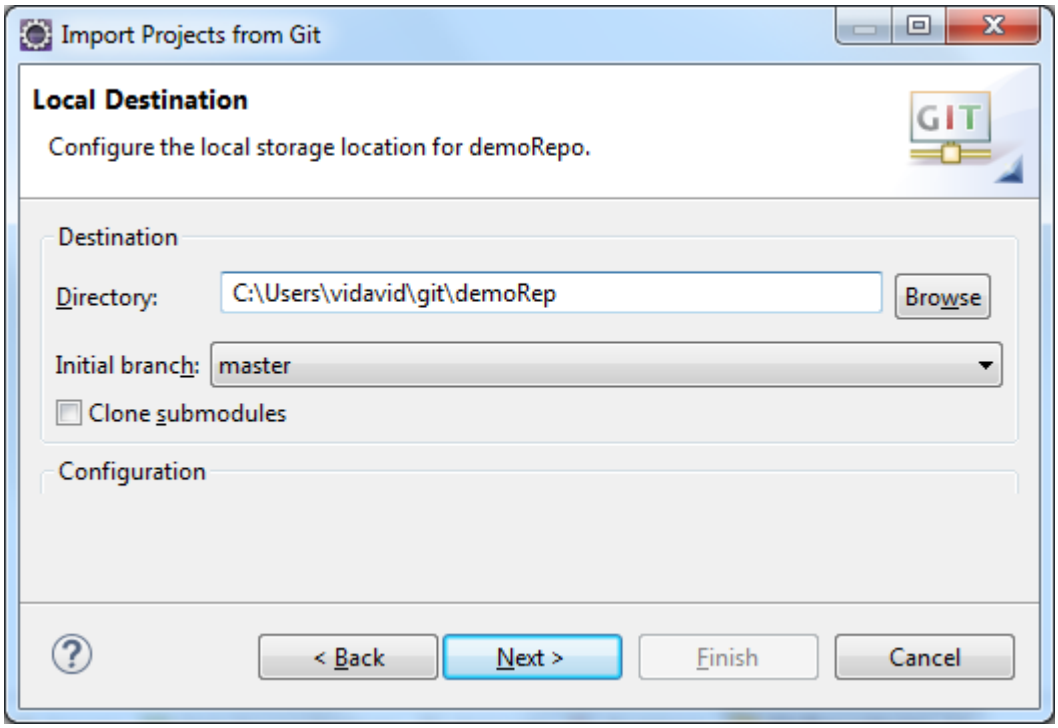
1. Open Eclipse IDE.
2. Go to File → Import
3. Select Git Category → **Projects from Git → Clone URI**
4. Select Paste your GitHub URL , from where you want to import the project.



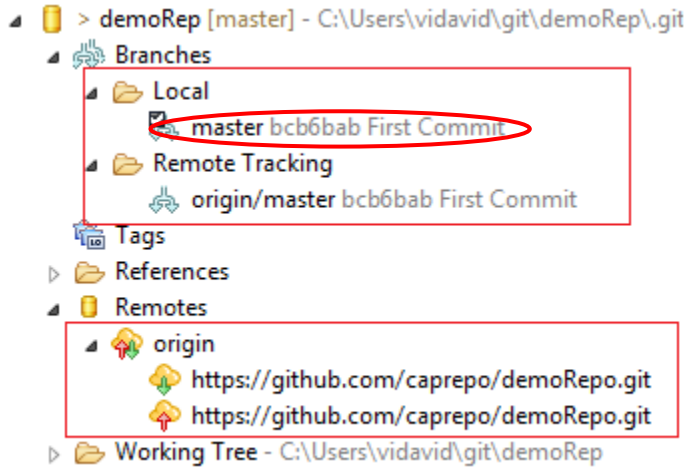
5. Click Next
6. Here select the master branch of the project to import.



7. Click next, in the next screen select your local repository to store the project:



8. Click Next → finish
9. Now you got the project from GitHub Repository. Please check your Git Repositories view, it will show all Remote Tracking, Remote and Branch details as shown below:



10. Now successfully we checked out the project from GitHub repository.
11. Try to modify the file, and give the commit with remote. See the changes in the GitHub.
12. For every commit, you could see git has generated one hashCode

Conclusion:

From the above example, we learnt how to check out existing project from GitHub Repository.

Note:

*In case in the current workspace if you have **MyJavaApplication** already, please remove the project. Then import the project in the workspace. If the same name of the project exists in the workspace, eclipse will not allow you to import the project.*

Lab 3:

Try the below commands in Eclipse – Git Repository. See the changes in your IDE.

Commands:

1. **Commit**
2. **Push**
3. **Merge**
4. **Pull**
5. **History**
6. **Disconnect**
7. **Rebase**

Conclusion:

From the above example we understand the different commands supported by Git.

Note:

Please open another IDE import the same project, and try to access the same file with different workspace. You can understand the conflict and rebasing structure.