Selenium Automation – Key Notes

Selenium IDE - Firefox add-on that will do simple record-and-playback of interactions with the browser. It’s browser based add-on to identify web elements like record and play back. To know the difference and further details between selenium IDE and Web Driver <http://www.seleniumhq.org/>

* **Looping -** To perform looping in Selenium Ide - <https://github.com/73rhodes/sideflow> , this file has to be loaded as external file into IDE.
* **Java source code from Selenium IDE** IDE code can be converted to many different formats by Selenium IDE -> File - Export Testcase As - Java / JUnit4 / WebDriver (Java source code from Selenium IDE - <https://stackoverflow.com/questions/7990922/how-to-generate-java-source-code-from-selenium-ide-ide-code-is-in-html-extensio>)

*Why are we going to Web Driver* – Due to lot of constraints like coding & Maintenance ease/ testing approach see here, [(https://stackoverflow.com/questions/19683100/why-do-we-use-webdriver-instead-of-selenium-ide]((https:/stackoverflow.com/questions/19683100/why-do-we-use-webdriver-instead-of-selenium-ide))

Selenium Web Driver – Using the Web driver (gecko driver acts as driver between your tests in Selenium and the Firefox browser) the browsers objects (web elements are identified in the browsers) and below are the pre-requisites and links to go through to learn more about Selenium web driver.

* Create robust, browser-based regression automation suites and tests and scale and distribute scripts across many environments

**Pre-Requisite –**

* Java / JDK <http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>
* Eclipse <https://www.eclipse.org/downloads/packages/>
* Selenium Standalone Server - JAR Files <http://www.seleniumhq.org/download/>
* *TestnG* **only if** you need this framework <http://www.softwaretestinghelp.com/testng-framework-selenium-tutorial-12/>

**To run selenium program in eclipse we need gecko driver if you are working on Firefox browser, get it from here** [**https://github.com/mozilla/geckodriver/releases**](https://github.com/mozilla/geckodriver/releases) **in the code to execute the scripts**

Step by step videos to start up with selenium web driver.

<https://www.youtube.com/watch?v=X6jc7UurmD4&index=48&list=PLwbuZz2PgXBrGx2kjegs-C4iOtoxXmxM2>

Channel Name - [Oresoft LWC](https://www.youtube.com/channel/UC1Wot8tO0-0TZZZxVhq-zcg)

Below are the frameworks to be used to run the Java programs on the IDE environments

**Junit** – By default Eclipse has this one.

**Sample step by step selenium program to run in Eclipse**

* <http://www.softwaretestinghelp.com/webdriver-eclipse-installation-selenium-tutorial-9/>
* <http://www.softwaretestinghelp.com/selenium-webdriver-tutorial-10/>

**Upgradtion from Junit to TestnG**

<http://www.softwaretestinghelp.com/testng-framework-selenium-tutorial-12/>

Test NG is **TDD approach** **Test Design & Develop** has a lot of advantages compared to Junit, like setting method prioritization, ease use of annotations , providing parameters as data providers class file.

**Selenium Automation Project Prerequisite and how to proceed** –

**Download (everything available in google online)** – **Java JDK** latest version to be installed in your machine; **Selenium Web Drivers (Jar files)** – used to automate the browser – functional / regression…etc testing ; Download **web driver for IE/Chrome/fire fox**; Download latest **Eclipse** (I used Eclipse NEON 4.0.4) which will be integrated with TestNG/Maven/ Junit / a lot more…remaining things like cucumber, gherkin, Natural plugin and other Jar files can be integrated under Help - > Eclipse Market place / Available softwares.

**Maven Project** - Maven is a powerful project / build management tool, based on the concept of a POM (Project Object Model) that includes project information and configuration information for Maven such as construction directory, source directory, dependency, test source directory, Goals, plugins, etc.

In real time, lots of challenges in transferring the Java dependencies files (Jar/ libraries / ..etc) between developers in BIG team. To overcome this we can go with **1. GITHUB repository** – this is Concurrent source code versioning system, like check in and check out, people (freshers) can fetch the code from here to start for their day. **2** Most widely used and currently used in the market is **MAVEN** under POM.xml , we can give the all the dependencies Jar/Lib/ or any kind of inputs (selenium/testng/cucumber/gherkin/…etc) in the xml format, to get the dependencies code just search in google like eg:- selenium – maven dependencies (<https://mvnrepository.com/>).

