**Module-7(Selenium Webdriver)**

• W.A.J.Script for Locating links by linkText() and partialLinkText()

// Locating link by linkText

driver.findElement(By.linkText("Full Link Text")).click();

// Locating link by partialLinkText

driver.findElement(By.partialLinkText("Partial Link")).click();

• W.A.J.Script for Selecting multiple items in a drop dropdown

// Selecting multiple items in a dropdown

Select dropdown = new Select(driver.findElement(By.id("dropdownId")));

dropdown.selectByVisibleText("Option 1");

dropdown.selectByVisibleText("Option 2");

• W.A.J. script to use different methods to manage the windows-alerts and pop ups.

// Handling alert

Alert alert = driver.switchTo().alert();

alert.accept(); // To accept the alert

alert.dismiss(); // To dismiss the alert

alert.sendKeys("Input Text"); // To send text to the alert

• W.A.J.script to register your self in Gmail.

// Registering in Gmail

driver.get("https://accounts.google.com/signup");

driver.findElement(By.id("firstName")).sendKeys("YourFirstName");

driver.findElement(By.id("lastName")).sendKeys("YourLastName");

driver.findElement(By.id("username")).sendKeys("YourUsername");

driver.findElement(By.name("Passwd")).sendKeys("YourPassword");

driver.findElement(By.name("ConfirmPasswd")).sendKeys("YourPassword");

driver.findElement(By.id("accountDetailsNext")).click();

• W.A.J. Script To perform the radio button to select one by one in loop <https://demo.automationtesting.in/Register.html>

// Selecting radio buttons one by one in a loop

List<WebElement> radioButtons = driver.findElements(By.name("radiooptions"));

for (WebElement radioButton : radioButtons) {

radioButton.click();

// Perform any additional actions here

}

• W.A.J. script To write the script for image of logo facebook using xpath.

// Locating the Facebook logo image using XPath

WebElement facebookLogo = driver.findElement(By.xpath("//img[@alt='Facebook']"));

• W.A.J.Script To write a script for drop down.

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.ui.Select;

public class DropDownExample {

public static void main(String[] args) {

// Set the path of the ChromeDriver executable

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

// Initialize WebDriver

WebDriver driver = new ChromeDriver();

// Navigate to the webpage with the drop-down

driver.get("URL\_of\_the\_webpage");

// Locate the drop-down element by its ID

WebElement dropDown = driver.findElement(By.id("dropDownElementID"));

// Initialize the Select class with the drop-down element

Select select = new Select(dropDown);

// Select an option by its visible text

select.selectByVisibleText("OptionText");

// Select an option by its value attribute

select.selectByValue("OptionValue");

// Select an option by its index (starting from 0)

select.selectByIndex(1);

// Close the browser

driver.quit();

}

}

• W.A.J.Script To use Mouse and Keyboard event using Action class

1. Mouse Hover Event

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.interactions.Actions;

public class MouseHoverExample {

public static void main(String[] args) {

// Set the path of the ChromeDriver executable

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

// Initialize WebDriver

WebDriver driver = new ChromeDriver();

// Navigate to the webpage with the element to hover over

driver.get("URL\_of\_the\_webpage");

// Locate the element to hover over by its ID

WebElement hoverElement = driver.findElement(By.id("hoverElementID"));

// Initialize the Actions class

Actions actions = new Actions(driver);

// Perform the mouse hover action

actions.moveToElement(hoverElement).perform();

// Close the browser

driver.quit();

}

}

2. Keyboard event

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.Keys;

public class KeyboardEventExample {

public static void main(String[] args) {

// Set the path of the ChromeDriver executable

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

// Initialize WebDriver

WebDriver driver = new ChromeDriver();

// Navigate to the webpage with the textbox

driver.get("URL\_of\_the\_webpage");

// Locate the textbox element by its ID

WebElement textbox = driver.findElement(By.id("textboxID"));

// Initialize the Actions class

Actions actions = new Actions(driver);

// Perform the keyboard event: Send text to the textbox

actions.sendKeys(textbox, "Sample text").perform();

// Perform the keyboard event: Press Enter key

actions.sendKeys(textbox, Keys.ENTER).perform();

// Close the browser

driver.quit();

}

}

• W.A.J. Script How to handled Alert in selenium

import org.openqa.selenium.Alert;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class AlertHandlingExample {

public static void main(String[] args) {

// Set the path of the ChromeDriver executable

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

// Initialize WebDriver

WebDriver driver = new ChromeDriver();

// Navigate to the webpage with the alert

driver.get("URL\_of\_the\_webpage");

// Locate the element that triggers the alert

WebElement triggerAlertButton = driver.findElement(By.id("triggerAlertButtonID"));

