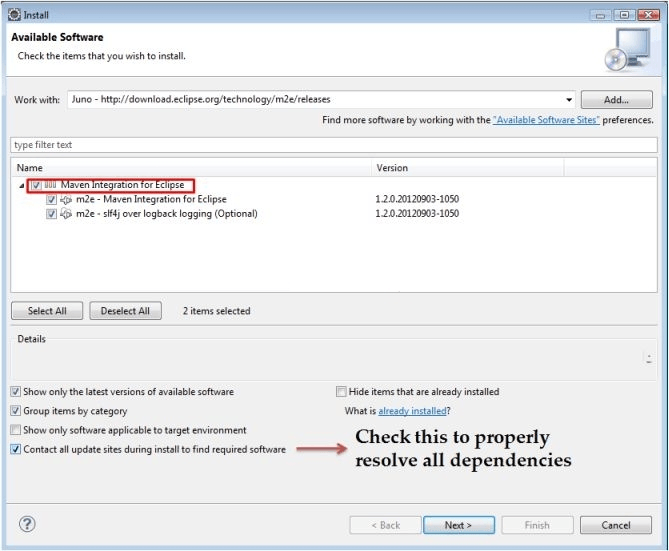
Selenium maven integration .

1. In Eclipse , select Help -> install new software .

2. On the install dialgoue box enter the url.

http://download.eclipse.org/technology/m2e/releases/.

3. 

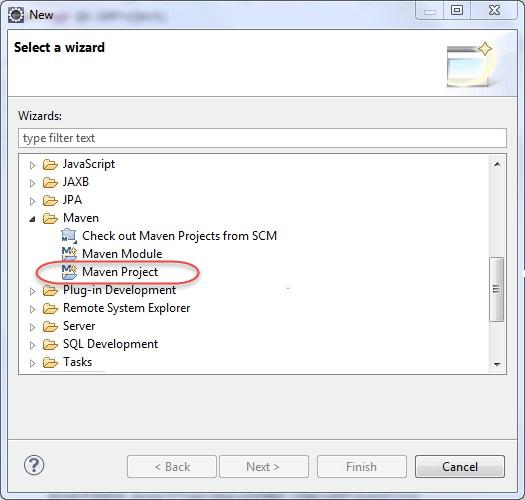
**Step 3)**Click on **Next** button and finish installation.

**Configure Eclipse with Maven**

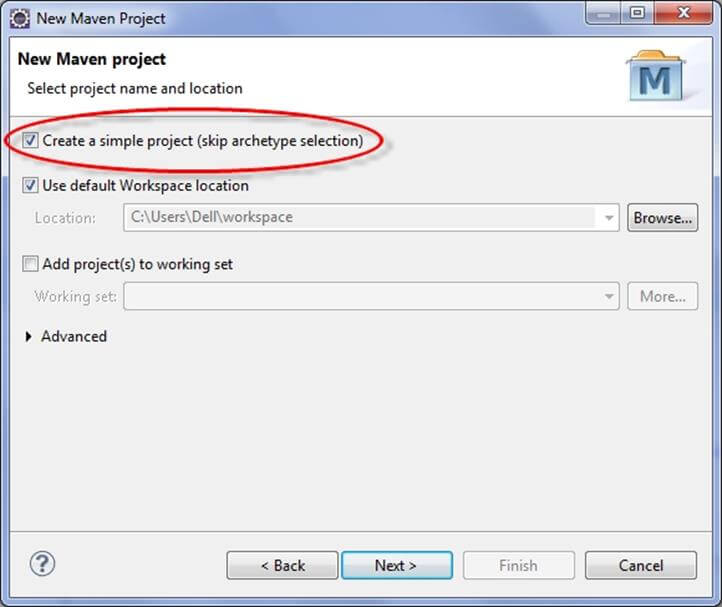
With m2e plugin is installed, we now need create Maven project.

