GIT

LAB ASSIGNMENTS

Pre-Requisitions:

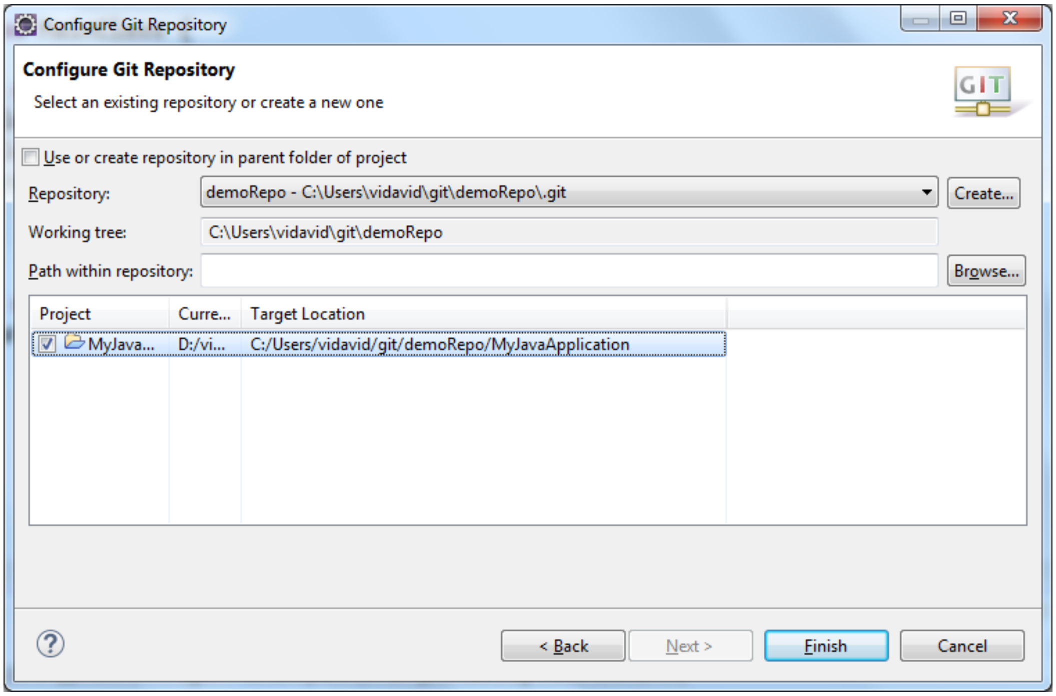
1. Create one account with GitHub (https://github.com/)
    a. Create one repository in the name of "**demoRepo**"
2. JDK 1.6 or higher
3. Git client should be installed. You can download GIT client from the below URL.
    a. https://git-scm.com/download/
4. Install EGIT plug-in in Spring Tool Suite / 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/Training2017/agile_dev)
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



Configure Git Repository

Select an existing repository or create a new one

□ Use or create repository in parent folder of project

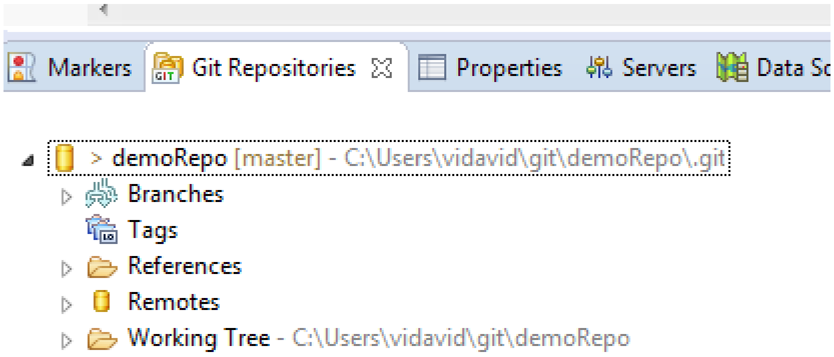
Repository: demoRepo - C:\Users\vidavid\git\demoRepo\.git

