**DevOps Driven Test Automation**

Overview: this document provides the guidelines how to setup the automated testing environment using open source tools.

**Tools used**

* **Maven (**build management tool) has a XML file called pom.xml (Project Object Model), and that file describes the software project being built, its dependencies on other external modules and components, the build order, directories, and required plug-ins.
* **Selenium (**testing tool**)** web driver Automates browsers with interacting the browsers objects.
* **Jenkins** is a powerful tool that allows continuous integration and continuous delivery of project.

**Prerequisites:**

1) Eclipse IDE (Neon)

2) JDK-1.8

3) apache-maven-3.3.9

4) Jenkins-2.7.3

**Steps to install:**

1) Download Eclipse (<https://eclipse.org/downloads/>) and install it.

2) Download JDK-1.8 (<http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>) and Set its path in environment variables.

3) Download apache-maven-3.3.9 (<https://maven.apache.org/download.cgi>) and Set its path in environment variable.

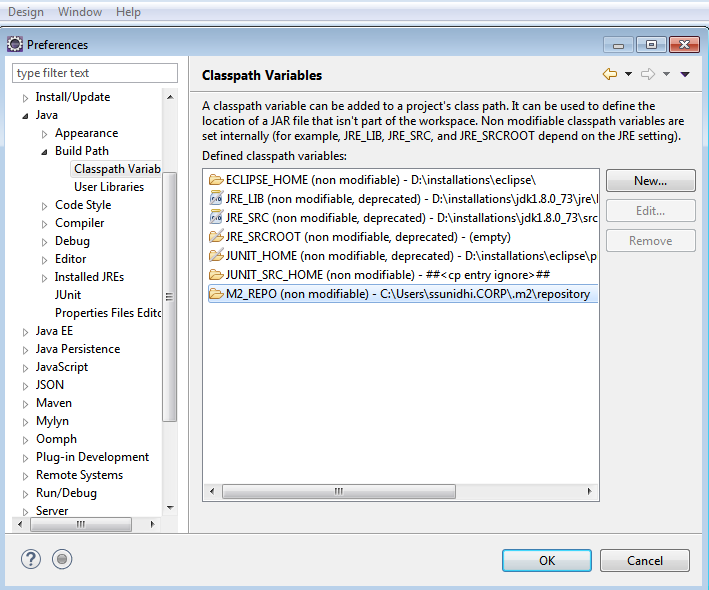
**Steps to use Set Maven Path in eclipse**

1) Eclipse IDE, menu bar

2) Select Window > Preferences.

3) Select Java > Build Path > Classpath Variables

4) Click on the new button > defined a new M2\_REPO variable and point it to your local Maven repository as shown in the screenshot:



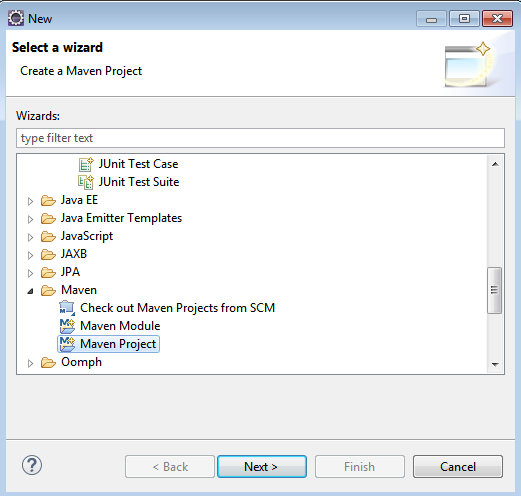
5) Click OK button.

**Configure Eclipse with Maven**

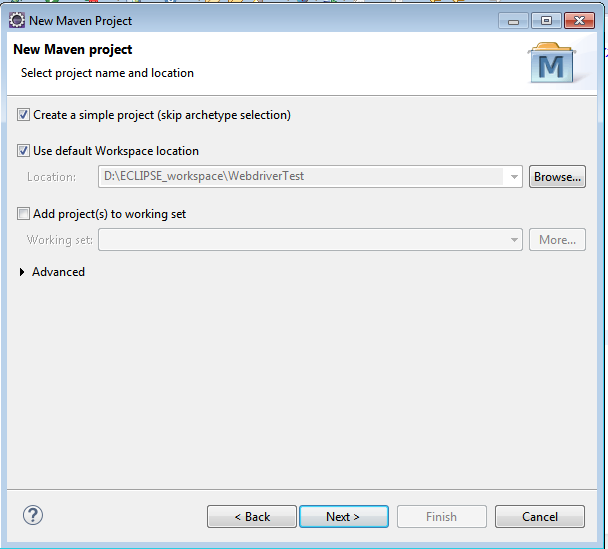
Now, we need to create a Maven project.