**For building & running simple Maven Project** – three things you **NEED** to remember –

1. Configure POM.xml with all the dependencies (selenium, TestNG, cucumber dependencies, gherkin…etc) to get the dependencies code just search in google like eg:- selenium – maven dependencies (<https://mvnrepository.com/>).
2. Set the build path in eclipse from JRE to JDK, in Eclipse Project, window- > preferences ->Installed JRE, change from JRE to JDK path usually it will be installed in C drive
3. Suffix all your java files with “Test”, then only Maven complier & surefire plugin will understand that these files to be compiled and executed accordingly
4. Class Variable set up , since maven download all the dependencies (selenium, TestNG, cucumber dependencies, gherkin…etc) from POM.xml (Project Object Model) file and store in the .m2 repository path in your machine, we need to set the class variable for the project, right click on the project ->properties-> java build path – paste the .M2 folder path.
5. Now all set you can run the Java program by right click on the editor ->Maven ->Maven Test

**Refer the below link -** <https://www.techbeamers.com/create-selenium-webdriver-maven-project/>

**For BDD (Behavioral design & Develop ) approach using Cucumber/Gherkin with Maven** – Please refer the videos.

**For building & running simple Maven Project with Cucumber and Gherkin**

1. Get all the dependencies/feature files/test step definitions/test runner class) for POM.xml for your project (<https://github.com/naveenanimation20/CucumberSeleniumFramework>) and update yours respectively
2. Setting up Feature file (contains what feature going to be automated , pre-requisites, steps with the keyword like Feature/ Scenario -outline/Given/When/Then/And/But/\*(asterisk). )(download **Natural plugin to reflect the text color change in the**  under Help - > Eclipse Market place / Available softwares or in Google first link, just drag and drop in your eclipse IDE editor.
3. Set up the Test step definition (here only will write the Java code for performing automation
4. Test Runner Class – using Junit, from here only the project / code will be executed. In this java file, we will be giving the path of the feature file and step definition.

**Watch this video to brush up and stay from the beginning**

<https://www.youtube.com/watch?v=vHzMJuc9Zuk&list=PLFGoYjJG_fqoBFPevCDZDCufDX5h-o3yO&index=1>

Cucumber Topics – Main three things needed for cucumber frame works is feature file, step-definition file and test runner class.

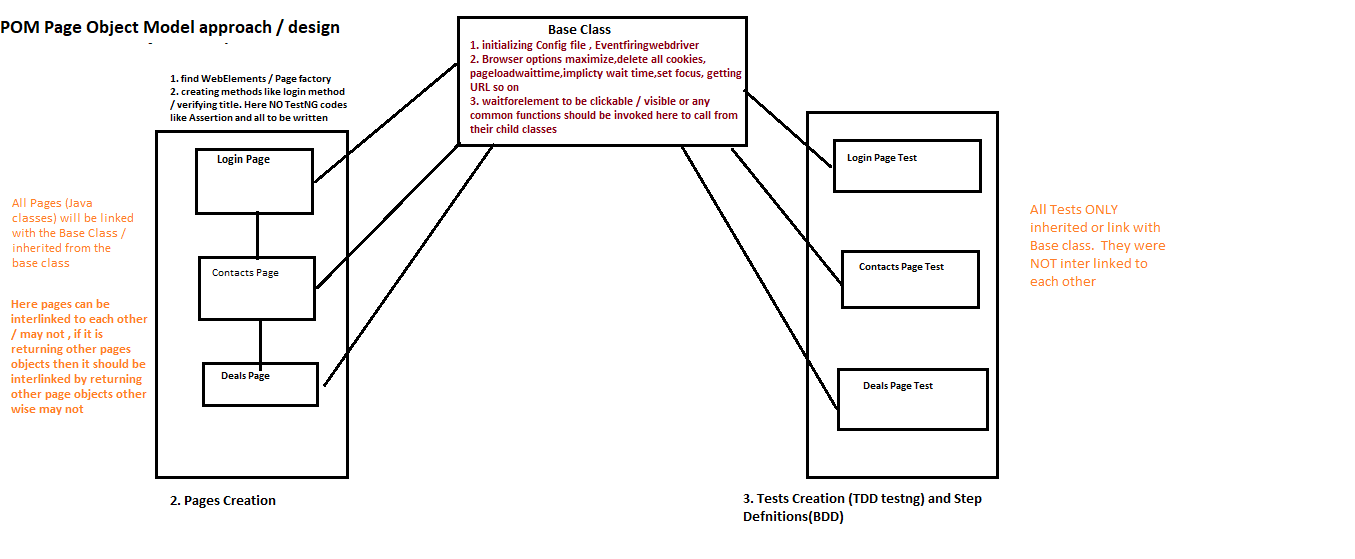
**Feature File** – where the scenarios and test steps were created with the keywords Feature/ Scenario -outline/Given/When/Then/And/But/\*(asterisk).