Working tree: C:\Users\vidavid\git\demoRepo

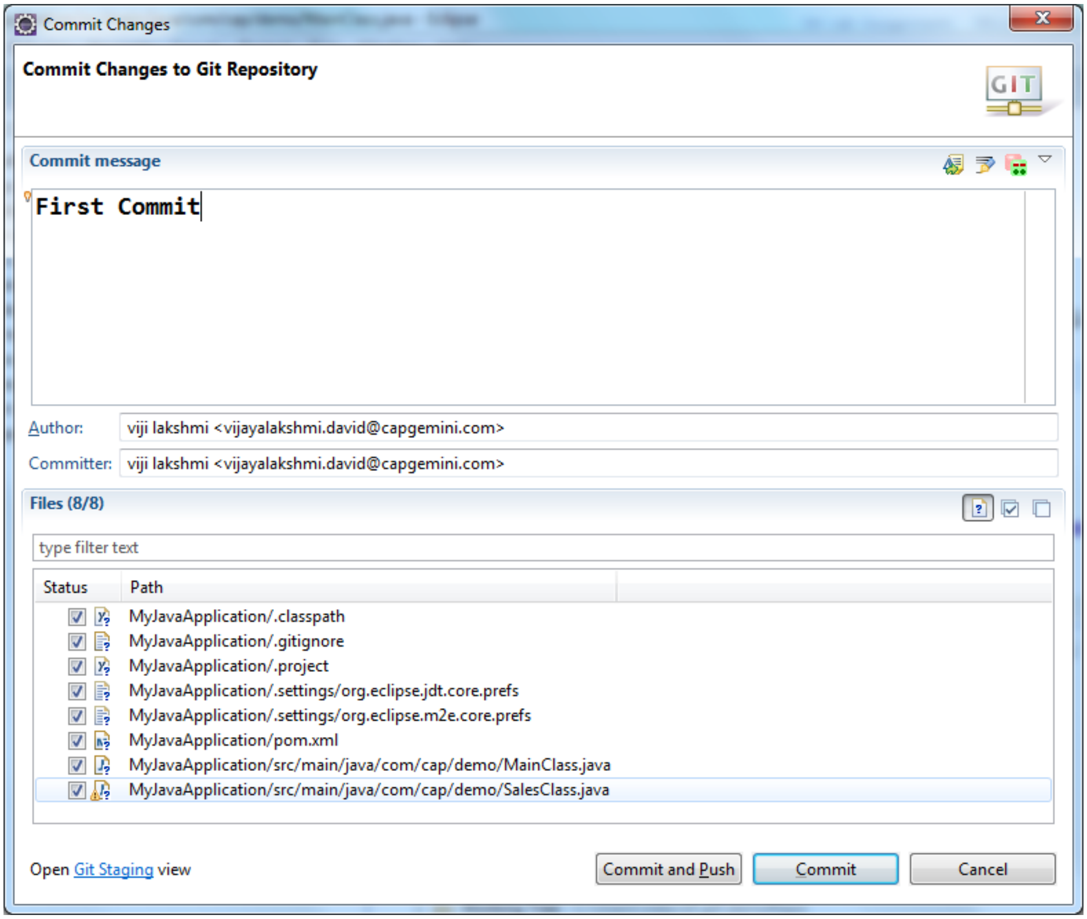
Path within repository:

Project	Curre...	Target Location
☑ MyJava...	D:/vi...	C:/Users/vidavid/git/demoRepo/MyJavaApplication

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


Markers
Git Repositories
Properties
Servers
Data Sc
demoRepo [master] - C:\Users\vidavid\git\demoRepo\.git
Branches
Tags
References
Remotes
Working Tree - C:\Users\vidavid\git\demoRepo

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.



First Commit

Author: viji lakshmi <vijayalakshmi.david@capgemini.com>

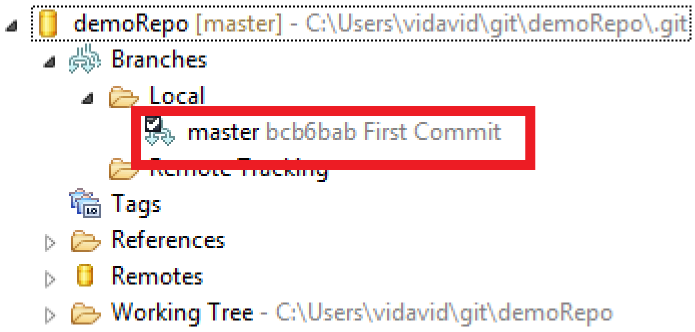
Committer: viji lakshmi <vijayalakshmi.david@capgemini.com>