**Step 1)** In Eclipse IDE, create a new project by selecting **File** | **New** | **Other** from Eclipse menu.

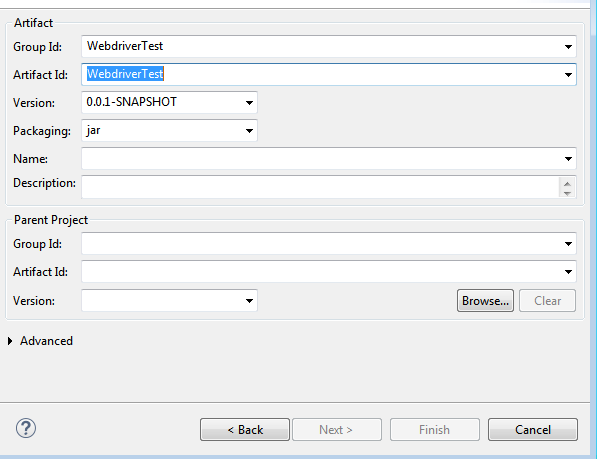
**Step 2)** On the **New** dialog, select **Maven** | **Maven Project** and click Next



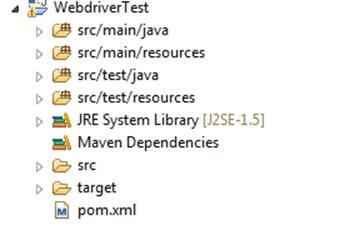
**Step 3)**On the **New Maven Project** dialog select the **Create a simple project** and click Next.



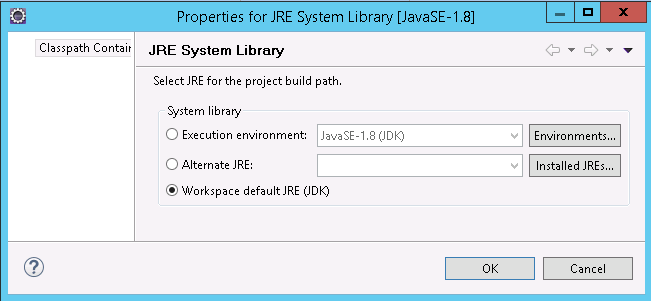
**Step 4)**Enter WebdriverTest in **Group Id**: and **Artifact Id**: and click finish



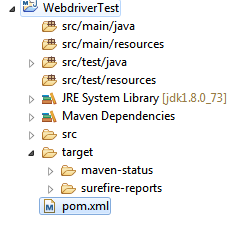
**Step 5)** Eclipse will create **WebdriverTest** with following structure:

[](http://cdn.guru99.com/images/5-2015/050115_1023_MavenJenkin5.jpg)

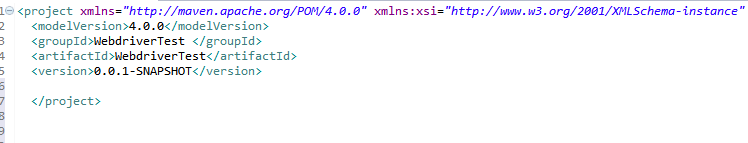
**Step 6)** Right click on **JRE System Library** and select the **Properties** option from the menu and select **workspace default JRE (JDK).**

****

**Step 7)** Select **pom.xml** from **Project Explorer**.



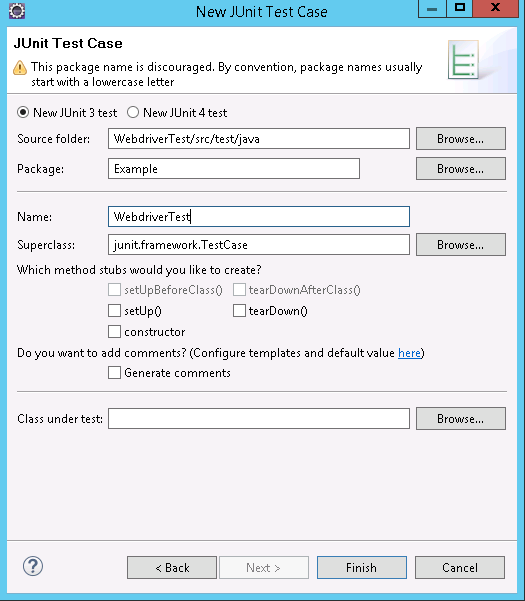
pom.xml file will Open in Editor section.



