# Git Setup –

Git is a free and open source distributed version control system. There are other SCM (Software Configuration Management) tools like Subversion(SVN), CVS, Perforce and ClearCase but these days Git has taken precedence over these being a lightweight footprint.

Following document gives detailed steps on Git set up and also covers 2 scenarios where…

1. You are uploading your Local Project to GitHub for the first time
2. You are downloading a remote repository from GitHub for the first time

# Uploading Local Project (Repository) to GitHub

# Steps –

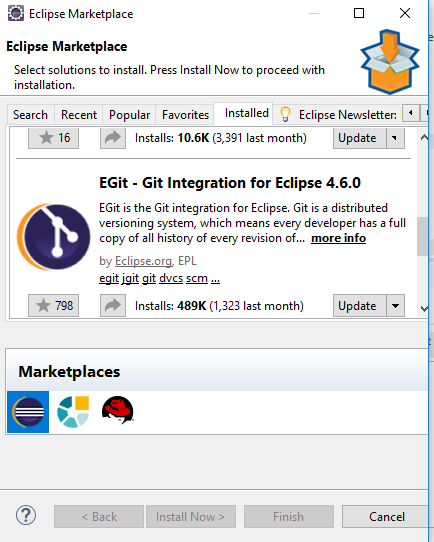
1. Install gitbash
2. Install EGit - Git Integration for Eclipse
3. Create a GitHub account
4. Move your local repository (project) to GitHub
5. Convert your Eclipse project into Git project in Eclipse
6. Upload (Push) your local changes to GitHub remote repository through Eclipse
7. Pull (download) GitHub remote repository changes to your local repository in Eclipse

## Install gitbash –

1. Link to download gitbash - <https://git-scm.com/>
2. Take all defaults while installing gitbash

## Install EGit- Git Integration for Eclipse

1. Go to Eclipse Marketplace
2. Search for EGit and click Install in front of EGit – Git Integration for Eclipse *ver*

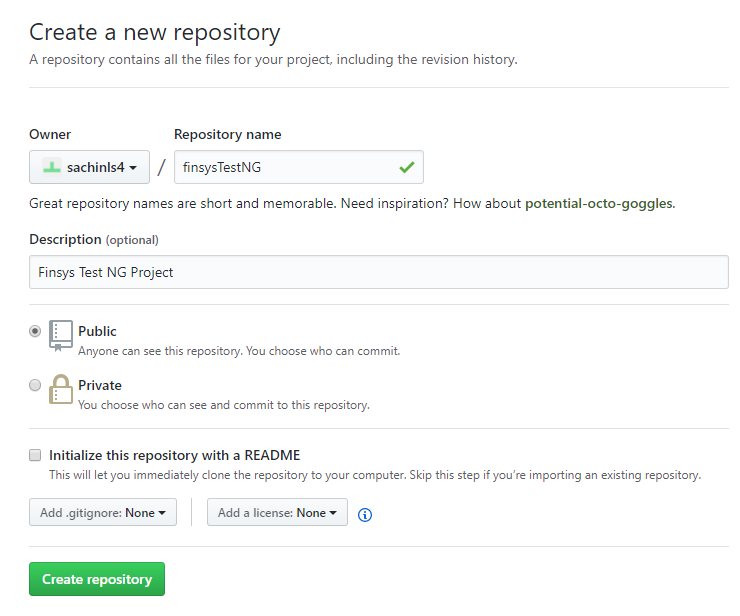


## Creating a GitHub account –

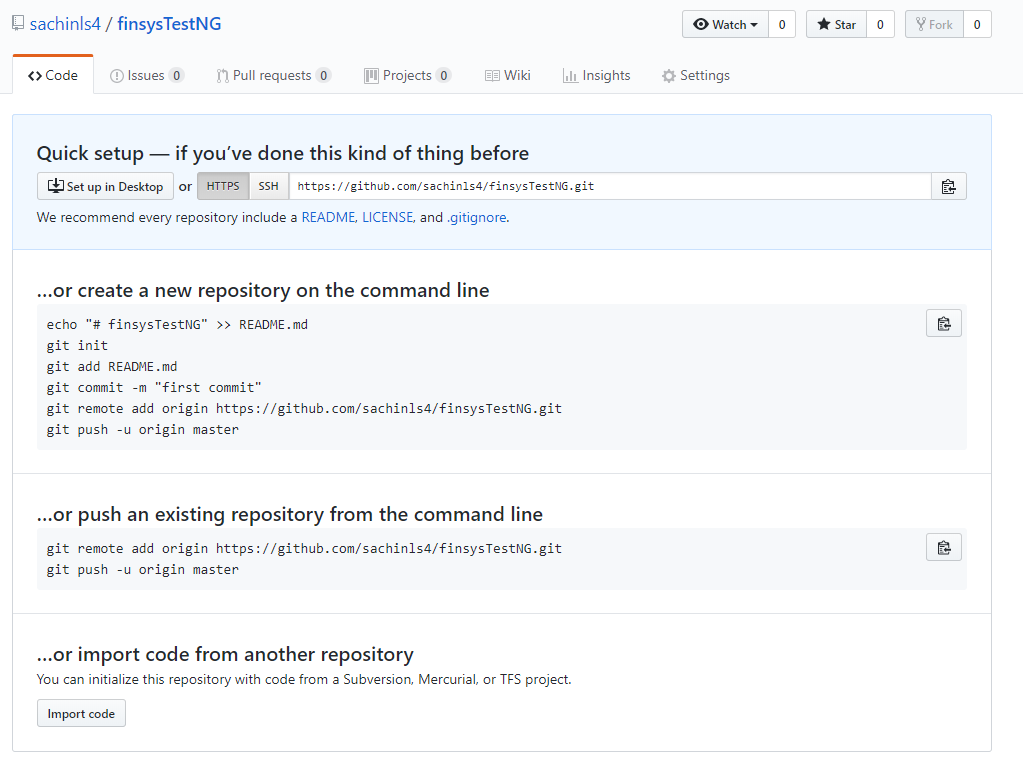
1. Open <https://github.com/>
2. Click on Sign up link on top right corner of the screen
   1. Fill a unique username
   2. Fill in your Email address
   3. Chose a password and click on Create an account button
3. On the next Step (Step 2) select Unlimited public repositories for free radio button
4. Click Continue button
5. On Step 3 click Submit button

Note: It’s not mandatory to select answers to the given questions on this page

1. On the next screen click on Start a project button
2. It’ll ask you to verify your email address.
3. Go to your email inbox and click on Verify email address link
4. Clicking on this link will take you to a page where you can name your repository, put a name for your repository and if you want you can put some Description, keep the Public radio button selected and click on Create repository button

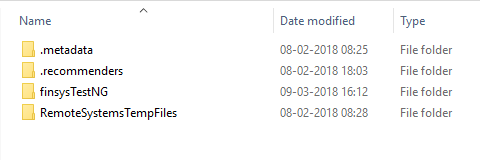


1. On the next screen you’ll see a link for your newly created repository, see following screen shot. Note down the URL, we’ll need it for pushing your local project (local repository) to the GitHub in later steps in this document.

