Selenium Web Driver – TestNG – Test automation

For this part you will be creating automated tests to validate a service for registering and editing customers. You will create a test script for each task described underneath. The delivery will be the test scripts and a screenshots of the test report generated.
The page that you will be using is this demo page: https://itera-qa.azurewebsites.net

Task 1: Create a new user

Test Case 1:

1. Launch new browser
2. Navigate to URL https://itera-qa.azurewebsites.net
3. Click on “Sign up”
4. Fill in the form
5. Click on Submit button
6. Check that the new user registered successfully
7. Close the browser

Task 2: Login, Create customer, update customer info, delete customer

You will run all test cases together because all test cases dependent each other.

Test Case 1 - Login:

1. Launch new browser
2. Navigate to URL https://itera-qa.azurewebsites.net
3. Click on “Login”
4. Login with valid user
5. Check that you logged in successfully

Test Case 2 – Create customer:

1. On the Dashboard click on “Create new” button
2. Fill in the form
3. Click on “Create”
4. Check that new customer created successfully

Test Case 3 – Update customer:

1. Find the customer that you created
2. Click on “Edit”
3. Change “Phone number”
4. Click on “Save”
5. Check that phone number is updated

Test Case 4 – Delete customer:

1. Find the customer that you created
2. Click on “Delete”
3. Check that customer that you created is deleted from the list
4. Log out

Selenium Automation set up with TestNG and Eclipse

Author: Özkan Murat

ozkan.murat@itera.no

Abstract: This article is a guide for the beginners when they use Selenium WebDriver, TestNG and Eclipse for automation testing

1. Introduction

Selenium Web driver has becoming a very popular testing tool as it is an open source. This is a GUI automation tool. This is beginner`s guide to install and use Selenium with TestNG and Eclipse IDE. This gives a step by step installation guide to Kick-start an Automation journey.

This document will describe one of the ways of setting up the environment automating Web Applications.

1.1. Install JDK 1.8:

- Download Windows/Mac JDK 1.8 (Java SE Development Kit 8u162) from
  http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
- Install JDK 1.8

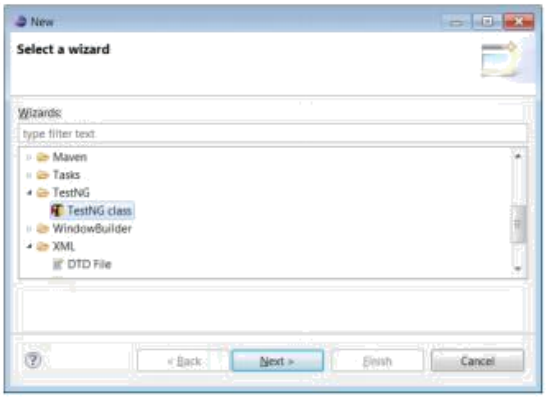
1.2. Install Eclipse IDE for Win/Mac

- Download Eclipse version Oxygen or latest version from here:
  https://www.eclipse.org/downloads/
- Install Eclipse IDE for Java Developers

- Give project location as Workspace

1.3. Install TestNG

- Open the eclipse.exe
- Go to Help – Install New Software
- Add link http://beust.com/eclipse to the field Work with... and click Add...
- Call the link TestNG and click OK
- It will after some time come up a selection called TestNG in the field underneath, select this.
- Click Next until you can finish the installation
- You might get a message asking if you are sure that you want to install this, click Install Anyway.
- And restart Eclipse
- Once the installation is complete, open File> New> Other(or Ctrl + N) > Testng Class must be listed in the window



