**setProperty**

1. Its is used set properties for the desired browser for test automation. Browser does not have any inbuilt server to run the automation code.
2. This method has 2 attribute:

* propertyName- name of the browser-specific driver
* value-path of browser driver

If not set -we will get exception: **IllegalStateException**We can set this using key value mapping in configuration file but it is not advisable to do so because if chrome browser update happens, it will not reflected in configuration files.

**System.setProperty(webdriver.chrome.driver,driverpath)**

FILE HANDLING

Type=file

Then use below format to upload files

Type=text()

ROBOT CLASS

1.

2. Add

Webpage navigation handling

1.navigate().forward()

2.navigate().back()

3.navigate().refresh()

4.navigate().to

TestNg

@DataPrivider

1.When we pass different parameter value in the testcase(@test) from compared to Dataprovider parameters, we get exception called as ‘MethodMatcherException”

**Types of Assertions**

* Hard Assertions
* Soft Assertions (Verify Method)

**Hard assertion**

To use hard assertion we use **predefined class ‘Assert’**

There are several types of hard assertions in Selenium, including:

* 1. **assertEquals: assertEquals()** is a method that takes a minimum of 2 arguments and compares actual results with expected results. If both match, the assertion is passed, and the test case is marked as passed. **assertEquals()** can compare Strings, Integers, Doubles, and many more variables, as shown in the image below.
  2. **assertNotEquals()** is a method that does the opposite of the **assertEquals()** method. In this case, the method compares the actual and expected result. But if the assertion condition is met if the two are not identical. The test case is marked as passed if actual and expected results are not the same.
  3. **assertTrue()**: This Assertion verifies the Boolean value returned by the condition. If the Boolean value is true, then the assertion passes the test case.
  4. **assertFalse()**: This method works the opposite of **assertTrue()**. The Assertion verifies the Boolean value returned by the condition. If the Boolean value is false, the assertion passes the test case.
  5. **assertNull():**This method verifies if the expected output is null. If not, the value returned is false. f the condition is not met, then it will throw “***java.lang.AssertionError***” error, as shown in the example.
  6. **assertNotNull()**: This method works opposite to the **assertNull()** method. The assertion condition is met when the method validates the expected output to be not null.

**Example:** AssertEqulas(actURL, expURL,”invalidcredentials”)

**Note:** When testcase fails, we get AssertionError

**Soft Assertions**

To user soft assertion, we **should define object for SoftAssert class**

SoftAssert softAssert = new SoftAssert();

softAssert.assertNotEquals(getActualTitle, "Most Reliable App & Cross Browser Testing Platform | BrowserStack");

softAssert.assertNull(verifyTitle);

softAssert.assertNotNull(verifyTitle);

softAssert.assertTrue("BrowserStack".equals("Browserstack"), "First soft assert failed");

softAssert.assertFalse("BrowserStack".equals("BrowserStack"), "Second soft assert failed");

* The code below uses **assertAll()** to see assertion results at the end of the test

**Code Line-27:**

softAssert.assertAll();

READ/WRITE EXCEL

1. Always while writing to excel make a copy of excel file to have backup if file gets corrupted during write to operation
2. When performing read and write to same excel , always configure fos object only after configuring file,fis,workbook,sheet

Cucumber

Remove property configu from line number

Githib link for pom.xml config

<https://github.com/cucumber/cucumber-java-skeleton/commit/d7249b50c570816eba27ce94557e1de7e9b0f97>

line number 11 from <properties> to line number 41 till </dependencies>

Add dependency of selenium java from

<https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java/4.23.1>

**Keywords:**

**Feature**: objective of test case

Scenario: test scenarios

Given -pre-requisite

When- test case steps

**Then** 🡪

And/But 🡪 to combine multiple conditions

**1.Feature file**

**NOTE**: Same **Given** statement **should not be used in different feature files**

Feature1.feature file

**Given Open google page**

Feature2.feature file

**Given Open google page--** is not allowed instead mention

Given Open google page in browser

**Execute feature file to generate glue code**

**NOTE**:

When we execute feature file is executed- gluecode will be generated

**2.StepDef file**

Define a normal class and paste glue code generated and then add required logic

**Execute :** No need to execute step def file

**3.RUNNER file**

Define a runner class and add below 2 annotation

@RunWith(Cucumber.class)

@CucumberOptions(features={“path of feature file excluding project name”},

glue={“package name”} )

Note: no need to mention stepdef class name , cucumber will automaticll understand the stepdef file based on steps written in feature file

**Execute :** runner.java file as –“ Junit test “