* BeforeSuite : This will execute before the Test Suite where as test suit is set of test cases. We define suit in testing.xml file
* BeforeTest : This will execute before the Test
* BeforeClass : This will execute before the Class
* BeforeMethod :This will execute before **every** Method
* Test Case 1 : This is the Test Case 1
* AfterMethod : This will execute after **every** Method
* AfterClass : This will execute after the Class
* AfterTest : This will execute after the Test
* AfterSuite :This will execute after the Test Suite

**NOTE** : If you create 2 different packages, in one project in a framework (like we have tests and tests2) still we can set inter dependent testing annotations like : priorities, depends on group etc. It works well in two different packages of same framework.

Below are the lists of attributes that we can pass to our Test method:

* **alwaysRun** : This is used when we want to make sure a method always runs even if the parameters on which the method depends, fails. If set to true, this test method will always run. Eg: @Test(alwaysRun = true)
* **dataProvider**: TestNG dataProvider is used to provide any data for parameterization. Eg. @Test(dataProvider = “Hello”).
* **dataProviderClass**: This is the class from where we pass the data to data provider. In our case dataProvider class name is “Hello”.
* **dependsOnGroups**: It is the list of groups this method depends on. Eg: @Test (groups = { “City” ,”State” }). @Test(groups="test1", dependsOnGroups={"test2"}, alwaysRun=**true**)
* **dependsOnMethods**: This command is used to execute a method based on its dependent method. Eg: @Test (dependsOnMethods = { “OpenBrowser” ,”database is up” })
* **description**: It is the description for the method. Eg: @Test(description = “test method”)

**eg**:@BeforeClass(**description** = "Set capabilities for your Firefox browser and set time it should wait for a page to load.")

* **invocationCount**: It refers to the number of times a method should be invoked. It will work as a loop. Eg: @Test(invocationCount = 7) . Hence, this method will execute 7 times.
* **invocationTimeOut**: This refers to the maximum number of milliseconds a method should take for all the invocationCount to complete. This attribute will be ignored if invocationCount is not specified. Eg: @Test(invocationCount =7,invocationTimeOut = 30 )
* **priority**: This command sets the priority of the test method. Lower priorities will be scheduled first. Eg: @Test(priority =1 )
* **\*\*\* IMP \*\*\* > if there is group of 3 TCs, no one has a priority then testing will alphabetically execute the 3 test cases**

|  |  |
| --- | --- |
| **Sr.No.** | **Annotation & Description** |
| 1 | **@BeforeSuite**  The annotated method will be run only once before all tests in this suite have run. |
| 2 | **@AfterSuite**  The annotated method will be run only once after all tests in this suite have run. |
| 3 | **@BeforeClass**  The annotated method will be run only once before the first test method in the current class is invoked. |
| 4 | **@AfterClass**  The annotated method will be run only once after all the test methods in the current class have run. |
| 5 | **@BeforeTest**  The annotated method will be run before any test method belonging to the classes inside the <test> tag is run. |
| 6 | **@AfterTest**  The annotated method will be run after all the test methods belonging to the classes inside the <test> tag have run. |
| 7 | **@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. |
| 8 | **@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. |
| 9 | **@BeforeMethod**  The annotated method will be run before each test method. |
| 10 | **@AfterMethod**  The annotated method will be run after each test method. |
| 11 | **@DataProvider**  Marks a method as supplying data for a test method. The annotated method must return an Object[ ][ ], where each Object[ ] can be assigned the parameter list of the test method. The @Test method that wants to receive data from this DataProvider needs to use a dataProvider name equals to the name of this annotation. |
| 12 | **@Factory**  Marks a method as a factory that returns objects that will be used by TestNG as Test classes. The method must return Object[ ]. |
| 13 | **@Listeners**  Defines listeners on a test class. |
| 14 | **@Parameters**  Describes how to pass parameters to a @Test method. |
| 15 | **@Test**  Marks a class or a method as a part of the test. |

VIMP : SEQUENCE : suit > test > class > method

<suite name=*"Suite"*> // b4 this > b4suit executes

<test hread-count=*"5"*name=*"Test"*> //b4 this > b4 test executes

<classes> // b4 this > b4 class executes

<class name=*"SelTest.abc"*/> (abc is class name and it has 2 methods. So before methods and after methods will be executed for each method.)