Files (8/8)

type filter text

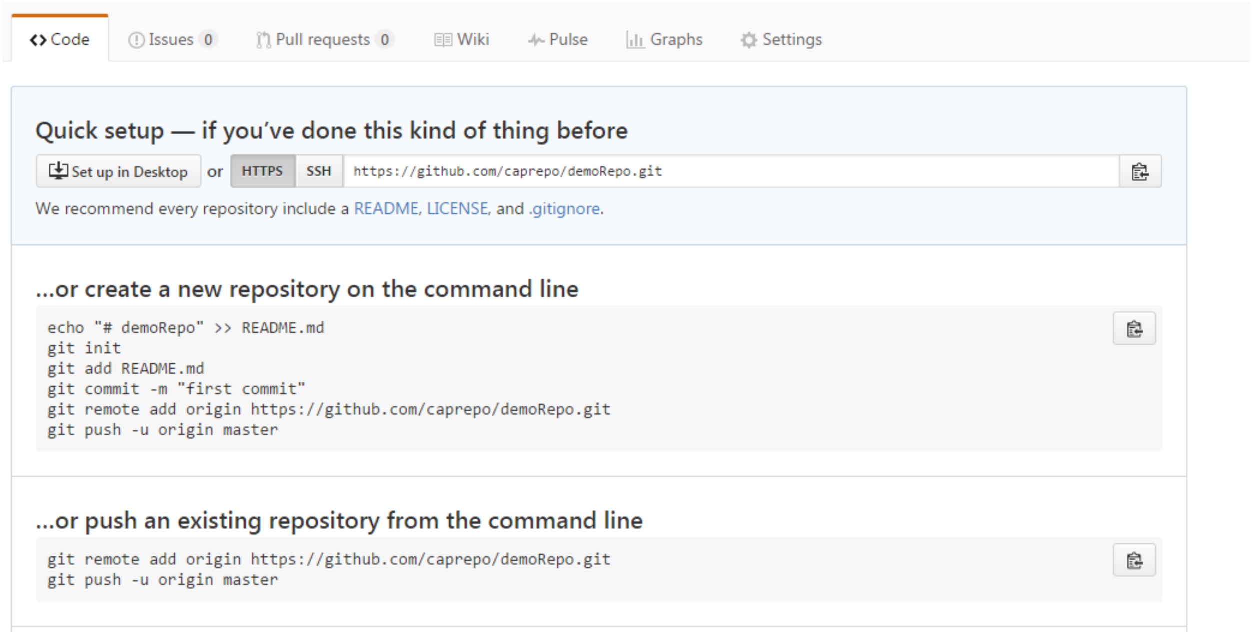
Status	Path
☑	MyJavaApplication/.classpath
☑	MyJavaApplication/.gitignore
☑	MyJavaApplication/.project
☑	MyJavaApplication/.settings/org.eclipse.jdt.core.prefs
☑	MyJavaApplication/.settings/org.eclipse.m2e.core.prefs
☑	MyJavaApplication/pom.xml
☑	MyJavaApplication/src/main/java/com/cap/demo/MainClass.java
☑	MyJavaApplication/src/main/java/com/cap/demo/SalesClass.java

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



demoRepo [master] - C:\Users\vidavid\git\demoRepo\.git
- Branches
  - Local
    - master bcb6bab First Commit
  - Remote Tracking
- Tags
- References
- Remotes
- Working Tree - C:\Users\vidavid\git\demoRepo

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



Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/caprepo/demoRepo.git

We recommend every repository include a README, LICENSE, and .gitignore.

