

[Selenium — Maven — Git — Jenkins]



This is a guide to integrate Selenium with Jenkins, Maven, and TestNG with GitHub. Even though there are some other tool, here I am using Selenium WebDriver, Java, TestNG, Maven, and I integrate these with Jenkins.

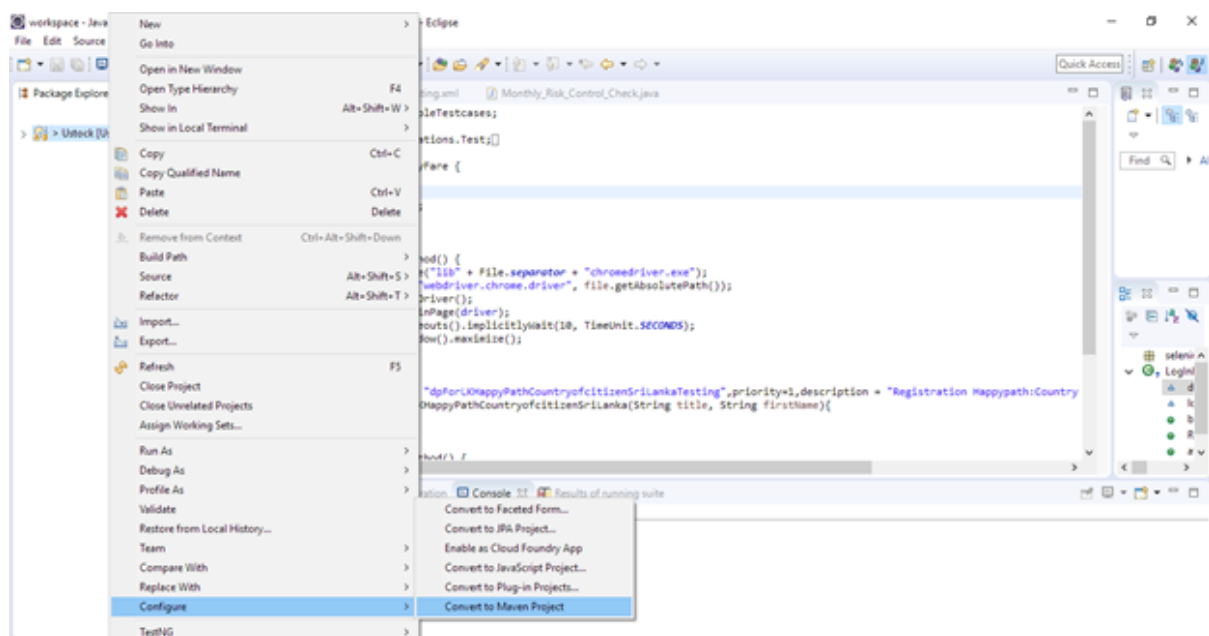
As Maven manage all project dependencies and ensure easy building process I have convert my java project to Maven. Before creating a maven project/converting existing java project to maven, you need to install Maven and TestNG in Eclipse IDE [Help >

Install New Software]. In the maven project, POM file is to add the required dependencies. Add all your dependencies related to **Selenium WebDriver, TestNG** in to your Maven Project.

Let's see how to convert your java project to Maven,

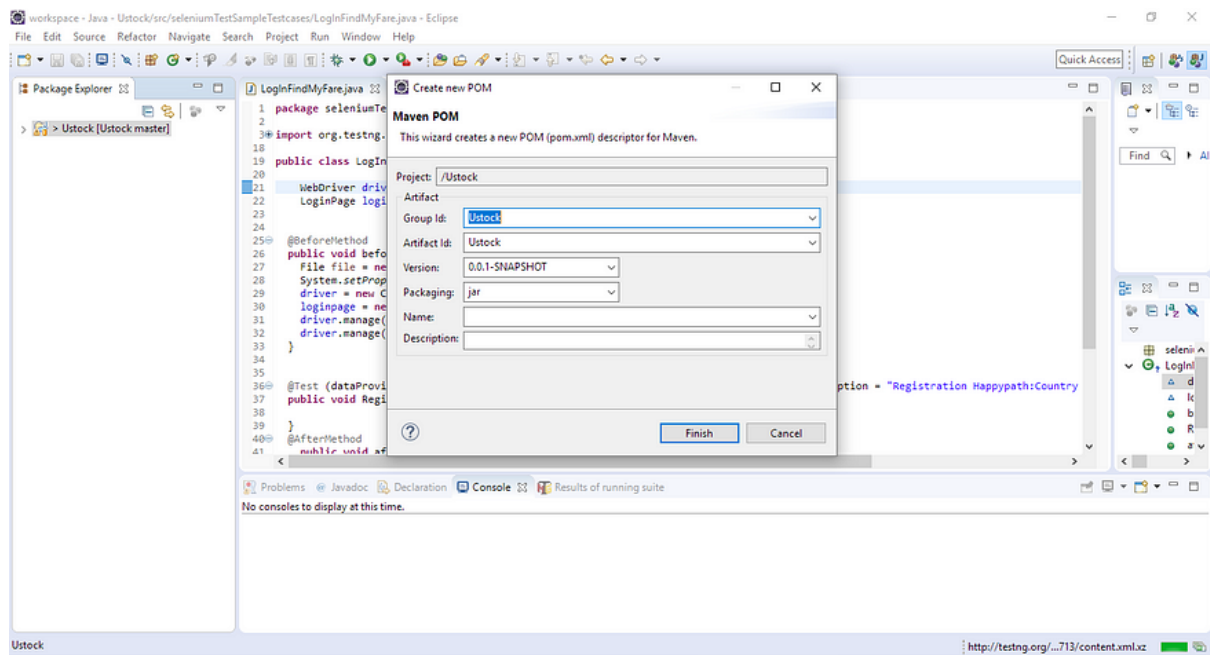
Java Project to Maven in Eclipse !!!