// Click the button to trigger the alert

triggerAlertButton.click();

// Switch to the alert

Alert alert = driver.switchTo().alert();

// Handle the alert

// 1. Accept the alert

alert.accept();

// 2. Dismiss the alert

// alert.dismiss();

// 3. Get the alert message

// String alertMessage = alert.getText();

// System.out.println("Alert Message: " + alertMessage);

// 4. Send text to the alert

// alert.sendKeys("Sample text");

// Close the browser

driver.quit();

}

}

• W.A.J. Script To find the total hyperlink from this web page

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.List;

public class HyperlinkCount {

public static void main(String[] args) {

// Set the path of the ChromeDriver executable

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

// Initialize WebDriver

WebDriver driver = new ChromeDriver();

// Navigate to the web page

driver.get("https://qavbox.github.io/demo/webtable/");

// Find all hyperlink elements by the tag name "a"

List<WebElement> hyperlinks = driver.findElements(By.tagName("a"));

// Get the total count of hyperlinks

int hyperlinkCount = hyperlinks.size();

// Print the total number of hyperlinks

System.out.println("Total number of hyperlinks on the web page: " + hyperlinkCount);

// Optionally, print each hyperlink's text and URL

for (WebElement link : hyperlinks) {

System.out.println("Link Text: " + link.getText());

System.out.println("Link URL: " + link.getAttribute("href"));

}

// Close the browser

driver.quit();

}

}

• W.a.junit program to handled Assert class with all method to check its pass or fail

import org.junit.Assert;

import org.junit.Test;

public class AssertExampleTest {

@Test

public void testAssertEquals() {

// Assert that two values are equal

int expected = 5;

int actual = 5;

Assert.assertEquals("Values should be equal", expected, actual);

}

@Test

public void testAssertNotEquals() {

// Assert that two values are not equal

int unexpected = 5;

int actual = 10;

Assert.assertNotEquals("Values should not be equal", unexpected, actual);

}

@Test

public void testAssertTrue() {

// Assert that a condition is true

boolean condition = (5 > 3);

Assert.assertTrue("Condition should be true", condition);

}

@Test

public void testAssertFalse() {

// Assert that a condition is false

boolean condition = (5 < 3);

Assert.assertFalse("Condition should be false", condition);

}

@Test

public void testAssertNull() {

// Assert that an object is null

Object obj = null;

Assert.assertNull("Object should be null", obj);

}

@Test

public void testAssertNotNull() {

// Assert that an object is not null

Object obj = new Object();

Assert.assertNotNull("Object should not be null", obj);

}

@Test

public void testAssertArrayEquals() {

// Assert that two arrays are equal

int[] expectedArray = {1, 2, 3};

int[] actualArray = {1, 2, 3};

Assert.assertArrayEquals("Arrays should be equal", expectedArray, actualArray);

}

@Test

public void testAssertSame() {

// Assert that two objects refer to the same instance

String str1 = "JUnit";

String str2 = str1;

Assert.assertSame("Both strings should refer to the same instance", str1, str2);

}

@Test

public void testAssertNotSame() {

// Assert that two objects do not refer to the same instance

String str1 = "JUnit";

String str2 = new String("JUnit");

Assert.assertNotSame("Both strings should not refer to the same instance", str1, str2);

}

}

• W.a. junit program to perform test with webdriver to login process of facebook

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class FacebookLoginTest {

private WebDriver driver;

@Before

public void setUp() {

// Set the path of the ChromeDriver executable

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

// Initialize WebDriver

driver = new ChromeDriver();

// Navigate to Facebook login page

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

}

@Test

public void testLogin() {

// Locate the email input field and enter the email

WebElement emailField = driver.findElement(By.id("email"));

emailField.sendKeys("your\_email@example.com");

// Locate the password input field and enter the password

WebElement passwordField = driver.findElement(By.id("pass"));

passwordField.sendKeys("your\_password");

// Locate the login button and click it

WebElement loginButton = driver.findElement(By.name("login"));

loginButton.click();

// Optional: Verify the login was successful by checking the presence of a specific element

// For example, the presence of the profile icon or the home page URL

WebElement profileIcon = driver.findElement(By.xpath("//div[@aria-label='Account']"));

Assert.assertTrue("Profile icon should be displayed after login", profileIcon.isDisplayed());

}

@After

public void tearDown() {

// Close the browser

driver.quit();

}

}

• W.a. junit program to check gmail login using with @before,@after,@Test

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class GmailLoginTest {

private WebDriver driver;

@Before

public void setUp() {

// Set the path of the ChromeDriver executable

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

// Initialize WebDriver

driver = new ChromeDriver();

