Asserts and Verify methods are commonly used in [Selenium](https://www.browserstack.com/selenium) for verifying or validating applications.

**Assertions (also known as Asserts)**

The word **Assert** means to state a fact or belief confidently or forcefully. In Selenium, Asserts are validations or checkpoints for an application. Assertions state confidently that application behavior is working as expected. One can say that Asserts in Selenium are used to validate the test cases. They help testers understand if tests have passed or failed.

**Types of Assertions**

* Hard Assertions
* Soft Assertions (Verify Method)

**Hard vs Soft Asserts in Selenium**

|  |  |
| --- | --- |
| **Hard Assertions** | **Soft Assertions** |
| Test Execution will be aborted if assert condition is not met | Test execution will continue till the end of the test case even if assert condition is not met |
| Does not have to invoke any methods to capture the assertions | To view assertions result at the end of the test, tester has to invoke assertAll() |

**Difference between Assert and Verify in selenium**

* In the case of assertions, if the assert condition is not met, test case execution will be aborted. The remaining tests are skipped, and the test case is marked as failed. These assertions are used as checkpoints for testing or validating business-critical transactions.
* In case of verify, tests will continue to run until the last test is executed even if assert conditions are not met. Verify or Soft Asserts will report the errors at the end of the test. Simply put, tests will not be aborted if any condition is not met. Testers need to invoke the assertAll() method to view the results.

Both Hard and Soft Assertions are very important for designing and running [Selenium webdriver](https://www.browserstack.com/guide/selenium-webdriver-tutorial) tests. They are instrumental in verifying application behavior at critical stages. By using assertions, testing teams can determine if an application is working as it is expected to. They can also save teams the trouble of running tests that don’t need to be run if a condition is not met.

**Hard Assertions**

Hard Assertions are ones in which test execution is aborted if the test does not meet the assertion condition. The test case is marked as failed. In case of an assertion error, it will throw the “***java.lang.AssertionError***” exception.

* **assertEquals()** is a method that takes a minimum of 2 arguments and compares actual results with expected results. If both match, then 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.

Below is an example of assertEquals().

**Code Snippet for assertEquals() in Selenium**

package com.tests;

import org.junit.Assert;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.Test;

public class BrowserStackTutorials {

@Test

public void testAssertFunctions() {

System.setProperty("webdriver.chrome.driver", "C:\\I2EWebsiteTest\\Driver\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

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

String ActualTitle = driver.getTitle();

String ExpectedTitle = "Most Reliable App & Cross Browser Testing Platform | BrowserStack";

Assert.assertEquals(ExpectedTitle, ActualTitle);

}

}

* **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. If actual and expected results are not the same, the test case is marked as passed.

**Code For assertNotEquals() in Selenium**

package com.tests;

import org.junit.Assert;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.Test;

public class BrowserStackTutorials {

@Test

public void testAssertFunctions() {

System.setProperty("webdriver.chrome.driver", "C:\\I2EWebsiteTest\\Driver\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

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

String ActualTitle = driver.getTitle();

String ExpectedTitle = "Most Reliable App & Cross Browser Testing Platform | browserstack";

Assert.assertNotEquals(ActualTitle, ExpectedTitle);

}

}

* **assertTrue()**: This Assertion verifies the Boolean value returned by the condition. If the Boolean value is true, then the assertion passes the test case.

**Code For assertTrue() in Selenium**

package com.tests;

import static org.testng.Assert.assertTrue;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.Test;

public class BrowserStackTutorials {

@Test

public void testAssertFunctions() {

System.setProperty("webdriver.chrome.driver", "C:\\I2EWebsiteTest\\Driver\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

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

Boolean verifyTitle = driver.getTitle().equalsIgnoreCase("Most Reliable App & Cross Browser Testing Platform | BrowserStack");

assertTrue(verifyTitle);

}

}

**Code For assertFalse() in Selenium**

package com.tests;

import static org.testng.Assert.assertFalse;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.Test;

public class BrowserStackTutorials {

@Test

public void testAssertFunctions() {

System.setProperty("webdriver.chrome.driver", "C:\\I2EWebsiteTest\\Driver\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

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

Boolean verifyTitle = driver.getTitle().equalsIgnoreCase("Most Reliable App & Cross Browser Testing Platform");

assertFalse(verifyTitle);

}

}

.

* **assertNull():** This method verifies if the expected output is null. If not, the value returned is false.

**Code Snippet For assertNull() in Selenium**

package com.tests;

import static org.testng.Assert.assertNull;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.Test;

public class BrowserStackTutorials {

@Test

public void testAssertFunctions() {

System.setProperty("webdriver.chrome.driver", "C:\\I2EWebsiteTest\\Driver\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

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

String verifyAssertNull = null;

assertNull(verifyAssertNull);

}

}

* **assertNotNull()**: This method works opposite to that of the assertNull() method. The assertion condition is met when the method validates the expected output to be not null.

**Code For assertNotNull() in Selenium**

package com.tests;

import static org.testng.Assert.assertNotNull;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.Test;

public class BrowserStackTutorials {

// Author: Chaitanya Pujari

@Test

public void testAssertFunctions() {

System.setProperty("webdriver.chrome.driver", "C:\\I2EWebsiteTest\\Driver\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

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

Boolean verifyTitle = driver.getTitle().equalsIgnoreCase("Most Reliable App & Cross Browser Testing Platform");

assertNotNull(verifyTitle);

}

}

**Example of Hard Assert in Selenium**

package com.tests;

import static org.testng.Assert.assertEquals;

import static org.testng.Assert.assertFalse;

import static org.testng.Assert.assertNotEquals;

import static org.testng.Assert.assertNotNull;

import static org.testng.Assert.assertNull;

import static org.testng.Assert.assertTrue;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.Test;

public class BrowserStackTutorials {

@Test

public void testAssertFunctions() {

System.setProperty("webdriver.chrome.driver", "C:\\I2EWebsiteTest\\Driver\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

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

String ActualTitle = driver.getTitle();

String verifyAssertNull=null;

String ExpectedTitle = "Most Reliable App & Cross Browser Testing Platform | BrowserStack";

Boolean verifyTitleIsPresent=driver.getTitle().equalsIgnoreCase("Most Reliable App & Cross Browser Testing Platform | BrowserStack");

Boolean verifyTitleIsChanged=driver.getTitle().equalsIgnoreCase("Testing Platform | BrowserStack");

assertEquals(ExpectedTitle, ActualTitle);

assertNotEquals(ExpectedTitle, "browserstack");

assertTrue(verifyTitleIsPresent);

assertFalse(verifyTitleIsChanged);

assertNotNull(verifyTitleIsPresent);

assertNull(verifyAssertNull);

}

}

**Verify in Selenium (also known as Soft Assertion)**

In a hard assertion, when the assertion fails, it terminates or aborts the test. If the tester does not want to terminate the script they cannot use hard assertions. To overcome this, one can use soft assertions.

Let’s explore the different types of soft assertions with examples (verify).

**Example of Soft Assert in Selenium (or Verify in Selenium)**

package com.tests;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.Test;

import org.testng.asserts.SoftAssert;

public class BrowserStackTutorials {

@Test

public void softAssert() {

System.setProperty("webdriver.chrome.driver", "C:\\I2EWebsiteTest\\Driver\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

SoftAssert softAssert = new SoftAssert();

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

String getActualTitle = driver.getTitle();

Boolean verifyTitle = driver.getTitle().equalsIgnoreCase("Most Reliable App & Cross Browser Testing Platform | BrowserStack");

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

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");

softAssert.assertAll();

}

}