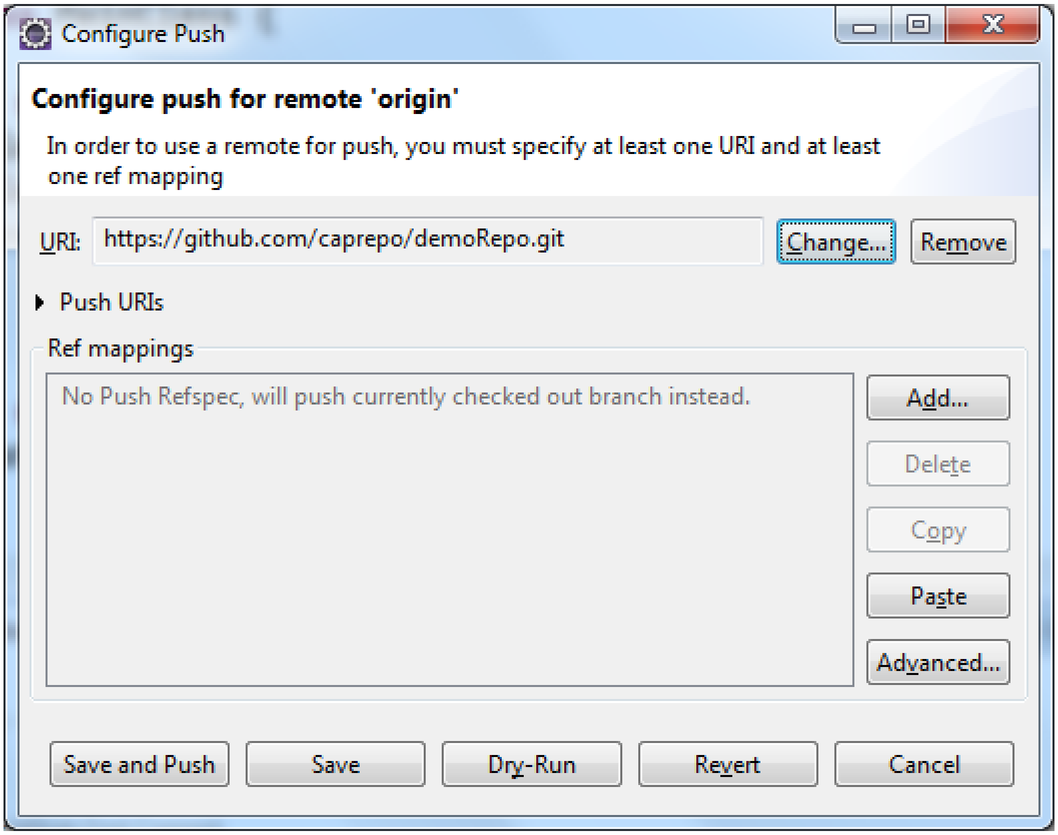
...or create a new repository on the command line

```
echo "# demoRepo" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/caprepo/demoRepo.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/caprepo/demoRepo.git
git push -u origin master
```

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



Configure Push

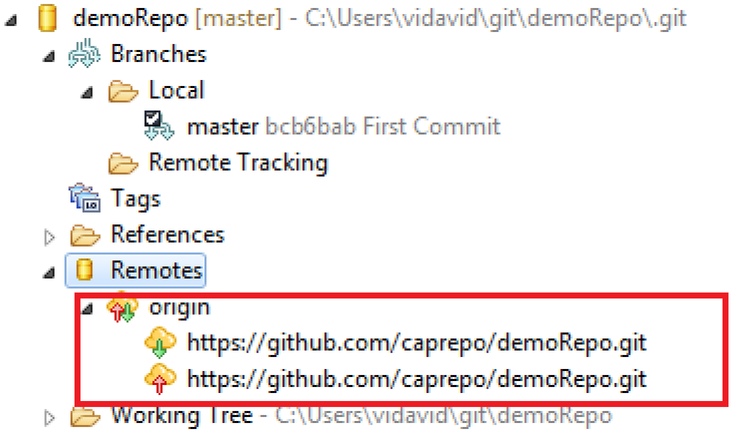
Configure push for remote 'origin'

In order to use a remote for push, you must specify at least one URI and at least one ref mapping