1.4. Install Chrome

- Download Google Chrome
- Install Google Chrome

1.5. Install Chromedriver for Selenium

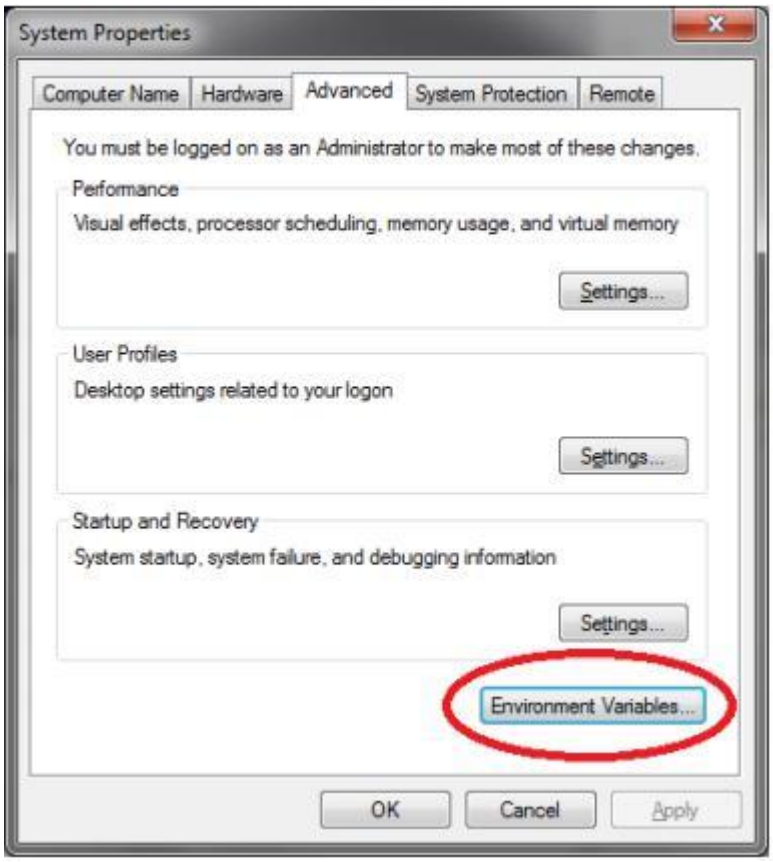
- Download chromedriver for Win/Mac/Linux from

  https://chromedriver.storage.googleapis.com/index.html?path=2.45/

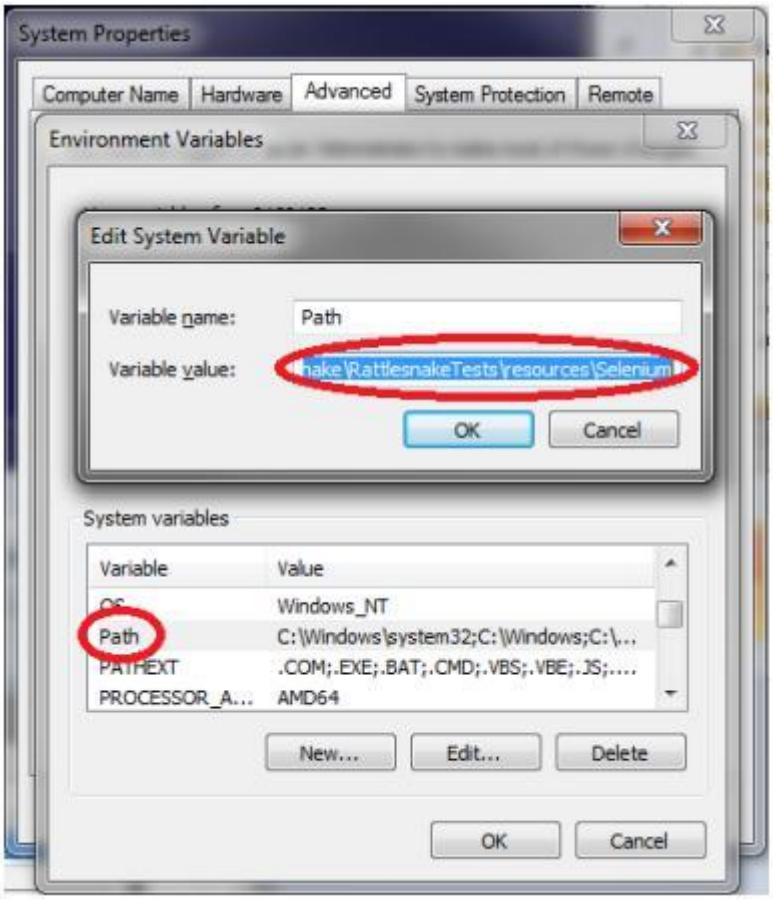
- Extract Files to your Eclipse workspace location or any other location

Add Chromedriver to your system`s path (WINDOWS):