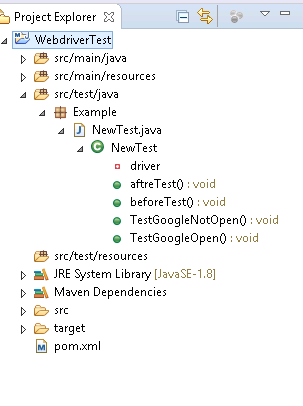
**Step 8)** Add the Selenium, TestNG, chrome Driver and JUnit dependencies to pom.xml in the <project> node:



**Step 9)**Create a New JUnit Class. Enter Package name as "example" and "NewTest" in the **Name**: textbox and click on the **Finish** button as shown in the following screenshot:



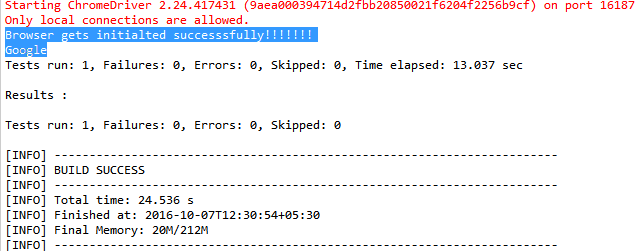
**Step 10)** Eclipse will create the NewTest class as shown in the following screenshot:



**Step 11)**Add the following code to the **NewTest** class:



**Step 12)** To run th**e** tests in the Maven lifecycle, Right-click on the WebdriverTest and select **Run As** | **Maven test**. Maven will execute test from the project.



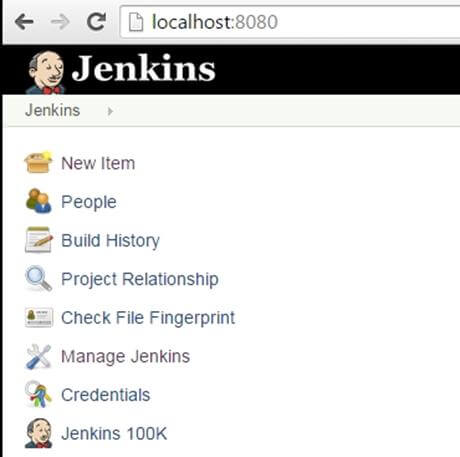
Make sure that build finished successfully

### Steps to Install Jenkins, Maven and Selenium Configuration

**Installation**

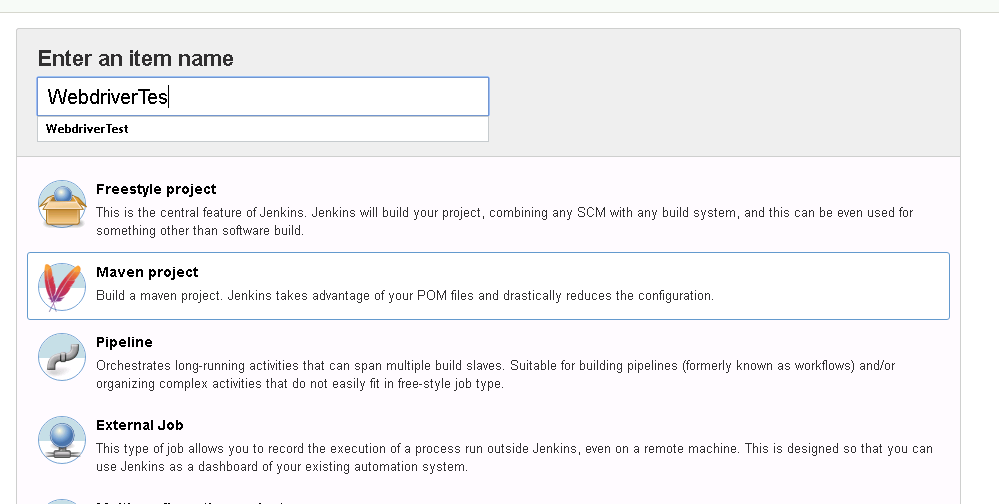
**Step 1)** Go to <http://jenkins-ci.org/>and download correct package for your OS. Install Jenkins

**Step 2)**once installation is done, navigate to the Jenkins Dashboard (http://localhost:8080 by default) in the browser window.