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

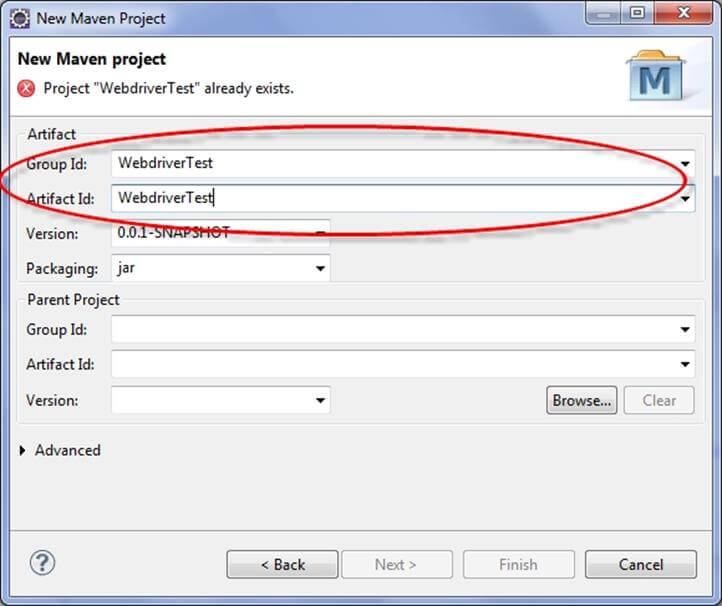
**Step 2)** On the **New** dialog, select **Maven** | **Maven Project** and click Next



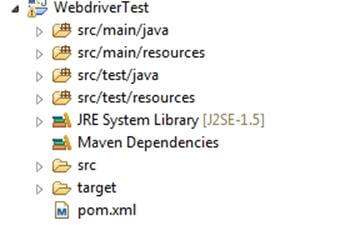
**Step 3)**On the **New Maven Project** dialog select the **Create a simple project** and click Next



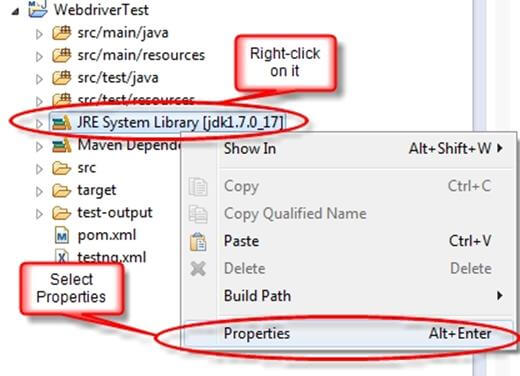
**step 4)**Enter WebdriverTest in **Group Id**: and **Artifact Id**: and click finish .



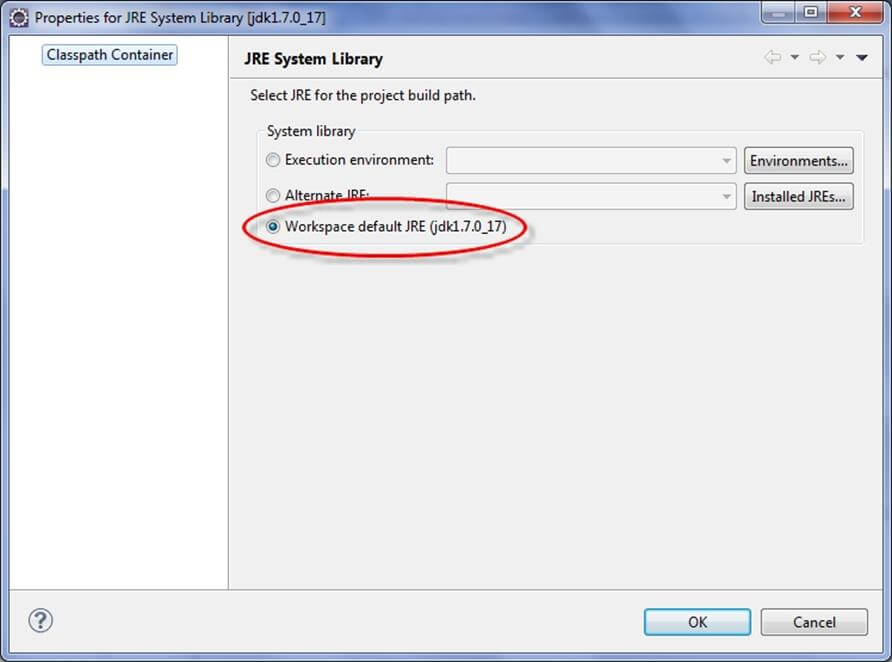
**Step 5)** Eclipse will create **WebdriverTest** with following structure:



