Selenium 4 Beginners

Rex Jones II

How To Install Selenium, Java, Eclipse, Drivers & TestNG

Table of Contents

Introduction

The Java Development Kit (JDK) is required for developing and running programs in Java.

Eclipse IDE will be used to compile and run our automation Test Scripts. IDE stands for Integrated Development Environment which contains a code editor for writing our automation Test Scripts.

Selenium has 4 components and all of those components include the word Selenium. The 4 components are Selenium IDE, Selenium RC, Selenium WebDriver, and Selenium Grid. Most of the times, when a person refers to Selenium, they are referring to Selenium WebDriver. We are going to focus on Selenium WebDriver which is mistakenly called an automation tool. In reality, for Java, Selenium WebDriver is a collection of jars. JAR is an acronym for Java ARchive which has many packages consisting of classes and methods.

A Browser Driver is sometimes called a Driver. It is a file that manages communication between the browser and Selenium WebDriver. We need a driver before we begin automation on a browser.

TestNG is one of the test frameworks for Java. TestNG stands for Test Next Generation which joins the programming and testing aspects of automation. In fact, TestNG was designed so testers can include a diverse group of test types with their automation efforts.

The first installation will be the Java Development Kit (JDK).

Demo

Java Development Kit (JDK)

We are going to search for JDK version 9. Select Java SE Development Kit 9. Now we are at Oracle's website. Click the radio button to Accept License Agreement. There is a JDK for each platform. The platforms are Linux, Mac, Windows, and Solaris. I have a Windows platform and will download the executable file for version 9.0.4. Save the executable file in my Downloads folder then wait for it to finish downloading.

It is finished downloading. Open the folder then prepare for installation by opening the JDK executable file. Click Next in the Installation Wizard. We are getting ready to install the Development Tools, Source Code, and Public JRE. This is the JRE that is included with JDK.

While the Status is in Progress, let's go to the JDK folder for Version 9 within the Java folder. We see a jre executable application within the JDK folder. After the Status is complete, we will be prompted to install JRE.

Click Next to install jre-9.0.4. Close the Complete dialog since jre has been Successfully Installed. Now, let's go back to the Java folder and we see 4 folders: 2 folders for version 9 and 2 folders for version 8. Both versions have a jdk folder and a jre folder.

Open JDK for Version 9 then open the bin folder which has many applications. The 2 primary applications within the bin folder are java.exe and javac.exe: java.exe is the standard Java interpreter while javac.exe is the Java compiler.

Let's copy the bin folder's path so we can add the bin folder's path to Environment Variables. To copy the bin folder's path, select Shift key and click the right mouse button at the same time to see Copy as path. You will not see Copy as path if you only click the right mouse button without selecting Shift.

To go to Environment Variables, we navigate to System, type System, select System, click Advanced system settings, within the Advanced tab, click Environment Variables, go to Path within System variables section, click Edit, then New and paste the bin folder's path. There

Rex Jones II, CSTE, TMap
https://www.linkedin.com/in/rexjones34

are 2 reasons for adding the bin folder to Environment Variables. The first reason is to help Windows Operating System search the path entries for Java's executable applications. The second reason is to bypass writing the full path to the executable file every time we run it.

Notice, I have a bin directory for Version 8 and Version 9. There should only be 1 bin directory for JDK in the path at a time. Therefore, I will delete Version 8 and keep Version 9. Click OK three times (1, 2, 3).

Now, let's verify through the command prompt that we have installed Java Version 9. I will go to the Run prompt then type cmd. Within the command prompt, type java space hyphen version (java -version) then hit Enter. We see java version "9.0.4". Java Version 9 has been successfully installed.

The next installation will be Eclipse IDE.

Eclipse IDE

I will search for Eclipse by typing Download Eclipse IDE then select Eclipse Downloads. Eclipse Oxygen is the current release for Eclipse IDE. We click the Download 64 Bit button. Next, we download the file which is the Eclipse Installer by clicking the Download button or the hyperlink eclipse-inst-win64.

Save the Eclipse Installer, go to my Downloads folder, then open the Eclipse Installer. We see 2 Java options within this Eclipse Installer: Eclipse IDE for Java Developers and Eclipse IDE for Java EE Developers. We can install either one of the Java options. However, I prefer to install the second option so I will select Eclipse IDE for Java EE Developers. The EE stands for Enterprise Edition which comes with many pre-installed plug-ins.