Right click your project and go to **Configure** and click **Convert Maven Project**.



configure > Convert to Maven Project

Fill the **Group Id, Artifact Id** and click the **Finish** button



Fill Group ID and Artifact ID

Configure ***pom.xml*** file, that was automatically created, when you create your Maven project or when you convert your java project to Maven. POM file is used to add all your dependencies, double click on pom.xml file and click on pom.xml tab.

You can get all the dependencies from Maven repository.

<https://mvnrepository.com/>

Indexed Artifacts (9.75M)

9739K
4869K
0
2004 2018

Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities
- Date and Time Utilities
- Dependency Injection
- Embedded SQL Databases
- HTML Parsers
- HTTP Clients
- I/O Utilities

Home » org.seleniumhq.selenium » selenium-java » 3.11.0

Selenium Java » 3.11.0

Selenium automates browsers. That's it! What you do with that power is entirely up to you.

License: Apache 2.0

Categories: Web Testing

HomePage: <http://www.seleniumhq.org/>

Date: (Mar 11, 2018)

Files: pom (3 KB) jar (293 bytes) View All

Repositories: Central

Used By: 829 artifacts

Note: There is a new version for this artifact

New Version: 3.12.0

Maven Gradle SBT Ivy Grape Leiningen Buildr

```
<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-java</artifactId>
  <version>3.11.0</version>
</dependency>
```

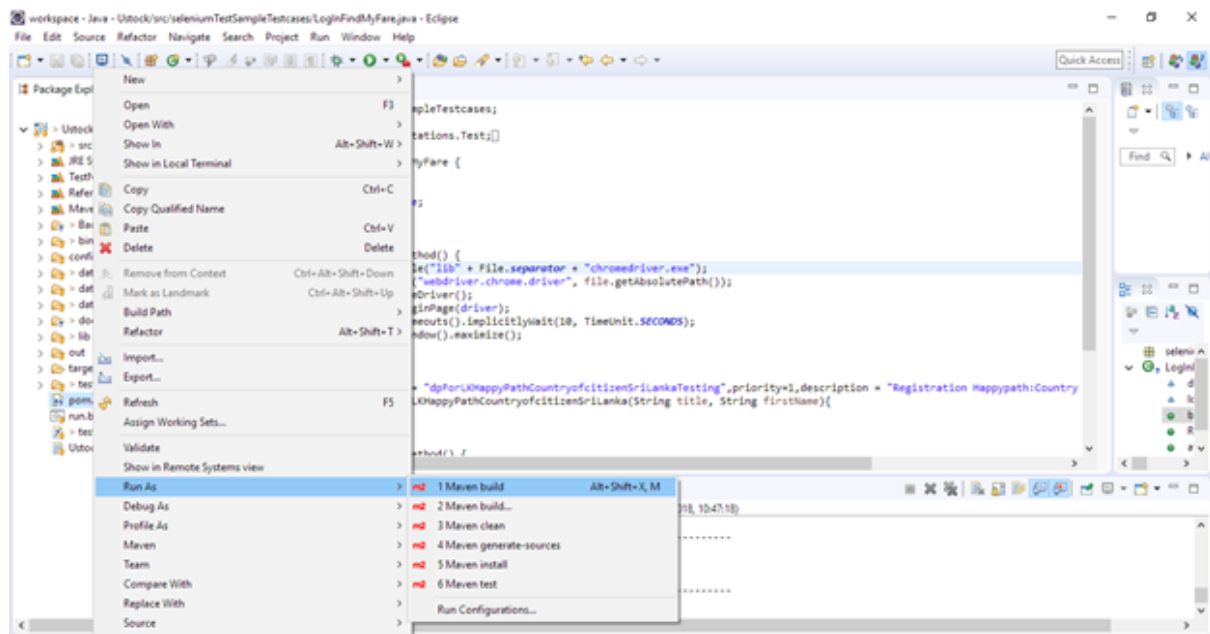
☒ Include comment with link to declaration