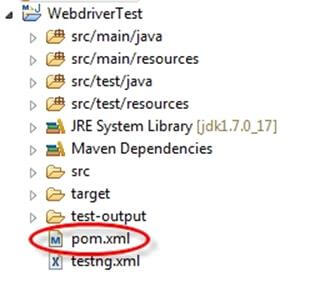
**Step 6)** Right-click on **JRE System Library** and select the **Properties** option from the menu.



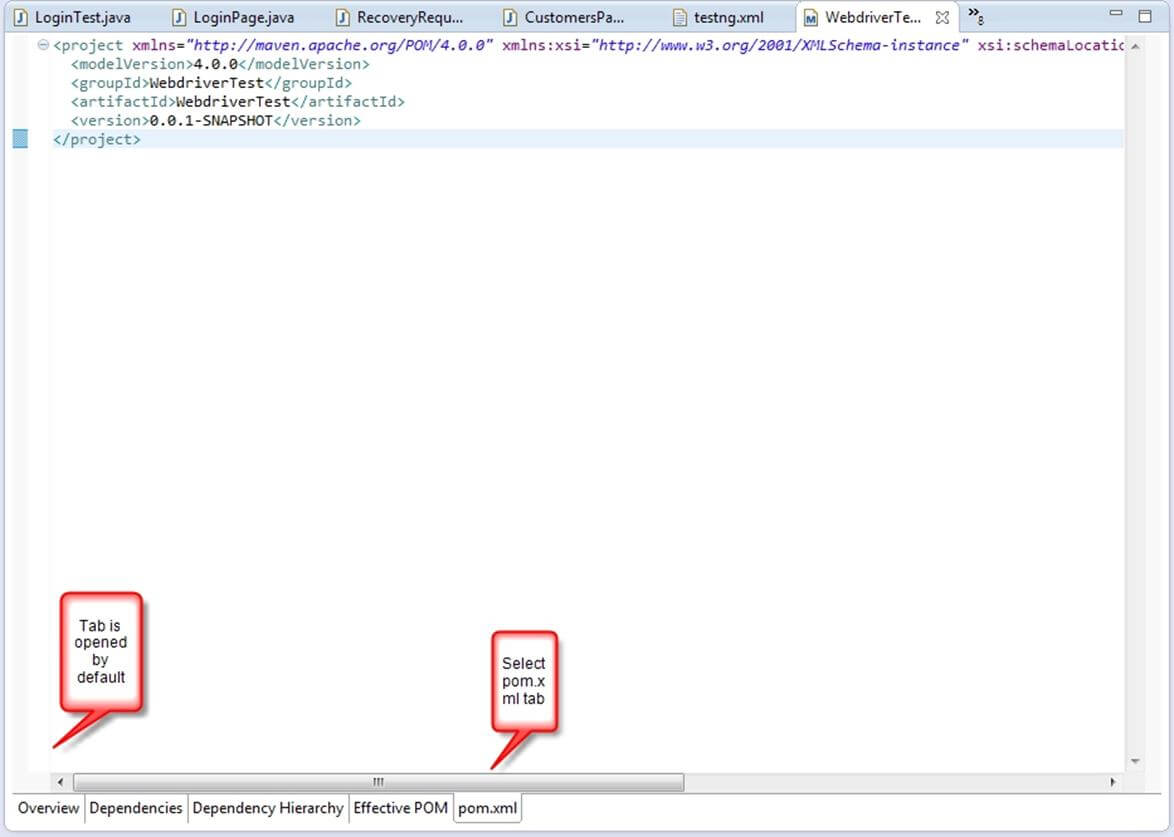
On the **Properties for JRE System Library** dialog box, make sure **Workspace default JRE** is selected and click OK



