**What are Assertions in TestNG?**

***Assertions in TestNG are a way to verify that the expected result and the actual result matched or not***. If we could decide the outcome on different small methods using assertions in our test case, we can determine whether our test failed or passed overall. An example of assertion can be logging into the website, checking the title of the webpage, verifying the functionality of an input box that takes only integers, etc.

We should remember that an assertion in TestNG is successful only if there are no exceptions thrown during the test case execution. TestNG asserts (*or assertions*) popularly validate the results in TestNG using selenium.

***Syntax for TestNG Assertions:***

Although there are many methods for assertions (*later in this article*), the generic syntax is:

***Assert.Method(actual, expected)***

The parameter as you see contains three values:

* ***Actual***: *The actual value that the tester gets like if the tester's assertion is on the title of the page then what was the actual title of the page goes here*.
* ***Expected***: *The value that you expect like if the tester's assertion is on the title of the page then what value of title do you expect goes here*.

Let's explore the assert in TestNG by writing a simple test case.

**How to use Asserts in TestNG using Selenium?**

As mentioned above, the assert statements just require the implementation of assert methods. However, since we perform assertions to verify the values, we will pass those as a parameter as "***actual\_value***" and "***expected\_value***". In the following code, we will try to assert conditions on the web page's title name using selenium webdriver.

import org.openqa.selenium.WebDriver;

import org.testng.Reporter;

import org.testng.annotations.Test;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

public class TestNG {

WebDriver driver = new ChromeDriver();

@Test (priority = 0)

public void CloseBrowser() {

driver.close();

Reporter.log("Driver Closed After Testing");

}

@Test (priority = -1)

public void OpenBrowser() {

Reporter.log("This test verifies the current selenium compatibility with TestNG by launching the chrome driver");

Reporter.log("Launching Google Chrome Driver version 81 for this test");

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

Reporter.log("The website used was DemoQA for this test", true);

String expectedTitle = "Free QA Automation Tools For Everyone";

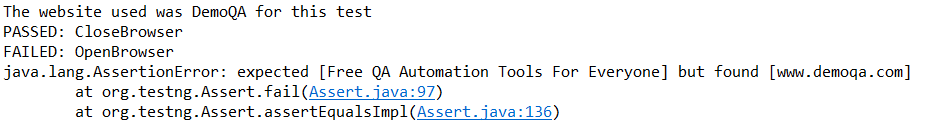
String originalTitle = driver.getTitle();

Assert.assertEquals(originalTitle, expectedTitle);

}

}

We have a combination of features of TestNG in the above code, such as [***TestNG Annotation***](https://www.toolsqa.com/testng/testng-annotations/), [***TestNG priority***](https://www.toolsqa.com/testng/testng-test-priority/), and [***TestNG Reporter***](https://www.toolsqa.com/testng/testng-reporter-log/). You can visit their respective tutorials to know their work.

Run the above-written code to execute the TestNG asserts.

So, our test method *OpenBrowser* failed because of the expected title, and the actual titles were different. It determined that our test failed. Since selenium tests require a lot of validations, you can practice using different things in the assert methods using TestNG.

***How to use Message as a Parameter in TestNG Asserts?***

In the above example, you may notice that an *AssertionError* is thrown by Java which is quite complete in its own form. But, in TestNG asserts, we can mention a message as well in the parameter that will be displayed when the assertion fails along with the assertion error. So, as a slight modification to the syntax, this will work as a third parameter making syntax as follows:

***Assert.Method(actual, expected, message)***

The parameter as you see contains three values:

* ***Actual***: *The actual value that the tester gets like if the tester's assertion is on the title of the page then what was the actual title of the page goes here*.
* ***Expected***: *The value that you expect like if the tester's assertion is on the title of the page then what value of title do you expect goes here*.
* ***Message***: *A string message to display only in case of an error when the assert fails*.

Let's explore the assert in TestNG by writing a simple test case.