URI: https://github.com/caprepo/demoRepo.git

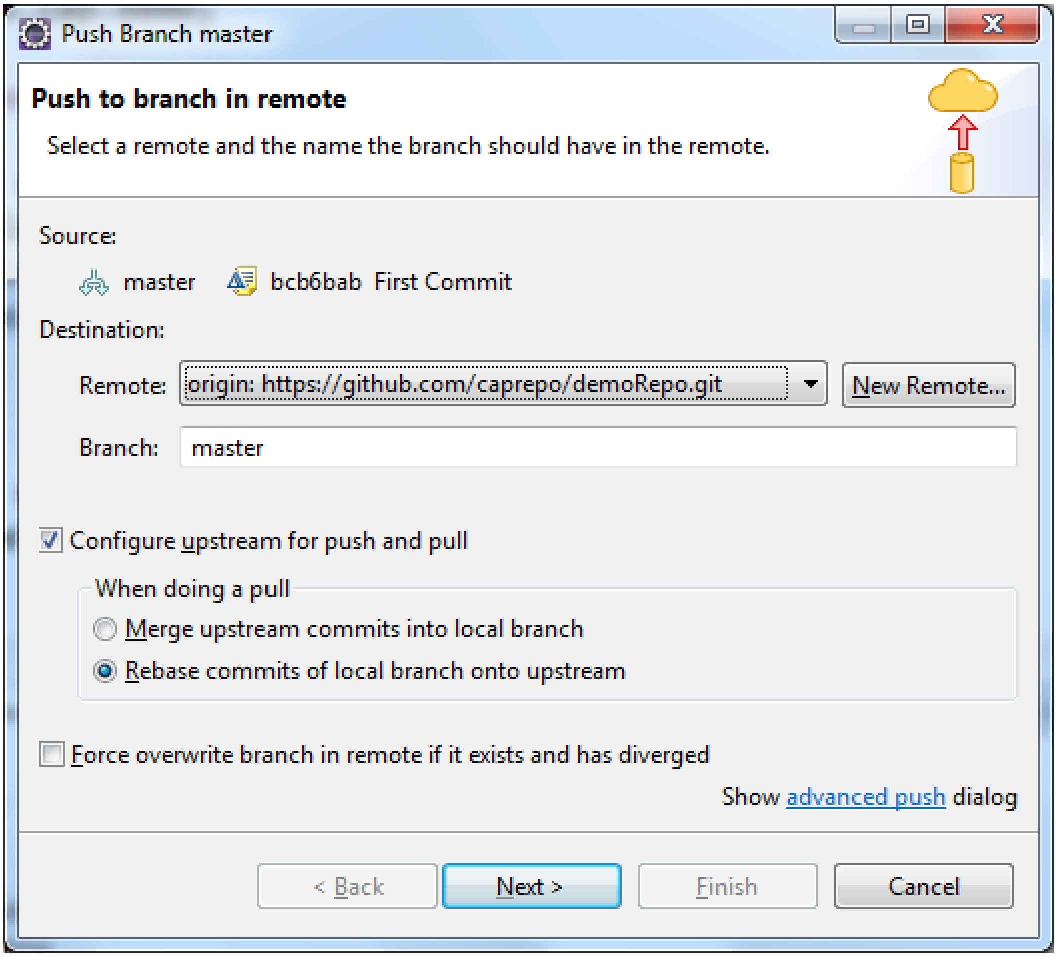
No Push Refspec, will push currently checked out branch instead.

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



```
demoRepo [master] - C:\Users\vidavid\git\demoRepo\.git
  Branches
    Local
      master bcb6bab First Commit
    Remote Tracking
  Tags
  References
  Remotes
    origin
      https://github.com/caprepo/demoRepo.git
      https://github.com/caprepo/demoRepo.git
  Working Tree - C:\Users\vidavid\git\demoRepo
```

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



Push Branch master

Push to branch in remote

Select a remote and the name the branch should have in the remote.

Source:

master bcb6bab First Commit

Destination:

Remote: origin: https://github.com/caprepo/demoRepo.git

Branch: master

☑ Configure upstream for push and pull

When doing a pull

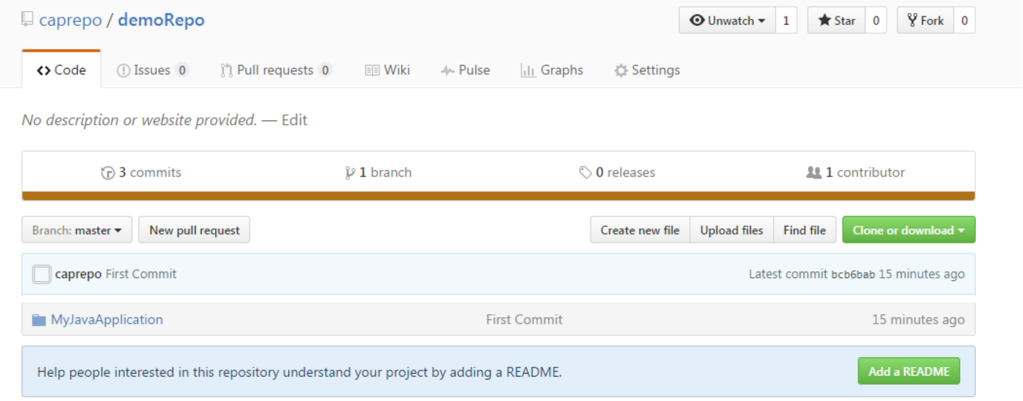
○ Merge upstream commits into local branch

● Rebase commits of local branch onto upstream

☐ Force overwrite branch in remote if it exists and has diverged

Show advanced push dialog

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:



caprepo / demoRepo

No description or website provided. — Edit

3 commits | 1 branch | 0 releases | 1 contributor

Branch: master

caprepo First Commit — Latest commit bcb6bab 15 minutes ago

MyJavaApplication — First Commit — 15 minutes ago

Help people interested in this repository understand your project by adding a README.

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:

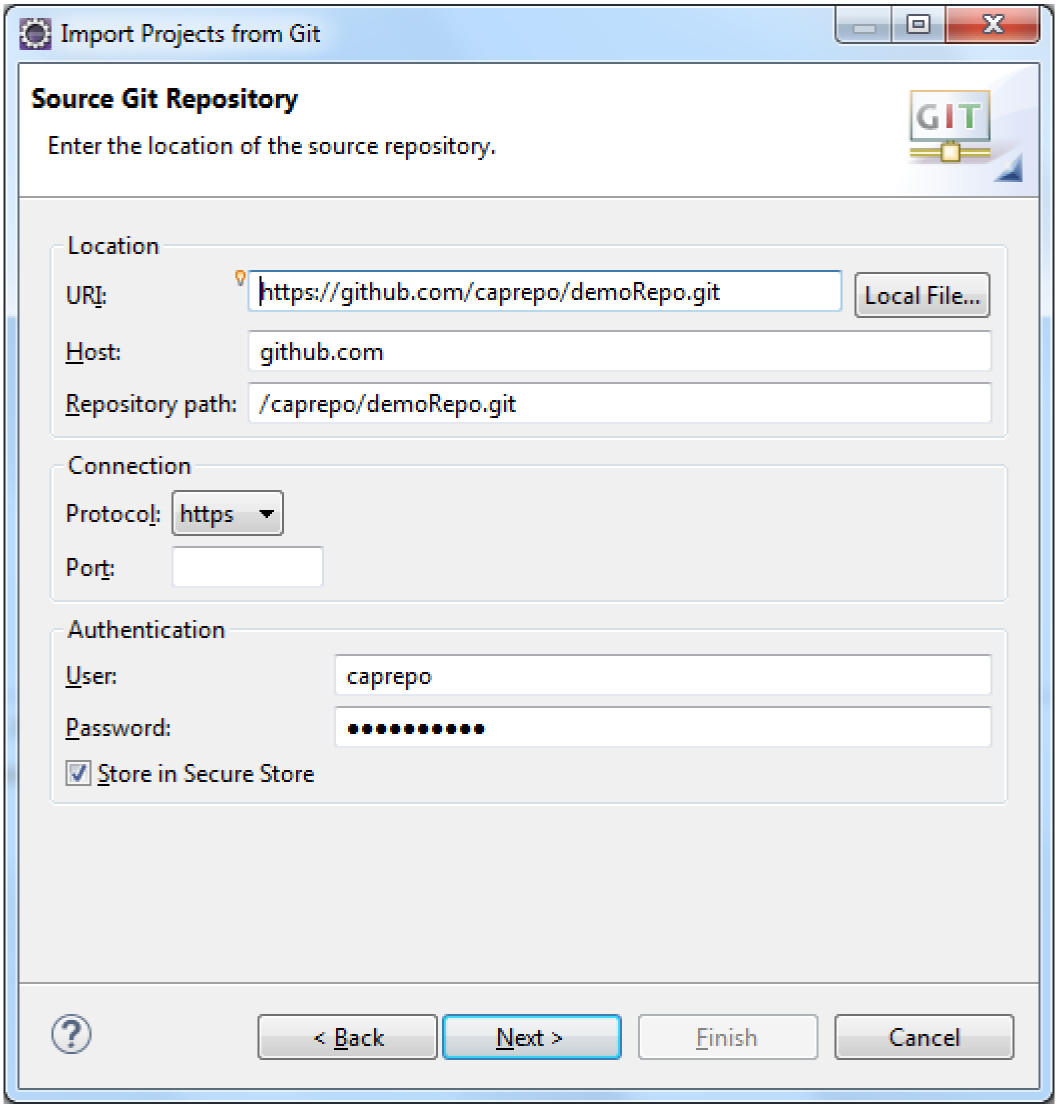
*In case in the current workspace if you have **MyJavaApplication** already, please remove the project. Then import the project in the workspace. If the same name of the project exists in the workspace, eclipse will not allow you to import the project.*