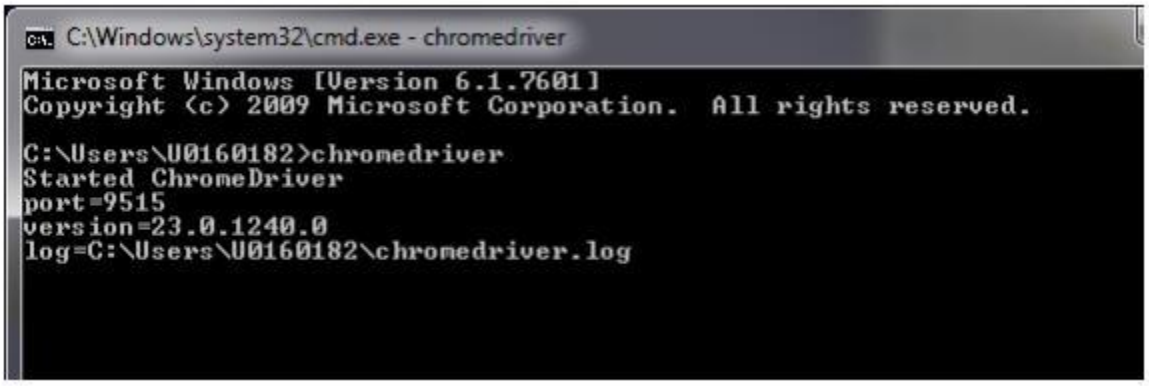
In Windows, goto System -> Advanced System Settings -> Advanced (Tab) -> Environment Variables (button)



Under System variables, scroll to the Variable named Path -> Edit… (button) -> Variable value -> Scroll to the end of the field, add a semicolon and append the local path of chromedriver.exe to the end of the value field. Click OK:



To verify, open the Command Line (Run cmd.exe) -> Type chromedriver ->Hit Enter -> ChromeDriver should start:



Add Chromedriver to your system (MAC):

- Complete the steps 1.6 and 1.7 first. When running the test files, include the following in @BeforeClass method:
- System.setProperty("webdriver.chrome.driver", "path_to_chromedriver/chromedriver");

Alternative 2 (MAC):

- Copy chromedriver
- Finder -> Go -> Go to folder
- Type -> /usr/local/bin
- Paste chromedriver to <bin> folder
- Restart Eclipse
- If you have manually put in the line "System.setProperty(……..)" in @BeforeClass Method of each test file, remove them. Ignore this step if you haven't written anything in the @BeforeClass Methods yet.

Add Chromedriver and Eclipse executable (LINUX):

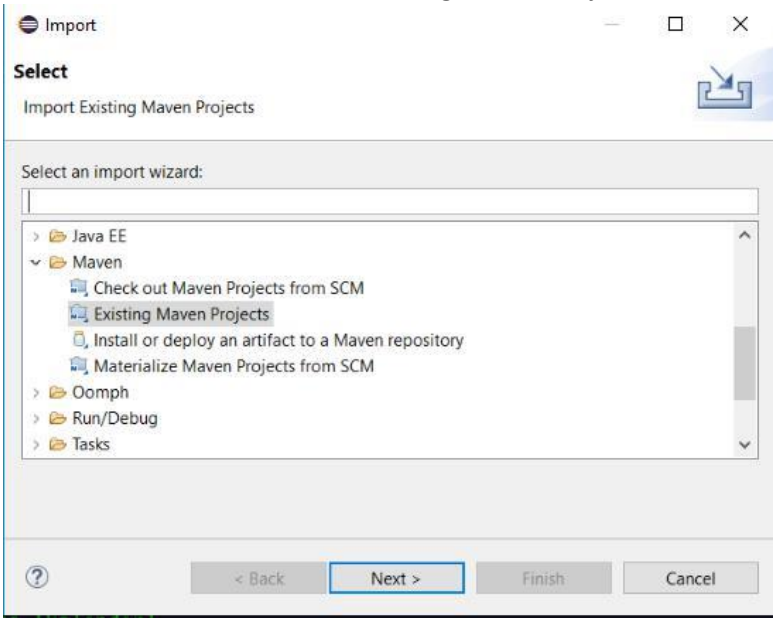
- Type in "gedit ~/.bashrc"
- Write the following lines on top of the file:

- export PATH = $PATH:/home/USERNAME/eclipse/java-2018-12/eclipse
    - For launching eclipse in terminal