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



pom.xml file will Open in Editor section



**Step 8)** Add the Selenium, Maven, TestNG,[Junit](https://www.guru99.com/junit-tutorial.html)dependencies to pom.xml in the <project> node:

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>2.45.0</version>

</dependency>

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

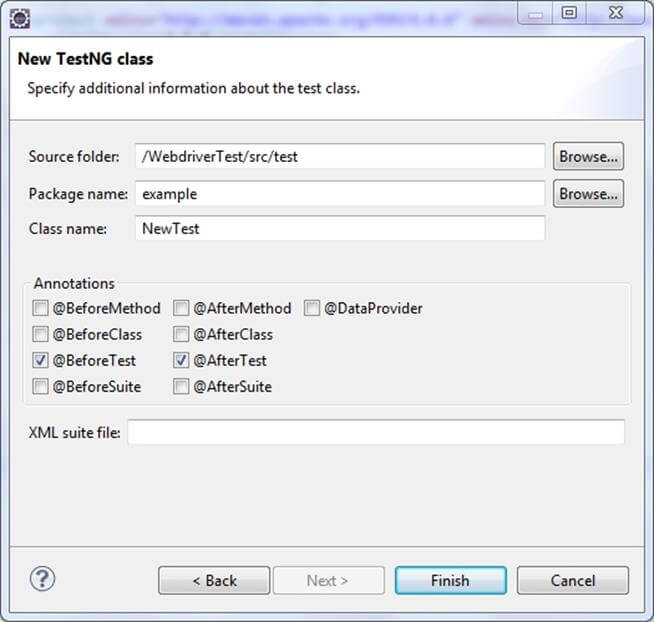
<version>6.8.8</version>

<scope>test</scope>

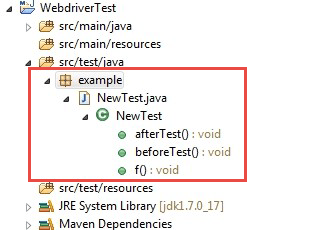
</dependency>

</dependencies>

**Step 9)**Create a New TestNG 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:

This code will verify the title of Guru99 Selenium Page

package example;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.Assert;

import org.testng.annotations.Test;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.AfterTest;

public class NewTest {

private WebDriver driver;

@Test

public void testEasy() {

driver.get("http://demo.guru99.com/test/guru99home/");

String title = driver.getTitle();

Assert.assertTrue(title.contains("Demo Guru99 Page"));

}

@BeforeTest

public void beforeTest() {

driver = new FirefoxDriver();

