**MODULE-8 (Adv. Selenium - Web Driver Advance)**

1. W.a.framework program for data driven framework -to get value from the excel and check into your website https://demo.automationtesting.in/SignIn.html Direct check with login which emaild through login successful or not

Pom.xml

<dependencies>

<!-- TestNG Dependency -->

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.4.0</version>

<scope>test</scope>

</dependency>

<!-- Selenium Dependency -->

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>4.1.2</version>

</dependency>

<!-- Apache POI Dependency -->

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>5.2.2</version>

</dependency>

</dependencies>

Java

import org.apache.poi.ss.usermodel.DataFormatter;

import org.apache.poi.ss.usermodel.Row;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.ss.usermodel.Workbook;

import org.apache.poi.ss.usermodel.WorkbookFactory;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

public class ExcelUtils {

public static List<String[]> getExcelData(String filePath, String sheetName) throws IOException {

List<String[]> data = new ArrayList<>();

FileInputStream fileInputStream = new FileInputStream(new File(filePath));

Workbook workbook = WorkbookFactory.create(fileInputStream);

Sheet sheet = workbook.getSheet(sheetName);

DataFormatter formatter = new DataFormatter();

for (Row row : sheet) {

String[] rowData = new String[row.getPhysicalNumberOfCells()];

for (int cellNum = 0; cellNum < row.getPhysicalNumberOfCells(); cellNum++) {

rowData[cellNum] = formatter.formatCellValue(row.getCell(cellNum));

}

data.add(rowData);

}

workbook.close();

fileInputStream.close();

return data;

}

}

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;

import java.io.IOException;

import java.util.List;

public class DataDrivenTest {

private WebDriver driver;

private String excelFilePath = "path/to/your/excel/file.xlsx";

private String sheetName = "Sheet1";

@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 the login page

driver.get("https://demo.automationtesting.in/SignIn.html");

}

@DataProvider(name = "loginData")

public Object[][] loginData() throws IOException {

List<String[]> data = ExcelUtils.getExcelData(excelFilePath, sheetName);

Object[][] testData = new Object[data.size()][2];

for (int i = 0; i < data.size(); i++) {

testData[i][0] = data.get(i)[0]; // Email

testData[i][1] = data.get(i)[1]; // Password

}

return testData;

}

@Test(dataProvider = "loginData")

public void testLogin(String email, String password) {

// Locate the email input field and enter the email

WebElement emailField = driver.findElement(By.xpath("//input[@placeholder='E mail']"));

emailField.clear();

emailField.sendKeys(email);

// Locate the password input field and enter the password

WebElement passwordField = driver.findElement(By.xpath("//input[@placeholder='Password']"));

passwordField.clear();

passwordField.sendKeys(password);

// Locate the login button and click it

WebElement loginButton = driver.findElement(By.id("enterbtn"));

loginButton.click();

// Check if login is successful or not

boolean loginSuccess;

try {

driver.findElement(By.xpath("//a[text()='SignOut']"));

loginSuccess = true;

} catch (Exception e) {

loginSuccess = false;

}

Assert.assertTrue(loginSuccess, "Login failed for email: " + email);

}

@AfterClass

public void tearDown() {

// Close the browser

if (driver != null) {

driver.quit();

}

}

}

1. W.a.framework program for keyword driven framework -given below keyword which you get from the Excel and check into youwebsite <https://demo.automationtesting.in/SignIn.html>

<dependencies>

<!-- TestNG Dependency -->

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.4.0</version>

<scope>test</scope>

</dependency>

<!-- Selenium Dependency -->

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>4.1.2</version>

</dependency>

<!-- Apache POI Dependency -->

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>5.2.2</version>

</dependency>

</dependencies>

import org.apache.poi.ss.usermodel.DataFormatter;

import org.apache.poi.ss.usermodel.Row;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.ss.usermodel.Workbook;

import org.apache.poi.ss.usermodel.WorkbookFactory;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

public class ExcelUtils {

public static List<String[]> getExcelData(String filePath, String sheetName) throws IOException {

List<String[]> data = new ArrayList<>();

FileInputStream fileInputStream = new FileInputStream(new File(filePath));

Workbook workbook = WorkbookFactory.create(fileInputStream);

