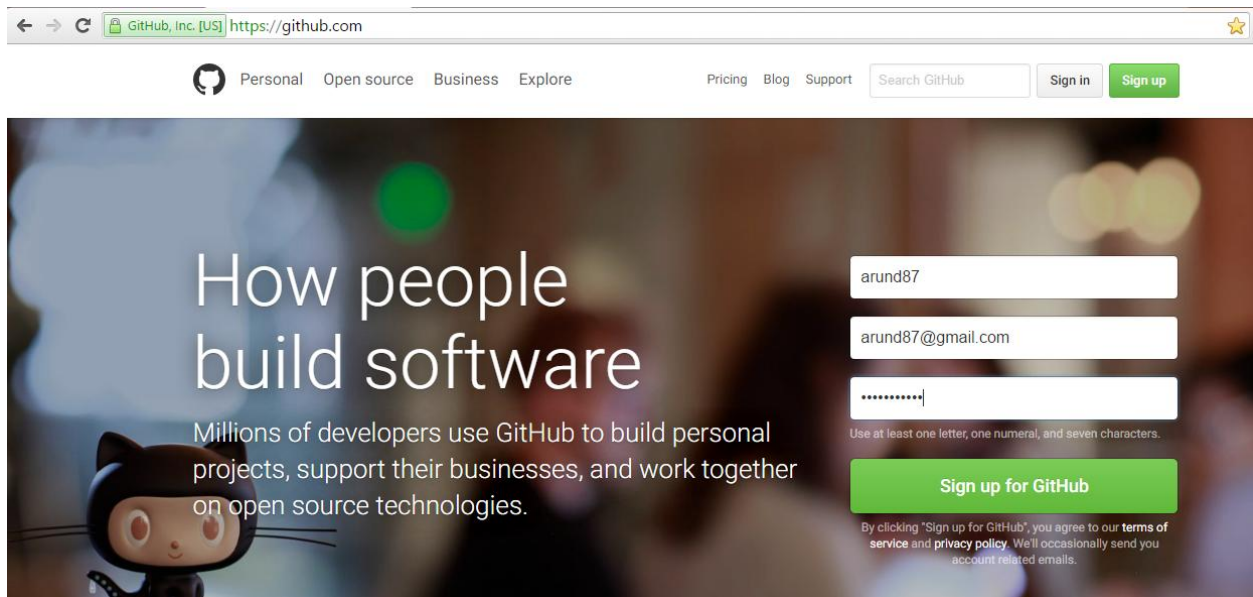


Selenium Maven Jenkins GITHUB Integration

GITHUB: Code sharing and publishing service for Developers and Testers.

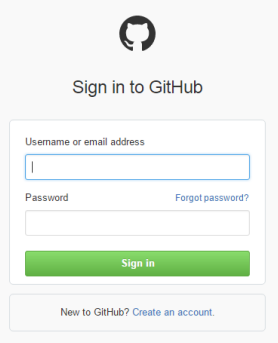
Create GIT account in the site <https://github.com>



The screenshot shows the GitHub homepage with the heading "How people build software". Below the heading, it says "Millions of developers use GitHub to build personal projects, support their businesses, and work together on open source technologies." On the right side, there is a sign-up form with three input fields: a username field containing "arund87", an email field containing "arund87@gmail.com", and a password field containing "*****". Below the password field, there is a note: "Use at least one letter, one numeral, and seven characters." A green "Sign up for GitHub" button is located below the form. At the bottom of the form, there is a small text: "By clicking 'Sign up for GitHub', you agree to our [terms of service](#) and [privacy policy](#). We'll occasionally send you account related emails."

Then do the E-mail verification.

* Login into GITHUB using the above credentials.



The screenshot shows the GitHub sign-in page. At the top, there is a GitHub logo and the text "Sign in to GitHub". Below this, there is a form with two input fields: "Username or email address" and "Password". There is a "Forgot password?" link next to the password field. A green "Sign in" button is located below the form. At the bottom of the form, there is a link: "New to GitHub? [Create an account.](#)"

***Upper right hand side you see New repositories ->click to create.**


Now->Enter repositories name , Description, Check public(free), Check Initialize this repository with a README.

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

Repository name

 arund87 /

Great repository names are short and memorable. Need inspiration? How about **glowing-train**.

Description (optional)



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.



Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾ ⓘ

Create repository

Now repository is created.

 arund87 / **MavenTutorialProject**

 Unwatch ▾

1

★ Star

0

🍴 Fork

0

↔ Code

📄 Issues 0

🔀 Pull requests 0

📖 Wiki

📈 Pulse

📊 Graphs

⚙ Settings

Learning Maven, Jenkins and GIT HUB for selenium project. — Edit

🕒 2 commits

🌿 1 branch

📦 0 releases

👤 1 contributor

Branch: master ▾

New pull request

New file

Upload files

Find file

HTTPS ▾

<https://github.com/arund87/MavenTutorialProject>

Download ZIP



arund87 Update README.md

Latest commit e4931d1 10 minutes ago



README.md

Update README.md

Readme file have description content.

10 minutes ago



README.md

Repository URL

MavenTutorialProject ➡ Repository Name

Learn Maven, Jenkins and GIT HUB for selenium project. ➡ Description

*Open eclipse .

If the Git functionality is missing in your Eclipse IDE installation, you can install it via the Eclipse installation manager. Start this manager via the *Help* → *Install new Software...* menu entry by using the following update site URL: <http://download.eclipse.org/egit/updates>

Available Software

Check the items that you wish to install.



Work with:

Add...

Find more software by working with the "[Available Software Sites](#)" preferences.

type filter text

Name	Version
<input checked="" type="checkbox"/> Eclipse Git Team Provider	
<input checked="" type="checkbox"/> Eclipse Git Team Provider - experimental features (incubation)	
<input checked="" type="checkbox"/> JGit	

Using Upload files option I uploaded three files

[arund87](#) / [MavenTutorialProject](#)

Unwatch 1

Star 0

Fork 0

Code

Issues 0

Pull requests 0

Wiki

Pulse

Graphs

Settings

Learning Maven, Jenkins and GIT HUB for selenium project. — Edit

7 commits

1 branch

0 releases

1 contributor

Branch: master

[New pull request](#)

[New file](#)

[Upload files](#)

[Find file](#)

HTTPS

<https://github.com/arund87/MavenTutorialProject>

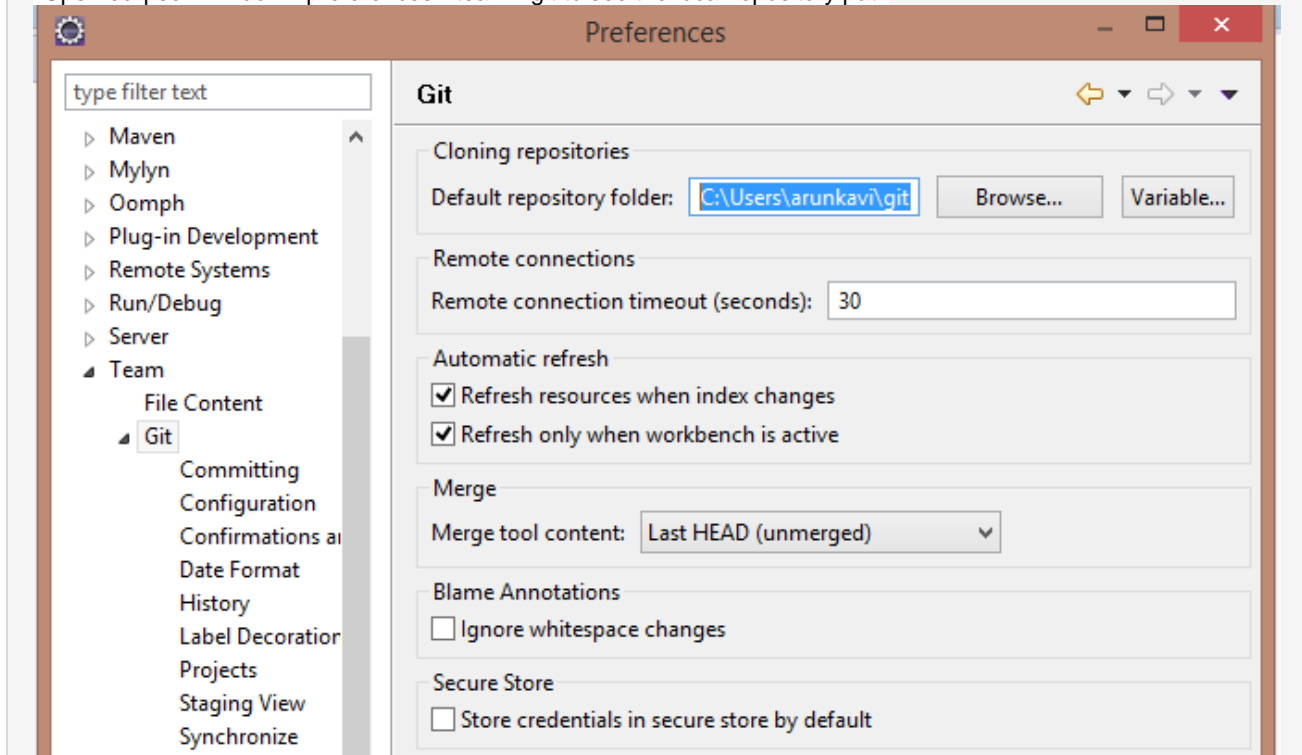
[Download ZIP](#)