// Navigate to Gmail login page

driver.get("https://mail.google.com/");

}

@Test

public void testLogin() {

// Locate the email input field and enter the email

WebElement emailField = driver.findElement(By.id("identifierId"));

emailField.sendKeys("your\_email@gmail.com");

// Locate the "Next" button and click it

WebElement nextButton = driver.findElement(By.id("identifierNext"));

nextButton.click();

// Wait for the password field to be visible

try {

Thread.sleep(2000); // This is a simple wait; use WebDriverWait for a better approach

} catch (InterruptedException e) {

e.printStackTrace();

}

// Locate the password input field and enter the password

WebElement passwordField = driver.findElement(By.name("password"));

passwordField.sendKeys("your\_password");

// Locate the "Next" button and click it

WebElement nextPasswordButton = driver.findElement(By.id("passwordNext"));

nextPasswordButton.click();

// Optional: Verify the login was successful by checking the presence of a specific element

// For example, checking the presence of the inbox

try {

Thread.sleep(5000); // Wait for login to complete

} catch (InterruptedException e) {

e.printStackTrace();

}

WebElement inboxElement = driver.findElement(By.xpath("//div[text()='Inbox']"));

Assert.assertTrue("Inbox should be displayed after login", inboxElement.isDisplayed());

}

@After

public void tearDown() {

// Close the browser

driver.quit();

}

}

• W.a. junit program to use parameterized demo with multiple parameter of Facebook login in junit.

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.junit.runner.RunWith;

import org.junit.runners.Parameterized;

import org.junit.runners.Parameterized.Parameters;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.Arrays;

import java.util.Collection;

import static org.junit.Assert.assertTrue;

@RunWith(Parameterized.class)

public class FacebookLoginParameterizedTest {

private WebDriver driver;

private String email;

private String password;

// Constructor to initialize parameters

public FacebookLoginParameterizedTest(String email, String password) {

this.email = email;

this.password = password;

}

// Define the parameters

@Parameters

public static Collection<Object[]> data() {

return Arrays.asList(new Object[][] {

{ "your\_email1@example.com", "your\_password1" },

{ "your\_email2@example.com", "your\_password2" },

{ "your\_email3@example.com", "your\_password3" }

});

}

@Before

public void setUp() {

// Set the path of the ChromeDriver executable

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

// Initialize WebDriver

driver = new ChromeDriver();

// Navigate to Facebook login page

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

}

@Test

public void testLogin() {

// Locate the email input field and enter the email

WebElement emailField = driver.findElement(By.id("email"));

emailField.sendKeys(email);

// Locate the password input field and enter the password

WebElement passwordField = driver.findElement(By.id("pass"));

passwordField.sendKeys(password);

// Locate the login button and click it

WebElement loginButton = driver.findElement(By.name("login"));

loginButton.click();

// Optional: Verify the login was successful by checking the presence of a specific element

// For example, the presence of the profile icon or the home page URL

// Assuming the test will fail if login is unsuccessful

try {

Thread.sleep(5000); // Wait for login to complete

} catch (InterruptedException e) {

e.printStackTrace();

}

WebElement profileIcon = driver.findElement(By.xpath("//div[@aria-label='Account']"));

assertTrue("Profile icon should be displayed after login", profileIcon.isDisplayed());

}

@After

public void tearDown() {

// Close the browser

driver.quit();

}

}

• W.a. TestNG program to perform test with webdriver to login processof facebook

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class FacebookLoginTestNG {

private WebDriver driver;

@BeforeClass

public void setUp() {

// Set the path of the ChromeDriver executable

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

// Initialize WebDriver

driver = new ChromeDriver();

// Navigate to Facebook login page

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

}

@Test

public void testLogin() {

// Locate the email input field and enter the email

WebElement emailField = driver.findElement(By.id("email"));

emailField.sendKeys("your\_email@example.com");

// Locate the password input field and enter the password

WebElement passwordField = driver.findElement(By.id("pass"));

passwordField.sendKeys("your\_password");

// Locate the login button and click it

WebElement loginButton = driver.findElement(By.name("login"));

loginButton.click();

// Optional: Verify the login was successful by checking the presence of a specific element

// For example, the presence of the profile icon or the home page URL

// Assuming the test will fail if login is unsuccessful

try {

Thread.sleep(5000); // Wait for login to complete

} catch (InterruptedException e) {

e.printStackTrace();

}

WebElement profileIcon = driver.findElement(By.xpath("//div[@aria-label='Account']"));

Assert.assertTrue(profileIcon.isDisplayed(), "Profile icon should be displayed after login");

}

@AfterClass

public void tearDown() {

// Close the browser

driver.quit();

