* **@Factory**: Used to execute a set of test classes.
* @Factory
* public Object[] factoryMethod() {
* return new Object[] { new TestClass1(), new TestClass2() };
* }
* **@Listeners**: Defines listeners on a test class.
* @Listeners(MyListener.class)
* public class TestClass {
* // test methods
* }
* **@Parameters**: Passes parameters to test methods.
* @Parameters({ "param1" })
* @Test
* public void testMethod(String param1) {
* System.out.println("Parameter: " + param1);
* }

**⚙️ TestNG Attributes**

Attributes modify the behavior of test methods:

* **priority**: Defines the order of test execution. Lower values have higher priority.
* @Test(priority = 1)
* public void testMethod() {
* // test code
* }
* **enabled**: If set to false, the test method is skipped.
* @Test(enabled = false)
* public void testMethod() {
* // this test will be skipped
* }
* **invocationCount**: Specifies the number of times a test method should be invoked.
* @Test(invocationCount = 5)
* public void testMethod() {
* // this test will run 5 times
* }
* **invocationTimeOut**: Specifies the maximum time allowed for all invocations.
* @Test(invocationCount = 5, invocationTimeOut = 1000)
* public void testMethod() {
* // this test will run 5 times with a total timeout of 1000ms
* }
* **timeOut**: Specifies the maximum time allowed for a single test method execution.
* @Test(timeOut = 1000)
* public void testMethod() {
* // this test must finish within 1000ms
* }
* **dependsOnMethods**: Specifies the methods that the test method depends on.
* @Test
* public void loginTest() {
* // login test code
* }
* @Test(dependsOnMethods = { "loginTest" })
* public void dashboardTest() {
* // dashboard test code
* }
* **dependsOnGroups**: Specifies the groups that the test method depends on.
* @Test(groups = "init")
* public void setupTest() {
* // setup code
* }
* @Test(dependsOnGroups = { "init" })
* public void mainTest() {
* // main test code
* }
* **groups**: Assigns the test method to one or more groups.
* @Test(groups = { "smoke", "regression" })
* public void testMethod() {
* // test code
* }
* **description**: Provides a description for the test method.
* @Test(description = "This test verifies the login functionality.")
* public void loginTest() {
* // test code
* }
* **alwaysRun**: If set to true, the test method will always run even if it depends on a method that failed.
* @Test(dependsOnMethods = { "initTest" }, alwaysRun = true)
* public void cleanupTest() {
* // cleanup code
* }

**🗂️ testng.xml Configuration**

The testng.xml file is used to configure test suites:

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

<suite name="MyTestSuite">

<test name="LoginTests">

<classes>

<class name="com.example.tests.LoginTest"/>

</classes>

</test>

</suite>

* **suite**: Defines a test suite.([LambdaTest](https://www.lambdatest.com/blog/asserts-in-testng/?utm_source=chatgpt.com))
* **test**: Defines a test.
* **classes**: Specifies the classes to be included in the test.
* **class**: Specifies a single class to be included.
* **methods**: Specifies methods to include or exclude.
* <methods>
* <include name="testMethod1"/>
* <exclude name="testMethod2"/>
* </methods>
* **groups**: Specifies groups to include or exclude.
* <groups>
* <run>
* <include name="smoke"/>
* <exclude name="regression"/>
* </run>
* </groups>
* **packages**: Specifies packages to include.
* <packages>
* <package name="com.example.tests"/>
* </packages>