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.



Source Git Repository

Enter the location of the source repository.

Location

URI: https://github.com/caprepo/demoRepo.git

Host: github.com

Repository path: /caprepo/demoRepo.git

Connection

Protocol: https

Port:

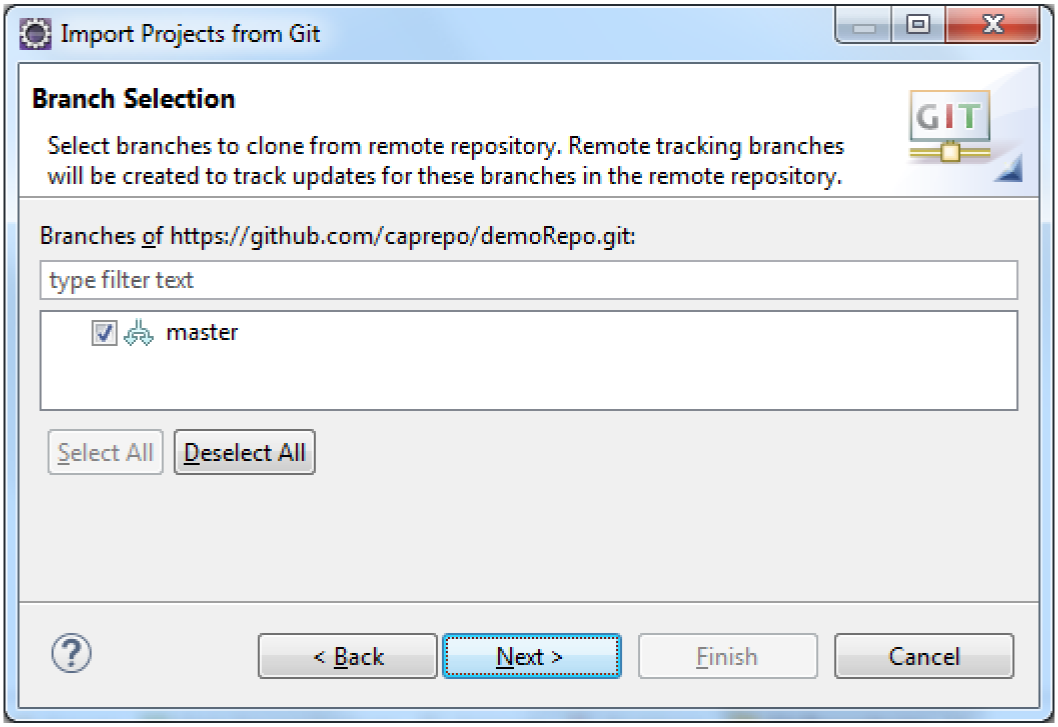
Authentication

User: caprepo

Password: ●●●●●●●●●●●

☑ Store in Secure Store

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



Import Projects from Git

Branch Selection

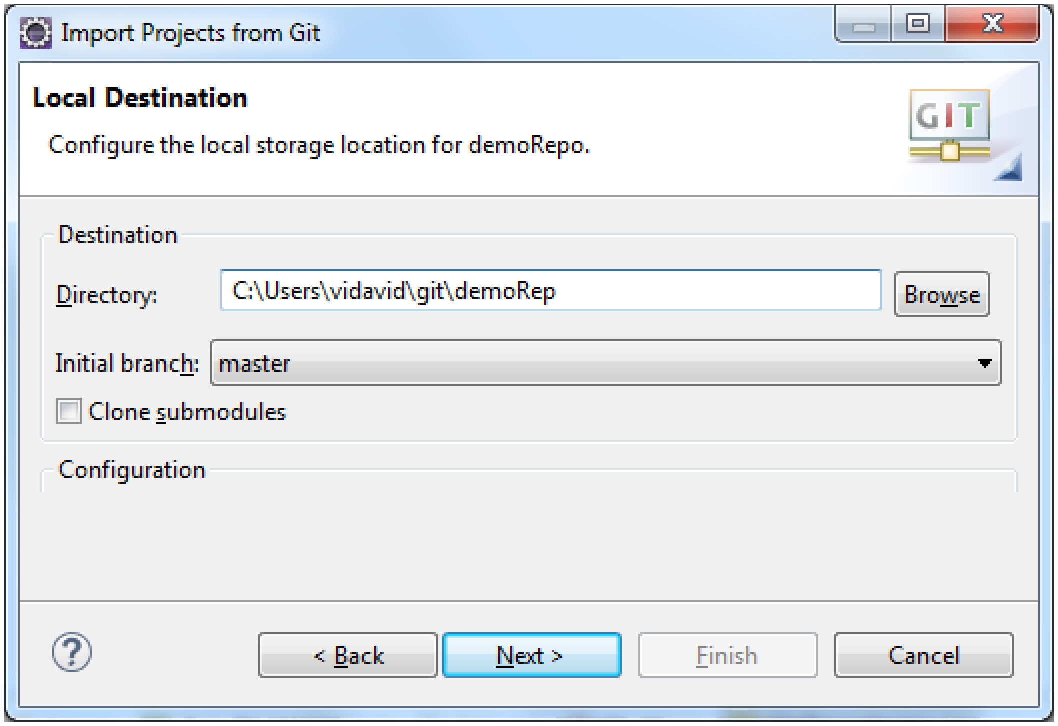
Select branches to clone from remote repository. Remote tracking branches will be created to track updates for these branches in the remote repository.

Branches of https://github.com/caprepo/demoRepo.git:

type filter text

☑ master