<class name=*"SelTest.testngPractice"*/> (*testngPractice* is class name and it has 2 methods. So before methods and after methods will be executed for each method.)

</classes> //After Class.

</test><!-- Test --> //After test.

</suite><!-- Suite --> //After suite.

**IMP** annotations :

1. Enabled : if = true : TC will be executed and if = false then TC won’t be executed.

Ex. @Test(enabled = **false**)

**Public void** testcase2()

{

System.***out***.println("In testcase 2");

}

1. To run specific **group** from XML : if group name is ‘Regression’ and we want to run the regression test cases from XML file then the XML file should be :

<suite name=*"Suite"*>

<groups>

<run>

<include name=*"regression"*/>

</run>

</groups>

<test hread-count=*"5"*name=*"Test"*>

<classes>

<!--<class name="SelTest.abc" /> -->

<classname=*"SelTest.testngPractice"*>

</class>

</classes>

</test><!-- Test -->

</suite><!-- Suite -->

Then the test cases with the group : test cases will be executed :

@Test(enabled = **true**, groups={"regression"}) // this will be executed

**publicvoid** testcase1()

{

System.***out***.println("In testcase 1 : regression");

}

@Test(enabled = **true**, groups={"smoke"}) // this will be **excluded**

**publicvoid** testcase2()

{

System.***out***.println("In testcase 2");

}

@BeforeSuite(groups={"regression", "smoke"})// this will be executed

**publicvoid**beforeSuit()

{

System.***out***.println("before suite..");

}

@AfterSuite(groups={"regression","smoke"})// this will be executed

**publicvoid**afterSuit()

{

System.***out***.println("After suite..");

}

NOTE : IMP : @Test(enabled = **true**, groups={"regression"}) // this will be executed

**Public void** testcase1() { … .. } this will be executed here and the other methods such as BeforeSuit and AfterMethod etc will **ONLY** be executed if all of them **have** groups={"regression"})

1. Include : in XML we can specify whether to include / exclude test cases from execution

Ex. :<suite name=*"Suite"*>

<groups>

<run>

<include name=*"regression"*/>

<exclude name=*"smoke"*/>

</run>

</groups>

<test thread-count=*"5"*name=*"Test"*>

<classes>

<!--<class name="SelTest.abc" /> -->

<class name=*"SelTest.testngPractice"*>

</class>

</classes>

</test><!-- Test -->

</suite><!-- Suite -->

Here if a method such as : @AfterSuite(groups={"regression","smoke"})

**publicvoid**afterSuit()

{

System.***out***.println("After suite..");

} will be **excluded** because it is a part of “regression” as well as “smoke”. Only methods with “regression” will be executed but if we remove the <exclude>tag then all with “regression” as well as “smoke” will be executed.

1. Parallel execution : We have set parameter parallel = class, so it executes the classes parallel.

Ex. :

<?xml version=*"1.0"*encoding=*"UTF-8"*?>

<!DOCTYPEsuiteSYSTEM"http://testng.org/testng-1.0.dtd">

<suite name=*"Suite"* parallel=*"classes"*> //the two classes will be parallaly executed.

<!--<groups><run><include name="regression" /><exclude name="smoke"

/></run></groups> -->

<test thread-count=*"5"*name=*"Test"*>

<classes>

<class name=*"SelTest.abc"*/>

<class name=*"SelTest.testngPractice"*>

</class>

</classes>

</test><!-- Test -->

</suite><!-- Suite -->

Op :

before suite..

before test.

before Class. // started parallel execution

before Class. // started parallel execution

before Method.

in abc test3

In testcase 1 : regression

After Class.

After Method.

before Method.

.

.

…

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WE WILL SEE ATTRIBUTES AT TIME OF FRAMEWORK \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*