}

}

• W.a. TestNG program to check gmail login using with @beforetest,@aftertest,@Test

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

public class GmailLoginTestNG {

private WebDriver driver;

@BeforeTest

public void setUp() {

// Set the path of the ChromeDriver executable

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

// Initialize WebDriver

driver = new ChromeDriver();

// Navigate to Gmail login page

driver.get("https://mail.google.com/");

}

@Test

public void testLogin() {

// Locate the email input field and enter the email

WebElement emailField = driver.findElement(By.id("identifierId"));

emailField.sendKeys("your\_email@gmail.com");

// Locate the "Next" button and click it

WebElement nextButton = driver.findElement(By.id("identifierNext"));

nextButton.click();

// Wait for the password field to be visible

try {

Thread.sleep(2000); // This is a simple wait; use WebDriverWait for a better approach

} catch (InterruptedException e) {

e.printStackTrace();

}

// Locate the password input field and enter the password

WebElement passwordField = driver.findElement(By.name("password"));

passwordField.sendKeys("your\_password");

// Locate the "Next" button and click it

WebElement nextPasswordButton = driver.findElement(By.id("passwordNext"));

nextPasswordButton.click();

// Optional: Verify the login was successful by checking the presence of a specific element

// For example, checking the presence of the inbox

try {

Thread.sleep(5000); // Wait for login to complete

} catch (InterruptedException e) {

e.printStackTrace();

}

WebElement inboxElement = driver.findElement(By.xpath("//div[text()='Inbox']"));

Assert.assertTrue(inboxElement.isDisplayed(), "Inbox should be displayed after login");

}

@AfterTest

public void tearDown() {

// Close the browser

driver.quit();

}

}

• W.a. TestNG program to use parameterized demo with multiple parameter of Facebook login with TestNG.

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.DataProvider;

import org.testng.annotations.Test;

public class FacebookLoginTestNG {

private WebDriver driver;

@BeforeClass

public void setUp() {

// Set the path of the ChromeDriver executable

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

// Initialize WebDriver

driver = new ChromeDriver();

// Navigate to Facebook login page

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

}

@DataProvider(name = "loginData")

public Object[][] getData() {

return new Object[][] {

{ "your\_email1@example.com", "your\_password1" },

{ "your\_email2@example.com", "your\_password2" },

{ "your\_email3@example.com", "your\_password3" }

};

}

@Test(dataProvider = "loginData")

public void testLogin(String email, String password) {

// Locate the email input field and enter the email

WebElement emailField = driver.findElement(By.id("email"));

emailField.sendKeys(email);

// Locate the password input field and enter the password

WebElement passwordField = driver.findElement(By.id("pass"));

passwordField.sendKeys(password);

// Locate the login button and click it

WebElement loginButton = driver.findElement(By.name("login"));

loginButton.click();

// Optional: Verify the login was successful by checking the presence of a specific element

// For example, the presence of the profile icon or the home page URL

// Assuming the test will fail if login is unsuccessful

try {

Thread.sleep(5000); // Wait for login to complete

} catch (InterruptedException e) {

e.printStackTrace();

}

WebElement profileIcon = driver.findElement(By.xpath("//div[@aria-label='Account']"));

Assert.assertTrue(profileIcon.isDisplayed(), "Profile icon should be displayed after login");

// Navigate back to the Facebook login page for the next iteration

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

}

@AfterClass

public void tearDown() {

// Close the browser

driver.quit();

}

}

• W.a. TestNG program to create group with testing.xml file

import org.testng.annotations.Test;

public class TestClass1 {

@Test(groups = { "group1" })

public void testMethod1() {

System.out.println("TestClass1 - testMethod1 in group1");

}

@Test(groups = { "group2" })

public void testMethod2() {

System.out.println("TestClass1 - testMethod2 in group2");

}

}

• w.a. TestNG program to create dataprovider.

import org.testng.Assert;

import org.testng.annotations.DataProvider;

import org.testng.annotations.Test;

public class DataProviderExampleTestNG {

// DataProvider method

@DataProvider(name = "loginData")

public Object[][] getData() {

return new Object[][] {

{ "user1@example.com", "password1" },

{ "user2@example.com", "password2" },

{ "user3@example.com", "password3" }

};

}

// Test method using DataProvider

@Test(dataProvider = "loginData")

public void testLogin(String email, String password) {

System.out.println("Testing login with Email: " + email + " and Password: " + password);

// Dummy logic to simulate login test

String expectedEmailDomain = "example.com";

Assert.assertTrue(email.endsWith(expectedEmailDomain), "Email should end with " + expectedEmailDomain);

Assert.assertNotNull(password, "Password should not be null");

}

}