The below-given code is exactly the same code as mentioned in the previous section. Therefore, I have mentioned just the *OpenBrowser* function since the changes are made only in this part of the code. In the following code, we have used "***Titles of the website do not match***" as the message parameter which is expected to be seen on the console if the assertion fails.

public void OpenBrowser() {

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

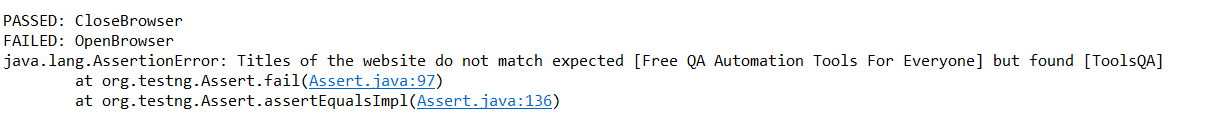
Reporter.log("The website used was DemoQA for this test", true);

String expectedTitle = "Free QA Automation Tools For Everyone";

String originalTitle = driver.getTitle();

Assert.assertEquals(originalTitle, expectedTitle, "Titles of the website do not match");

}

Run this on Eclipse to find the message along with the assertion exception error thrown by TestNG (*Java*).

As expected, the message that we passed is visible. Applying a message to the assertion makes it more meaningful and easy to understand for other team members.

**Different types of Asserts in TestNG**

There are two types of TestNg Assert:

* Hard Assert
* Soft Assert

***Hard Assert in TestNG***

***Hard Asserts*** are those asserts that stop the test execution when an assert statement fails, and the subsequent assert statements are therefore not validated. It plays a vital role in projects where we have an element without whose validation, asserting other elements is useless. One good example in such cases is the login functionality. If I want to see my past orders, for example, then what is the point of checking this test case when the login validation already failed? ***Hard asserts are the default type of asserts in TestNG***, and what we used in the previous section was Hard Assert.

***Soft Assert in TestNG***

Soft asserts are just the opposite of hard asserts. In soft asserts, the subsequent assertions keep on running even though one assert validation fails, i.e., the test execution does not stop. ***Soft assert does not include by default in TestNG***. For this, you need to include the package ***org.testng.asserts.Softassert***. So, when should we use soft asserts in TestNG? We use soft asserts when we do not care about the failure of specific validations and want the test execution to proceed and also want to see the exception errors.

A good example is multiple validations on an input form. Also, to note that on many platforms, you will see "***verify***" while learning about asserts. ***Soft asserts are also known as*** "***Verify***" and hence do not get confused about the same.

**How to use Soft Assert in TestNG?**

The following code will demonstrate the use of soft asserts in TestNG. In this code, we are validating the title of the web page, bypassing two different expected titles.

import org.openqa.selenium.WebDriver;

import org.testng.asserts.SoftAssert;

import org.testng.Reporter;

import org.testng.annotations.Test;

import org.openqa.selenium.chrome.ChromeDriver;

public class TestNG {

WebDriver driver = new ChromeDriver();

@Test (priority = 0)

public void CloseBrowser() {

driver.close();

Reporter.log("Driver Closed After Testing");

}

@Test (priority = -1)

public void OpenBrowser() {

Reporter.log("This test verifies the current selenium compatibility with TestNG by launching the chrome driver");

Reporter.log("Launching Google Chrome Driver version 81 for this test");

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

SoftAssert softassert = new SoftAssert();

Reporter.log("The website used was DemoQA for this test", true);

String expectedTitle = "Free QA Automation Tools For Everyone";

String originalTitle = driver.getTitle();

softassert.assertEquals(originalTitle, expectedTitle);

System.out.println("\*\*\* Checking For The Second Title \*\*\*");

// Checking title for ToolsQA – Demo Website to Practice Automation – Demo Website to Practice Automation

softassert.assertEquals(originalTitle, "ToolsQA – Demo Website to Practice Automation – Demo Website to Practice Automation" );