Maven Repository

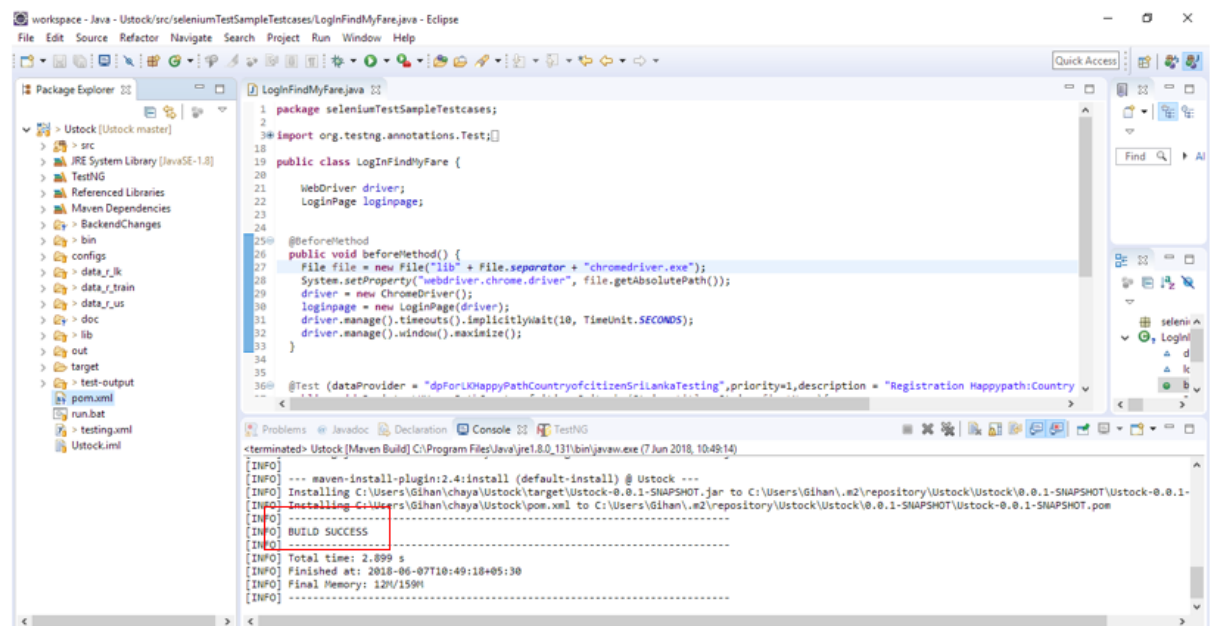
All my dependencies are listed below.

POM.java file

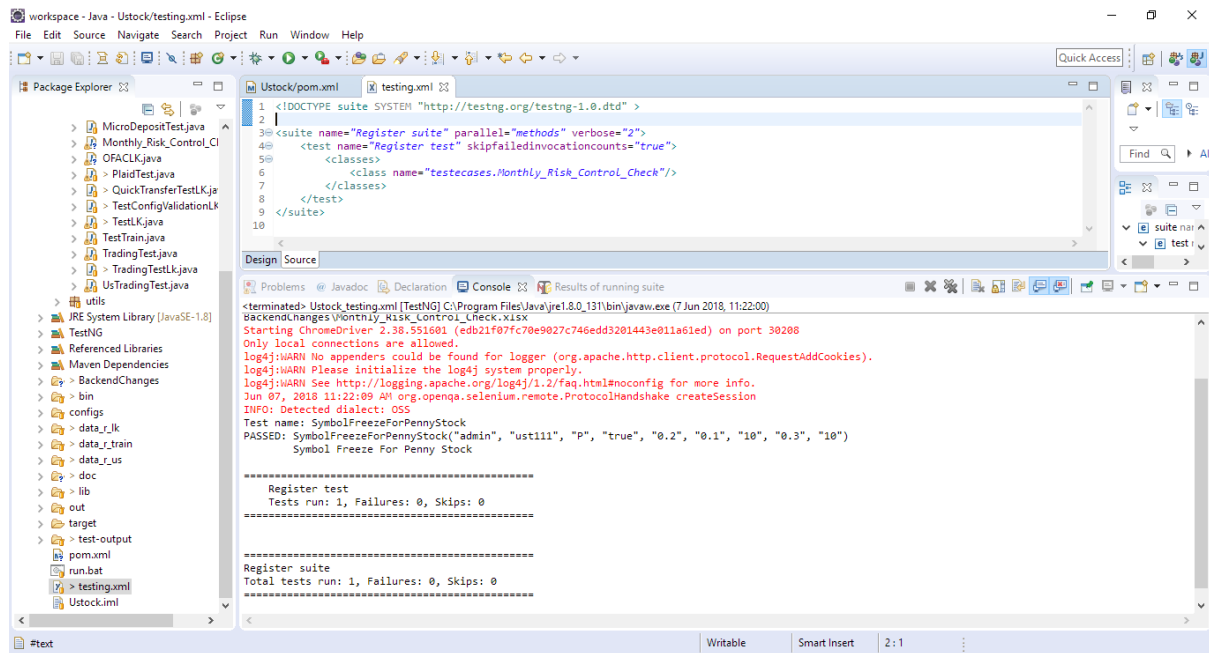
Then right click the project and run as a Maven build.



Run as Maven Build



Build Success



Run successfully : Test cases

Execute Maven Project Using Jenkins:

You can go through below article in medium to setup jenkins.