arund87 Selenium Maven Jenkins Integration Document is updated.	Latest commit 1e68f22 8 hours ago
com.dakinfotech.maventutorial	List program is updated to the project. 8 hours ago
README.md	Update README.md 23 hours ago
SeleniumMavenJenkinsIntegrations.pdf	Selenium Maven Jenkins Integration Document is updated. 8 hours ago

***Now my repository having three file(project, readme, document).**

*Copy the github URL: <https://github.com/arund87/MavenTutorialProject.git>

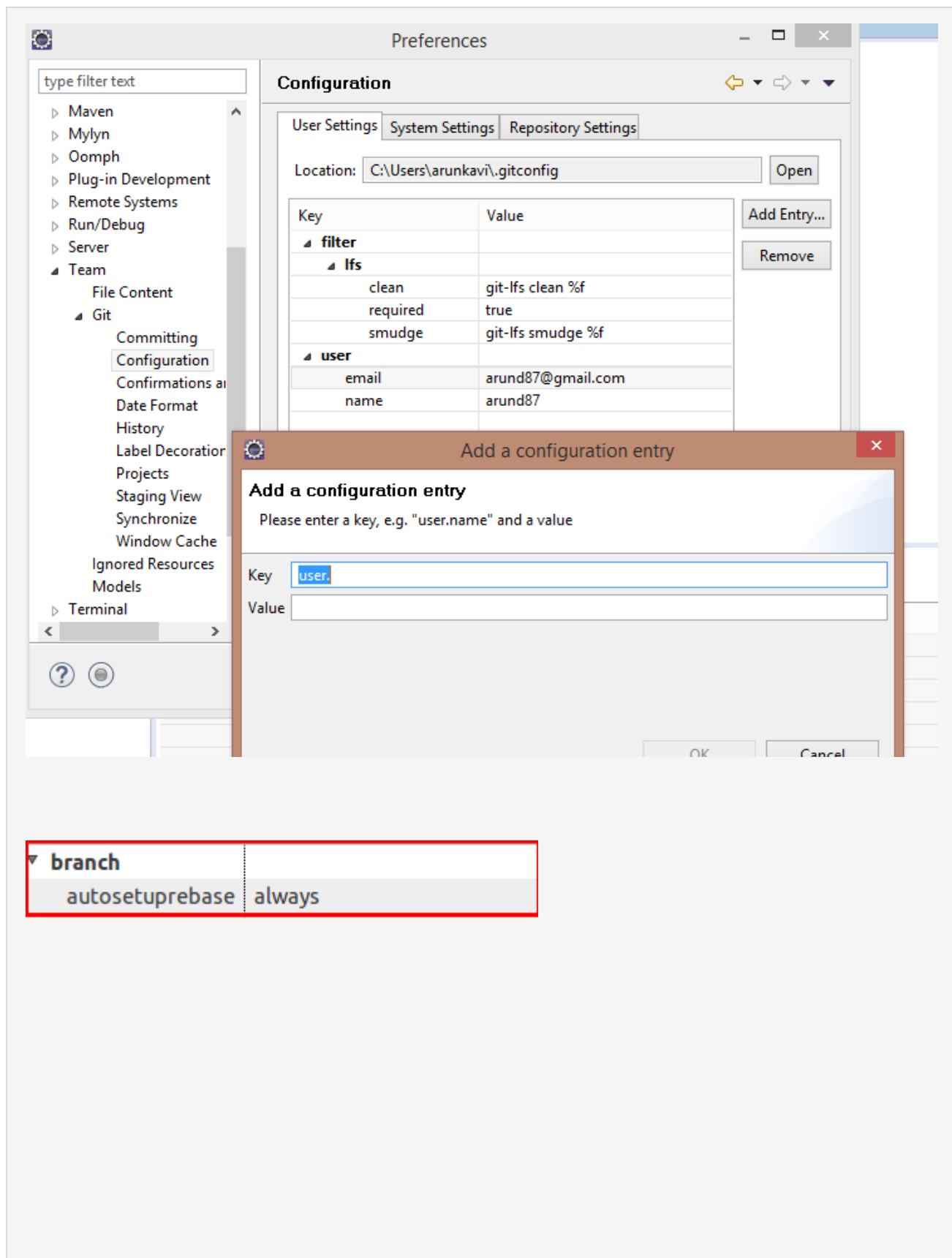
*Open eclipse->window->preferences->team->git to see the local repository path



