Selenium WebDriver

Duration: 48 + 18 Hrs

* WebDriver
  + Concept of Automation
  + Introduction to Selenium
  + Configuration
  + Basic methods of WebDriver
    - Launching different browsers
    - get()
    - getTitle()
    - getCurrentUrl()
    - getPageSource()
    - close()
    - quit()
  + Locators
    - Name
    - Id
    - ClassName
    - CssSelector
    - LinkText
    - PartialLinkText
    - Xpath
    - TagName
  + Handling controls (WebElements)
    - Text Box
    - Command Button
    - Links
    - Checkbox
    - Radio button
    - Drop down list
    - List box
  + Synchronization
    - Thread.Sleep()
    - Implicit Wait
    - Explicit Wait
    - Fluent Wait
    - PageLoadTimeout
  + Handling Table
  + Handling Multiple Windows
  + Handling Alerts
  + Handling File uploads
  + Robot Class
  + Handling Menus
  + Mouse Actions
    - Hovering the mouse
    - Click
    - Right Click
    - Double Click
    - Drag and drop
  + Headless browser
  + Screen shot
* TestNG
  + What is Framework
  + Introduction to TestNG
  + Configuration / Installation
  + Executing single test
  + Executing multiple tests
  + Setting the priority
  + Reports
    - Normal Report
    - HTML Report
  + Annotations
    - @Test
    - @BeforeTest
    - @AfterTest
    - @BeforeMethod
    - @AfterMethod
    - @Parameters
  + Data Driven Testing using @DataProvider
  + Modular Framework
    - Execute / Skip single test / multiple tests
    - Execute single / multiple classes
    - Execute / skip single / multiple groups
  + Keyword Driven Framework
    - Read data from .properties
    - Read from Excel file
  + Page Object Model (POM)
  + Data Driven Framework (Apache POI)
    - Read the data from Excel
    - Write the data to Excel
    - Pass the data to test case from Excel
  + Hybrid Framework
* Maven
  + Introduction
  + Configuration
  + Configuring the pom.xml file
  + Adding dependencies
  + Execution
  + Extent Report
* Cucumber
  + Introduction to BDD
  + Configuration
  + Components of BDD
    - Feature File
    - Step Definition
    - Runner Class
  + Keyword in feature file
    - Feature
    - Scenario
    - Given
    - When
    - Then
    - And
    - Background
    - Examples
    - Scenario Outline
  + Multiple Scenarios
  + Passing parameters to Step Definition
  + Tags in cucumber
  + Hooks in Cucumber
  + Reports
  + Data Driven Testing

**Software Testing**

Process of checking the **C**orrectness, **C**ompleteness, **S**ecurity and **Q**uality of a developed software.

Manual Testing

* Test Scenario
* Test Case
* Test Data
* RTM

Process

* Entering data in text box
* Clicking on buttons
* Selecting the value/s from check box, radio button, drop down list, list box
* Navigating from one page to another
* Verifying actual result with expected result
* Marking the test case as Pass or Fail

Manual tester uses their Hand-Eye-Brain co-ordanitation

**Automation testing is performing all above actions via a Machine.**

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

Every test tool is a software.

Every tool understands specific programming language, so needs to give instructions via any one of the supporting programming language. (Java, C#, Python, Ruby, VBScript, JavaScript)

**Advantages**

* Time consuming
* Correctness
* Less human efforts
* Reusability
* Regression Testing
* Less human errors
* 100% test coverage
* Cost cutting

**Process of Automation**

* Planning
* Selection of Tool
  + Type of application
  + Cost of tool
  + Support availability
  + Report generation
  + HR Availability
* Test script creation
* Test data creation
* Execute the script
* Generate the report
* Maintance

**Selenium**

It is suite / bundle of test automation tools to test **Web Based Applications. (Web Sites)**

**Components of Selenium**

1. Selenium IDE – Record and playback mech
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 applications (Web site)
* API
* Interface in Java

Create 2 folders

1. YourName\_Selenium Demos
2. Selenium Jar Files

**Pre-Requisite for Selenium WebDriver**

* Windows 10 (Minimum)
* Java 11 (Minimum)
* At least one **Updated Browser**
* 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**