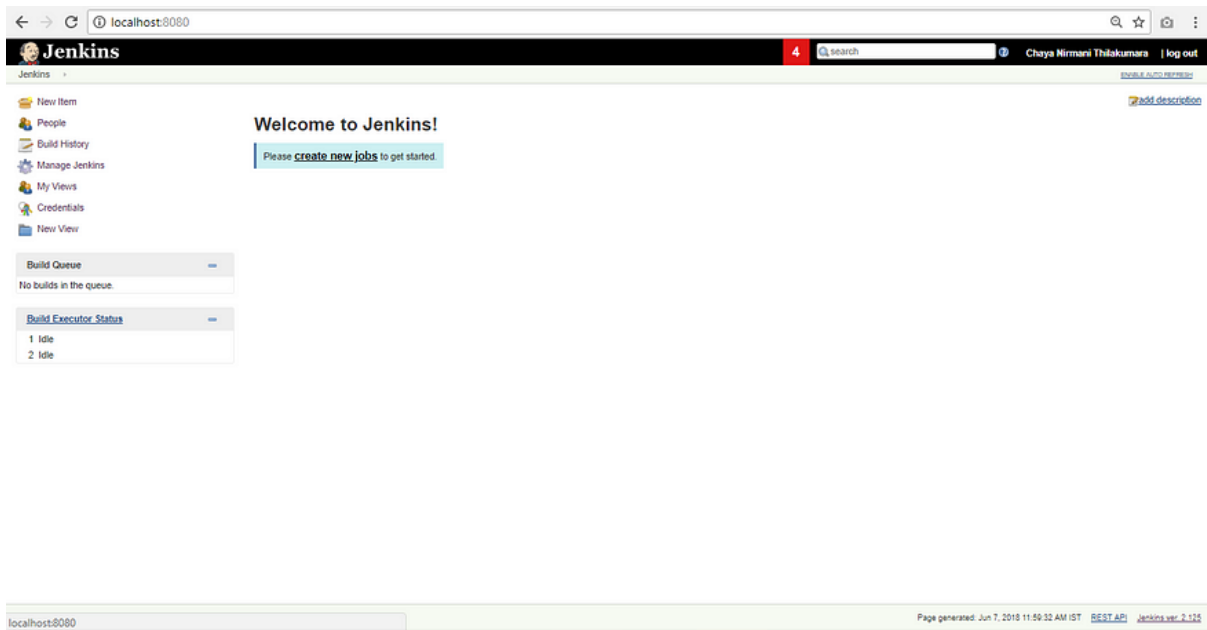
How to Integrate your Selenium Project with Jenkins

In this article, we will examine what Jenkins is, and how to set it up. How to integrate selenium with Jenkins and some...

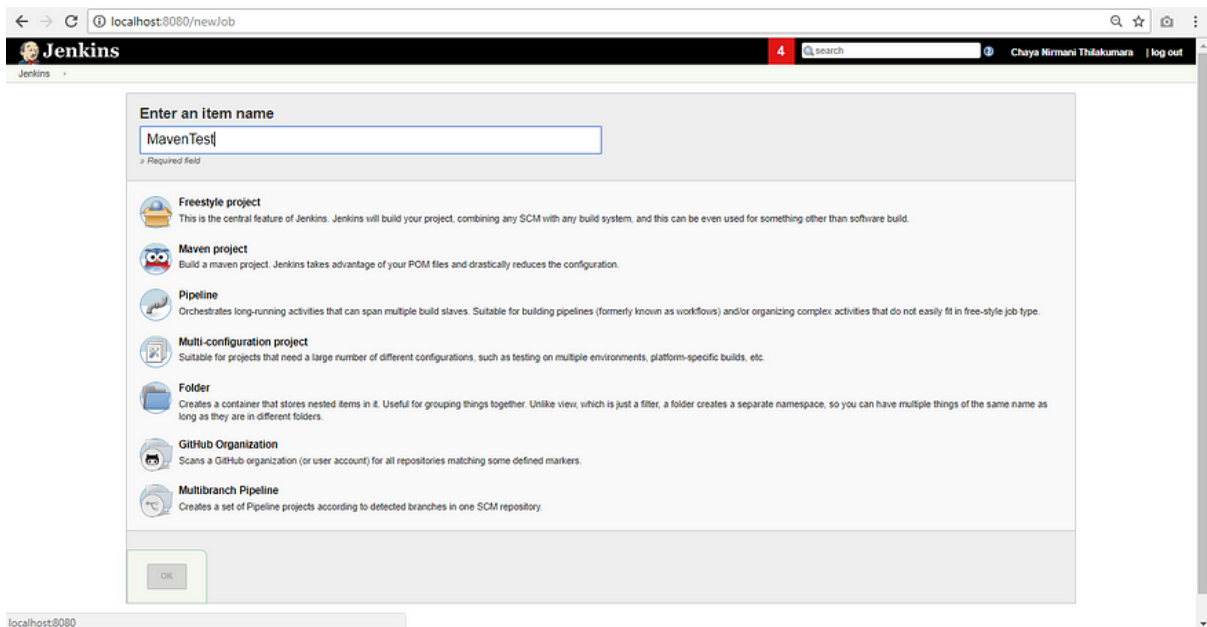
medium.com

Let's see how to execute maven project using jenkins.