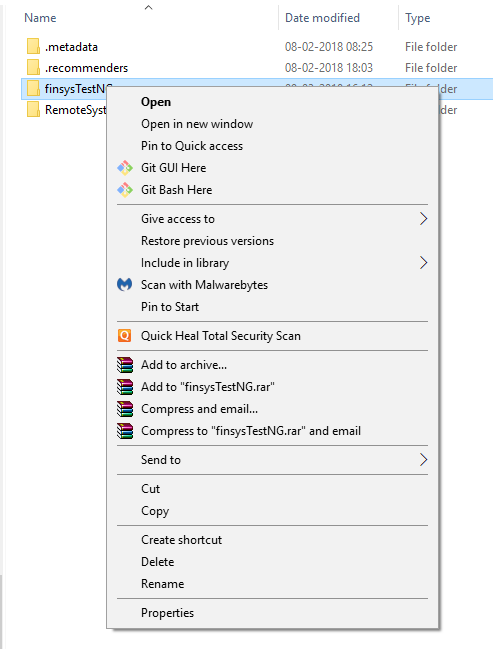


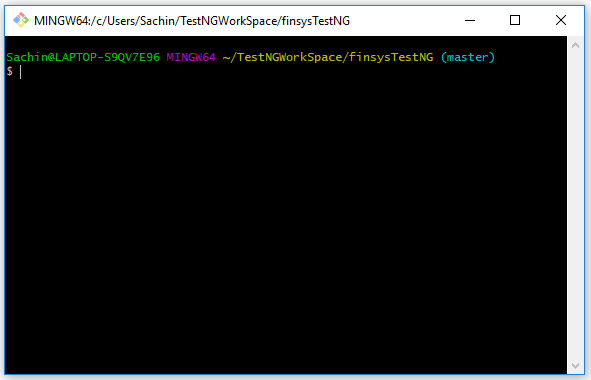
# Move your local repository (project) to GitHub

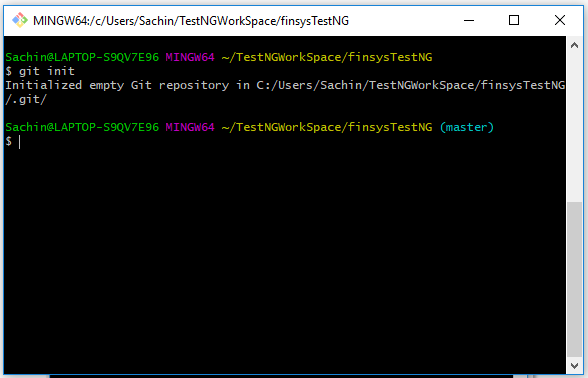
1. Navigate to your directory where your project lies like in following screen shot, finsysTestNG directory is the project

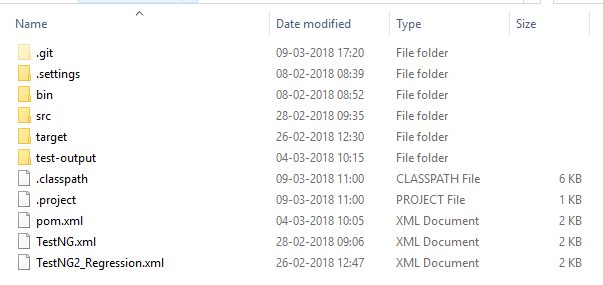


1. Right click to see Git GUI Here and Git Bash Here menu options on the right click menu



1. Click on Git Bash Here, you’ll get a prompt similar to command prompt called GitBash Terminal
2. We need to initialize the local directory as a Git repository. Type in *git init* in this window and hit Enter, this will initialize the local repository as Git repository. Meaning it’ll add a .git directory into your local project, see following screen shots.





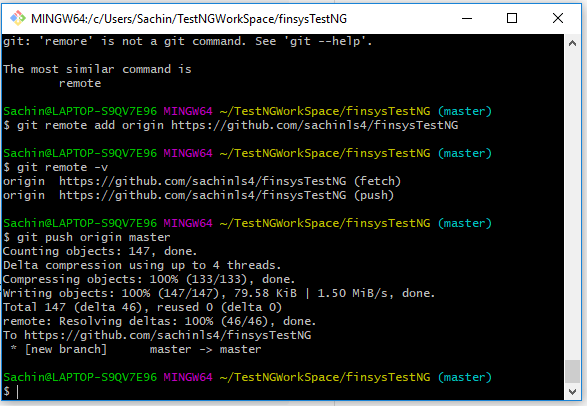
1. In the gitbash terminal window, type in *git add . (*the dot(.) with space is required) and hit Enter. This will stage the files in your project for first commit
2. Then give a command *git commit -m ‘First commit’* and hit Enter.This will commit your project files.
3. Copy your remote repository URL field from your GitHub repository that we created in above steps and type in following commandin the GitBash terminal *– git remote add origin <remote repository URL>* for example*– git remote add origin* [*https://github.com/sachinls4/finsysTestNG*](https://github.com/sachinls4/finsysTestNG)and hit Enter

Note: it’ll ask you your GitHub User Name and Password

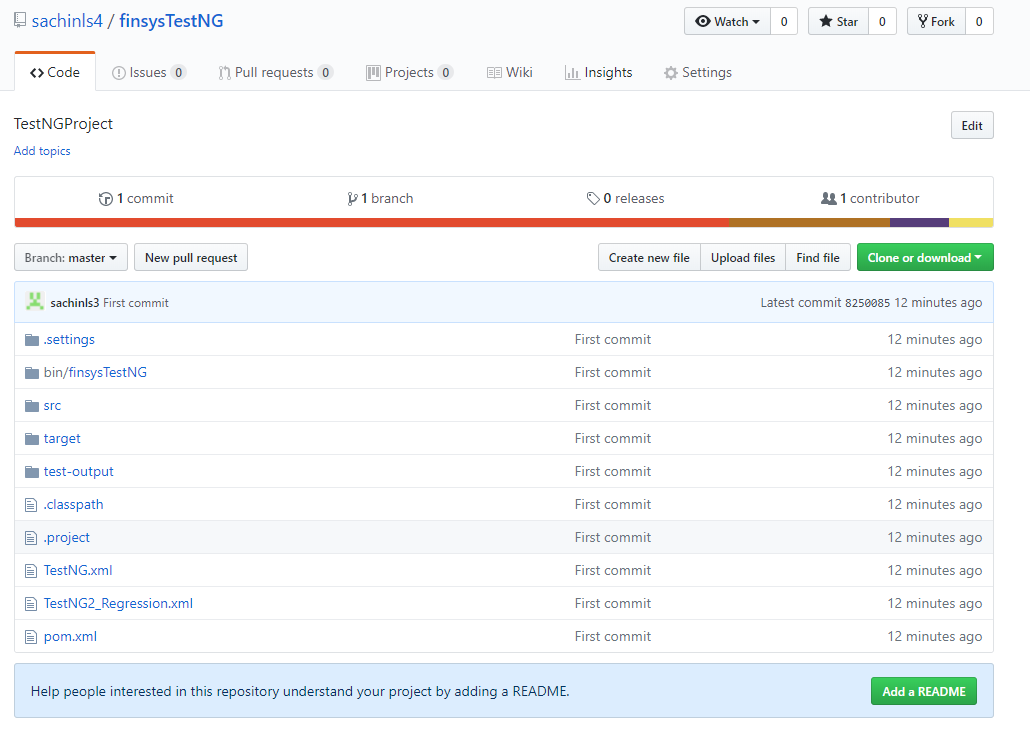
7a. Type following command and hit enter