[](http://cdn.guru99.com/images/5-2015/050115_1023_MavenJenkin21.jpg)

**Step 3)**Click on the **New Item** link to create a new job.

**Step 4)**Select the Maven project radio button.

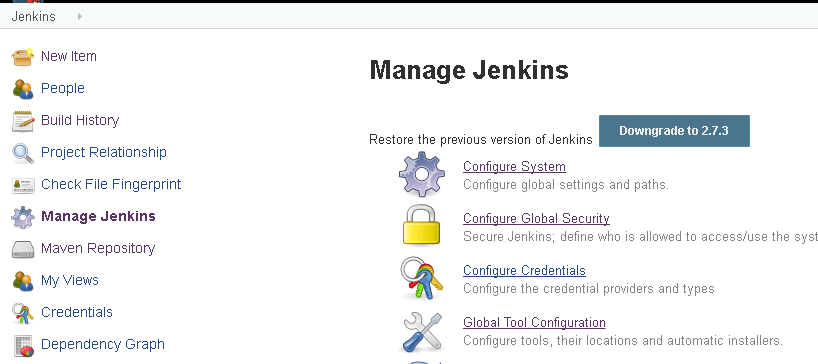


Using the Build a **Maven Project** option, Jenkins supports building and testing Maven projects.

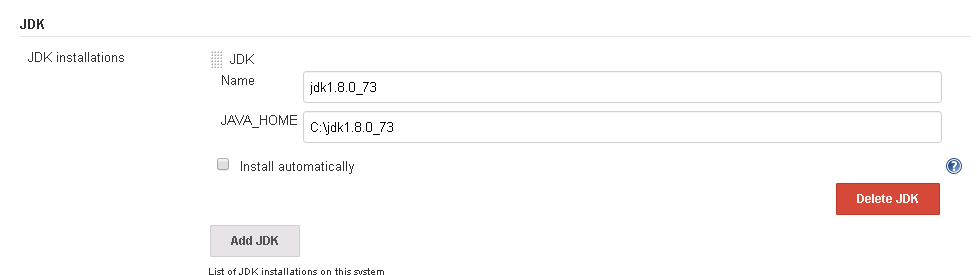
**Step 5)** Click on OK button. A new job with name "WebdriverTest" is created in Jenkins Dashboard.

[Maven & Jenkins with Selenium: Complete Tutorial](http://cdn.guru99.com/images/5-2015/050115_1023_MavenJenkin24.jpg)

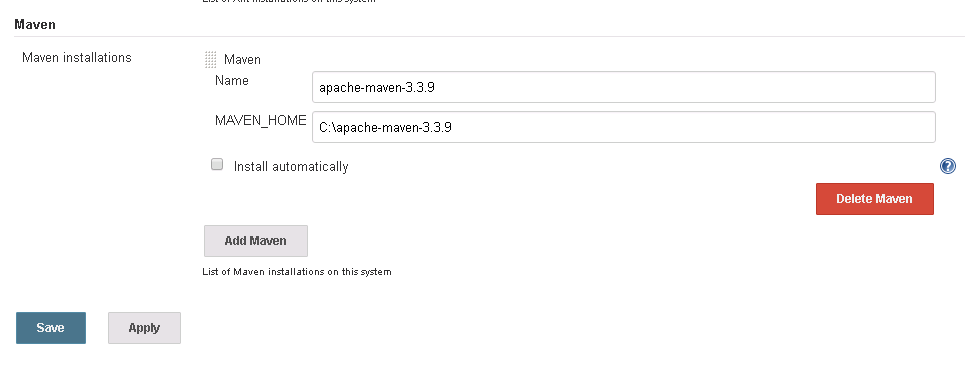
**Step 6)** Go to **Manage Jenkins** => **Global Tool Configuration**.



**Step 7)** Click on JDK installations and configure JDK.



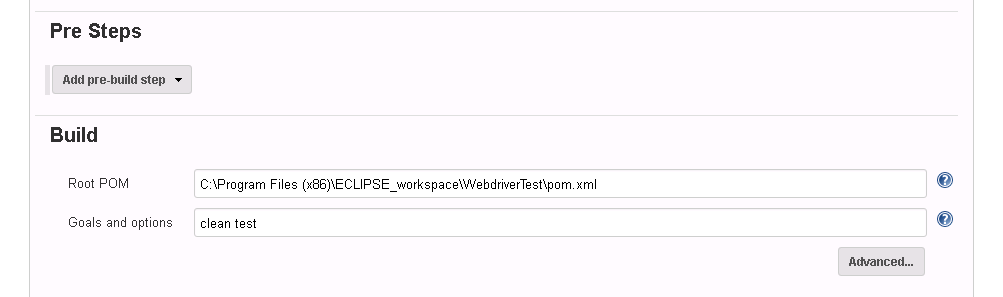
**Step 8)** Click on Add Maven and configure it.