Select *Window* → *Preferences* → *Team* → *Git* → *Configuration* and add your full name and email setting to your user settings if this is not yet configured.

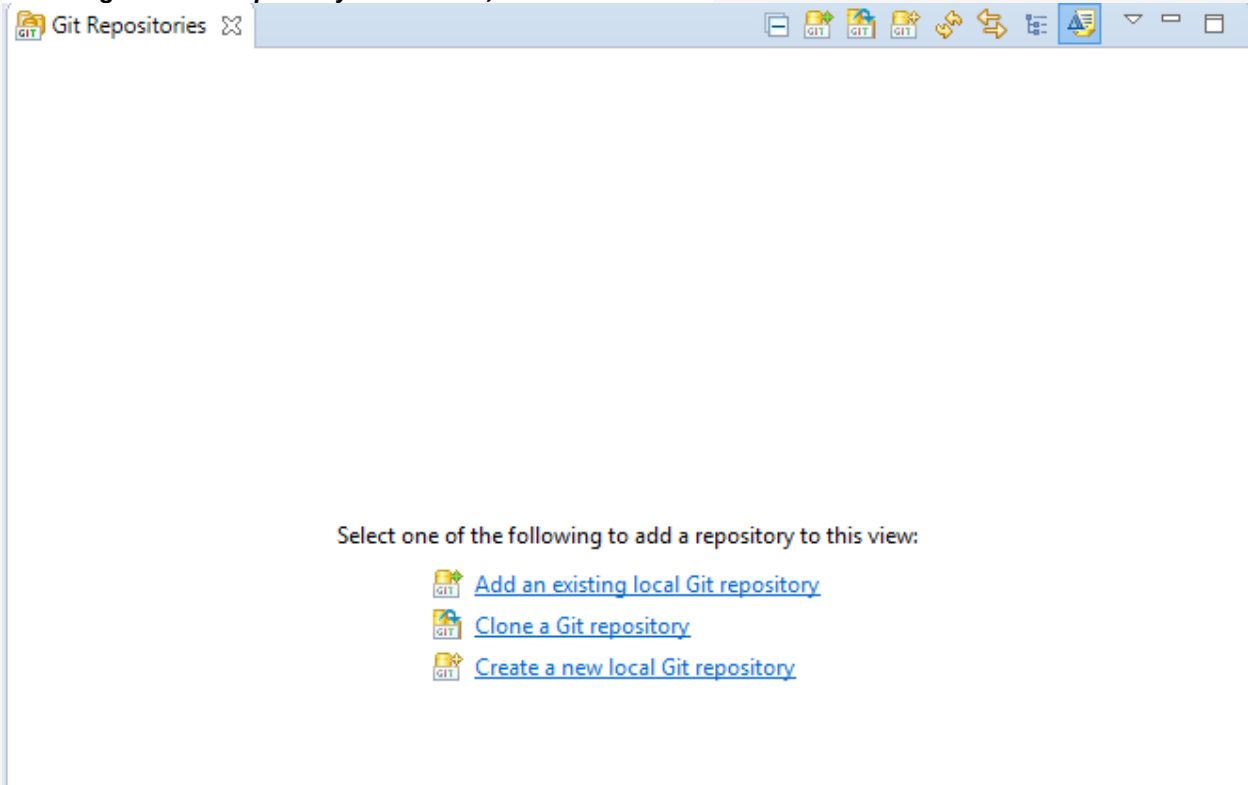
The user should be stored under the ***user.name*** key with the full name as value. If it is not available press the *Add Entry...* button. Repeat the procedure for your email address via the ***user.email*** key.

Unless you have a different opinion about rebase and merge commits, also set the ***branch.autosetuprebase*** parameter to ***always***.



Window->Perspective → open Perspective->other->Add GIT.

You will get the GIT repository. Leave this, it's like a GUI tool.