}

@AfterTest

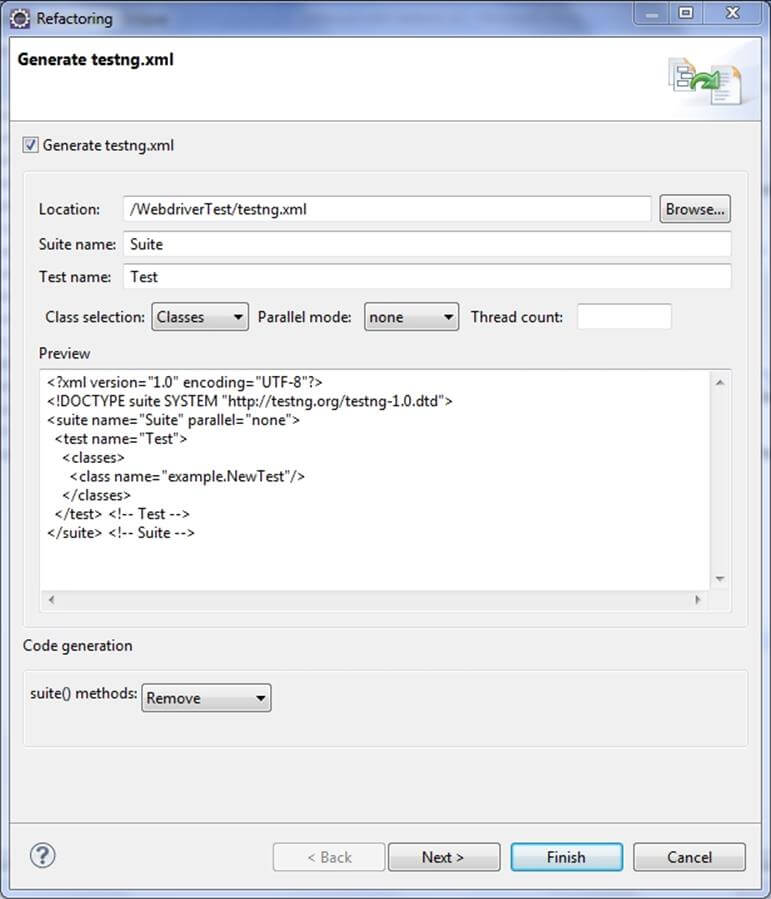
public void afterTest() {

driver.quit();

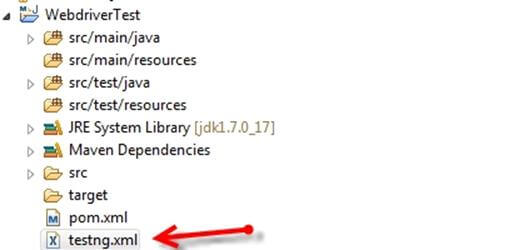
}

}

**Step 12)** Right-click on the WebdriverTest and select **TestNG** | **Convert to TestNG**.  
Eclipse will create testng.xml which says that you need to run only one test with the name **NewTest** as shown in the following screenshot:

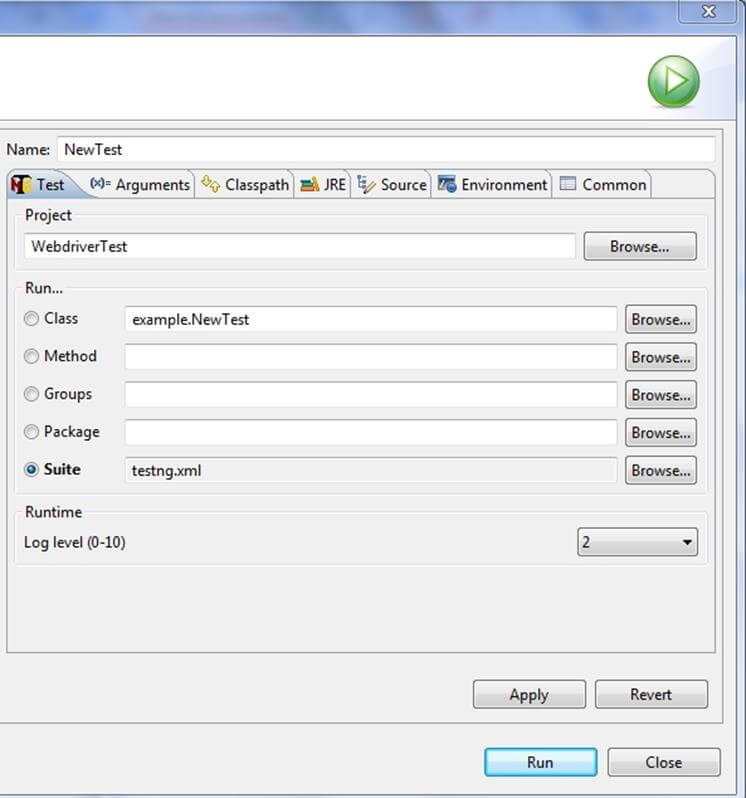


Update the project and make sure that file appears in the tree **Package Explorer** (right click on the project – Refresh).