Click on **New Item** link to create a job on Jenkins



Enter an item name click on FreeStyle Project and click on OK



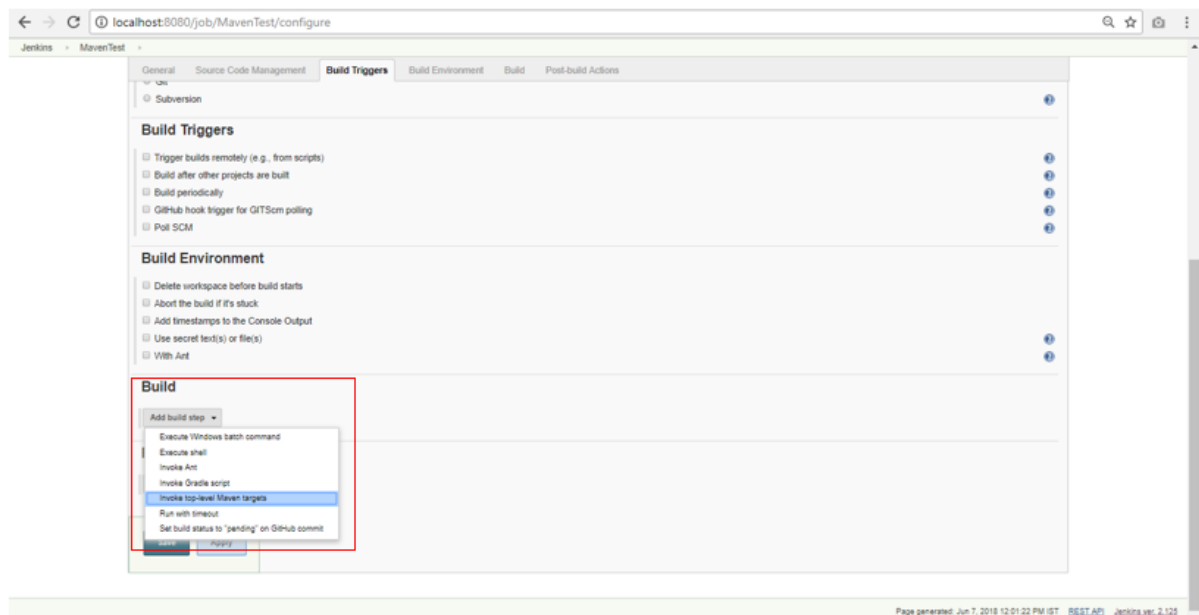
Scroll down to '**Build**' option. Click on '**Add Build Step**' and choose the value '**Invoke top-level Maven targets**' from the drop down list.

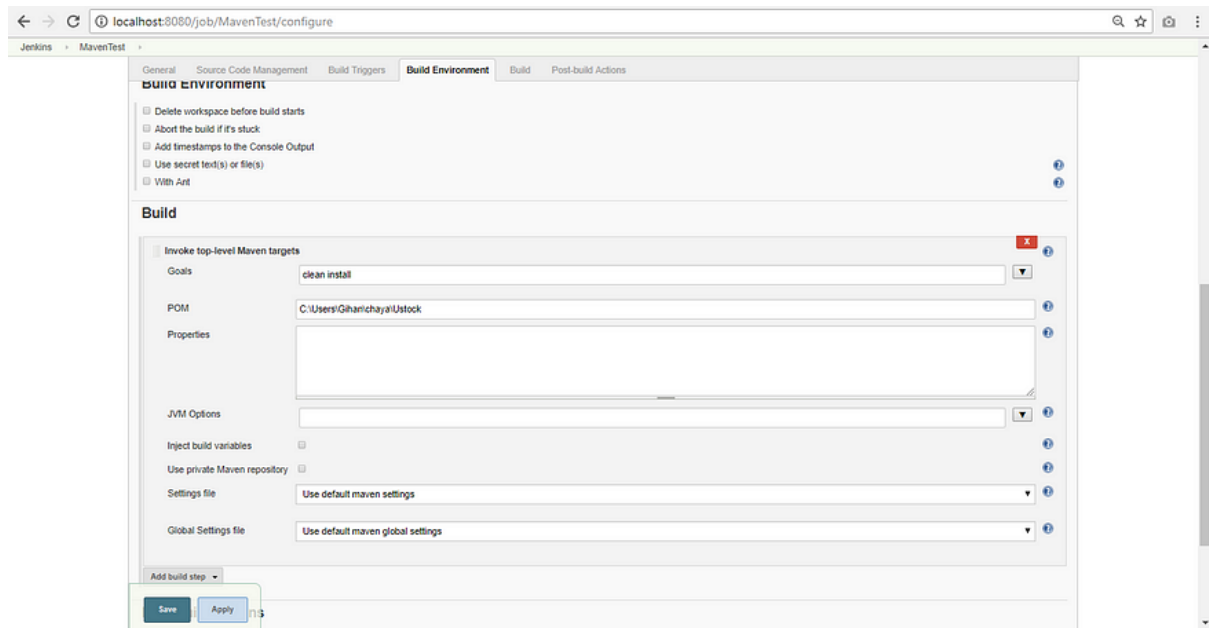
Enter Goals "**clean install**"