Click the Install button. We click the Launch button to open the Eclipse application. In this demo, I did not have to accept a license agreement. If you come across a checkbox and button, click both to accept the agreement. After launching Eclipse, we provide a name for our workspace. The purpose of workspace is to group projects. It's important we group projects that are similar to each other in a workspace. I will name the workspace Social Media then click the Launch button.

Now, let's create a new Java Project. I will minimize the Welcome tab, click File then navigate to New. We have the option of selecting Project or Other. Both options will take us to the same dialog which is the Select a Wizard. In this dialog, we select Java Project then click the Next button to add a Project Name. Notice how the JRE section defaults and displays our Java Version 9. The JRE is used for compiling and launching our Java Programs. I created a separate video showing how to change the default version to jre 9 if you do not see it.

Let's call this project Online Training - Teaching. Click Finish then Open Perspective. Perspective is a group of views. A view is a sub window such as Package Explorer, Task List, Outline, and Problems. We have the option of customizing our Perspective by adding or removing a view. We can add a view by going to Window -> Show View. However, I am going to minimize the views to show more of the work area.

Within Package Explorer, we see our project Online Training – Teaching, JRE System Library, and src. src is the source folder.

Our project is used for holding our folders, libraries, and files.

The JRE System Library is a library added by default to Eclipse. We see the path of each jar file starting with C:\Program Files.

The source folder is used for storing sub folders and source files.

Next, we are going to configure the Selenium jars.

Selenium

I am going to search for Selenium Download then go to Selenium's website. We navigate to the Selenium Client & WebDriver Language Bindings section to see the most recent Selenium versions. Five languages are available for download: Java, C Sharp, Ruby, Python, and Javascript.

Rex Jones II, CSTE, TMap
https://www.linkedin.com/in/rexjones34

We will download Java but will not download the latest version of Selenium 3.11. Let's navigate back to the top and go to Previous Releases. The Previous Release page shows releases for Selenium 2 and Selenium 3: starting with version 2.39 and ending with version 3.9.

I will download Selenium 3.9. Out of habit, I tend not to use the latest version of most products until it has been in the market for awhile. That's why I did not download Selenium 3.11. There are some people that still use Selenium 2.53 which was the last version of Selenium 2. They use Selenium 2.53 because it is a stable release and Selenium 3.0 had a ton of problems when it came to the market.

On this Index of 3.9 page, you see several zip and jar files. We have the option of downloading the selenium-java or selenium-server zip files. These 4 zip files will help us automate using Selenium WebDriver. I am going to download selenium-java-3.9.0 zip file and selenium-server-standalone-3.9.0 jar file to my Downloads folder. Go to the location of both files. We see the zip and the jar file.

Next, we must unzip all of the files by clicking the right mouse button and selecting Extract All. The zip file must be extracted so we can configure all of the Selenium jars. Now, we see a folder and zip file for selenium-java-3.9.0.

Let's go back to Eclipse and configure Selenium. We configure Selenium by navigating to Build Path. To get to Build Path, we can right click the project, src, or JRE System Library, Build Path followed by Configure Build Path. Select the Libraries tab and click Modulepath.

Here we see the JRE System Library has already been added as a Library. Now, let's add the Selenium jars by clicking the Add External JARs button. Open the selenium-java-3.9.0 folder, add both client-combined-3.9.0 jars. It is optional to add the client-combined-3.9.0-sources jar. Click the Add External JARs button again but this time we are going to add all of the jar files within the libs folder. Select shortcut keys Control A (CTRL + A) to select all jar files then click Open. Finally, we add the selenium-server-standalone jar by going to the Downloads folder. Let me show you what happens if someone attempts to add the selenium-

java zip file. We need the jar files but only the zip file shows up. I will remove the zip file by clicking the Remove button.

Click the Apply and Close button. Another library called Referenced Libraries has been added to include the jar files we configured for Selenium.

Next, we are going to download the Browser Drivers for Internet Explorer, Google Chrome, and Firefox.