Git push origin master

After #7a step, the terminal window will look like following…

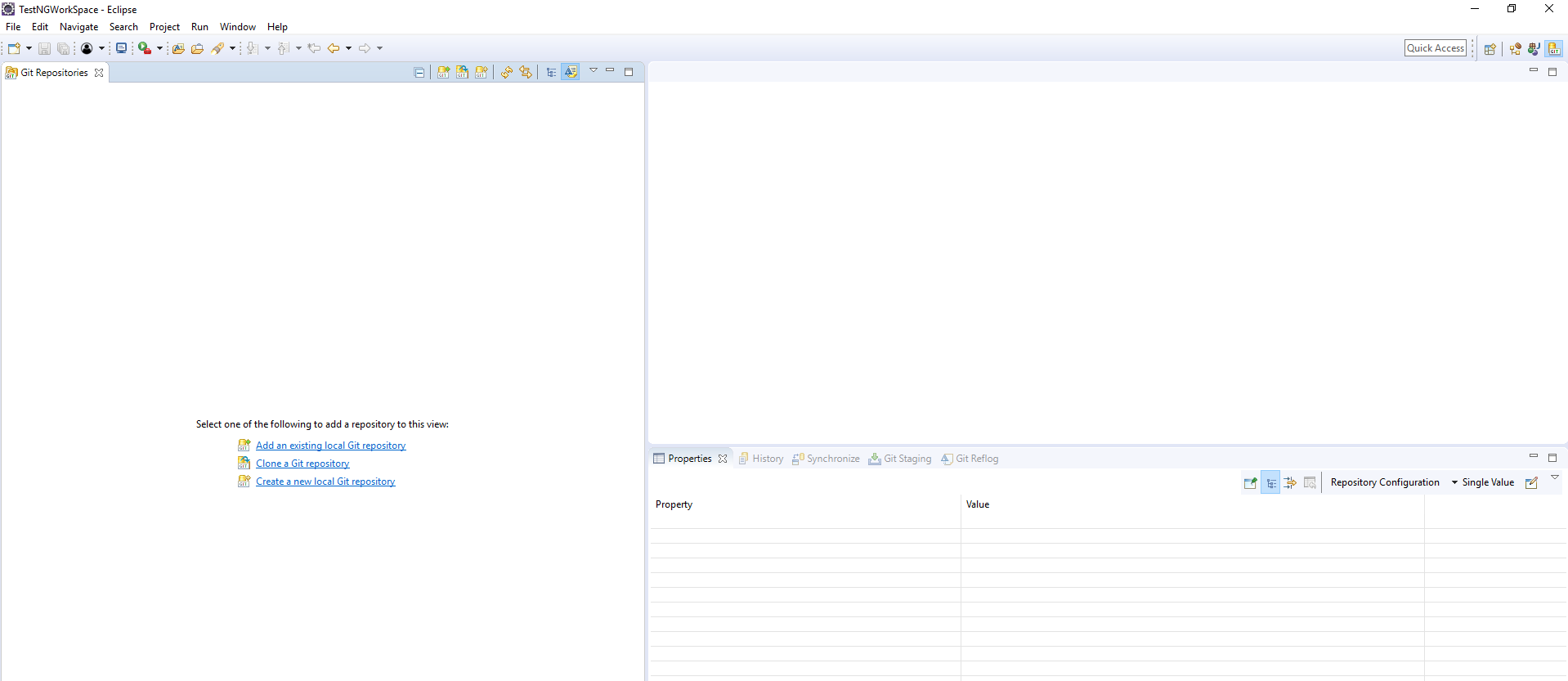


1. Refresh your GitHub page, and you’ll see your project uploaded, see following screen shot –

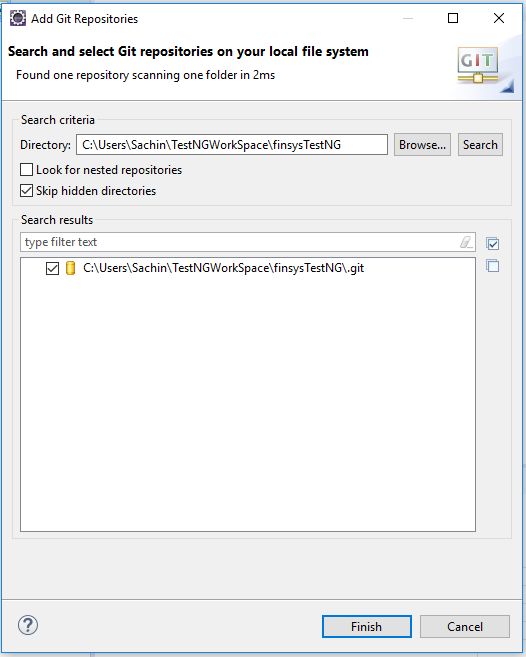


# Convert your Eclipse project into Git project in Eclipse

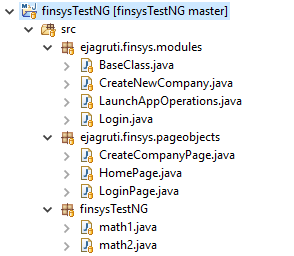
1. Open Eclipse with the Workspace where the project lies which we just moved to GitHub
2. Open Git perspective in Eclipse. Here you’ll see option to Add an existing local Git repository in the left-hand side screen. See following screen shot…



1. Click on Add an existing local Git repository link which will open Add Git Repositories window. Put in the Directory path where your project is saved locally and click on Search button which will show list your project converted as Git repository. Screen shot below –



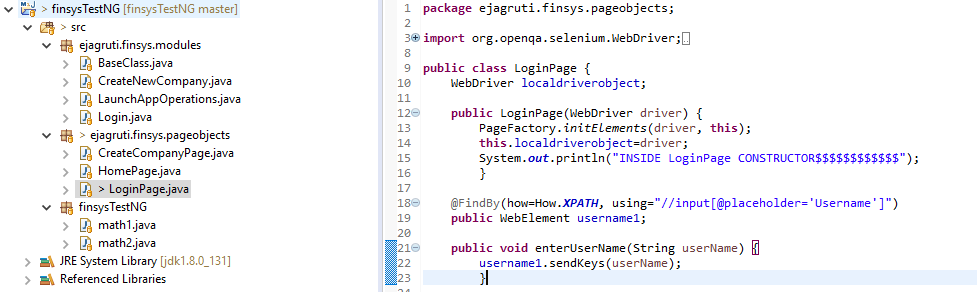
1. Clicking Finish will convert your Eclipse workspace project as a Git repository project in Eclipse, the indication of this is that the project hierarchy will have a small server sign for each folder, file in it which is a Git artefact now i.e. taking part in syncing with remote repository.

