Module Name: Selenium WebDriver (Automation Testing)

Trainer Name: Ankush Vankore

Total Sessions: 19

Selenium WebDriver

* Introduction of Automation
* Need of Automation
* Process of Automation
* Introduction Selenium
  + Selenium IDE
  + Selenium Grid
  + Selenium RC
  + Selenium WebDriver
* Introduction to Selenium WebDriver
* Configuration
* Basic methods of WebDriver
  + Launching the different browsers
  + get()
  + getTitle()
  + getCurrentUrl()
  + getPageSource()
  + close()
  + quit()
* Locators (Identifying the controls)
  + Name
  + Id
  + ClassName
  + CssSelector
  + XPath
  + LinkText
  + PartialLinkText
  + TagName
* Handling Controls
  + Text Box
  + Command Button
  + Radio Button
  + Check box
  + Dropdown list
  + List Box
  + Links
* Synchronization
  + Thread.Sleep()
  + ImplicitWait
  + ExplicitWait
  + FluentWait
  + PageLoadTimeout
* Handling table
* Handling multiple browser windows
* Handling file upload
* Robot class
* Handling alerts
* Handling mouse actions
  + Hover
  + Click
  + Right click
  + Double click
  + Drag and drop
* JavascriptExecutor
* Handling menus
* Screenshots

TestNG

* Introduction
* Installation / Configuration
* Writing single tests
* Writing multiple tests
* Setting the priority
* Annotations
  + @Test
  + @BeforeTest
  + @AfterTest
  + @BeforeMethod
  + @AfterMethod
* Data driven testing via @DataProvider
* Reports
  + Normal report
  + HTML Report
* Modular framework
  + Execute single test / multiple tests
  + Skipping single test / multiple tests
  + Executing / skipping groups
  + Executing single class /multiple classes
* Parameterization
* Keyword Driven framework
* Page Object Model (POM)
* Data Driven Framework via Apache POI

Maven

* Introduction
* Creating the project
* Adding dependencies
* Maven Architecture
* Extent Report
* Execution of project

Cucumber

* Introduction to BDD
* Configuration
* Components of Cucumber
  + Feature file
  + Step Definition
  + Runner Class
* Keywords
  + Feature
  + Scenario
  + Background
  + Given
  + When
  + Then
  + And
  + Examples
  + Scenario Outline
* Tags in Cucumber
* Hooks in Cucumber
* Data driven framework via Cucumber

Java Revision

* Basics of language
  + Variables
  + Data Types
  + Conditions
    - If
    - If else
    - Nested if else
    - Switch case
  + Loops
    - For Loop
    - While
    - Do – While
  + Writing and calling the function / methods
* Creating class and object
  + Constructor
  + Getter and setter methods
  + Static members and static methods
* Basics of Inheritance
  + extends keyword
  + Concrete class
  + Abstract Class
  + Interface
* Exception Handling
  + throws keyword
  + try catch block
* Collections
  + List
  + Set
  + Map

What is Software Testing?

It is the process of checking the **C**orrectness, **C**ompleteness, **S**ecurity and **Q**uality of developed software application.

**Process**

* Scenario Creation
* Test Case creation
* Test Data Creation
* RTM Creation

**Operations while testing**

Entering some data in text box

Clicking the buttons

Selecting the values from check box, radio buttons, drop down list, list box

Navigating from one page to another

Verifying the expected result

Marking the test case as pass or fail

During manual testing tester uses their hand – eye – brain co-ordination

**Automation testing** is performing above actions via a machine.

Machine in this context is **Test Automation Tool.**

**Every automation tool is a Software.**

Every tool understands some specific programming language like Java, C#, Python, Javascript, Ruby etc

**Advantages of Automation**

* Reduces the time
* Faster execution
* No human errors
* More accuracy
* Reduces the cost
* Efficiency
* Regression testing
* Huge amount of data
* 100% Test overage
* Reusability of the script
* Easy reporting

**When to Automate**

* Stable requirement
* Regression
* Data testing
* Performance testing
* Security testing
* Compability testing

**Types of Automation Testing**

1. Functional Testing
   1. Selenium WebDriver
   2. Tosca
   3. QTP
   4. Appium
2. API Testing
   1. Postman
   2. RESTFull API
3. Unit testing
   1. JUnit
   2. NUnit

**Process of Automation**

1. Planning
2. Selection of Tool
   1. Technology / Type of AUT
   2. Cost of the tool
   3. Support availability
   4. Automated reports
   5. Tester availability
3. Test Script Creation
4. Test Data Creation
5. Execution
6. Report
7. Maintance

**Selenium**

It is suite / bundle of test automation tools those are used to test **Web Based Application (Web Sites).**

**Components of Selenium**

1. Selenium IDE – Record and playback
2. Selenium Grid – Parallel Execution on multiple platform, multiple browsers
3. ~~Selenium RC – Remote Control~~
4. Selenium WebDriver

**Selenium WebDriver**

* Test automation tool to test Web Based application (Web Sites)
* It an API for automation testing
* It is an interface in Java

**Create 2 folders**

1. YourName\_Selenium Demos
2. Selenium Jar File

Pre-Requisite for Selenium WebDriver

* Windows 10 (Minimum)
* Any one updated browser
* Java 11 (JDK 11) Minimum
* Java Editor
  + Eclipse
  + Idea Intellj
* **Selenium Jar file**
  + Open selenium.dev website in any browser
  + Click on Downloads link
  + Download Latest stable version [4.34.0](https://github.com/SeleniumHQ/selenium/releases/download/selenium-4.34.0/selenium-server-4.34.0.jar)
  + Open your downloads folder
  + Copy this file and paste in the 2nd folder (Selenium Jar Files)

**Eclipse Configuration**

* Open Eclipse
* Select your 1st folder as a workspace using browse button
* Click on Launch button
* Go to File 🡪 New 🡪 Java Project
* Give the name to Project
* Select Java version (Minimum 11)
* **Uncheck Create module-info.java file checkbox**
* Click on Finish
* Create a Package
* Create a class in this package
* Right click on you project (in package explorer) 🡪 Build path 🡪 Configure Build path
* Click on Libraries Tab
* Click on Classpath
* Click on Add Extern JARs… button
* Open your 2nd folder and select the file which you have downloaded in earlier steps. (selenium-server-4.34.0.jar)
* Click on Open button  
  **Make sure that this file is inside the class path**
* Click on Apply and Close