**Groups**

* Grouping tests
* Running tests in a group
* Tests belonging to multiple groups
* Including/excluding groups
* Using regular expressions
* Default group
* Group of groups

Example of Tests defined

**package** pack3;

**import** org.testng.annotations.Test;

**public** **class** TestGroup {

**// Test belongs to multiple groups**

@Test(groups = { "Group1", "Group2" })

**public** **void** LoginTest() {

System.***out***.println("LoginTest");

}

@Test(groups = { "Group2" })

**public** **void** SendEmailTest() {

System.***out***.println("SendEmailTest");

}

@Test(groups = { "Group1" })

**public** **void** SendTextMessageTest() {

System.***out***.println("SendTextMessageTest");

}

}

Below is corresponding testng.xml

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

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

<suite name=*"Suite"* time-out=*"500"*>

<test name=*"Test"*>

<groups>

<run>

<include name=*"Group2"*></include>

</run>

</groups>

<classes>

<class name=*"pack3.TestGroup"*></class>

</classes>

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

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

The preceding XML file contains only one test inside a suite. This contains the groups section defined by using the groups tag as shown in the code. The run tag represents the group that needs to be run. The include tag represents the name of the group that needs to be executed.

Below is the output when the tesestng.xml is run

[RemoteTestNG] detected TestNG version 6.14.2

LoginTest

SendEmailTest

===============================================

Suite

Total tests run: 2, Failures: 0, Skips: 0

===============================================

Note:

If a test method belongs to both included and excluded group, the excluded group takes the priority and the test method will be excluded from the test execution.

**Regular Expression**

To use regular expressions to include and exclude groups you have to use .\* for matching names. We can also use it for searching groups that contains a certain string in their names by using the expression at start and end of the search string (for example, .\*name.\*).

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

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

<suite name=*"Suite"* time-out=*"500"*>

<test name=*"Test"*>

<groups>

<run>

<include name=*"Group.\*"*></include>

</run>

</groups>

<classes>

<class name=*"pack3.TestGroup"*></class>

</classes>

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

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

**Default group**

Sometimes we may need to assign a default group to a set of test methods that belong to a class. This way all the public methods that belong to the said class will automatically become TestNG test methods and become part of the said group.

This can be achieved by using the @Test annotation at class level and defining the default group in the said @Test annotation.

**package** pack3;

**import** org.testng.annotations.Test;

@Test(groups = { "default-group" })

**public** **class** DefaultGroup {

**public** **void** testMethodOne() {

System.***out***.println("Test method one.");

}

**public** **void** testMethodTwo() {

System.***out***.println("Test method two.");

}

@Test(groups = { "test-group" })

**private** **void** testMethodThree() {

System.***out***.println("Test method three.");

}

@Test(groups = { "test-group" })

**protected** **void** testMethodFour() {

System.***out***.println("Test method four.");

}

@Test(groups = { "test-group" })

**void** testMethodFive() {

System.***out***.println("Test method five.");

}

}

The preceding class contains five methods that print a message onto console when executed. All of the methods are considered as test methods by the use of the @Test annotation on the class. All of the methods belong to the group default-group by mentioning the group name at the class level. Three test methods also belong to the group test-group, this is done by using the @Test annotation at the method level.

**Corresponding testng.xml**

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

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

<suite name=*"Suite"* time-out=*"500"*>

<test name=*"Test"*>

<groups>

<run>

<include name=*"default-group"*></include>

</run>

</groups>

<classes>

<class name=*"pack3.DefaultGroup"*></class>

</classes>

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

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

**Below is the output**

[RemoteTestNG] detected TestNG version 6.14.2

Test method five.

Test method four.

Test method one.

Test method three.

Test method two.

===============================================

Suite

Total tests run: 5, Failures: 0, Skips: 0

===============================================

**Note: It does not matter if a method in** DefaultGroup **class is private, public or protected. All will be executed by testng.xml**

**Group of groups**

TestNG allows users to create groups out of existing groups and then use them during the creation of the test suite. You can create new groups by including and excluding certain groups and then use them.

Let's create a sample test program and learn how to create group of groups called MetaGroups.

**package** pack3;

**import** org.testng.annotations.Test;

@Test(groups = { "default-group" })

**public** **class** GroupofGroup {

@Test(groups = { "TelephonyMxOne", "RadioESN" }, priority = 1)

**public** **void** testMethodOne() {

System.***out***.println("Test method one.");

}

@Test(groups = { "TelephonyAvaya" }, priority = 2)

**public** **void** testMethodTwo() {

System.***out***.println("Test method two.");

}

@Test(groups = { "test-group", "RadioESN" }, priority = 3)

**private** **void** testMethodThree() {

System.***out***.println("Test method three.");

}

@Test(groups = { "test-group", "TelephonyAvaya" }, priority = 4)

**protected** **void** testMethodFour() {

System.***out***.println("Test method four.");

}

@Test(groups = { "test-group", "RadioAirwave" }, priority = 5)

**void** testMethodFive() {

System.***out***.println("Test method five.");

}

}

Corresponding testng.xml

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

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

<suite name=*"Suite"*>

<test name=*"Test1"* parallel=*"tests"*>

<groups>

<define name=*"Radio\_Group"*>

<include name=*"RadioAirwave"*></include>

<include name=*"RadioESN"*></include>

</define>

<define name=*"Telephony\_Group"*>

<include name=*"TelephonyAvaya"*></include>

<include name=*"TelephonyMxone"*></include>

</define>

<run>

<include name=*"Radio\_Group"*></include>

</run>

</groups>

<classes>

<class name=*"pack3.GroupofGroup"*>

</class>

</classes>

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

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