TestNG Cheatsheet

A quick reference guide for TestNG annotations, features, and best practices for Java testing.

Core Annotations

Test Suite Setup/Teardown

`@BeforeSuite`	Executed before all tests in the suite.
`@AfterSuite`	Executed after all tests in the suite.
`@BeforeTest`	Executed before all tests within a `<test>` tag.
`@AfterTest`	Executed after all tests within a `<test>` tag.
`@BeforeGroups`	Executed before a specific group of tests.
`@AfterGroups`	Executed after a specific group of tests.

Test Class Setup/Teardown

`@BeforeClass`	Executed before the first test method in the class.
`@AfterClass`	Executed after all test methods in the class have run.
`@BeforeMethod`	Executed before each test method.
`@AfterMethod`	Executed after each test method.

Test Method

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

Parameterization & Data Providers

Data Providers

`@DataProvider(name = "testData")`	Marks a method as a data provider. **Example:** `@DataProvider(name = "testData") public Object[][] provideData() { return new Object[][] { { "data1" }, { "data2" } }; }`
`@Test(dataProvider = "testData")`	Specifies the data provider for a test method. **Example:** `@Test(dataProvider = "testData") public void testMethod(String data) { // Test logic }`

Data providers are a powerful way to parameterize tests and run the same test method with different sets of data.

Parameters in XML

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

Advanced Features

Groups

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

Dependencies

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

Listeners

`ITestListener`	Interface for listening to test execution events. **Common Methods:** `onTestStart`, `onTestSuccess`, `onTestFailure`, `onTestSkipped`
`@Listeners(MyListener.class)`	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 `@Test` methods to verify expected outcomes.

Common Assertions:

`Assert.assertEquals(actual, expected)`: Checks that two values are equal.
`Assert.assertTrue(condition)`: Checks that a condition is true.
`Assert.assertFalse(condition)`: Checks that a condition is false.
`Assert.assertNull(object)`: Checks that an object is null.
`Assert.assertNotNull(object)`: Checks that an object is not null.
`Assert.fail(message)`: Fails the test with the given message.

testng.xml Configuration

The `testng.xml` file is the primary configuration file for TestNG. It defines the test suite, test classes, parameters, and other settings.

Key Elements:

`<suite>`: Root element that defines the test suite.
`<test>`: Defines a set of test classes to be executed.
`<classes>`: Contains a list of `<class>` elements, specifying the test classes to be included in the test.
`<parameter>`: Defines parameters that can be passed to test methods.
`<listeners>`: Specifies listeners to be used during test execution.

Example structure of `testng.xml`:

```
<!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>
```