Sheet sheet = workbook.getSheet(sheetName);

DataFormatter formatter = new DataFormatter();

for (Row row : sheet) {

String[] rowData = new String[row.getPhysicalNumberOfCells()];

for (int cellNum = 0; cellNum < row.getPhysicalNumberOfCells(); cellNum++) {

rowData[cellNum] = formatter.formatCellValue(row.getCell(cellNum));

}

data.add(rowData);

}

workbook.close();

fileInputStream.close();

return data;

}

}

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.DataProvider;

import org.testng.annotations.Test;

import java.io.IOException;

import java.util.List;

public class KeywordDrivenTest {

private WebDriver driver;

private String excelFilePath = "path/to/your/excel/file.xlsx";

private String sheetName = "Sheet1";

@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 the login page

driver.get("https://demo.automationtesting.in/SignIn.html");

}

@DataProvider(name = "keywordData")

public Object[][] keywordData() throws IOException {

List<String[]> data = ExcelUtils.getExcelData(excelFilePath, sheetName);

Object[][] testData = new Object[data.size()][];

for (int i = 0; i < data.size(); i++) {

testData[i] = data.get(i);

}

return testData;

}

@Test(dataProvider = "keywordData")

public void testKeywords(String action, String locatorType, String locatorValue, String value) {

performAction(action, locatorType, locatorValue, value);

}

public void performAction(String action, String locatorType, String locatorValue, String value) {

WebElement element = findElement(locatorType, locatorValue);

switch (action.toLowerCase()) {

case "click":

element.click();

break;

case "sendkeys":

element.sendKeys(value);

break;

case "clear":

element.clear();

break;

case "verify":

assert element.getText().equals(value) : "Verification failed!";

break;

// Add more actions as needed

default:

throw new IllegalArgumentException("Invalid action: " + action);

}

}

public WebElement findElement(String locatorType, String locatorValue) {

switch (locatorType.toLowerCase()) {

case "id":

return driver.findElement(By.id(locatorValue));

case "name":

return driver.findElement(By.name(locatorValue));

case "xpath":

return driver.findElement(By.xpath(locatorValue));

case "css":

return driver.findElement(By.cssSelector(locatorValue));

// Add more locators as needed

default:

throw new IllegalArgumentException("Invalid locator type: " + locatorType);

}

}

@AfterClass

public void tearDown() {

// Close the browser

if (driver != null) {

driver.quit();

}

}

}

1. W.a.framework program for hybrid driven framework -given below keyword and Data both which you get from the Excel andcheck into you website <https://demo.automationtesting.in/SignIn.html>

<dependencies>

<!-- TestNG Dependency -->

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.4.0</version>

<scope>test</scope>

</dependency>

<!-- Selenium Dependency -->

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>4.1.2</version>

</dependency>

<!-- Apache POI Dependency -->

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>5.2.2</version>

</dependency>

</dependencies>

import org.apache.poi.ss.usermodel.DataFormatter;

import org.apache.poi.ss.usermodel.Row;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.ss.usermodel.Workbook;

import org.apache.poi.ss.usermodel.WorkbookFactory;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

public class ExcelUtils {

public static List<String[]> getExcelData(String filePath, String sheetName) throws IOException {

List<String[]> data = new ArrayList<>();

FileInputStream fileInputStream = new FileInputStream(new File(filePath));

Workbook workbook = WorkbookFactory.create(fileInputStream);

Sheet sheet = workbook.getSheet(sheetName);

DataFormatter formatter = new DataFormatter();

for (Row row : sheet) {

String[] rowData = new String[row.getPhysicalNumberOfCells()];

for (int cellNum = 0; cellNum < row.getPhysicalNumberOfCells(); cellNum++) {

rowData[cellNum] = formatter.formatCellValue(row.getCell(cellNum));

}

data.add(rowData);

}

workbook.close();

fileInputStream.close();

return data;

}

}

import org.apache.poi.ss.usermodel.DataFormatter;

import org.apache.poi.ss.usermodel.Row;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.ss.usermodel.Workbook;

import org.apache.poi.ss.usermodel.WorkbookFactory;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

public class ExcelUtils {

public static List<String[]> getExcelData(String filePath, String sheetName) throws IOException {