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



Import Projects from Git

Local Destination

Configure the local storage location for demoRepo.

Destination

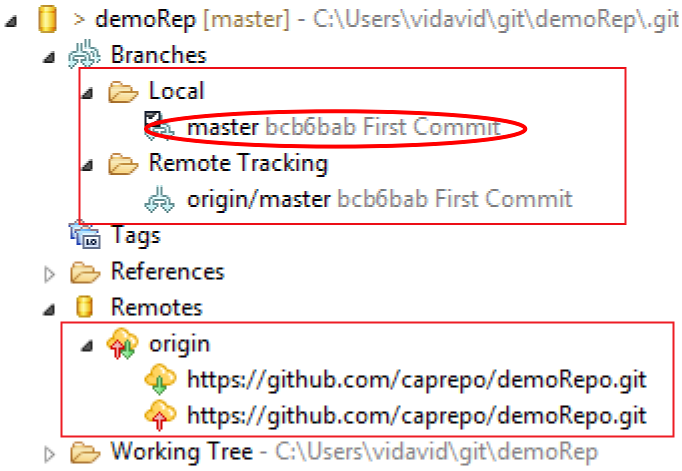
Directory: C:\Users\vidavid\git\demoRep

Initial branch: master

□ Clone submodules

Configuration

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:



```
> demoRep [master] - C:\Users\vidavid\git\demoRep\.git
  Branches
    Local
      master bcb6bab First Commit
    Remote Tracking
      origin/master bcb6bab First Commit
  Tags
  References
  Remotes
    origin
      https://github.com/caprepo/demoRepo.git
      https://github.com/caprepo/demoRepo.git
  Working Tree - C:\Users\vidavid\git\demoRep
```

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.