Test Automation

(Eclipse / Selenium / JBehave / Spring)

for Test Engineers

Table of Contents

[Downloads and installs 3](#_Toc452990915)

[Eclipse IDE for Java EE Developers 3](#_Toc452990916)

[JDK (Java SE Development Kit) 3](#_Toc452990917)

[Apache Maven 3](#_Toc452990918)

[Selenium Webdrivers 3](#_Toc452990919)

[Configure Eclipse 4](#_Toc452990920)

[Create a project 4](#_Toc452990921)

[Configure JBehave plugin 5](#_Toc452990922)

[JBehave Run 6](#_Toc452990923)

[Create Story 6](#_Toc452990924)

[Stories folder 6](#_Toc452990925)

[Math.story 6](#_Toc452990926)

[Create steps 6](#_Toc452990927)

[Steps package 6](#_Toc452990928)

[MathSteps.java 6](#_Toc452990929)

[Create runner 6](#_Toc452990930)

[Stories package 6](#_Toc452990931)

[JBehaveRunner.java 6](#_Toc452990932)

[Run test 7](#_Toc452990933)

[Selenium Run 8](#_Toc452990934)

[Create pages (Page Objects) 8](#_Toc452990935)

[Pages package 8](#_Toc452990936)

[AbstractPage.java 8](#_Toc452990937)

[SearchPage.java 8](#_Toc452990938)

[Create runner 8](#_Toc452990939)

[SeleniumRunner.java 8](#_Toc452990940)

[Run test 9](#_Toc452990941)

[Selenium + JBehave Run 10](#_Toc452990942)

[Create Story 10](#_Toc452990943)

[PerformSearch.story 10](#_Toc452990944)

[Create steps 10](#_Toc452990945)

[PerformSearchSteps.java 10](#_Toc452990946)

[Create runner 10](#_Toc452990947)

[StoriesRunner.java 10](#_Toc452990948)

[Run test 10](#_Toc452990949)

[Project Explorer Structure 11](#_Toc452990950)

# Downloads and installs

## Eclipse IDE for Java EE Developers

1. Download the *Eclipse IDE for Java EE Developers* from:

<https://eclipse.org/downloads/>

1. Unzip anywhere.

## JDK (Java SE Development Kit)

1. Download the latest *Java SE JDK* from:

[Home] <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

[Wipro] [\\10.197.72.32\Mastercard\Programas\Java](file:///\\10.197.72.32\Mastercard\Programas\Java)

1. Run installer (take note where it’ll be installed).

## Apache Maven

1. Download the *Binary zip archive* from:

<https://maven.apache.org/download.cgi>

1. Unzip under the same folder as Eclipse’s folder.
2. [Wipro] Add proxy settings:
   1. Open C:\<maven folder>\conf\settings.xml
   2. Add the following in the <proxies> section:

<proxies>

<proxy>

<id>optional</id>

<active>true</active>

<protocol>http</protocol>

<username>wipro\[YOUR AD ID]</username>

<password>[YOUR WIPRO PASSWORD]</password>

<host>10.197.81.254</host>

<port>8080</port>

<nonProxyHosts>local.net|some.host.com</nonProxyHosts>

</proxy>

</proxies>

## Selenium Webdrivers

1. Download the *Internet Explorer Driver Server* for your IE version from:

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

More info/configuration if needed: <https://github.com/SeleniumHQ/selenium/wiki/ChromeDriver>

1. Download the *Google Chrome Driver* from:

[Home] <https://sites.google.com/a/chromium.org/chromedriver/>

[Wipro] [\\10.197.72.32\Mastercard\Programas\Selenium](file:///\\10.197.72.32\Mastercard\Programas\Selenium)

