



SELENIUM JUNIT CLASS-1

TRAINER

SALMAN RAHMAN

ROAD TO SDET

CONTENTS

- **Setup Selenium JUnit**
- **Difference between Gradle and Maven**
- **First Script: Get website title**
- **Selenium Wait Commands**
- **Locators**
- **Chropath**
- **Evaluating Locators in browser console**
- **Relative XPath vs Absolute XPath**
- **Check if element exists**

SETUP SELENIUM JUNIT

- Install JDK LTS version. Java SE 8 (LTS) or Java SE 11 (LTS)
- Set JAVA_HOME in system environment variables
- Check java version from cmd
 - > *Java -version*
- Download Maven or Gradle
- Setup MAVEN_HOME or GRADLE_HOME
- Download IntelliJ community edition
- Start project as Maven or Gradle
- Import selenium package by Maven or Gradle
 - - testCompile group: 'junit', name: 'junit', version: '4.12'
 - // *https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java*
 - compile group: 'org.seleniumhq.selenium', name: 'selenium-java', version: '3.141.59'
 - // *https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-firefox-driver*
 - compile group: 'org.seleniumhq.selenium', name: 'selenium-firefox-driver', version: '3.141.59'
 - // *https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-chrome-driver*
 - compile group: 'org.seleniumhq.selenium', name: 'selenium-chrome-driver', version: '3.141.59'
- Download Firefox and Chrome driver and put them into
 - **src > test > resources**
 - If resources folder not found, create a folder named “**resources**” under src/test and mark directory as **Test Resources Root**
- Create a class under **test > java** and start writing code

DIFFERENCE BETWEEN GRADLE AND MAVEN

Basis	Gradle	Maven
Definition	Gradle is a build automation tool for multi-language software development, deployment, and publishing. Supported languages include Java, Groovy, C/C++, and JavaScript.	Maven is a build automation tool used primarily for Java projects. Also, it can be used for C#, Scala, Ruby etc
Configuration	It uses a Groovy-based Domain-specific language (DSL) for creating project structure.	It uses Extensible Markup Language (XML) for creating project structure.
Performance	It performs better than maven as it is optimized for tracking only the current running task.	It does not create local temporary files during software creation hence uses less time.
Customization	Gradle is highly customizable	Maven is not much customizable/
UI Experience	It is very user friendly	It is likely user friendly

First Script: Get website title

```
public class FirstTestCase {
    WebDriver driver;
    @Before
    public void setup(){
        System.setProperty("webdriver.gecko.driver",
            "./src/test/resources/geckodriver.exe");
        FirefoxOptions ops=new FirefoxOptions();
        ops.addArguments("--headed");
        driver=new FirefoxDriver(ops);
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(30));
    }
    public void getTitle(){
        driver.get("https://demoqa.com");
        String title=driver.getTitle();
        System.out.println(title);
        Assert.assertTrue(title.contains("ToolsQA"));
    }
    @After
    public void finishTest(){
        driver.close();
    }
}
```

SELENIUM WAIT COMMANDS

- When a web page loads on a browser, various web elements (buttons, links, images) that someone wants to interact with may load at various intervals.
- Selenium WebDriver provides three commands to implement waits in tests.
- **Implicit wait:**
 - Implicit Wait directs the Selenium WebDriver to wait for a certain measure of time before throwing an exception. Once this time is set, WebDriver will wait for the element before the exception occurs.
- **Syntax:**
 - `driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);`

CHECK IF ELEMENT EXISTS

```
public void checkifElementExists() throws InterruptedException {  
    driver.get("https://demoqa.com");  
    wait=new WebDriverWait(driver,30);  
    Boolean status=wait.until(ExpectedConditions.elementToBeClickable(By.xpath  
    ("//img[@src='/images/Toolsqa.jpg']"))).isDisplayed();  
    Assert.assertEquals(status,true);  
  
}
```

- **Explicit wait:**

- By using the Explicit Wait command, the WebDriver is directed to wait until a certain condition occurs before proceeding with executing the code.
- Setting Explicit Wait is important in cases where there are certain elements that naturally take more time to load. If one sets an implicit wait command, then the browser will wait for the same time frame before loading every web element. This causes an unnecessary delay in executing the test script.
- Explicit wait is more intelligent, but can only be applied for specified elements. However, it is an improvement on implicit wait since it allows the program to pause for dynamically loaded Ajax elements.

- **Syntax:**

- `WebDriverWait wait = new WebDriverWait(driver, timeout);`
- `wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//xpath")));`

LOCATORS

Locator is a command that tells Selenium IDE which GUI elements (Text Box, Buttons, Check Boxes etc) its needs to operate on.

The different types of CSS Locator in Selenium IDE:

1. ID (Faster)
2. Name
3. ClassName
4. Tag Name
5. Link Text
6. Attributes
7. Relative Xpath
8. Absolute Xpath (Faster)

CHROPATH

- Install chropath plugin in chrome browser

EVALUATING LOCATORS IN BROWSER CONSOLE

1. `$('#id')`
2. `$('[name="locatorName"]')`
3. `$('[type="locatorType"]')`
4. `$('.class')`
5. `$x("//span[contains(text(),'Success')]")`

RELATIVE XPATH VS ABSOLUTE XPATH

Point of Difference	Absolute Path	Relative Path
Starts with	Single Forward Slash. Select the element from the root <html> and cover the whole path to the element. It is also known as complete or Full Xpath.	Double Forward Slash. Expression can start in the middle of the HTML DOM structure.
Speed	Faster. It identifies the element very fast.	Slower compare to absolute. it will take more time in identifying the element as we specify the partial path not (exact path).
Failure Chances	More. It Changes Frequently, if there are any changes made in the path of the element then XPath gets failed.	Failure chance of well written relative path is very less
Example	/html/head/body/form/table/tbody/tr/th	//span[contains(text(),"Success")]