Enter POM path (in my case the path is

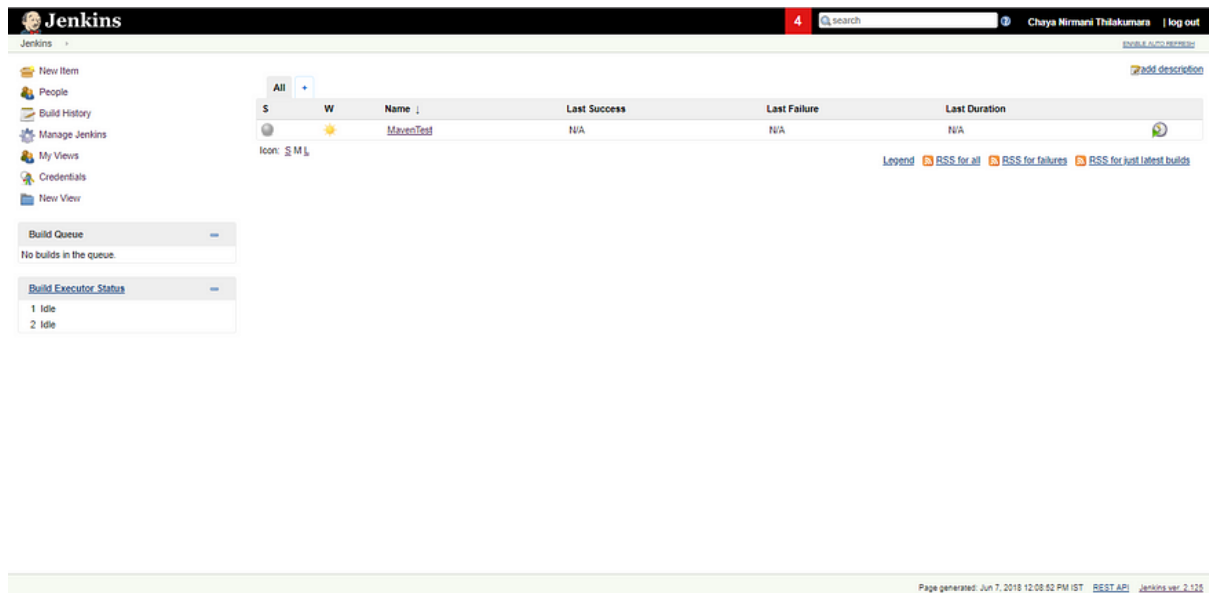
D:\CH\CBlog\workspace\MavenProject\pom.xml)

Click on '**Apply**' and '**Save**'



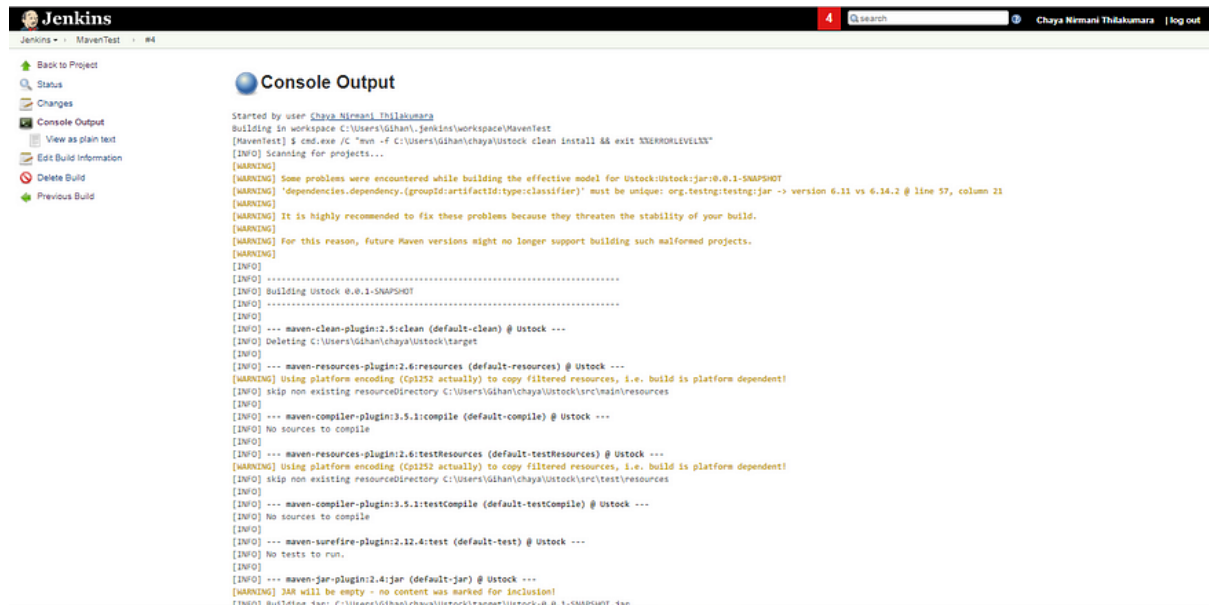


We have created a new project '**MavenProject**' with the configuration to execute Maven Project using Jenkins.



Let's execute it now. Click on '**Build Now**' button.