**Step-definition** – Here we write our selenium and Java code for doing the automation. Each step mentioned in the feature will be defined here

**Test Runner Class** – From this file only we will execute our scripts . This one contains the cucumber options to specify the path of the feature file and step definitions to run with.

Explore in google for these cucumber important topics **POM(Page Object Model), EventFiringwebdriver, webdriver event listener, Data Parameterization, Tagging , Hooks & Tagged Hook**, **tagged hook concept is missing TestNG** to implement in project.

**POM – Page Object Model Approach design**



### **Java & Selenium favorites**

[Java Inheritance](https://www.tutorialspoint.com/java/java_inheritance.htm)

[Selenium with Cucumber (BDD Framework): Tutorial with Example](https://www.guru99.com/using-cucumber-selenium.html)

[Selenium Locators: Identify Web Elements Using XPath in Selenium](https://www.softwaretestinghelp.com/using-selenium-xpath-and-other-locators-selenium-tutorial-5/#Types_of_Locators)

[4. Locating Elements — Selenium Python Bindings 2 documentation](https://selenium-python.readthedocs.io/locating-elements.html)

[How to find element in Selenium using FindElement Command?](https://www.toolsqa.com/selenium-webdriver/find-element-selenium/)

[Create Selenium Webdriver Maven Project in TestNG and Eclipse](https://www.techbeamers.com/create-selenium-webdriver-maven-project/)

[Assertions in Selenium Using Junit and TestNG Frameworks](https://www.softwaretestinghelp.com/assertions-in-selenium/)

[Selenium – Cucumber – Passing Parameters | Software-Automata](https://softwareautomata.wordpress.com/2017/08/02/selenium-cucumber-passing-parameters/)

[Assertions in Selenium Using Junit and TestNG Frameworks](https://www.softwaretestinghelp.com/assertions-in-selenium/)

[Specify different cucumber options in Java using Eclipse](https://www.toolsqa.com/cucumber/cucumber-options/)

[How to create Cucumber Step Definition class - AutomationTestingHub](http://www.automationtestinghub.com/cucumber-step-definition/)

[Data Driven Testing in Cucumber | Passing parameter in Cucumber](https://www.toolsqa.com/cucumber/data-driven-testing-in-cucumber/)

[How To Write Dynamic XPath In Selenium WebDriver | Software Testing Material](https://www.softwaretestingmaterial.com/dynamic-xpath-in-selenium/)

[Handling Web Tables, iFrame, and Dynamic Elements in Selenium Script – Selenium Tutorial #18](https://www.softwaretestinghelp.com/selenium-tutorial-18/)

[How To Handle Drop Down And Multi Select List Using Selenium WebDriver](https://www.softwaretestingmaterial.com/handle-drop-down-and-multi-select-list-using-selenium/)

[How to do Data Driven Testing using Json with Cucumber](https://www.toolsqa.com/selenium-cucumber-framework/data-driven-testing-using-json-with-cucumber/)

[log4j.properties example - Log4j properties file example - HowToDoInJava](https://howtodoinjava.com/log4j/how-to-configure-log4j-using-properties-file/)

[Setup for Selenium with Cucumber Using Maven](https://www.axelerant.com/resources/team-blog/setup-for-selenium-with-cucumber-using-maven)

<https://www.edureka.co/blog/java-bytecode/>

<https://www.javatpoint.com/properties-class-in-java>

<https://www.toolsqa.com/selenium-cucumber-framework/selenium-end-to-end-automation-test/>