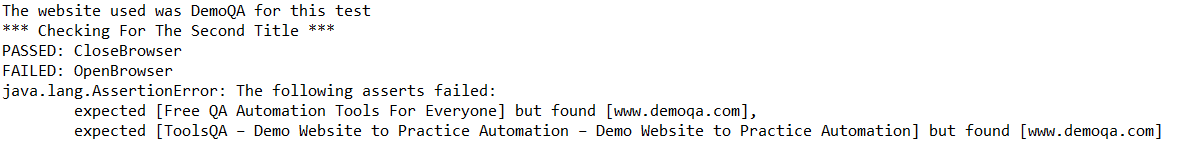
softassert.assertAll();

}

}

Important points to remember concerning soft assert that we can notice in the above code are:

* *Soft assert requires the external import of the package****import org.testng.asserts.SoftAssert****;.*
* *An object of the SoftAssert runs the assert statements*.
* *The object should have a life within the same test method in which we declared it*.
* *object.assertAll() statement is required to see the exceptions; otherwise, the tester won't know what passed and what failed*.

Run the above code and see the output:

It shows that our assertions executed and which ones failed. But the test execution did not stop.

**Commonly used TestNG Assert Methods**

All the TestNG Assert statements work in the same vein when we talk about the basic structure of its execution. But, like any other method, it can take different parameters and perform various types of validations in the test case methods. Since asserts are so crucial in TestNG and used so commonly, we will discuss some of the most common assertion methods.

* ***Assert.assertEqual(String actual, String expected)***: *Pass the actual string value and the expected string value as parameters. Validates if the actual and expected values are the same or not*.
* ***Assert.assertEqual(String actual, String expected, String message)***: *Similar to the previous method just that when the assertion fails, the message displays along with the exception thrown*.
* ***Assert.assertEquals(boolean actual, boolean expected)***: *Takes two boolean values as input and validates if they are equal or not*.
* ***Assert.assertTrue(condition)***: *This method asserts if the condition is true or not. If not, then the exception error is thrown*.
* ***Assert.assertTrue(condition, message)***:  *Similar to the previous method with an addition of message, which is shown on the console when the assertion fails along with the exception*.
* ***Assert.assertFalse(condition)***: *This method asserts if the condition is false or not. If not, then it throws an exception error*.
* ***Assert.assertFalse(condition, message)***: Similar to the previous method but with an addition of a message string which is shown on the console when the assertion fails, i.e., the condition is true\*.
* ***public static void assertEquals(Object actual, Object expected, String message)***: *Asserts whether the two objects passed are equal or not. If not, the message and the exception error appears. The message parameter is optional*.
* ***public static void assertEquals(String actual, String expected, String message)***: *Asserts whether two strings are equal or not. If not, the message along with the exception error displays. The message parameter is optional*.

And there are many more which you can easily find on the internet (*I recommend the*[***https://www.javadoc.io/doc/org.testng/testng/6.8.17/org/testng/Assert.html***](https://www.javadoc.io/doc/org.testng/testng/6.8.17/org/testng/Assert.html)*for the same*), but the ones above are the most common. TestNG asserts are very important for someone working on TestNG. They are the core of test case methods, and since a lot of the times we are using TestNG with selenium projects, they help us reduce the chances of error to a minimum. As far as I know, making efficient use of asserts in a test case is a sign of a perfect tester. So keep practicing TestNG asserts, and with this note, we will head on to our next tutorial.

**Commonly Questions on TestNG Asserts**

***How do I implement verify in TestNG?***  
*Verify is another name for soft asserts. Soft asserts can implement in the test code with the help of SoftAssert class in TestNG. Read this tutorial to know more*.

***What does "assert fail" mean in TestNG?***  
*Assert fail refers to the failure of the assertion test method. The conditions for failing depends totally on the assertion methods. When an assertion fails, they throw an exception error onto the console describing the failed test (only in hard asserts)*.

***Can we apply the assert statements inside the if-statements?***  
*Yes, assert statements are just like any other methods that return a boolean true or false value. A tester can leverage the return value and can use assert statements at any place in the code*.

References

https://www.toolsqa.com/testng/testng-asserts/