Eclipse->File->Import->GIT->Project from Git->clone URL.

github URL: <https://github.com/arund87/MavenTutorialProject.git>

Source Git Repository
Enter the location of the source repository.

Location
 URI:
 Host:
 Repository path:

Connection
 Protocol:
 Port:

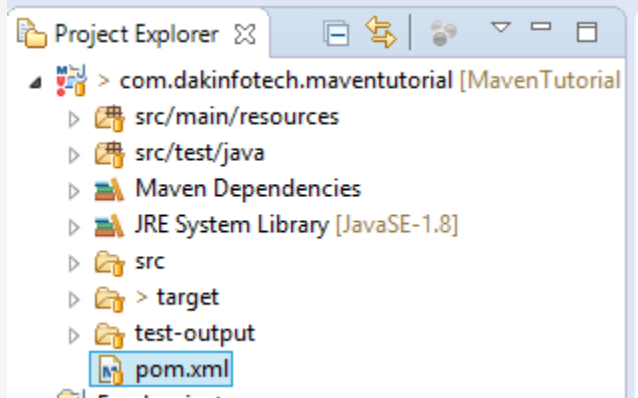
Authentication
 User:
 Password:
☒ Store in Secure Store

Next->next...->Finish the job.

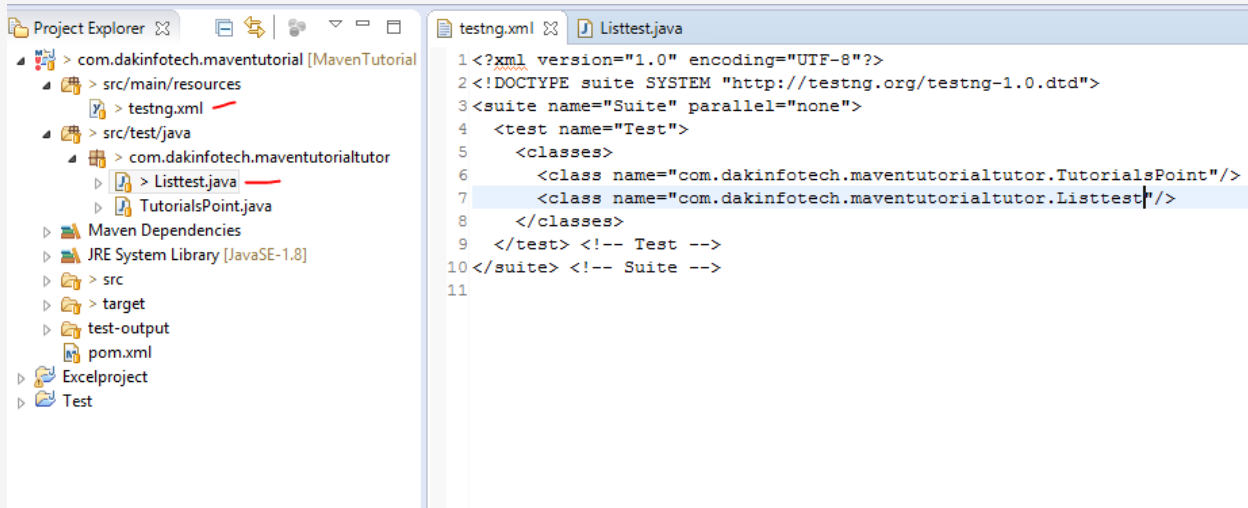
Cloning and import successful to the local and Eclipse.

📁 This PC > Windows (C:) > Users > arunkavi > git > MavenTutorialProject

<input type="checkbox"/>	Name	Date modified	Type	Size
	.git	5/6/2016 12:03 AM	File folder	
	com.dakinfotech.maventutorial	5/6/2016 12:03 AM	File folder	
	README.md	5/6/2016 12:03 AM	MD File	1 KB
	SeleniumMavenJenkinsIntegrations	5/6/2016 12:03 AM	PDF File	1,177 KB



Created new class Listtest in src/test/java and modified testNG file in src/main/resources.



the ">" symbols appeared on this project, it denotes file modification is ready to commit and push.

Note: Before commit please check locally(eclipse), program is executing without any errors.