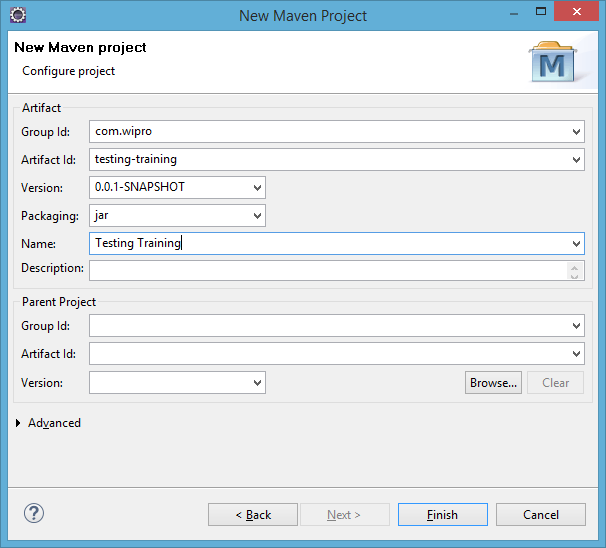
1. Unzip both into C:\Selenium\

# Configure Eclipse

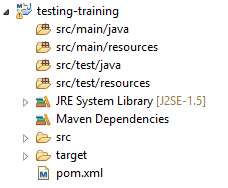
1. Open eclipse.exe.
2. Chose a workspace folder for your projects.
3. Open menu Window > Preferences > Maven > User Settings.
4. Map the User Settings field to C:\<maven folder>\conf\settings.xml.
5. Click Update Settings; Apply; OK.
6. Open menu Window > Preferences > Java > Installed JRE.
7. Click Add… > Select Standard VM > Click Next.
8. Click Directory… > Select the folder where Java was installed > Click Finish.
9. On the Installed JREs list, leave only the jdk you just added selected.
10. Click Apply; OK.

# Create a project

1. Open menu File > New > Maven Project.
2. Check options:
   1. *Create a simple project (skip archetype selection)*
   2. *Use default Workspace location*.
3. Click Next.
4. Fill the next step fields as below:



1. Click Finish; the structure below should be created:



1. Open the pom.xml file, add the following inside the <project> section:

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

</dependency>

<dependency>

<groupId>org.jbehave</groupId>

<artifactId>jbehave-spring</artifactId>

<version>3.9.1</version>

</dependency>

<dependency>

<groupId>org.jbehave.web</groupId>

<artifactId>jbehave-web-selenium</artifactId>

<version>3.6-beta-2</version>

</dependency>

<dependency>

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

<artifactId>selenium-java</artifactId>

<version>2.53.0</version>

</dependency>

<dependency>

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

<artifactId>selenium-firefox-driver</artifactId>

<version>2.53.0</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.3</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</build>

1. Save All.
2. In the Project Explorer pane, right click the project > Maven > Update Project; click OK.
3. JRE System Library should now show [JavaSE-1.8].
4. In the Project Explorer pane, right click the project > Run As > Maven install.
5. Console should show BUILD SUCCESS.

# Configure JBehave plugin

1. In Eclipse, open menu Help > Install New Software…
2. Fill the Work with: with [*http://jbehave.org/reference/eclipse/updates/*](http://jbehave.org/reference/eclipse/updates/); click Add…
3. Select the JBehave feature containing JBehave Eclipse; click Finish.
4. Follow standard installation procedure (an Eclipse restart will be required).

# JBehave Run

## Create Story

### Stories folder

1. In the Project Explorer pane, right click the src/main/resources folder > New > Other.
2. Select General > Folder from the list; click Next.
3. Fill the Folder Name as *stories*; click Finish.

### Math.story

1. In the Project Explorer pane, right click the stories folder > New > File.
2. Fill the File Name as *Math.story*; click Finish.
3. The new Math.story is created; content attached:



## Create steps

### Steps package

1. In the Project Explorer pane, right click the src/main/java folder > New > Package.
2. Fill the Name as *com.wipro.testing.training.steps*; click Finish.

### MathSteps.java

1. In the Project Explorer pane, right click the com.wipro.testing.training.steps package > New > Class.
2. Fill the Name as *MathSteps;* click Finish.
3. The new MathSteps.java is created; should contain:
   1. Extends Steps (import from org.jbehave.core.steps).
   2. Declaration of int x.
   3. Public @Given step to get the first value.
   4. Public @When step to multiply the first value with a second value.
   5. Public @Then step to validate the expected result.