Browser Drivers

The first driver we are going to download is the IEDriverServer. IEDriverServer can be downloaded from 2 locations. The 2 locations are Index of your Selenium version page and Selenium's Download page. We have Selenium 3.9 so we are going to download the driver from this Index of 3.9 page. Do you see how there are 2 driver options? IEDriverServer Win32 and IEDriverServer x64.

The difference between 32 and 64 is how your computer's processor better known as CPU handles information. There are several ways to verify if you have a 32-bit or 64-bit Operating System. One of the ways is through System whereby you can look at System type. In my case, I have a 64-bit Operating System, x64-based processor. Most people probably have a 64-bit if they have at least 4 GigaBytes (GB) of RAM. If you have a 32-bit Operating System then your System type will show x86-based processor.

Let's go back to the Index page and download IEDriverServer x64. We can also download the driver from Selenium's Download page. The Internet Explorer Driver Server shows driver version 3.11.1 which is the most recent version. You would use this version if you downloaded Selenium 3.11.

We are going to scroll down to the Third Party Drivers, Bindings, and Plugins section for the other drivers. There are many drivers but we will focus on Firefox and Chrome. It is important to know your browser version so you can know which driver version to download.

Let's take a look at the browser version for Firefox and Chrome. In Firefox, select the ALT key or click the right mouse button and check Menu Bar to bring up the menu. Go to Help then select About Firefox to view Firefox's browser version.

Rex Jones II, CSTE, TMap
https://www.linkedin.com/in/rexjones34

In Chrome, we click the 3 three vertical dots, then navigate to Help, and select About Google Chrome. We see the browser version for Chrome is 66.

Check your browser version, if you come across a problem, where you execute your Test Scripts and they pass then you execute them again and they Fail. It may be a situation where you must update your driver version to match your browser version. For example, my Chrome version is 66. Imagine if I execute some Test Scripts today and they pass but next week they fail. It's probably a result of Chrome automatically updating to version 67. Therefore, my driver version and browser version are no longer compatible.

I'll show you how to verify which driver and browsers are compatible. Click Change Log for Google Chrome Driver. The Change Log shows us which browser version is supported by the driver verison. We see ChromeDriver v2.36 Supports Chrome v63 – 65. The next section shows ChromeDriver v2.35 Supports Chrome v62 – 64. Hmm, Hmm, okay, notice we do not see what ChromeDriver version supports 67.

Sometimes these websites are not updated with the updated information. Let's try this. I will change 36 to 37 and see what happens. There it is. ChromeDriver v2.37 Supports Chrome v64 – 66. I will go back to Selenium's Download page and we see it shows 2.36. Let's do the same by changing 36 to 37 and we see Index of 2.37. Do you see how there is not a 64-bit for Windows? There's only a 32-bit for chromedriver? It is no big deal, we can download chromedriver_win32 if we have a 64-bit Operating System.

Now, let's download the driver for Firefox. The driver for Firefox is called GeckoDriver. It is called GeckoDriver and not FirefoxDriver because Gecko is the web browser engine name that is developed as part of Firefox. We click 0.20.0 to download and view documentation for the GeckoDriver. I am going to download version 0.19.0. Firefox 55.0 and greater and Selenium 3.5 and greater are supported by version 0.19.0. Click the link for win64 zip. Save the file.

In the Downloads folder, we see all 3 zip files. I prefer to create a folder called Drivers then add all of the drivers to the Drivers folder. This makes it easier to organize my drivers in one location. Next, we can unzip the zip files or copy and paste the executable files. Each of these

drivers helps us automate on Chrome, Firefox, and Internet Explorer. The driver operates like a bridge between Selenium WebDriver and the browser. I plan to use Chrome and Firefox although I have 3 drivers.

Finally, we have the last installation which is TestNG, the Java Test Framework.

TestNG

There is more than 1 way to install TestNG. However, I believe the quickest way to install TestNG is through Eclipse Marketplace. We navigate to Eclipse Marketplace by going to the Help menu then selecting Eclipse Marketplace. Type TestNG in the Find text field then click the Go button. TestNG shows up with an Install button. Click the Install button, make sure all checkboxes are checked, click the Confirm button, select the radio button called "I accept the terms of the license agreement" then click the Finish button. Click Install Anyway on the Security Warning prompt. Finally click Restart Now on the Software Updates prompt. Launch the Workspace. We verify TestNG is installed by making sure the TestNG library is located under our project.

