



Core Annotations

Test Suite Setup/Teardown

<code>@BeforeSuite</code> e	Executed before all tests in the suite.
<code>@AfterSuite</code> e	Executed after all tests in the suite.
<code>@BeforeTest</code> t	Executed before all tests within a <code><test></code> tag.
<code>@AfterTest</code> t	Executed after all tests within a <code><test></code> tag.
<code>@BeforeGroup</code> ps	Executed before a specific group of tests.
<code>@AfterGroup</code> s	Executed after a specific group of tests.

Test Class Setup/Teardown

<code>@BeforeClass</code> ss	Executed before the first test method in the class.
<code>@AfterClass</code> s	Executed after all test methods in the class have run.
<code>@BeforeMethod</code> hod	Executed before each test method.
<code>@AfterMethod</code> od	Executed after each test method.

Test Method

<code>@Test</code>	Marks a method as a test method. Example: <code>@Test public void myTest() {}</code>
<code>@Test(priority = 1)</code>	Defines the execution order of tests within the class.
<code>@Test(enabled = false)</code>	Skips the test.
<code>@Test(timeOut = 1000)</code>	Fails the test if it exceeds 1000ms.
<code>@Test(expectedExceptions = {Exception.class})</code>	Passes the test if the specified exception is thrown.

Parameterization & Data Providers

Data Providers

<code>@DataProvider(name = "testData")</code>	Marks a method as a data provider. Example: <pre>@DataProvider(name = "testData") public Object[][] provideData() { return new Object[][] { { "data1" }, { "data2" } }; }</pre>
<code>@Test(dataProvider = "testData")</code>	Specifies the data provider for a test method. Example: <pre>@Test(dataProvider = "testData") public void testMethod(String data) { // Test logic }</pre>
Data providers are a powerful way to parameterize tests and run the same test method with different sets of data.	

Parameters in XML

<code><parameter name="paramName" value="paramValue"/></code>	Define parameters in the <code>testng.xml</code> file. Accessible using <code>@Parameters</code> annotation.
<code>@Parameters({"paramName"})</code>	Inject parameters defined in the <code>testng.xml</code> into the test method. Example: <pre>@Parameters({"paramName"}) @Test public void testMethod(String paramName) { // Test logic }</pre>

Advanced Features

Groups

<code>@Test(groups = {"group1", "group2"})</code>	Assigns a test method to one or more groups. Groups can be used for selective test execution.
<code><groups></code> in <code>testng.xml</code>	Define groups to include or exclude in <code>testng.xml</code> .
Example:	
<pre><groups> <run> <include name="group1"/> <exclude name="group2"/> </run> </groups></pre>	

Dependencies

<code>@Test(dependsOnMethods = {"method1", "method2"})</code>	Specifies that a test method depends on other methods. If the dependent methods fail, this method will be skipped.
<code>@Test(dependsOnGroups = {"group1"})</code>	Specifies that a test method depends on a group of methods. If any method in the group fails, this method will be skipped.

Listeners

<code>ITestListener</code>	Interface for listening to test execution events.
Common Methods:	
<code>onTestStart</code> , <code>onTestSuccess</code> , <code>onTestFailure</code> , <code>onTestSkipped</code>	
<code>@Listeners(MyListener.class)</code>	Specifies listeners to be used for test execution.

Assertions and XML Configuration

Assertions

TestNG provides built-in assertions for validating test results. These methods are typically used within <code>@Test</code> methods to verify expected outcomes.
Common Assertions:
<code>Assert.assertEquals(actual, expected)</code> : Checks that two values are equal.
<code>Assert.assertTrue(condition)</code> : Checks that a condition is true.
<code>Assert.assertFalse(condition)</code> : Checks that a condition is false.
<code>Assert.assertNull(object)</code> : Checks that an object is null.
<code>Assert.assertNotNull(object)</code> : Checks that an object is not null.
<code>Assert.fail(message)</code> : Fails the test with the given message.

testng.xml Configuration

The <code>testng.xml</code> file is the primary configuration file for TestNG. It defines the test suite, test classes, parameters, and other settings.
Key Elements:
<code><suite></code> : Root element that defines the test suite.
<code><test></code> : Defines a set of test classes to be executed.
<code><classes></code> : Contains a list of <code><class></code> elements, specifying the test classes to be included in the test.
<code><parameter></code> : Defines parameters that can be passed to test methods.
<code><listeners></code> : Specifies listeners to be used during test execution.
Example structure of <code>testng.xml</code> :

```
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="MyTestSuite">
  <test name="MyTest">
    <classes>
      <class name="com.example.MyTestClass"/>
    </classes>
  </test>
</suite>
```