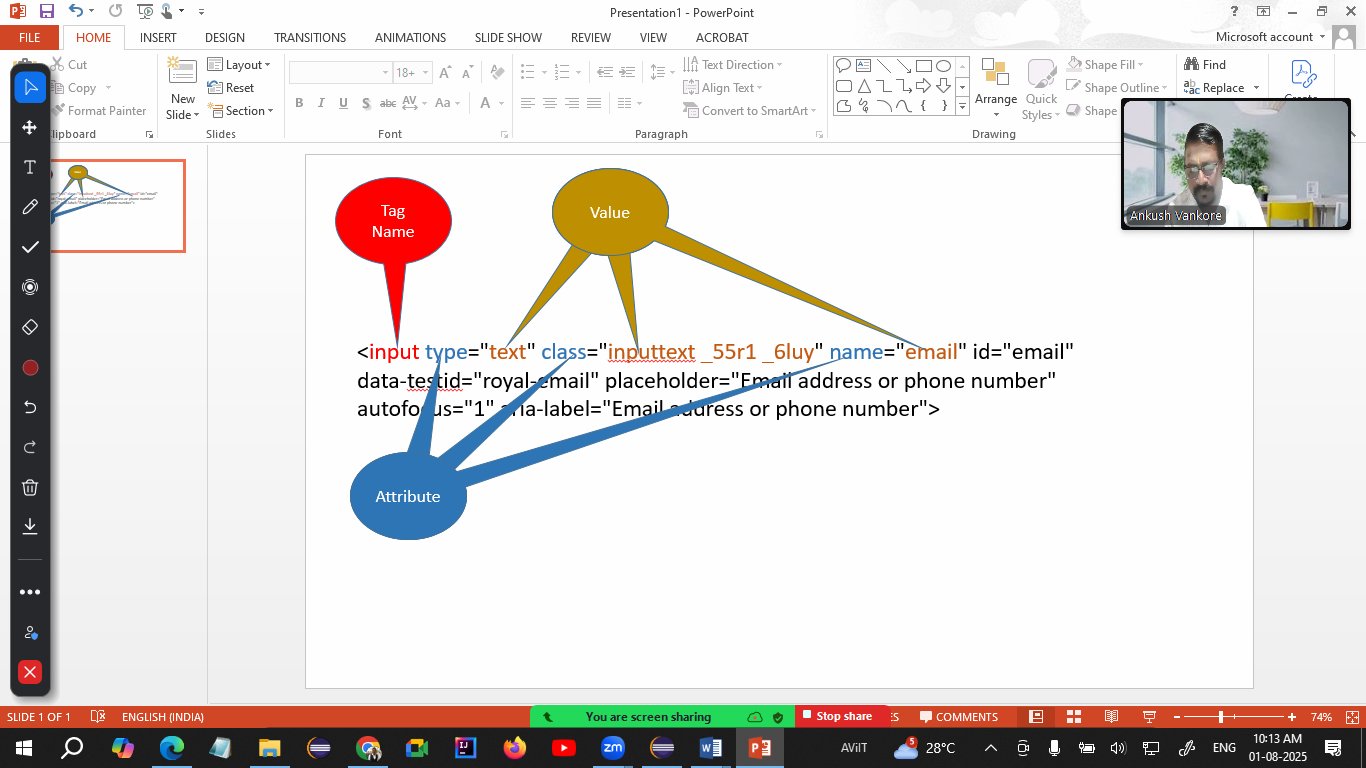
* Open Eclipse
* Select your 1st folder as a workspace using browse button
* Click on Launch button
* Go to File 🡪 New 🡪 Java Project
* Give the name to Project
* Select Java version (Minimum 11)
* **Uncheck Create module-info.java file checkbox**
* Click on Finish
* Create a Package
* Create a class 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

**Methods of Selenium**

1. Creating Object of WebDriver 🡪 Launch the blank browser window
2. get() 🡪 Launch the specific URL. URL should be Absolute
3. close() 🡪 Will close the current browser window which is opened by WebDriver object.
4. getTitle() 🡪 Will return the title of the Web Page opened in browser by WebDriver object. (String)
5. getCurrentUrl() 🡪 Will return the URL of the Web Page (String)
6. getPageSource() 🡪 Will return the rendered HTML of the page. (String)
7. findElement() 🡪 Will read and return the **single control** on the page. Always search for the 1st occurrence. (WebElement)
8. findElements() 🡪 Will read and return **multiple controls** on the page. (List<WebElement>)