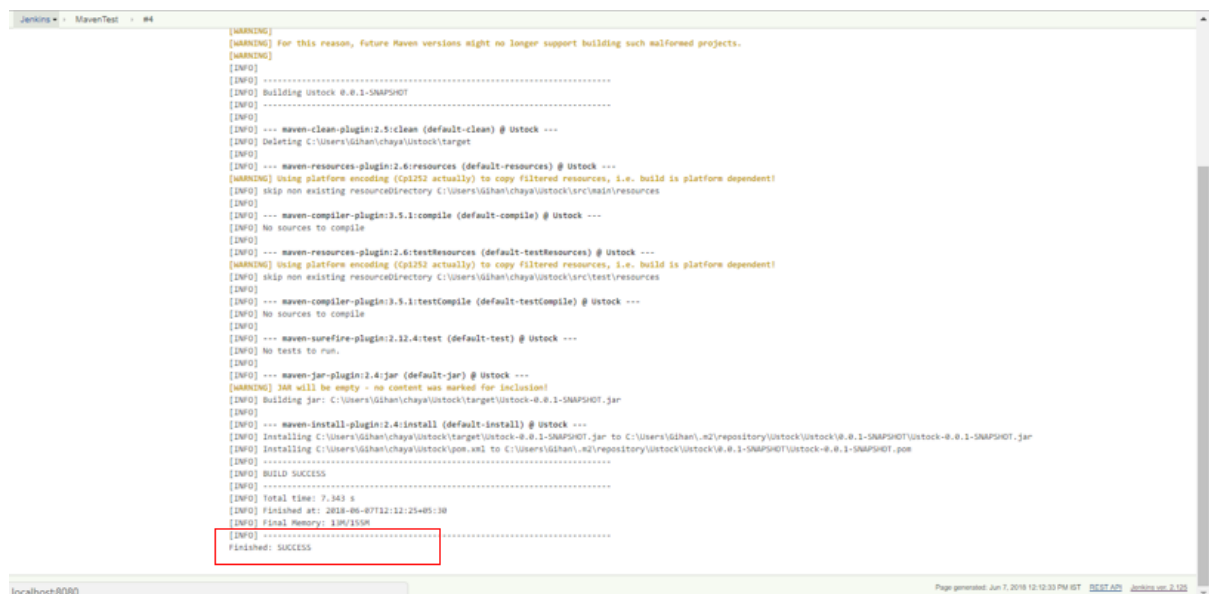
Right click on Build Number (here in my case it is #1) and click on Console Output to see the result. You could see Build Status 'Success' on Console Output.



```
Jenkins • MavenTest • #4
Back to Project
Status
Changes
Console Output
View as plain text
Edit Build Information
Delete Build
Previous Build

Console Output

Started by user Chaya Nirmani Thilakumara
Building in workspace C:\Users\Gihan\Jenkins\workspace\MavenTest
[MavenTest] $ cmd.exe /C "run -F C:\Users\Gihan\chaya\Ustock clean install && exit %ERRORLEVEL%"
[INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model for Ustock:ustock:jar:0.0.1-SNAPSHOT
[WARNING] 'dependencies.dependency.(groupId:artifactId:type:classifier)' must be unique: org.testng:testng:jar -> version 6.11 vs 6.14.2 @ line 57, column 21
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threaten the stability of your build.
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support building such malformed projects.
[WARNING]
[INFO]
[INFO] -----
[INFO] Building Ustock 0.0.1-SNAPSHOT
[INFO] -----
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ Ustock ---
[INFO] Deleting C:\Users\Gihan\chaya\Ustock\target
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ Ustock ---
[WARNING] Using platform encoding (cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Users\Gihan\chaya\Ustock\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.5.1:compile (default-compile) @ Ustock ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ Ustock ---
[WARNING] Using platform encoding (cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Users\Gihan\chaya\Ustock\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.5.1:testCompile (default-testCompile) @ Ustock ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ Ustock ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ Ustock ---
[WARNING] JAR will be empty - no content was marked for inclusion!
[INFO] Building jar: C:\Users\Gihan\chaya\Ustock\target\Ustock-0.0.1-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 7.343 s
[INFO] Finished at: 2018-06-07T12:12:25+05:30
[INFO] Final Memory: 13M/155M
[INFO] -----
[INFO] Finished: SUCCESS
```



```
Jenkins • MavenTest • #4
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support building such malformed projects.
[WARNING]
[INFO]
[INFO] -----
[INFO] Building Ustock 0.0.1-SNAPSHOT
[INFO] -----
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ Ustock ---
[INFO] Deleting C:\Users\Gihan\chaya\Ustock\target
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ Ustock ---
[WARNING] Using platform encoding (cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Users\Gihan\chaya\Ustock\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.5.1:compile (default-compile) @ Ustock ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ Ustock ---
[WARNING] Using platform encoding (cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Users\Gihan\chaya\Ustock\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.5.1:testCompile (default-testCompile) @ Ustock ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ Ustock ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ Ustock ---
[WARNING] JAR will be empty - no content was marked for inclusion!
[INFO] Building jar: C:\Users\Gihan\chaya\Ustock\target\Ustock-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ Ustock ---
[INFO] Installing C:\Users\Gihan\chaya\Ustock\target\Ustock-0.0.1-SNAPSHOT.jar to C:\Users\Gihan\m2\repository\ustock\ustock\0.0.1-SNAPSHOT\Ustock-0.0.1-SNAPSHOT.jar
[INFO] Installing C:\Users\Gihan\chaya\Ustock\pom.xml to C:\Users\Gihan\m2\repository\ustock\ustock\0.0.1-SNAPSHOT\Ustock-0.0.1-SNAPSHOT.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 7.343 s
[INFO] Finished at: 2018-06-07T12:12:25+05:30
[INFO] Final Memory: 13M/155M
[INFO] -----
[INFO] Finished: SUCCESS
```

This way, we could execute Maven Project using Jenkins.

Configure GIT Plugin in Jenkins:

Go To Manage Plugins → Filter list of plugins available with 'Git Plugin'. Check the Git Plugin and click on the button 'Install without restart'. After the installation is done, restart Jenkins by using the command in the browser.

<http://localhost:8080/jenkins/restart>

Once Jenkins is restarted, Git option should be available under Source Code Management when configuring a job in Jenkins.