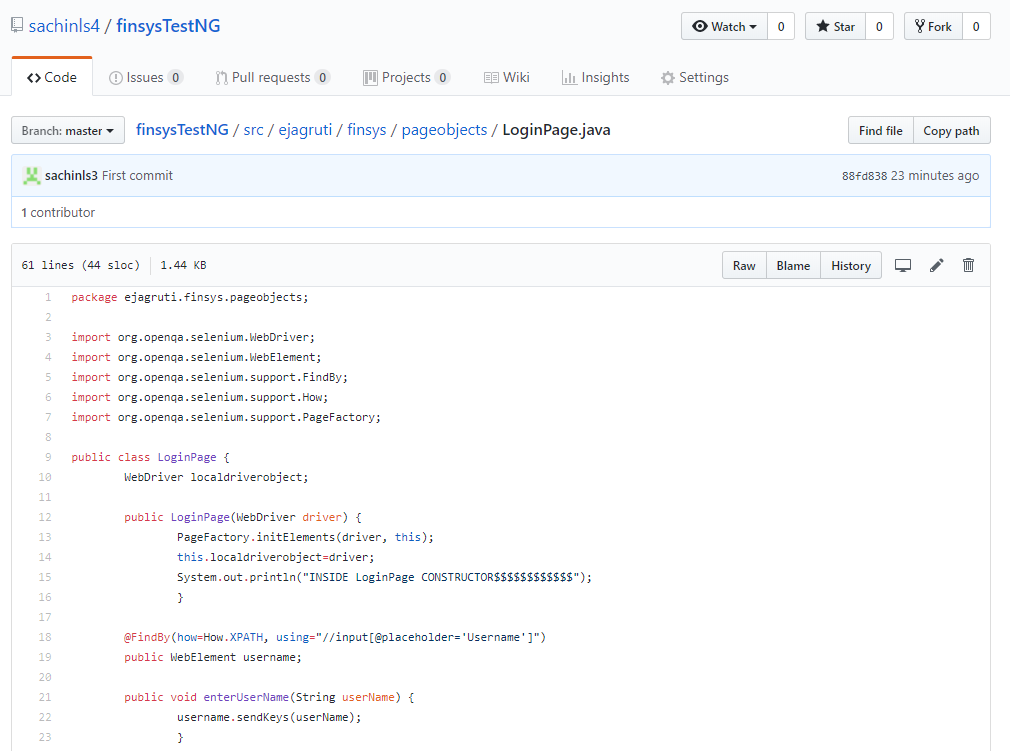


Your Eclipse workspace is ready to push and pull changes from GitHub

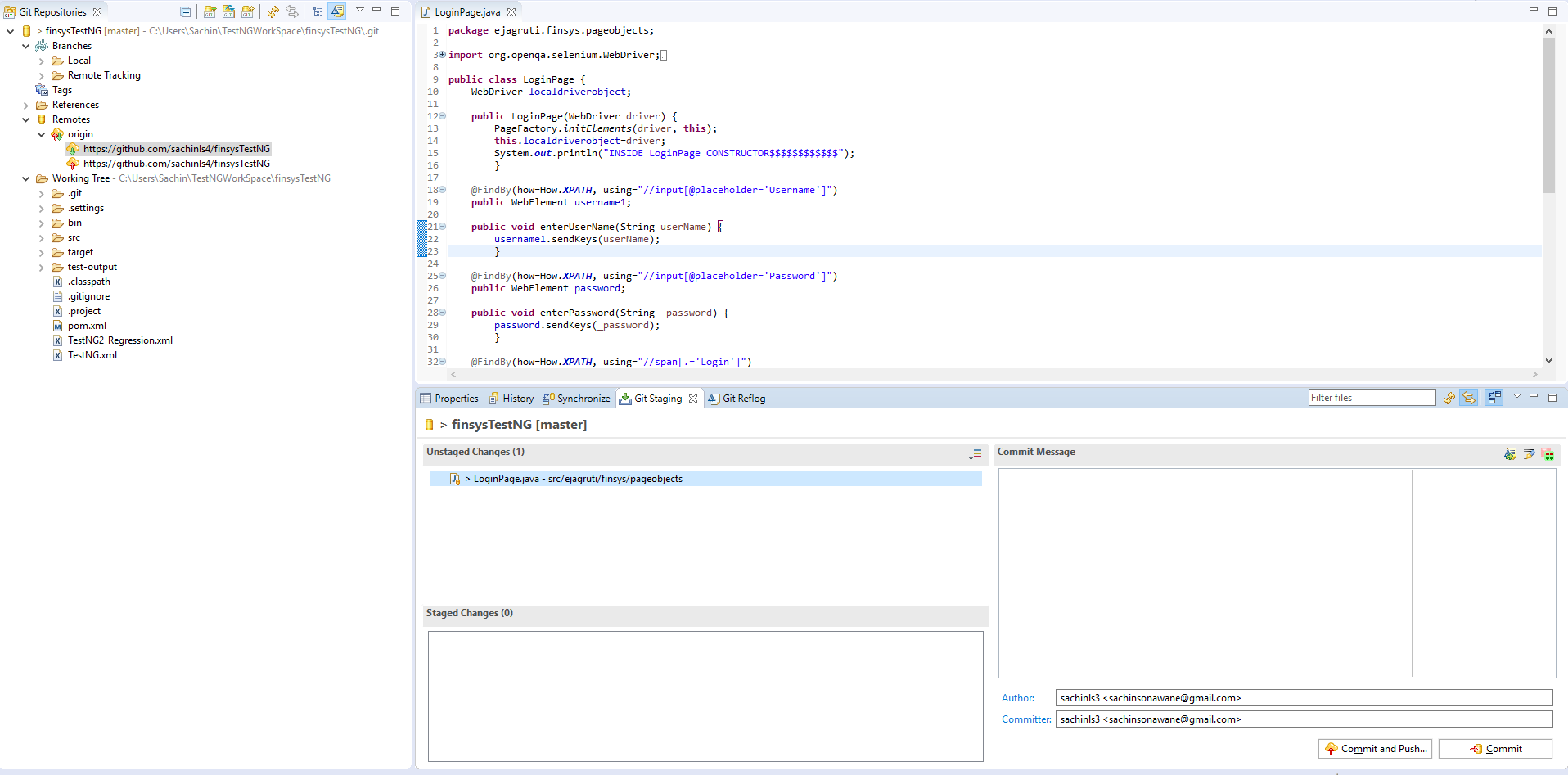
# Upload (Push) your local changes to GitHub remote repository through Eclipse

If you make changes to your local repository then follow below steps to upload your changes to GitHub

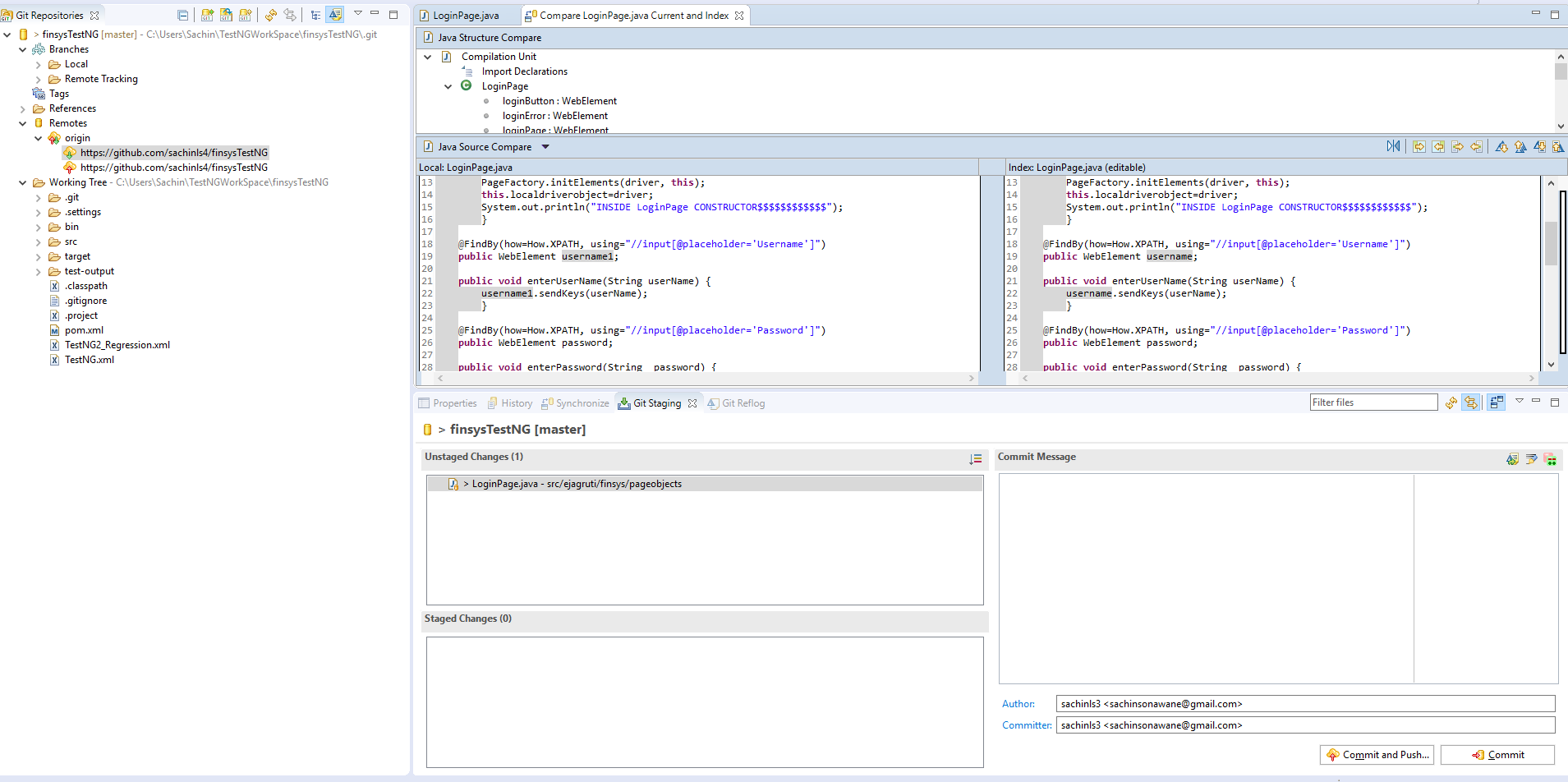
1. Here is a small change in LoginPage.java class where *username* variable is changed to *username1* at line number 19 and 22
2. Looking at the same file on GitHub you’ll see that the changes are not there, see following screenshot from GitHub at line number 19 and 22



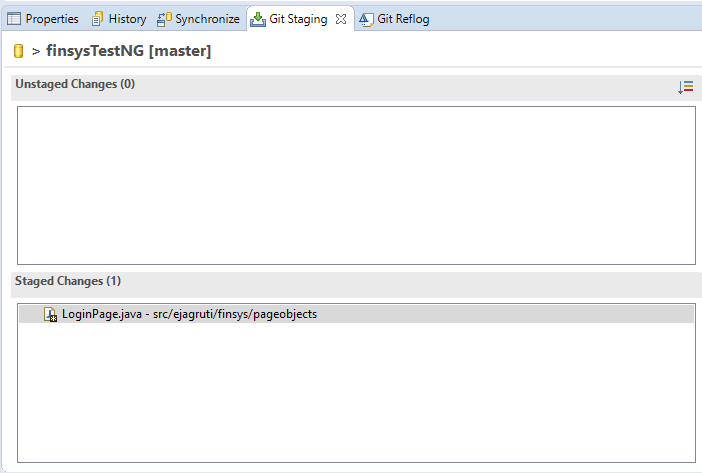
1. In Eclipse, open Git Perspective and then navigate to Window>Show Git>Git Staging. Resulting window will look like following with Git Staging pane opened with Unstaged Changes section having my recently made updates.



