Selenium – First Test Case in Selenium

Lecture Notes

A Complete Guide for Beginners

Softech Solutions Inc.

www.softechnosolutionsgroup.com

saney.alam@softechnosolutionsgroup.com

Selenium – First Test Case in Selenium

Table of Contents	*Page*

First Selenium WebDriver Test Case

In this chapter we will write a very basic automation script of Selenium and try to understand the meaning of import statements, comments in java, print statements and how to instantiate a browser using WebDriver object.

Prerequisites for writing Test Case in Eclipse using Selenium WebDriver are:

- Download and Install Java
- Download and Start Eclipse
- Download WebDriver Java Client
- Configure Eclipse with Selenium WebDriver

Once you are done with the above steps, the next steps need to do for running the First Test Case is:

- Create a New Project as "LearnSelenium" in Eclipse
- Create a New Package "automationFramework" in the Project
- Create a New Class as "FirstTestCase" in the Package
- Add Selenium WebDriver Jars to the Project

Now you are all set to write your First Selenium Automation Test Script. To practice automation we will try the following web application "***http://ebfs.bruteforcesolution.net/ebfs/index.php***". Let's just automate a very simple scenario and see if everything is set up correctly.

Scenario to Automate

- Launch the Firefox browser
- Open website "***http://ebfs.bruteforcesolution.net/ebfs/index.php***"
- Print a Message to display that the website is opened successfully
- Wait for 5 Seconds
- Close the Browser

So far we have not done any learning, I would suggest to create class named FirstTestCase and type the below code in the class and run the test.

```
package automationFramework;

public class FirstTestCase {

	public static void main(String[] args) {

	// Create a new instance of the Firefox driver
	WebDriver driver = new FirefoxDriver();

	//Launch the Online Store Website
	driver.get("http://ebfs.bruteforcesolution.net/ebfs/index.php")
;

	// Print a Log In message to the screen
	System.out.println("Successfully opened the website…");

	//Wait for 5 Sec
	Thread.sleep(5000);

	// Close the driver
	driver.quit();
    }
}
```

How to Resolve the Java Compilation Error?

Notice that there are two errors in the code at the statement '**WebDriver driver = new FirefoxDriver();**', as it is underlined with the red color line. ***Red colored Underlined word*** represents the **error** in the Eclipse IDE. Eclipse is a powerful tool and it gives us the very wonderful features and one of the best features is the '**Solution Suggestion'**. To use this feature, just **mouse hover** on the error keyword '**WebDriver'** and Eclipse itself will try to give you the possible solutions to fix the error.

Note: This suggestion will work only when the WebDriver jars are added/ imported to the project, if not then Eclipse would not be able to suggest the correct solution.

Select the very first solution provided by the Eclipse IDE. You will notice that after selecting the first solution, it will add one **import** statement at the top saying '**`import org.openqa.selenium.WebDriver;`**'

What is *import* Statement?

The *import statement* in *Java* allows referring to classes which are declared in other packages to be accessed without referring to the full package name. It means that rather declaring statement like:

```
org.openqa.selenium.WebDriver driver = new FirefoxDriver();
```

You can simple ***import*** package at the top of the class and write the statement like:

```
import org.openqa.selenium.WebDriver;
public class TestCase {
      public static void main(String[] args) {
            WebDriver driver = new FirefoxDriver();
        }
}
```

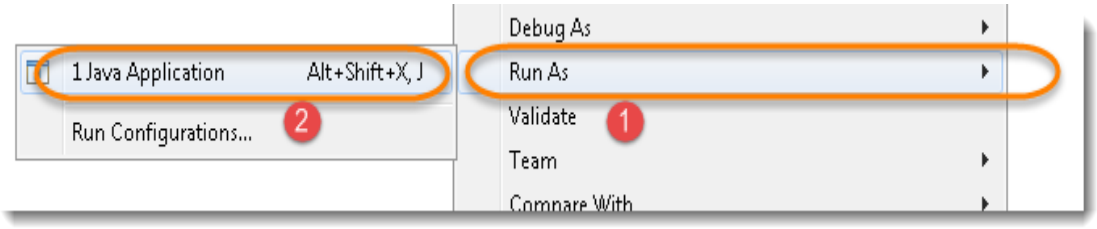
The same way try resolving the second error of the same statement which is at **'FirefoxDriver()'** *keyword. Just do the same thing, bring the cursor over the error keyword and again Eclipse will bring up few possible solutions and select the very first solution.*

This will add the statement 'import org.openqa.selenium.firefox.FirefoxDriver;' and the error will be gone.
Once you resolve these error one more error will pop up at the statement 'Thread.sleep(5);', now we know what to do right? Just mouse hover the error and choose the best solution. This time it will ask to select the 'Add throws declaration' and after selecting that it will add 'throws InterruptedException' at the method declaration.
For now just do not focus on what it is and what does it do, just ignore this part and we will take this later in the course.

Run the Java Test case

By now all the errors are resolved and we are good to go with the execution of our first automated test cases.

1) Now, to start the test just select **Run > Run As > Java Application** Or **Right Click** on Eclipse code and Click **Run As >** *Java Application***.**



2) After a few Seconds a **Firefox Browser** will open and you will see that with the help of your script, Selenium will **Launch** the **Online Store**, display the **Successful Message** and **Close** the browser.
Once the execution is finished, you will see the message in **Console** section of the **Eclipse IDE** displaying:

Successfully opened the website …

Explanation of the Code

Import Packages/Statements

In java we use import statements to use the classes present in another packages. In a way ***import*** keyword is used to import built-in and user-defined packages into your java source file. So that your class can refer to a class that is in another package by directly using its name. To get started, you need to import following two packages:

1. ***`org.openqa.selenium.WebDriver`*** References the WebDriver interface which is required to instantiate a new web browser.
2. ***`org.openqa.selenium.firefox.FirefoxDriver`*** References the FirefoxDriver class that is required instantiate a Firefox specific driver on the browser instance instantiated using WebDriver interface.

If your test needs more complicated actions such as accessing another class, taking browser screenshots, or manipulating external files, definitely you will need to import more packages and that we are going to learn in following chapters.

Instantiating an Objects

This is the way to instantiate a driver object in Selenium. We will not get in to the complexity of the code in the first chapter. We have a detailed chapter on this in the later tutorial.

```
WebDriver driver = new FirefoxDriver();
```

So for the sake of simplicity just remembers that this code wills instantiate a new Firefox driver in safe mode which will have no plugins and extensions loaded. This instant of Firefox browser will be different from the Firefox which is installed on your desktop.

Print Statement

Make a note of this statement, as you are going to use this statement in the selenium code again and again. This will display the message in the **Console Window**.

```
System.out.println("Successfully, opened the website…");
```

Shortcut for print statement is, type '**sysout**' and then press **'CTRL + SPACE'** button. This will bring the complete print statement automatically. Define message with in the parenthesis.

Sleep Statement

This is again more often used statement. As selenium is very fast and sometimes it is difficult to see the operations done or the action performed by selenium. One can use ***Thread.sleep()*** statement to delay the execution. Here **5** is the number of seconds.

Thread.sleep(5);

Although this method is not suggested and we have much smarter option available in Selenium which is **Smart Waits** and that will be introduced later in this tutorial.

Comments

In java the comments are marked by double slashes '//'.

```
//Comments
```

Whatever statement comes just after the double slashed, Java execution engine will ignore that and move to the following statement mentioned just below the commented statement.