**Step 9)**Go to the **Build** section of new job.

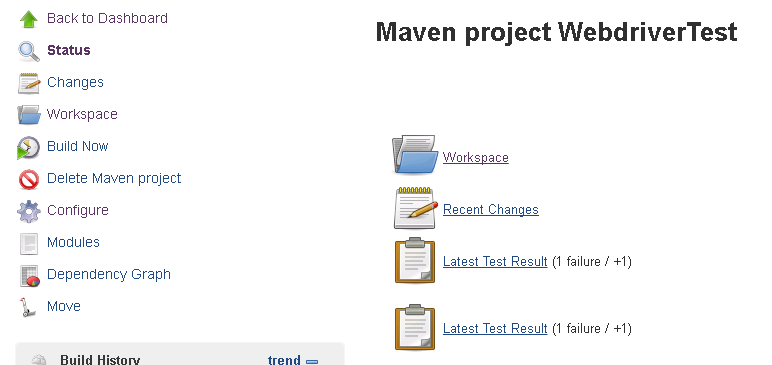
* In the **Root POM** textbox, enter full path to pom.xml
* In Goals and options section, enter Goals and options as "clean test"

Goal ‘**test’** will run the test case.



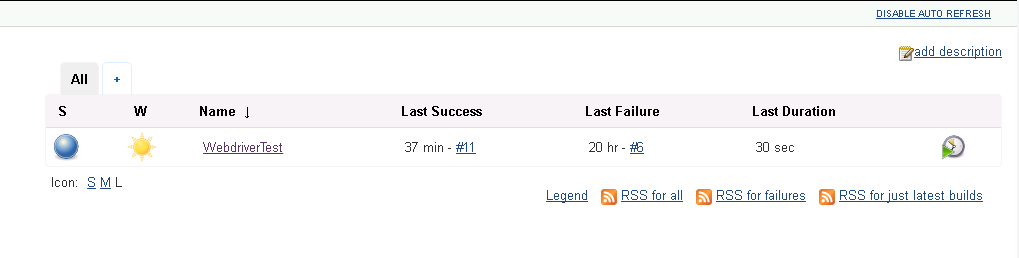
**Step 10)** Click on **Apply** button below the page.

**Step 11)**On the WebdriverTest project page, click on the **Build Now** link.

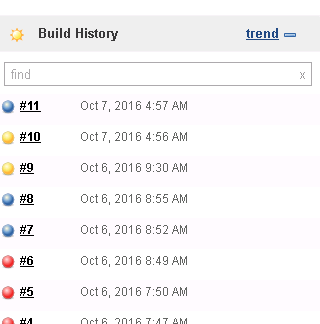


Maven will build the project. It will execute the test cases.

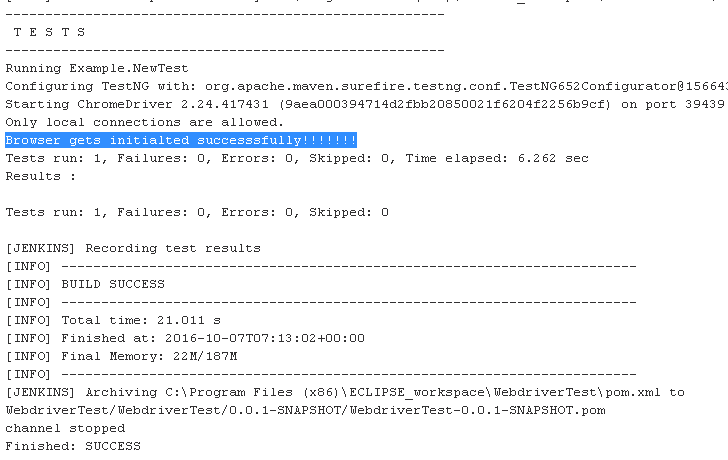
**Step 11)**once the build process is completed, in Jenkins Dashboard click on the **WebdriverTest** project



**Step 12)**The WebdriverTest project page displays the build history and links to the results as shown in the following screenshot:



**Step 13)** Select specific build and you will see the current status by clicking on "**console output**"



References:

1) <http://www.guru99.com/maven-jenkins-with-selenium-complete-tutorial.html>

2) <https://www.tutorialspoint.com/jenkins/jenkins_maven_setup.htm>

3) <http://www.guru99.com/selenium-tutorial.html>

4) <http://www.tutorialspoint.com/maven/>