**Common Exceptions in WebDriver**

1. InvalidArgumentException 🡪 Your URL is in wrong format. URL should be absolute URL. Which starts with http
2. SessionNotCreatedException 🡪 Your browser and Selenium versions are not compatible with each other. Check both versions (Both should be latest).
3. NoSuchElementException 🡪 Selenium is not able to locate the control due to
   1. Value of locator is wrong
   2. Value of locator is dynamic
   3. Synchronization issue
4. InvalidSelectorException 🡪 The locator is in wrong format.
5. SessionTimeoutException 🡪 The page is not loaded within 30 seconds.



**Locators in Selenium**

Locator is the way of identifying any control / WebElement on the page

1. Name
2. Id
3. ClassName
4. CssSelector
5. LinkText
6. PartialLinkText
7. Xpath
8. TagName

**WebElement**

* Every control on the page is treated as WebElement
* It is an **interface** in WebDriver API to store any control.
* **Methods**
  + sendKeys() 🡪 Will enter some text in the text box. It appends the text to existing text.
  + click() 🡪 Will click on any control.
  + getText() 🡪 Will return the text on the control. (String)
  + isSelected() 🡪 will check that whether the checkbox / radio button is selected or not. (boolean)
  + isEnabled() 🡪 Will check that whether the control is enabled or disabled. (boolean)
  + isDisplayed() 🡪 Will check that whether the control is visible or not. (boolean)
  + getAttribute() 🡪 Will return the value of any attribute. (String)

**CssSelector**

This the locator via which you can locate any control using any one or multiple attributes of the control.

Types of CssSelector

1. Single Attribute  
   Syntax:  
   tagName[attribute=”value”]  
   **input[data-testid="royal-email"]**
2. Multiple Attributes  
   Syntax:  
   tagName[attribute1=”value”][attribute2=”value”]
3. Special Characters
   1. ^ 🡪 Starts with  
      tagName[attribute^=”value”]
   2. $ 🡪 Ends with  
      tagName[attribute$=”value”]
   3. \* 🡪 Contains  
      tagName[attribute\*=”value]

**XPath (XML Path)**

Swargate Pune 🡪 Kothrud 🡪 Kothrud Demo 🡪 Shubheccha Managal Karyalay 🡪 Viom Building 🡪 Flat No B203

Types

1. Absolute XPath  
   starts with html
2. Relative XPath
   1. Taking the reference of parent tag
   2. Taking the reference of current tag / control

**Handling Dropdown List / List Box**

* If the control is having <select> tag then only it will be treated as dropdown list / list box.
* Selenium has provided special class to handle this control ie. **Select** class

Methods of Select class

1. getFirstSelectedOption() 🡪 Returns the selected item / option / element from the drop down list. (WebElement)
2. getOptions() 🡪 Return list of all the options / elements from the drop down list / list box. (List<WebElement>)
3. selectByContainsVisibleText()
4. selectByIndex()
5. selectByVisibleText()
6. selectByValue()
7. getAllSelectedOptions() 🡪 Returns list of all selected options / elements from the list box (List<WebElement>)
8. isMultiple() 🡪 Checks that whether this control (list box / drop down list) allows to select multiple options or not. (boolean)
9. deSelectAll()

**Synchronization (Waits in Selenium)**

It is process of adjusting speed of tool with speed of application.

1. Thread.Sleep() 🡪 Will pause the execution of script for specified milliseconds.
   1. It is applicable to single statement only.
   2. It takes mandatory delay.
2. ImplicitWait
   1. It is applicable throughout the script
   2. It doesn’t take mandatory delay.
3. ExplicitWait (WebDriverWait class)
   1. It is applicable to the single statement only.
   2. It doesn’t take mandatory delay.
   3. We can handle some condition like, visibility of element, element to be clickable, alert to be present etc.
4. FluentWait
   1. It is applicable to the single statement only.
   2. It doesn’t take mandatory delay.
   3. We can handle some condition like, visibility of element, element to be clickable, alert to be present etc.
   4. You can handle an exception as well.  
      w – withTimeout   
      i – ignoring   
      p – pollingEvery  
      u – until
5. PageLoadTimeout – It increases the timeout for loading the page to avoid SessionTimeoutException.