Module: Selenium WebDriver

Trainer: Ankush Vankore

No of Sessions: 19

Contains

* WebDriver
  + Introduction
  + Installation
  + WebDriver Methods
    - get()
    - getTitle()
    - getCurrentUrl()
    - getPageSource()
  + Locators
    - Name
    - Id
    - ClassName
    - CssSelector
    - Xpath
    - LinkText
    - PartialLinkText
    - TagName
    - RelativeLocator
  + Handling different Controls
    - Text Box
    - Command Button
    - Radio button
    - Check Box
    - Dropdown List
    - List Box
    - Hyperlinks
    - FileUpload
  + Synchronization
    - Thread.Sleep()
    - ImplicitWait
    - ExplicitWait
    - FluentWait
    - PageLoadTimeout
  + Handling Tables
  + Handling Multiple Windows
  + Handling Alerts
  + JavascriptExecutor
    - Clicking
    - Scrolling
  + Taking Screenshot
  + Robot Class
  + Action Class
    - Click
    - Right Click
    - Double Click
    - Drag and Drop
* TestNG
  + Configuration
  + Single Test
  + Multiple Tests
  + Setting priority
  + Annotations
    - BeforeTest
    - AfterTest
    - BeforeMethod
    - AfterMethod
    - DataProvider
    - Parameters
  + Assertions
  + Parallel Execution
  + Reports
  + Liner Framework
  + Modular Framework
  + Keyword Driven Framework
  + Page Object Model (POM)
  + Data Driven Testing
    - Reading data from Excel file
    - Writing data to Excel file
* Maven & Cucumber
  + Configuration
  + Configuration of pom.xml file
  + Adding dependencies
  + Creating Feature file
  + Creating Step Definition
  + Creating Runner Class
  + Keywords
  + Hooks
  + Tags
  + Reports
  + Data driven testing via cucumber

Revision of Java

* Basics
  + Variables
  + Data types
  + Conditions
    - If, if-else, switch case
  + Loops
    - for loop
    - for each / enhanced for loop
  + Writing and calling Methods
  + Arrays
  + Creating class and Objects
  + Static Members and static methods
  + Collections
    - **List**
    - Set
    - Map
  + Exception Handling
    - throws keyword

Software Testing

Process of checking **C**orrectness, **C**ompleteness, **S**ecurity & **Q**uality of developed software application.

Process

* Creating Test Scenarios
* Creating Test cases
* Creating Test Data
  + ECP
  + BVA
  + EG
* Execution
* Report the bug / defect

Manual Testing: Hand-Eye-Brain

* Entering some data (in text box)
* Selecting the values (from list box, dropdown list, check box, radio buttons)
* Taking action (Clicking on button)
* Navigation (Clicking on a link)
* Mark the test case as Pass or Fail

**Automation Testing:**

Performing above actions with the help of a machine. A machine in this context is nothing but Automation Testing Tool.

Every automation tool is a software.

Instructions needs to be provided in specific programming language.

Advantages:

* Faster
* Time Saving
* Avoids repetitions
* Less human efforts
* Ensure Quality
* Less human errors
* Accuracy
* Saves money
* Regression testing
* 24 / 7 test possible
* Reusability
* Test report
* Productivity

Need

* Faster execution
* Accuracy
* Less human errors
* Compatibility Testing
* Easy reporting

When to automate?

* Stable requirements
* Repeated test scenarios
* Large amount of data to be tested
* Performance testing
* More accuracy
* Compatibility Testing
* CICD

Types of automation Testing

* Unit testing
  + JUnit
  + NUnit
* API Testing
  + Postman
  + RESTAPI
* GUI (Functional Testing)
  + Selenium WebDriver
  + Tosca
  + QTP
  + Cucumber
  + Playwright
* Mobile Testing
  + Appium

Process of Automation

* Planning
* Tool selection
  + Technology
  + Market presence
  + Ease of the tool
  + Cost of tool
  + Support availability
  + Automated Report
* Generate the script
* Create the test data
* Execution
* Report
* Maintance

**Selenium**

Suite / Bundle of test automation tools to test web / browser based (Web Sites) application.

**Components of Selenium**

1. Selenium IDE (Record and Play)
2. Selenium Grid (Parallel Execution)
3. Selenium RC – Deprecated
4. Selenium WebDriver

**Selenium WebDriver**

* Tool to test Web Based application
* An interface in Java
* An API

Selenium WebDriver Configuration

1. Create 2 folders (Except on C: )
   1. YourName\_WebDriverDemos
   2. WebDriver Jar Files

**Pre-Requisite for Selenium WebDriver**

1. Minimum Windows 10
2. Minimum Java 11 (Check on command prompt via java –version)
3. Any 1 updated Browser
4. Editor for scripting
   * Eclipse
   * Idea Intellij
5. Selenium Jar File
   * Launch selenium.dev site
   * Click on Downloads
   * Download Latest stable version (Latest stable version [4.30.0](https://github.com/SeleniumHQ/selenium/releases/download/selenium-4.30.0/selenium-server-4.30.0.jar))
   * Cut paste this file to 2nd folder we have created.

**Configuration of WebDriver with Eclipse**

* Launch Eclipse
* Select the 1st folder which you have created earlier as a workspace.
* Create a new Java Project
  + File 🡪 New 🡪 Java Project
  + While Creating make Create module-info.java file check box OFF.
  + Click on Finish
* Create a package inside this folder
* Create a class
* Right click on your project 🡪 Build Path 🡪 Configure Build Path 🡪 Libraries 🡪 Click on Class Path 🡪 Click on Add External Jars… 🡪 Open the 2nd folder created earlier and select the Jar file that we have downloaded. 🡪 Click on Apply and Close

WebDriver Methods

1. Launching the browser 🡪 Create object of WebDriver interface.  
   This will launch the blank browser window
2. get() 🡪 Launch the specific website (URL)
3. close() 🡪 Close the current browser window which is launched by WebDriver object.
4. driver.manage().window().maximize() 🡪 Maximize the browser window.

Common Exceptions in WebDriver

1. InvalidArgumentException 🡪 Your URL is not in the correct format. URL should be Absolute (start with http / https)