If the library is not located under our project, then we can add the library by right clicking the project, src, JRE System Library, or Referenced Libraries then selecting Build Path and Configure Build Path. Click the Add Library button, select TestNG, click Next, Finish, then the Apply and Close button. Before clicking the Apply and Close button, we see TestNG has been added as a Library. TestNG is located under our project as a Library.

Now, we are finished setting up the Java Development Kit (JDK), Eclipse IDE, Selenium, Browser Drivers, and TestNG.

I created a document with step-by-step instructions to show you How To Set-Up Selenium, Java, Eclipse, Browser Drivers, and TestNG. Feel free, to download the document for free if you want a copy at https://tinyurl.com/How-To-Set-Up-Selenium. You will see this page "How To Set-Up Selenium And Its Required Components" after going to the URL.

Here's the document starting at the Table of Contents page: Verify the System Type, Install Java Development Kit (JDK), Install Eclipse IDE, Configure Selenium WebDriver, Download Browser Drivers, and Install TestNG. I will show you the Configure Selenium WebDriver section which have steps for downloading Selenium 3.11.

(Step 1) Go to Selenium's Download page (Step 2) Download the Selenium Standalone Server (Step 3) Download the client driver for Java (Step 5) Unzip the zip files (Steps 9 through 21) Add the External Jars to Eclipse

Print Statements

I will end this demo tutorial with a print statement saying "Hello to the World and Welcoming you to Selenium 4 Beginners".

We go to Eclipse and start by creating a class. To create a class, we select File, New, then Class. We will name the class PrintStatements then click the Finish button. Let's rename the class by adding an underscore.

First you must select the item then go to File and select Rename. Do you see F2? That's the shortcut key for renaming an item if you do not want to navigate through the menu. Here's the Rename dialog. The second way is to use the shortcut key F2 to bring up the Rename dialog. The third way is to right click the item, select Refactor then Rename. I am going to add an underscore in between Print and Statements then click the Finish button.

Minimize the Package Explorer, write a TestNG annotation called At Test "@Test" and create a Java method called printHelloWorld. Starting with public void printHelloWorld.

Next, we will add a print statement using shortcut sysout - sysout Control Space (CTRL + Space), add 2 double quotes. Now, we add Hello World !!! See the red X, that's because we have not imported the "@Test" annotation. So we hover over "@Test" then select Import 'Test' which is org.testng.annotations. I will create another TestNG annotation of At Test "@Test" and a Java method called printWelcomeMessage - printWelcomeMessage. Print Statement - sysout, Control Space (CTRL + SPACE). The message will say Welcome To Selenium 4 Beginners. We run by going to the Run menu then Run again. The Console shows both print statements:

Hello World !!! and Welcome To Selenium 4 Beginners. Both PASSED:
The Test Results tab also shows both print statements Passed.

Questions, Answers & Practice

This is the Questions, Answers, and Practice part. I forgot to mention this part at the beginning of the video but there will be a few questions and answers plus something for you to practice.

Questions, Answers, and Practice.

Questions

1. What is required for developing and running programs in Java?
2. Is Selenium WebDriver an automation tool?
3. What is the driver name for Internet Explorer, Google Chrome, and Firefox?

Answers

1. What is required for developing and running programs in Java? **Java Development Kit (JDK)**

2. Is Selenium WebDriver an automation tool? **No, it is not an automation tool**

3. What is the driver name for Internet Explorer, Google Chrome, and Firefox? **Internet Explorer is IEDriverServer, Google Chrome is chromedriver, Firefox is geckodriver**

Practice

- Install Java Development Kit (JDK)
- Install Eclipse IDE
- Configure Selenium
- Install Browser Drivers
- Install TestNG

That's it for How To Set-Up Selenium and It's Required Components. In the download zip file, you will find the step-by-step instructions, the PowerPoint presentation, and the Print

Rex Jones II, CSTE, TMap
https://www.linkedin.com/in/rexjones34

Statements code. Next Tuesday, I will demo the Building Blocks for Selenium covering an overview of Java, TestNG, and Selenium. Once again, my name is Rex Allen Jones II and Thank You.