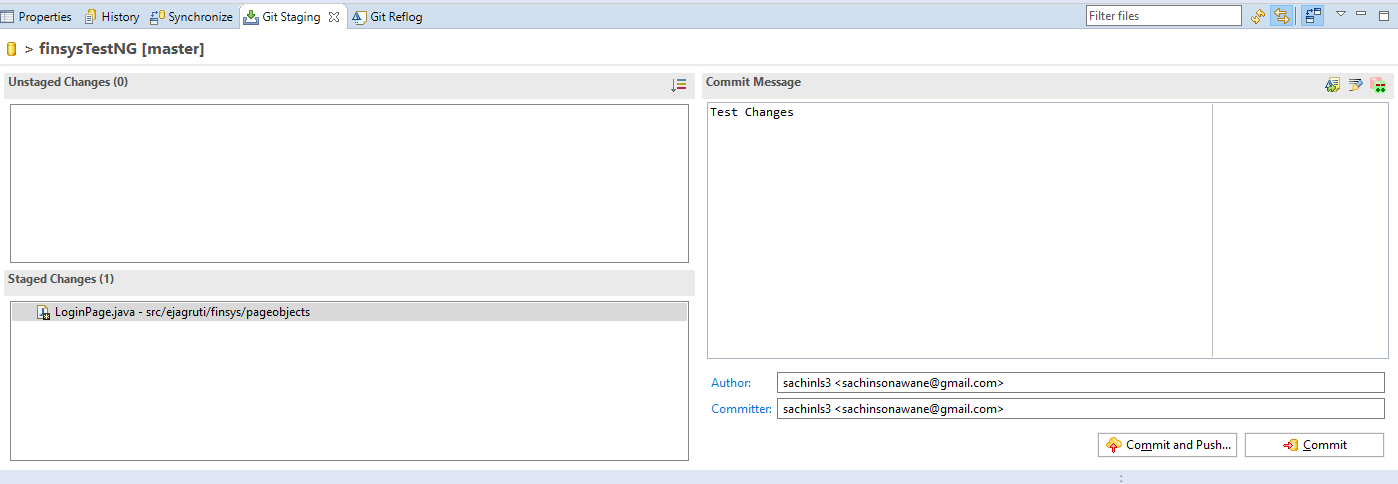
1. If you doble click on the Unstaged Changes you’ll get a resulting comparison screen which compares between Local and Remote repositories (here called Index on the right hand side) in following screen shot…



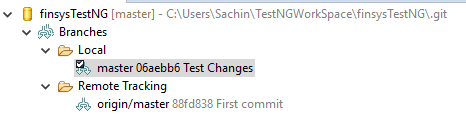
1. Drag your changes from Unstaged Changes pane down to Staged Changes pane –



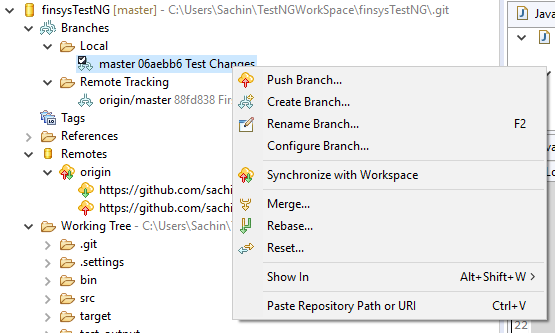
1. Add a comment (any) in Comment Pane (which is mandatory) and then click on Commit button (or you could click on Commit and Push… button as well, the steps will be different but easy and quick)



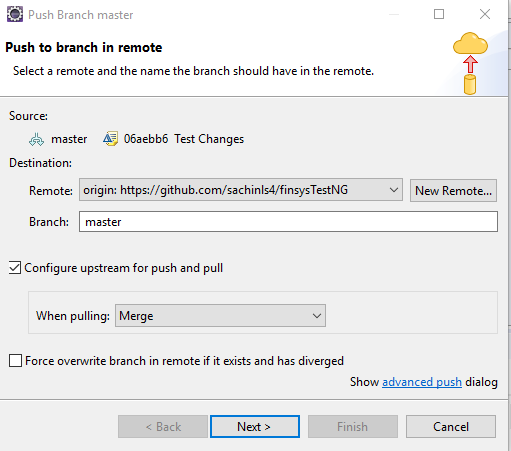
1. Now on the left hand pane, your changes (with comment) will be shown under Local Branch folder



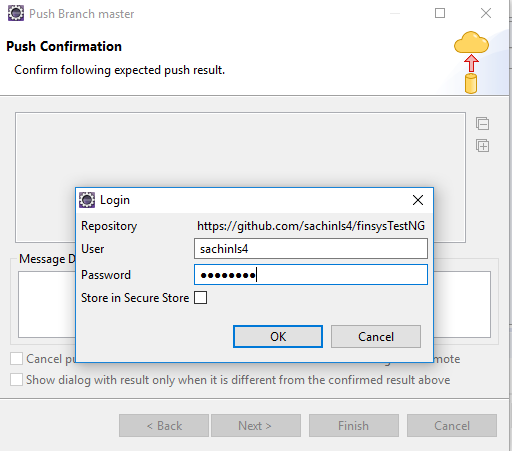
1. Right click on Local changes



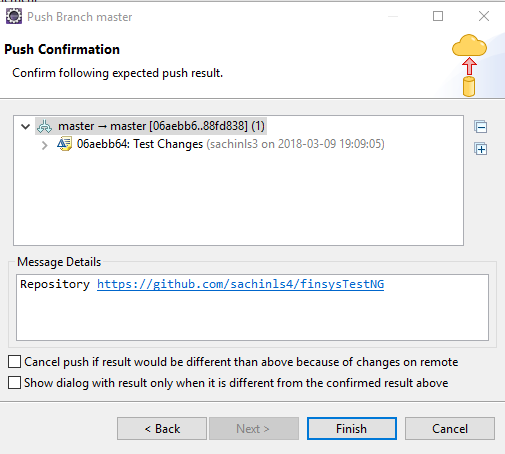
1. Click on Push Branch… which will open a popup with details of the remote branch where the changes are to be pushed…



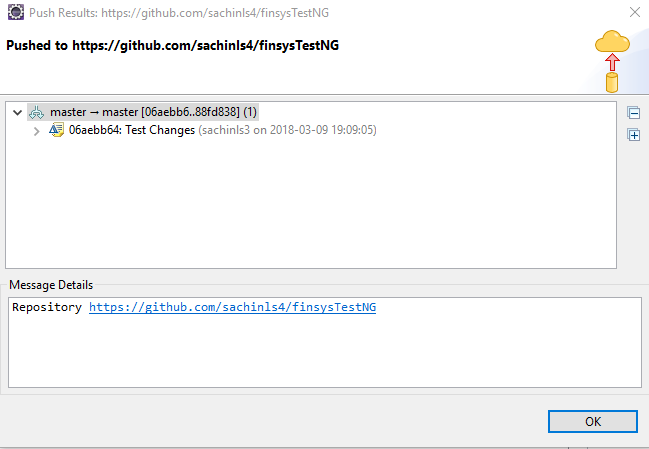
1. Click Next, here it’ll ask for User Name and Password, put your GitHub User Name and Password, click Ok



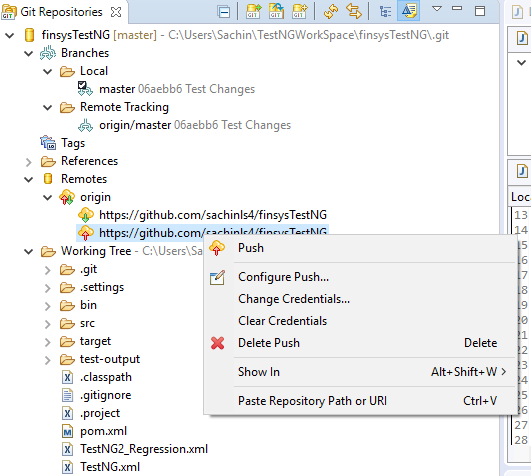
1. This will take you to the Finish screen, see following…



1. Clicking Finish will again ask for User Name and Password, fill in those and click Ok on the pop up.
2. This will again poup up another screen with details where you’ll have to click on Ok button



1. Click Ok and go to left hand side herarchy and to Remotes>origin path, right click on Push option (one with red up arrow), click on Push on the right click menu.



1. This will again prompt for User Name and Password. Click Ok after filling in User Name and Password.
2. This will take you to the final screen on Pushing your local changes to GitHub where you’ll see the master branch (in our case) says it’s up to date.



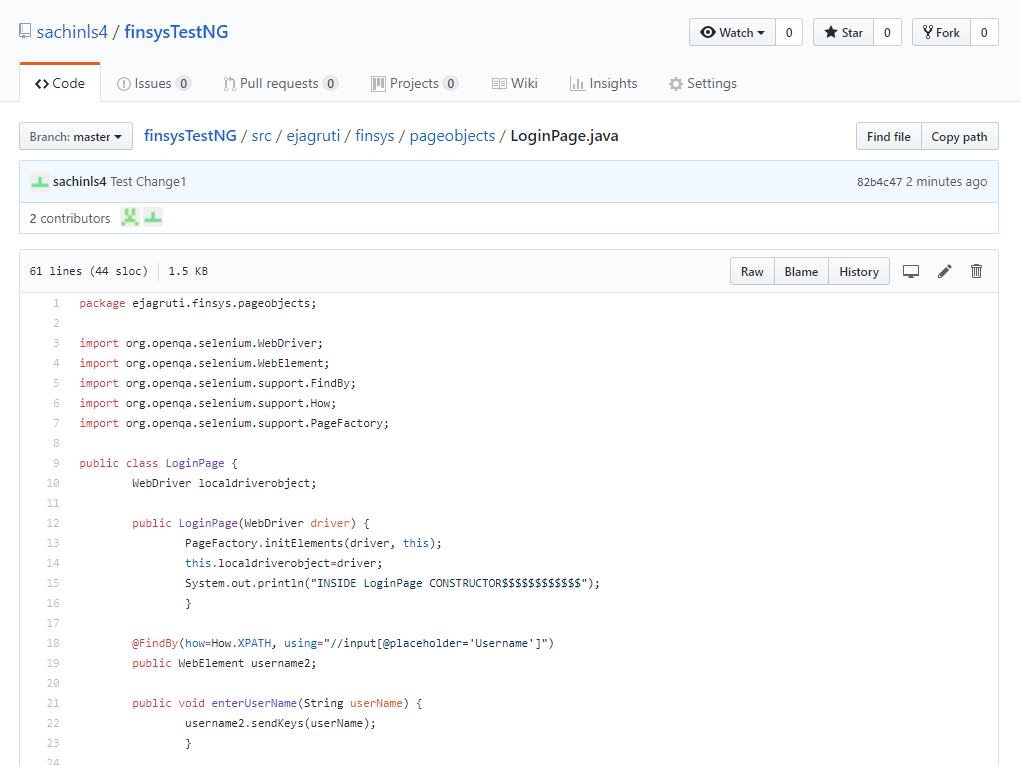
1. Now your Local Repository and Remote respository on GitHub are in sync. Following is the Remote respository screen shot after refreshing the page. Check line number 19 and 22 where the *username* is changed to *username1*

# 

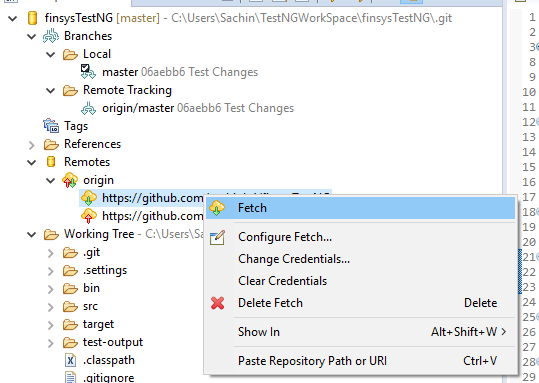
# Pull (download) GitHub remote repository changes to your local repository in Eclipse

Let us suppose we make changes to the remote repository on the same lines where we made above changes i.e. in line 19 and 22 from earlier *username1* to *username2*.

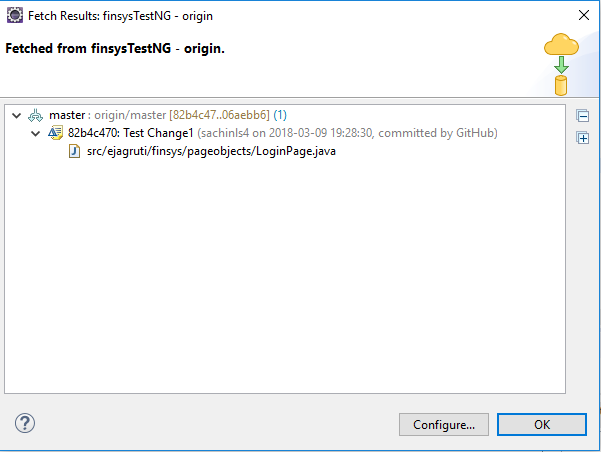
1. See screen of the changes made on GitHub. Check line 19 and 22 where the user name is now *username2*. The change was made with a comment added as Test Change1 on GitHub page.



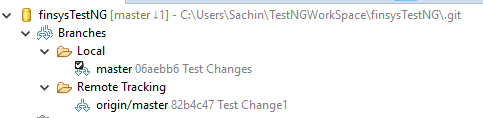
1. Go to Eclipse and switch to Git perspective. On the left hand side hierarchy, navigate to your project>Remotes>origin right click on the pull origin that is with green arrow and click Fetch.



