**Selenium Class**

# Chapter#21 – TestNG Annotations

1. @Test
2. @BeforeSuite
3. @AfterSuite
4. @BeforeClass
5. @AfterClass
6. @BeforeTest
7. @AfterTest
8. @BeforeMethod
9. @AfterMethod
10. @BeforeGroups
11. @AfterGroups
12. @DataProvider
13. @Parameter
14. Attributes

## @Test

The @Test is the most important and commonly used annotation of TestNG. It is used to mark a method as Test. So, any method over which we see @Test annotation is considered as a TestNG test.

**@Test**

public void sampleTest() {

//Any test logic

System.out.println("Hi! Welcome!");

}

## @BeforeSuite

The annotated method will run only once before all tests in this suite have run.

## @AfterSuite

The annotated method will run only once after all tests in this suite have run.

## @BeforeClass

The annotated method will run only once before the first test method in the current class is invoked.

## @AfterClass

The annotated method will run only once after all the test methods in the current class have been run.

## @BeforeTest

The annotated method will run before any test method belonging to the classes inside the <test> tag is run.

## @AfterTest

The annotated method will run after all the test methods belonging to the classes inside the <test> tag have run.

## @BeforeMethod

The annotated method will be run before each test method.

## @AfterMethod

The annotated method will be run after each test method.

## @BeforeGroups

The list of groups that this configuration method will run before. This method is guaranteed to run shortly before the first test method that belongs to any of these groups is invoked.

## @AfterGroups

The list of groups that this configuration method will run after. This method is guaranteed to run shortly after the last test method that belongs to any of these groups is invoked.

## @DataProvider

Using @DataProvider we can create a data driven framework in which data is passed to the associated test method and multiple iteration of the test runs for the different test data values passed from the @DataProvider method. The method annotated with @DataProvider annotation return a 2D array or object.

//Data provider returning 2D array of 3\*2 matrix

@DataProvider(name = "dataProvider1")

public Object[][] dataProviderMethod1() {

return new Object[][] {{"selenium ","java"}, {"k1","r1"},{"k2","r2"}};

}

//This method is bound to the above data provider

//The test case will run 3 times with different set of values

@Test(dataProvider = "dataProvider1")

public void sampleTest(String str1, String str2) {

System.out.println(str1 + " " + str2);

}

## @Parameter

The @Parameter tag is used to pass parameters to the test scripts. The value of the @Parameter tag can be passed through testng.xml file.

Sample testng.xml with parameter tag-

@Test (timeOut = 500)<suite name="sampleTestSuite">

<test name="sampleTest">

<parameter name="sampleParamName" value="sampleParamValue"/>

<classes>

<class name="TestFile" />

</classes>

</test>

</suite>

Sample Test Script with @Parameter annotation-

public class TestFile {

@Test

@Parameters("sampleParamName")

public void parameterTest(String paramValue) {

System.out.println("sampleParamName = " + sampleParamName);

}

## Attributes

**1. description** – The ‘description’ attribute is used to provide a description to the test method. It generally contains a one-liner test summary.

**@Test(description = "Test summary")**

**2.** **dataProvider** – This attribute helps in creating a data driven tests. It is used to specify the name of the data provider for the test.

**@Test(dataProvider = "name of dataProvider")**

**3. priority** – This attribute helps in prioritizing the test methods. The default priority starts with 0 and tests execute in ascending order.

**@Test(priority = 2)**

**4. enabled** – This attribute is used to specify whether the given test method will run with the suite or class or not.

**@Test(enabled = false)**

**5. groups** – Used to specify the groups, the test method belongs to.

**@Test(groups = { "sanity", "regression" })**

**6. dependsOnMethods** – Used to specify the methods on which the test method depends. The test method only runs after successful execution of the dependent tests.

**@Test(dependsOnMethods = { "dependentTestMethodName" })**

**7. dependsOnGroups** – Used to specify the groups on which the test method depends.

**@Test(dependsOnGroups = { "dependentGroup" })**

**9. alwaysRun** – When set as True, the test method runs even if the dependent methods fail.

**@Test(alwaysRun=True)**

**10. timeOut** – This is used to specify a timeout value for the test(in milli seconds). If test takes more than the timeout value specified, the test terminates and is marked as failure.

**@Test (timeOut = 500)**