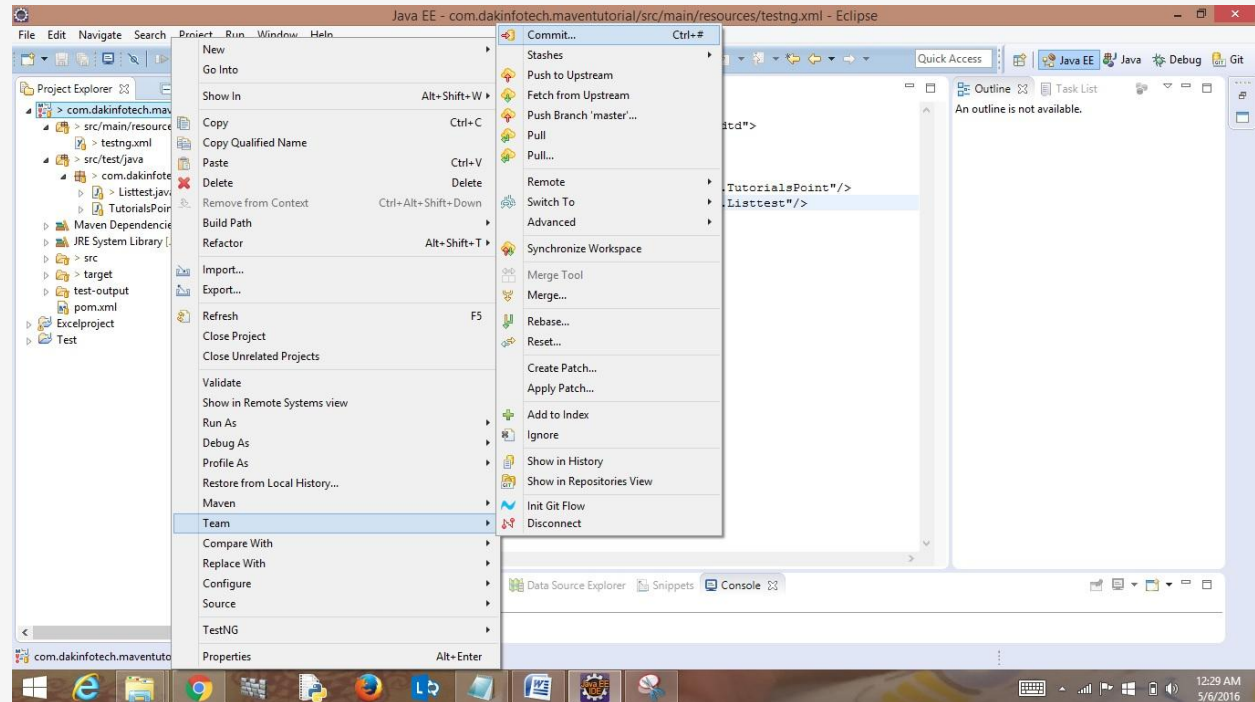
Then

Commit: modification done, File is ready to push.

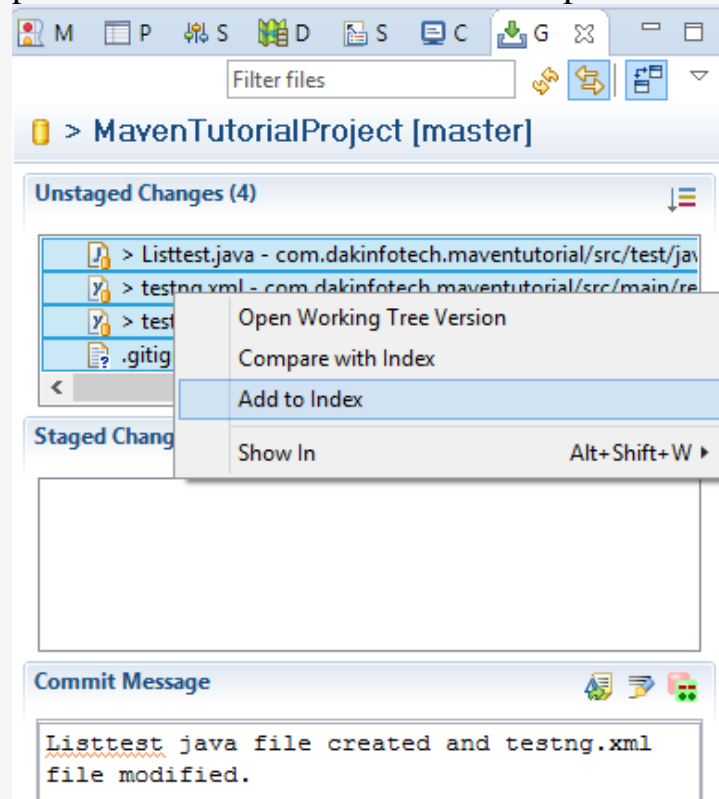
Push: Copy to local/remote repository.