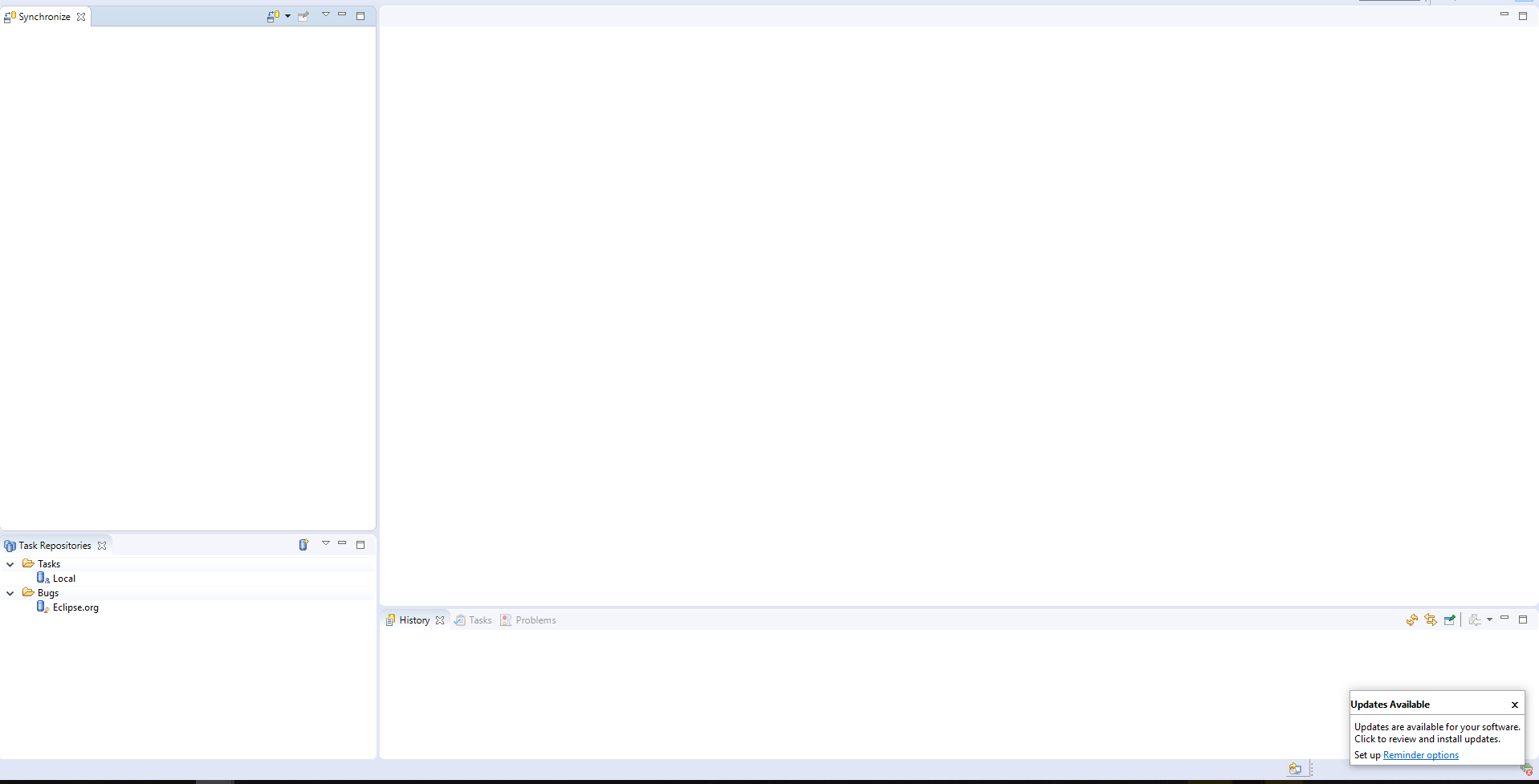
1. This will result in a popup as following…



1. Click Ok after verifying the details. This will bring the changes to Remote Tracking folder under Branches, you can see it with comment ‘Test Change1’.

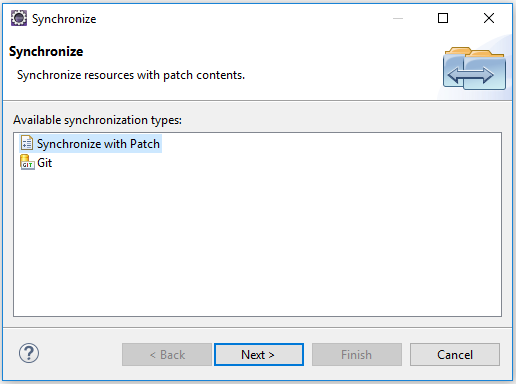


1. Open Team Synchronizing perspective which will initially look like following…

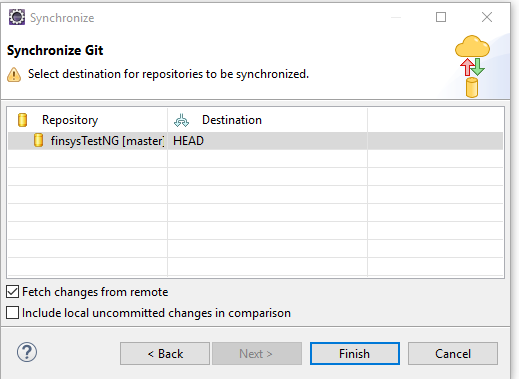


1. Click on Synchronize button on the top right hand corner of Synchronize pane which will open a popup. (*This is a one time activity)*

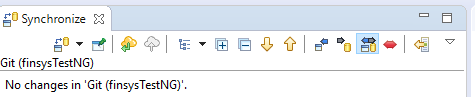




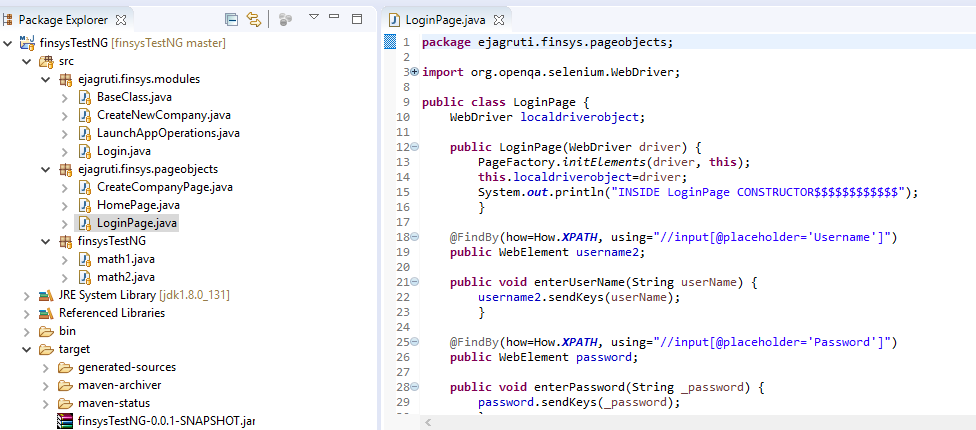
1. Select Git and click Next. Select HEAD under Destination column in the next screen…



1. Click Finish button, this will bring some buttons on the Synchronize pane. Click on the Pull button (Cloud with Green down arrow).



1. This will bring the updates from Remote to Local repository, as shown below, see line 19 and 22.



Note: Steps 6, 7 and 8 should be one time activity.

# Downloading existing Remote Repository to local Eclipse

Following steps will guide you to download existing Remote Repository to your local Eclipse IDE

# Steps –

1. Install gitbash – Follow steps from above
2. Install EGit - Git Integration for Eclipse - Follow steps from above
3. Create a GitHub account
4. Download Remote repository to your local machine
5. Import downloaded repositories (project) to Eclipse

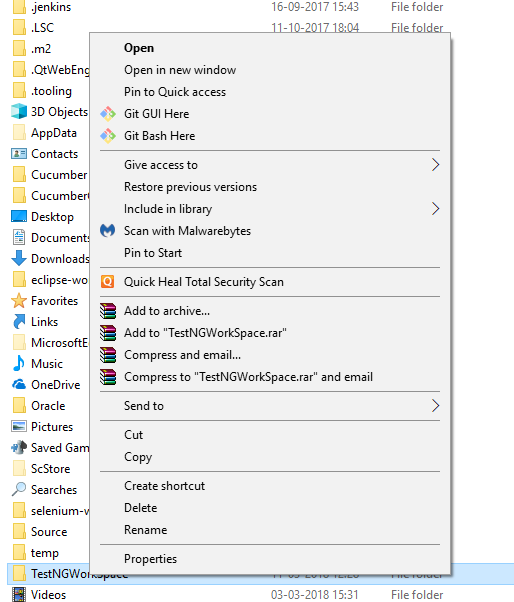
# Create a GitHub account

1. Follow steps from above except for steps 9 i.e. follow till step 8 but don’t create a repository

# Download Remote repository to your local machine

Pre-requisite: You must have remote repository URL before your start following steps