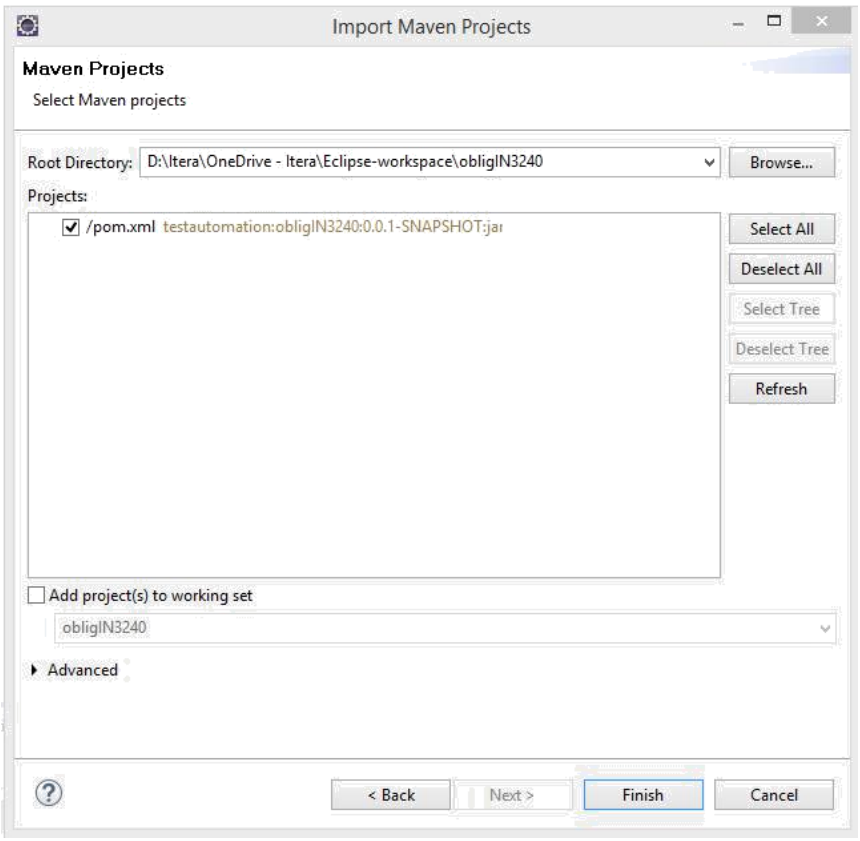
- export PATH = $PATH:/home/USERNAME/eclipse-workspace/
    - If Chromedriver is located in eclipse-workspace, use the path above.
    - If not, modify path to target the folder containing chromedriver

1.6. Download Maven Project

- Download « weeksExercises» Maven project from
- https://github.com/iteraozkan/assignmentIN3240
- Extract Files to your Eclipse workspace location
- Open Eclipse IDE
- Go to File → Import
- Under Maven Folder choose «Existing Maven Project»



- Click Next
- Browse «assigmentIN3240» folder that you download from https://github.com/iteraozkan/assignmentIN3240



- Choose /pom.xml.. and click Next

1.7. Pom.xml file

Make sure that your pom.xml file is include below dependencies:


```
<dependencies>
		<dependency>
			<groupId>com.beust</groupId>
			<artifactId>jcommander</artifactId>
			<version>1.27</version>
		</dependency>
		<!-- https://mvnrepository.com/artifact/org.testng/testng --
		> <dependency>
			<groupId>org.testng</groupId>
			<artifactId>testng</artifactId>
```

```
            <version>6.14.2</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.seleniumhq.selenium</groupId>
            <artifactId>selenium-java</artifactId>
            <version>3.9.1</version>
        </dependency>
        <dependency>
            <groupId>com.relevantcodes</groupId>
            <artifactId>extentreports</artifactId>
            <version>2.40.1</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/commons-io/commons-io --
        > <dependency>
            <groupId>commons-io</groupId>
            <artifactId>commons-io</artifactId>
            <version>2.6</version>
        </dependency>

    </dependencies>
```

Inspectors Tools

Here are some useful Chrome extensions that you can locate element by xpath

1. Xpath Helper
2. ChroPath

Conclusion:

This article provides the outline for setting up all the tools required to start automation using Selenium Webdriver with Java Eclipse IDE. Additional to this, knowledge of Core JAVA is required to test the functionalities.