**What is a TestNG XML File?**

A **TestNG XML file** is used to configure and execute test cases in a structured way. It allows users to define test suites, test cases, and their execution order.

**Why Use a TestNG XML File?**

* Run multiple test classes together
* Define groups and include/exclude tests
* Set test execution order
* Parameterize test execution

**Basic Structure of a TestNG XML File**

xml

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">

<suite name="MyTestSuite">

<test name="MyTest">

<classes>

<class name="com.example.TestClass1"/>

<class name="com.example.TestClass2"/>

</classes>

</test>

</suite>

**Explanation:**

* <suite>: Defines a test suite.
* <test>: Represents a test group.
* <classes>: Lists test classes to execute.
* <class>: Specifies test classes by package and name.

**Running Tests Using TestNG XML**

1. Right-click on the **testng.xml** file in your project.
2. Select **Run As → TestNG Suite** in your IDE.

### **TestNG Assertions – Importance, Hard vs. Soft Assertions**

#### ****What is an Assertion in TestNG?****

Assertions in TestNG are used to validate test case results. They help verify whether the actual output matches the expected output, ensuring the correctness of test execution.

#### ****Importance of Assertions****

✅ Ensures expected vs. actual results match  
✅ Helps identify failures in test cases  
✅ Improves automation script reliability  
✅ Avoids unnecessary test execution if a critical validation fails

## **Types of Assertions in TestNG**

### **1. Hard Assertions (org.testng.Assert)**

* Hard assertions **immediately stop** the test execution when they fail.
* They are useful for **critical validations** where execution should not proceed if a condition fails.

#### ****Example: Hard Assertion****

java

import org.testng.Assert;

import org.testng.annotations.Test;

public class HardAssertionExample {

@Test

public void testHardAssertion() {

int actual = 5;

int expected = 10;

Assert.assertEquals(actual, expected, "Test Failed: Values do not match!");

System.out.println("This line will not be executed if assertion fails.");

}

}

✅ If the assertion fails, the remaining test code **will not execute**.

### **2. Soft Assertions (org.testng.asserts.SoftAssert)**

* Soft assertions **do not stop** the test execution on failure.
* Even if an assertion fails, the remaining test steps continue executing.
* To check all assertions at the end, we call softAssert.assertAll().

#### ****Example: Soft Assertion****

java

import org.testng.asserts.SoftAssert;

import org.testng.annotations.Test;

public class SoftAssertionExample {

@Test

public void testSoftAssertion() {

SoftAssert softAssert = new SoftAssert();

int actual = 5;

int expected = 10;

softAssert.assertEquals(actual, expected, "First Check Failed!");

System.out.println("This line will be executed even if assertion fails.");

softAssert.assertTrue(false, "Second Check Failed!");

softAssert.assertAll(); // Reports all assertion failures at the end

}

}

✅ Soft assertions **continue execution** even if some checks fail.  
✅ The softAssert.assertAll() method reports all assertion failures at once.

### **Key Differences: Hard vs. Soft Assertions**

| **Feature** | **Hard Assertion (Assert)** | **Soft Assertion (SoftAssert)** |
| --- | --- | --- |
| Execution | Stops immediately on failure | Continues execution even after failures |
| Use Case | Critical validations | Multiple independent checks |
| assertAll() Required? | ❌ No | ✅ Yes (to report failures) |

### **When to Use Hard or Soft Assertions?**

✅ **Use Hard Assertions** when test failure should stop execution immediately.  
✅ **Use Soft Assertions** when multiple checks should be validated together.