Pull: Copy from remote repository to local.

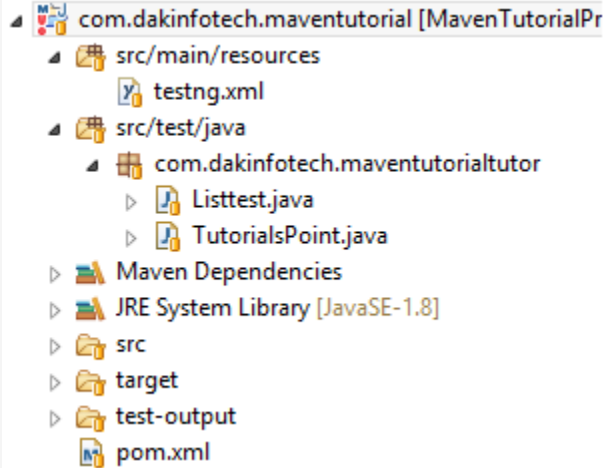
Do commit



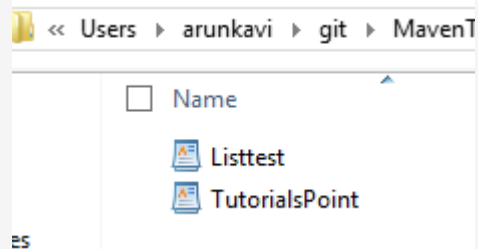
Once you press commit "Git staging" window appears. In unstaged changes it will shows the modified files. Right click those ->add to index then do commit and push. Before commit enter the description in commit message field.







See, Now ">" symbol gone(File modification updated to repository).



**Check the repository is updated:
Local:**



Remote:

 arund87	Listtest java file created and testng.xml file modified.	Latest commit 1adf80c 3 minutes ago
 com.dakinfotech.maventutorial	Listtest java file created and testng.xml file modified. ✓	3 minutes ago
 README.md	Update README.md	a day ago
 SeleniumMavenJenkinsIntegratios.pdf	Selenium Maven Jenkins Integration Document is updated.	9 hours ago

Yes it is Updated.

Now Run the Maven project in Jenkins.

Right click on the pom.xml file and copy the path.

C:\Users\arunkavi\git\MavenTutorialProject\com.dakinfotech.maventutorial\pom.xml

Open Jenkins->project->configuration->assign this path and save.

Build

Root POM

C:\Users\arunkav\git\MavenTutorialProject\com.dakinfotech.maventutorial\pom.xml

Goals and options

clean install

Post Steps

☐ Run only if build succeeds ☐ Run only if build succeeds or is unstable ☒ Run regardless of build result

Should the post-build steps run only for successful builds, etc.

Local repository path.

Now click Build Now to Run the program.

[Back to Dashboard](#)
[Status](#)
[Changes](#)
[Workspace](#)
[Build Now](#)
[Delete Maven project](#)
[Configure](#)
[Modules](#)

Maven project MavenJenkinsIntegration

This is the sample project to understand the selenium-maven-jenkins integration.

[edit description](#)
[Disable Project](#)

Test Result Trend

count

#1

#2

(just show failures) enlarge

Workspace

Recent Changes

Latest Test Result (no failures)

Latest Test Result (no failures)

Build History

find

#2	May 6, 2016 12:48 AM
#1	May 4, 2016 11:57 PM

[RSS for all](#) [RSS for failures](#)

