iamneo

WebDriver and WebElements Commands

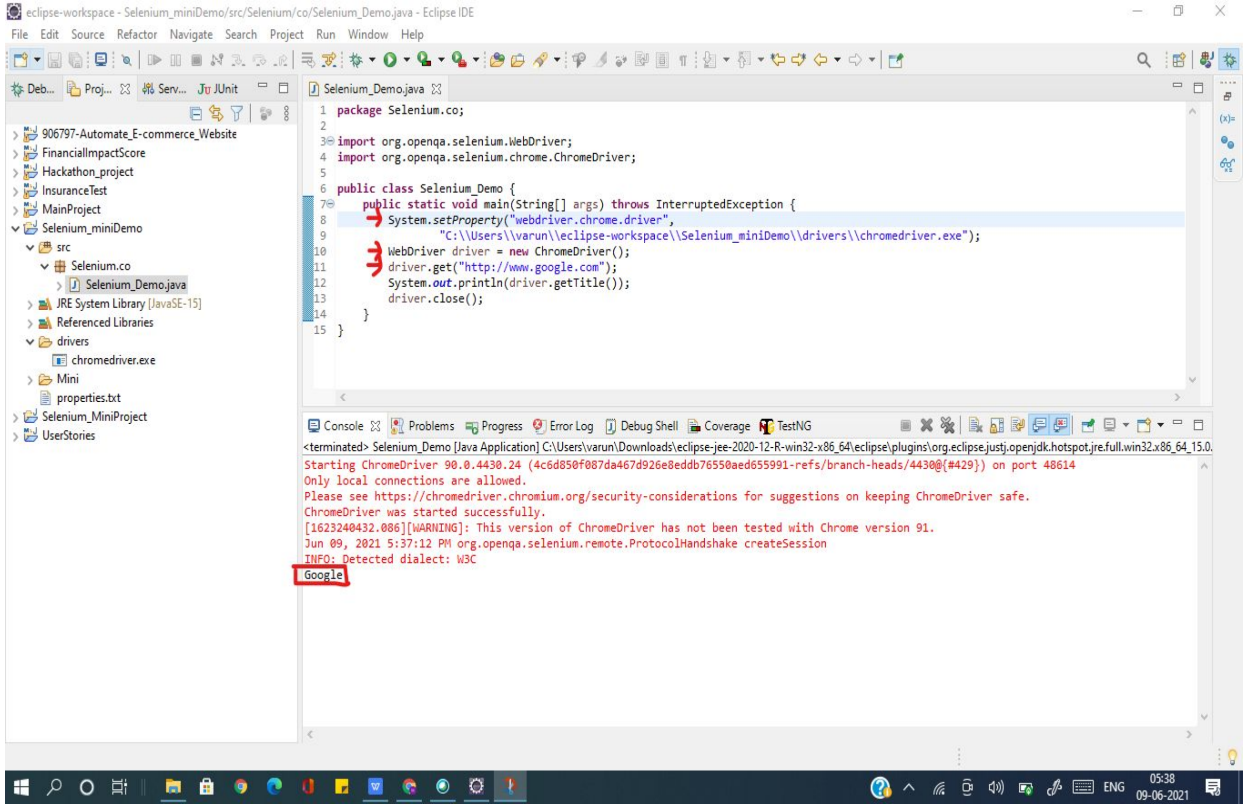
Launching Chrome Browser (Cmd)

- ✔ Open Google

  - ✔ Driver Launched and the Browser gets opened.

  - ✔ Here **driver.get ("https://www.google.com")** open the google by using the driver.

“System.out.println(driver.getTitle())” prints the title of the website opened in the chrome browser


eclipse-workspace - Selenium_miniDemo/src/Selenium/co/Selenium_Demo.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Deb... Proj... Serv... JUnit
906797-Automate_E-commerce_Website
FinancialImpactScore
Hackathon_project
InsuranceTest
MainProject
Selenium_miniDemo
src
Selenium.co
Selenium_Demo.java
JRE System Library [JavaSE-15]
Referenced Libraries
drivers
chromedriver.exe
Mini
properties.txt
Selenium_MiniProject
UserStories
Selenium_Demo.java
Console Problems Progress Error Log Debug Shell Coverage TestNG
ENG
05:38
09-06-2021

```java
package Selenium.co;

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

public class Selenium_Demo {
    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.chrome.driver",
                "C:\\Users\\varun\\eclipse-workspace\\Selenium_miniDemo\\drivers\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("http://www.google.com");
        System.out.println(driver.getTitle());
        driver.close();
    }
}
```

```
<terminated> Selenium_Demo [Java Application] C:\Users\varun\Downloads\eclipse-jee-2020-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.
Starting ChromeDriver 90.0.4430.24 (4c6d850f087da467d926e8eddb76550aed655991-refs/branch-heads/4430@{#429}) on port 48614
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
[1623240432.086][WARNING]: This version of ChromeDriver has not been tested with Chrome version 91.
Jun 09, 2021 5:37:12 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Google
```

Handling Browser Navigation

- ✔ WebDriver provides some basic Browser Navigation Commands that allows the browser to move backwards or forwards in the browser's history.
- ✔ Two Different kinds of browser navigation are available:

  1) Navigate using URL

  2) Navigate using commands

Navigate using URL

- ✔ Command used for navigation with the help of URL

  driver.navigate().to(appUrl)

- ✔ Here appUrl is a Website url that the user need to navigate.

For Example :

- ✔ The driver is present in

  driver.get(“https://www.Google.com”).

- ✔ The user need to navigate to facebook use

  driver.navigate().to("https://www.facebook.com");

Navigate using commands

Command used for navigation with the help of command

driver.navigate().forward()-Takes you forward by one page on the browser’s history.

driver.navigate().back()-Takes you back by one page on the browser’s history.

driver.navigate().refresh()-Takes you back the refreshed current URL in the browser.

Handling Browser Window

A window handle is a unique identifier that holds the address of all the windows. This is basically a pointer to a window, which returns the string value. This window handle function helps in getting the handles of all the windows.

- ✔ **get.windowhandle():** This method helps to get the window handle of the current window
- ✔ **get.windowhandles():** This method helps to get the handles of all the windows opened
- ✔ **set:** This method helps to set the window handles in the form of a string. set<string> **set= driver.get.windowhandles()**
- ✔ **switch to:** This method helps to switch between the windows
- ✔ **action:** This method helps to perform certain actions on the windows

What is Web Element?

A Web Element represents an HTML element. We see the elements as buttons, text, links, images, etc. on a web page.

Types of Web Element

1) Textbox

2) Button

3) Radio button

4) Checkbox

5) Dropdown

6) Link

7)Label

List of Selenium WebElement Commands

The Few commands are listed below:

sendKeys():allows the user to type content automatically into an editable field while executing tests.

element.sendKeys("text");

isDisplayed():If the element is displayed, then the value returned is true. If not, then the value returned is a NoSuchElementFound exception.

element.isDisplayed();

isSelected():This command only works on input elements such as radio buttons, checkboxes, select options, and menu items.

element.isSelected();

List of Selenium WebElement Commands

submit():This command is handy when interacting with forms (or elements within a form) on a web page.

element.submit();

isEnabled():This WebElement in Selenium command verifies if an element is enabled on the web page.If the element is enabled, it returns a true value.

element.isEnabled();

clear( ):When using this WebElement in Selenium command, its value will be cleared if the element in question is a text entry.

element.clear();

click():The click() command lets the tester replicate the click action on a button, link, radio button or checkbox.

element.click();