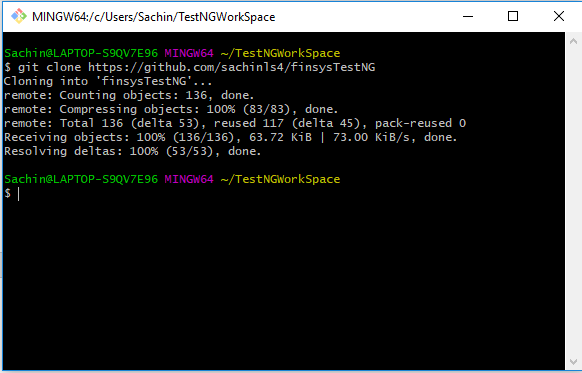
1. On your local machine, navigate to the directory where you want to keep the repository, in this case, it’s TestNGWorkSpace directory
2. Right click on the directory and click on Git Bash Here



1. Type in following command in the GitBash terminal *– git clone <remote repository URL>* for example*– git clone* [*https://github.com/sachinls4/finsysTestNG*](https://github.com/sachinls4/finsysTestNG)and hit Enter

Note: it’ll ask you your GitHub User Name and Password

Once successfully done, GitBash terminal will look like following screen shot…

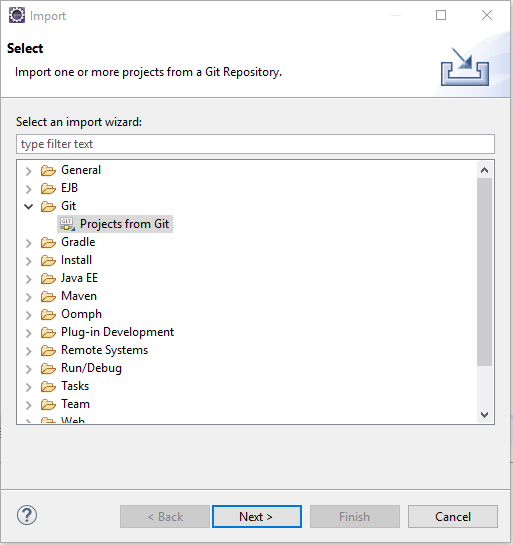


This means the remote repository is copied on your local machine successfully.

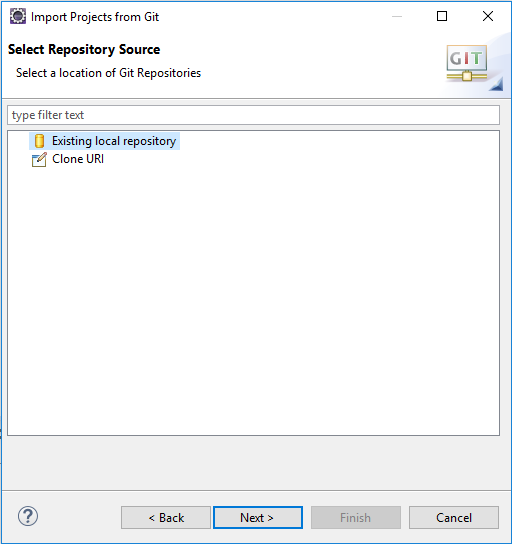
# Import downloaded repositories (project) to Eclipse

Open Eclipse and perform import an existing project steps

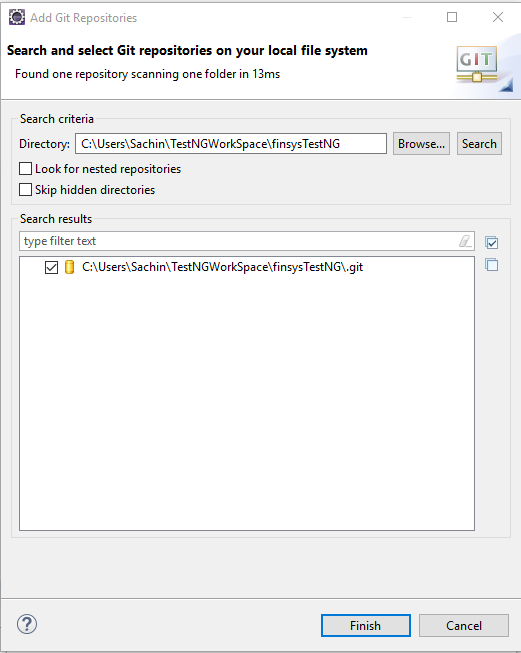
1. In Eclipse, navigate to File>Import
2. From the Import dialog popup, select Git>Projects from Git, click Next



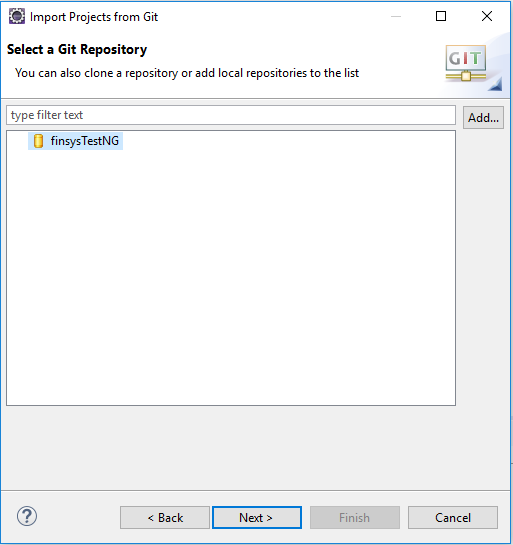
1. Select Existing local repository and click Next



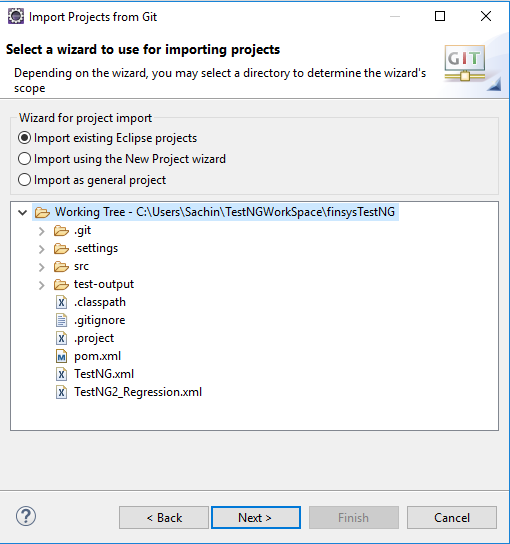
1. Click Add button, on the next screen i.e. Add Git Repositories, navigate to the directory where remote repository is cloned (in above steps)
2. Click on Search button and select the local repository from Search results
3. Select the resulting .git repository. Click Finish on Add Git Repositories, this will get back the Import Projects from Git screen



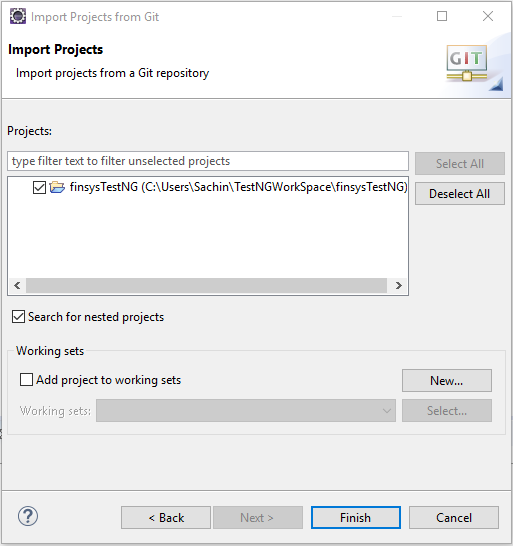
1. Click on Next button



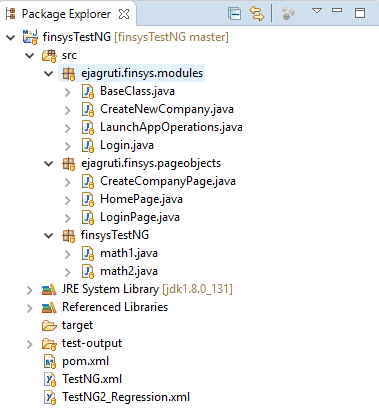
1. Click Next button again



1. Click Finish button on the next screen



1. Import is finished and ready to work as Git repositories



Now normal push (upload) and pull (download) operations can continue.