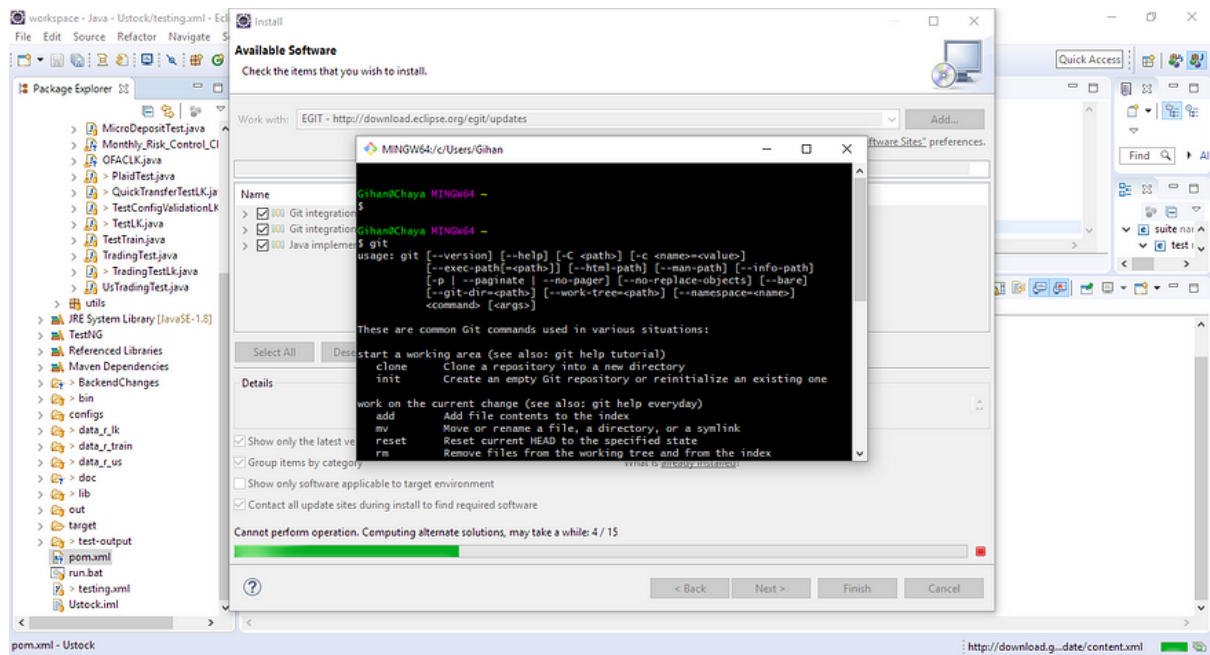
The screenshot shows the Jenkins Plugin Manager interface. The top navigation bar includes the Jenkins logo, a search bar, and the user name 'Chaya Hirmani Thilakumara'. The left sidebar has links for 'Back to Dashboard' and 'Manage Jenkins'. The main content area has tabs for 'Updates', 'Available', 'Installed', and 'Advanced'. A search filter 'git plugin' is applied. The table below lists various plugins, with the 'git' plugin highlighted in blue.

Enabled	Name	Version	Previously installed version	Uninstall
<input checked="" type="checkbox"/>	Bouncycastle API	2.18.2		Uninstall
<input checked="" type="checkbox"/>	Command Agent Launcher	1.1		Uninstall
<input checked="" type="checkbox"/>	Credentials Plugin	2.1.10		Uninstall
<input checked="" type="checkbox"/>	Git client plugin	2.7.2	Downgrade to 2.6.0	Uninstall
<input checked="" type="checkbox"/>	Git plugin	3.9.1	Downgrade to 3.6.4	Uninstall
<input checked="" type="checkbox"/>	GitHub Branch Source Plugin	2.3.0	Downgrade to 2.3.1	Uninstall
<input checked="" type="checkbox"/>	GitHub plugin	1.29.1	Downgrade to 1.28.1	Uninstall
<input checked="" type="checkbox"/>	JDK Tool	1.0		Uninstall
<input checked="" type="checkbox"/>	Mailer Plugin	1.21	Downgrade to 1.20	Uninstall
<input checked="" type="checkbox"/>	Matrix Project	1.12		Uninstall
<input checked="" type="checkbox"/>	Pipeline: GitHub Groovy Libraries	1.0		Uninstall
<input checked="" type="checkbox"/>	Pipeline: SCM Step	2.8		Uninstall
<input checked="" type="checkbox"/>	Pipeline: Step API	2.13		Uninstall
<input checked="" type="checkbox"/>	SCM API	2.2.5		Uninstall

Let's see how to add our project to Git,

You can download and install the latest stable release “Git Binaries” from the URL- <https://git-scm.com/>. Once git is installed successfully, you can access the git.

Open Gitbash and type “git” and hit “Enter”. If you see below screen that means it is installed successfully



Now go to your project folder and open git bash there.

From this article I just go with the commands of Git directly. Let's discuss more on Git from my next article.

So to initialize the repository, type below code in git bash.

```
git init
```

Then type,

```
git status
```

This will show you all the files which are not added to git. So you can add your file using below code.

```
git add .
```

And now if you type the command “git status” and hit enter, you can see that all files are added and then type below code to commit your changes.

```
git commit -m "initial commit"
```

Now you have committed your files but now you need to push these files to a remote repository. So for that you need to go to your github account and create a public repository and copy the location of the repository. Then in git bash type the below command.

```
git remote add origin <location of your git repository>
```

```
git push -u origin master
```

The let's see how to configure your jenkins job

Go to [Configure > General > Source code management] and select Git option and provide the repository URL. Then go to [Configure > General > Poll SCM] and provide your cron job pattern.

You can give any expression based on your requirement. As an example, you can give `*****` as the cron job pattern. This job will check the Jenkins repository every minute and if there is any change, Jenkins will trigger this job.

Then Apply the changes and Save.