

Ex6:- Opening Google.co.in using Chrome browser with Selenium Java WebDriver.

AIM:-

The objective of this lab Script is to demonstrate how to use Selenium WebDriver in Java to open a webpage(Google.co.in), interact with it using various WebDriver methods, and manage browser actions.

Requirements:

1. **JDK (Java Development Kit):** To compile and run Java programs, install JDK version 8 or later.
2. **Selenium WebDriver:** Selenium WebDriver is used for automating web applications for testing purposes. It interacts with web elements like buttons, links, and forms.
- 3.
4. **Chrome browser:** The latest version of Chrome installed.
5. **ChromeDriver:** A separate executable that WebDriver uses to control Chrome. The version of ChromeDriver must match the version of the installed Chrome browser.
6. **Eclipse IDE (or any other IDE):** For writing and executing Script

Theory:

Introduction to Selenium WebDriver:

Selenium WebDriver is an open-source tool that allows automated testing of web applications. It supports multiple browsers like Chrome, Firefox, Safari, and Edge. WebDriver interacts with the browser by sending commands such as opening a URL, clicking on elements, entering text into fields, and so on. This makes it a powerful tool for functional testing of web applications.

WebDriver allows the execution of automation scripts in various programming languages like Java, Python, C#, etc. In this lab, we will use Java to interact with the Google India homepage through WebDriver.

How Selenium WebDriver Works:

Selenium WebDriver directly interacts with the web browser. It communicates with the browser's native automation support to perform operations. In the case of Chrome, Selenium uses ChromeDriver, a standalone server, which WebDriver uses to send commands to the browser.

The WebDriver interface defines methods such as:

- `get()`: Opens the specified URL.
- `manage().window().maximize()`: Maximizes the browser window.
- `quit()`: Closes the browser.

Steps for Setting Up Selenium WebDriver with ChromeDriver:

1. Install JDK and Configure Environment Variables:

- Download the latest version of JDK from [Oracle's official site](https://www.oracle.com/in/java/technologies/javase-downloads.html).
- Install JDK and configure the JAVA_HOME environment variable.
- To verify, open the command prompt and type:

2. Download and Install Eclipse IDE:

- Download Eclipse IDE for Java developers from Eclipse official site.
- Install Eclipse and open it.

<https://www.eclipse.org/downloads/>

3. Download Selenium WebDriver:

- Go to the Selenium official website and download the WebDriver for Java.
- Extract the files and include the .jar files into the Eclipse project libraries.

4. Download ChromeDriver:

- Visit ChromeDriver official page to download the ChromeDriver that matches your Chrome version.
- Place the downloaded chromedriver.exe in a known directory (e.g., C://chromedriver.exe).

5. Set Up Selenium in Eclipse:

- Create a new Java project in Eclipse.
- Add external JAR files from the Selenium WebDriver .zip file to your project's build path.
- Include ChromeDriver by setting up the path in the script.

Selenium Java WebDriver Script

// Importing required libraries

```
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class OpenGoogleIndia {
```

```
    public static void main(String[] args) {
```

// Setting up ChromeDriver path

```
        System.setProperty("webdriver.chrome.driver", "C://chromedriver.exe");
```

// Initializing WebDriver instance

```
        WebDriver driver = new ChromeDriver();
```

// Opening Google India homepage

```
        driver.get("https://www.google.co.in");
```

// Maximize the browser window

```
        driver.manage().window().maximize();
```

// Print the title of the page

```
        System.out.println("Page title is: " + driver.getTitle());
```

// Pause for observation (5 seconds)

```
        Thread.sleep(5000);
```

// Close the browser

```
        driver.quit();
```

```
    }  
}
```

Procedure for Running the Script:

1. **Open Eclipse IDE:**
 - Launch Eclipse and open the Java project you created.
2. **Create a New Class:**
 - In the project's `src` folder, create a new Java class named `OpenGoogleIndia`.
3. **Write the Code:**
 - Write the code into the class.
4. **Run the Program:**
 - Right-click on the class file and select `Run As -> Java Application`.
5. **Observe the Output:**
 - The Chrome browser should open automatically, navigate to Google India, and display the title of the page in the console.

Common Issues and Troubleshooting:

1. **ChromeDriver Version Mismatch:** If ChromeDriver does not match the version of your installed Chrome browser, the browser may fail to launch. Ensure that the ChromeDriver version matches the version of the browser.
2. **Path to ChromeDriver:** If the path to ChromeDriver is incorrect, WebDriver won't be able to locate it. Double-check the path you've provided in `System.setProperty()`.
3. **Timeout Issues:** The script may terminate too quickly to observe the result. Increasing the sleep time allows better observation.
4. **Selenium WebDriver Exceptions:** Common exceptions include:
 - `SessionNotCreatedException`: Occurs if WebDriver cannot create a session due to ChromeDriver issues.
 - `NoSuchElementException`: Happens if you try to interact with an element that isn't present on the page.

Output:

- The Chrome browser opened automatically.
- It navigated to the URL "<https://www.google.co.in>".
- The browser maximize itself.
- The title of the Google India page will be printed in the console (e.g., "Google").
- The browser will remain open for 5 seconds and then close.

Result:-

we have successfully opened the Google India homepage using Chrome browser in Selenium WebDriver