**Step 13)**Now you need to run test through this **testng.xml.**

So, go to the **Run Configurations**and create a new launch **TestNG**, select the project and field **Suite** as **testng.xml** and click Run



Make sure that build finished successfully.

**Step 14)**. Additionally, we need to add

1. maven-compiler-plugin
2. maven-surefire-plugin
3. testng.xml

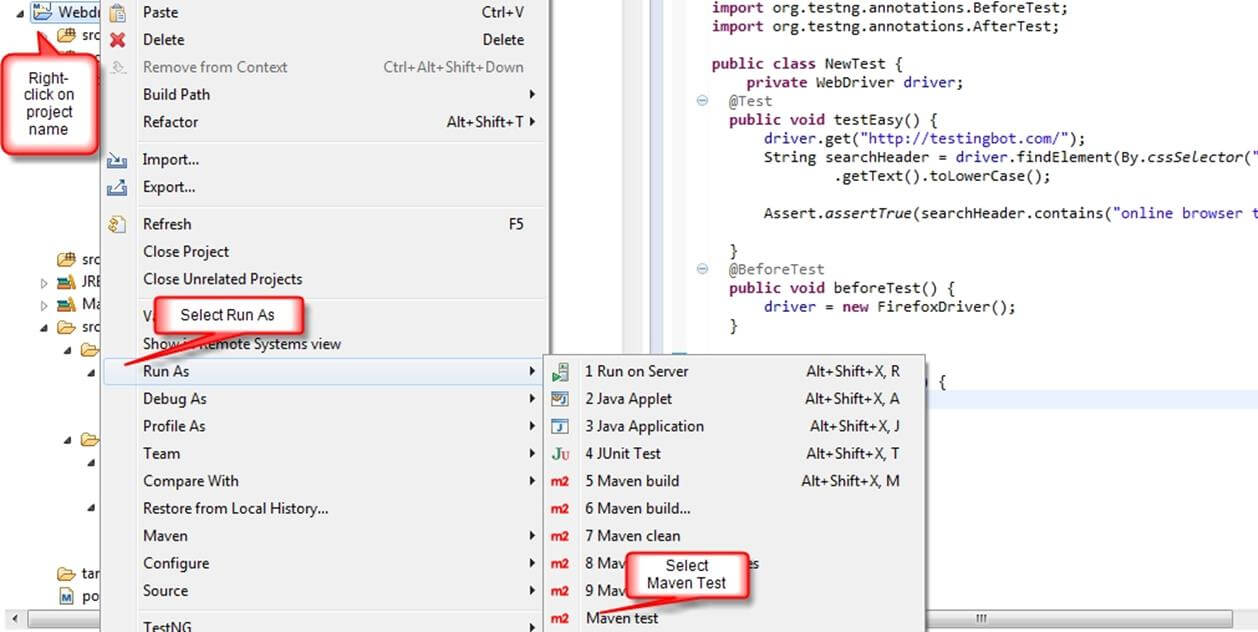
to pom.xml**.**

The maven-surefire-plugin is used to configure and execute tests. Here plugin is used to configure the testing.xml for TestNG test and generate test reports.

The maven-compiler-plugin is used to help in compiling the code and using the particular JDK version for compilation. Add all dependencies in the following code snippet, to pom.xml in the <plugin> node:



**Step 15)** 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.