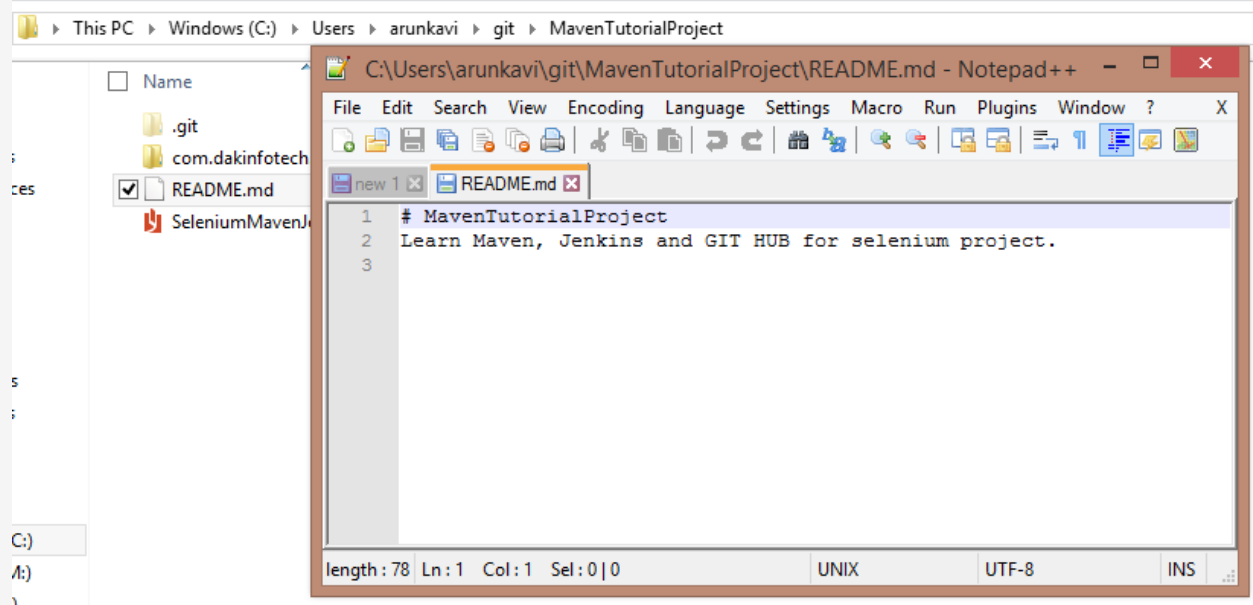
Permalinks

- [Last build \(#2\). 1 min 5 sec ago](#)
- [Last stable build \(#2\). 1 min 5 sec ago](#)
- [Last successful build \(#2\). 1 min 5 sec ago](#)
- [Last completed build \(#2\). 1 min 5 sec ago](#)

Congratulations your program successfully ran in Jenkins from Github repository.

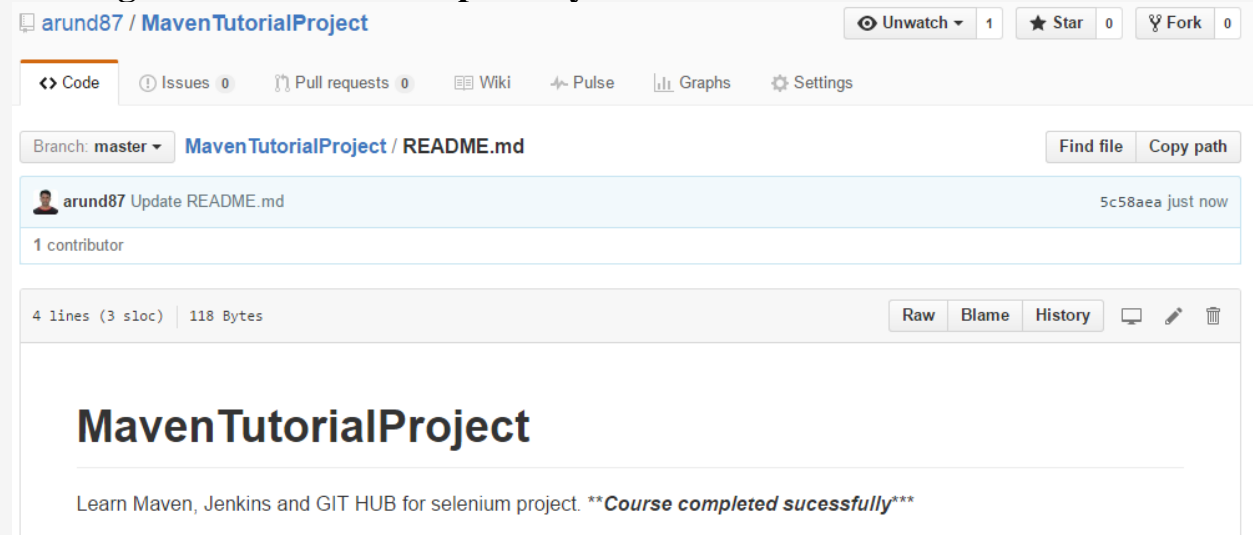
PULL:

Local repository readme file content.



I am going to change the content of readme file in remote repository and do pull to update in local repository.

***Changes done in remote repository.**



Go to eclipse right click the project->team->pull.
Good, data copied from remote successfully.