List<String[]> data = new ArrayList<>();

FileInputStream fileInputStream = new FileInputStream(new File(filePath));

Workbook workbook = WorkbookFactory.create(fileInputStream);

Sheet sheet = workbook.getSheet(sheetName);

DataFormatter formatter = new DataFormatter();

for (Row row : sheet) {

String[] rowData = new String[row.getPhysicalNumberOfCells()];

for (int cellNum = 0; cellNum < row.getPhysicalNumberOfCells(); cellNum++) {

rowData[cellNum] = formatter.formatCellValue(row.getCell(cellNum));

}

data.add(rowData);

}

workbook.close();

fileInputStream.close();

return data;

}

}

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.DataProvider;

import org.testng.annotations.Test;

import java.io.IOException;

import java.util.List;

public class HybridDrivenTest {

private WebDriver driver;

private String excelFilePath = "path/to/your/excel/file.xlsx";

private String sheetName = "Sheet1";

@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 the login page

driver.get("https://demo.automationtesting.in/SignIn.html");

}

@DataProvider(name = "keywordData")

public Object[][] keywordData() throws IOException {

List<String[]> data = ExcelUtils.getExcelData(excelFilePath, sheetName);

Object[][] testData = new Object[data.size()][];

for (int i = 0; i < data.size(); i++) {

testData[i] = data.get(i);

}

return testData;

}

@Test(dataProvider = "keywordData")

public void testKeywords(String action, String locatorType, String locatorValue, String value) {

performAction(action, locatorType, locatorValue, value);

}

public void performAction(String action, String locatorType, String locatorValue, String value) {

WebElement element = findElement(locatorType, locatorValue);

switch (action.toLowerCase()) {

case "click":

element.click();

break;

case "sendkeys":

element.sendKeys(value);

break;

case "clear":

element.clear();

break;

case "verify":

assert element.getText().equals(value) : "Verification failed!";

break;

// Add more actions as needed

default:

throw new IllegalArgumentException("Invalid action: " + action);

}

}

public WebElement findElement(String locatorType, String locatorValue) {

switch (locatorType.toLowerCase()) {

case "id":

return driver.findElement(By.id(locatorValue));

case "name":

return driver.findElement(By.name(locatorValue));

case "xpath":

return driver.findElement(By.xpath(locatorValue));

case "css":

return driver.findElement(By.cssSelector(locatorValue));

// Add more locators as needed

default:

throw new IllegalArgumentException("Invalid locator type: " + locatorType);

}

}

@AfterClass

public void tearDown() {

// Close the browser

if (driver != null) {

driver.quit();

}

}

}

1. W.a.maven program to create simple webdriver Program

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>simple-webdriver</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<!-- Selenium Dependency -->

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>4.1.2</version>

</dependency>

</dependencies>

</project>

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class SimpleWebDriver {

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 a website

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

// Print the title of the page

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

// Close the browser

driver.quit();

}

}

1. W.a.maven program to create Junit with webdriver Program

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>junit-webdriver</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<!-- Selenium Dependency -->

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>4.1.2</version>

</dependency>

<!-- JUnit Dependency -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import static org.junit.Assert.assertEquals;

public class JUnitWebDriverTest {

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 a website

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

}

@Test

public void testTitle() {

// Verify the title of the page

String expectedTitle = "Google";

String actualTitle = driver.getTitle();

assertEquals(expectedTitle, actualTitle);

}

@After

public void tearDown() {

// Close the browser

if (driver != null) {

driver.quit();

}

}

}

1. W.a.maven program to create TestNG with Webdriver Program.

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>testng-webdriver</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<!-- Selenium Dependency -->

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>4.1.2</version>

</dependency>

<!-- TestNG Dependency -->

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.4.0</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

import org.openqa.selenium.WebDriver;

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 TestNGWebDriverTest {

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 a website

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

}

@Test

public void testTitle() {

// Verify the title of the page

String expectedTitle = "Google";

String actualTitle = driver.getTitle();

Assert.assertEquals(actualTitle, expectedTitle);

}

@AfterClass

public void tearDown() {

// Close the browser

if (driver != null) {

driver.quit();

}

}

}