## Create runner

### Stories package

1. In the Project Explorer pane, right click the src/main/java folder > New > Package.
2. Fill the Name as *com.wipro.testing.training.stories*; click Finish.

### JBehaveRunner.java

1. In the Project Explorer pane, right click the com.wipro.testing.training.stories package > New > Class.
2. Fill the Name as *JBehaveRunner*; click Finish.
3. The new JBehaveRunner.java is created; should contain:
   1. Private static Embedder declaration.
   2. Private static List<String> declaration of storyPaths mapping the Math.story file.
   3. Public main method that:
      1. Adds the MathSteps steps to the embedder.
      2. Runs the stories from the storyPaths list.



## Run test

1. In the Project Explorer pane, right click the JBehaveRunner.java class > Run As > Java Application.
2. The application should run: the Console will show the application was run with no errors.
3. You can force an error by changing the expected result in the Math.story file.

# Selenium Run

## Create pages (Page Objects)

### Pages package

1. In the Project Explorer pane, right click the src/main/java folder > New > Package.
2. Fill the Name as *com.wipro.testing.training.pages*; click Finish.

### AbstractPage.java

1. In the Project Explorer pane, right click the com.wipro.testing.training.pages package > New > Class.
2. Fill the Name as *AbstractPage*; select the *abstract* modifier; click Finish.
3. The new AbstractPage.java is created; should contain:
   1. Private static WebDriver declaration.
   2. Public constructor, checking if the driver already exists. If it doesn’t exist, creates a new WebDriver (For each of the browsers).
   3. Public accessor to read the WebDriver.



### SearchPage.java

1. In the Project Explorer pane, right click the com.wipro.testing.training.pages package > New > Class.
2. Fill the Name as *SearchPage*; click Finish.
3. The new SearchPage.java is created; should contain:
   1. Extends AbstractPage.
   2. Find searchField WebElement.
   3. Public constructor with a call to the superclass constructor and initiating the page elements.
   4. Method to send keys to the searchField WebElement.
   5. Method to navigate to <https://www.google.com>.



## Create runner

### SeleniumRunner.java

1. In the Project Explorer pane, right click the com.wipro.testing.training.stories package > New > Class.
2. Fill the Name as *SeleniumRunner*; click Finish.
3. The new SeleniumRunner.java is created; should contain:
   1. Public @Test method that:
      1. Creates a SearchPage instance.
      2. Uses the method to navigate to google.
      3. Uses the method to type something in the searchField.



## Run test

1. In the Project Explorer pane, right click the SeleniumRunner.java class > Run As > Junit Test.
2. The test should run: a browser will open and perform a search on Google of what was specified in the SeleniumRunner.java class.

# Selenium + JBehave Run

## Create Story

### PerformSearch.story

1. In the Project Explorer pane, right click the stories folder > New > File.
2. Fill the File Name as *PerformSearch.story*; click Finish.
3. The new PerformSearch.story is created; content attached:



## Create steps

### PerformSearchSteps.java

1. In the Project Explorer pane, right click the com.wipro.testing.training.steps package > New > Class.
2. Fill the Name as *PerformSearchSteps;* click Finish.
3. The new PerformSearchPage.java is created; should contain:
   1. Declaration of a new SearchPage() – this will use the previously created SearchPage.java methods.
   2. Public @Given step to navigate to google.
   3. Public @When step to type something in the searchField.
   4. Public @Then step to validate a dummy assertion.



## Create runner

### StoriesRunner.java

1. In the Project Explorer pane, right click the com.wipro.testing.training.stories package > New > Class.
2. Fill the Name as *StoriesRunner*; click Finish.
3. The new StoriesRunner.java is created; should contain:
   1. Extends JUnitStories.
   2. Public constructor with a call to the superclass constructor.
   3. Public InjectableStepsFactory mapping the PerformSearchSteps() class.
   4. Protected List<String> declaration of storyPaths mapping the PerformSearch.story file.



## Run test

1. In the Project Explorer pane, right click the StoriesRunner.java class > Run As > Junit Test.
2. The test should run: a browser will open and perform a search on Google of what was specified in the PerformSearch.story file.

# Project Explorer Structure

Project Explorer structure so far:

