**TestNG Methods, Groups, dependsOnMethods, dependsOnGroups, alwaysRun, include**

**Include, Exclude**

**Scenario:** Suppose we have 3 @Test Method in a class and want to run 2 method and skip one. We can control this easily in 2 ways. One is i.e: **@Test(enabled = false/true)** here if mention “false” it will skip the method and if use “true” will include/run the method.

Another way is we can mention Include and Exclude methods’ name inside the **“.XML”** file and can run the test from there.

Code for **“.XML”** file for the above scenario:

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

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

<suite name="Suite">

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

<classes>

**// For class name need to write the code class name with the package name by .**

<class name="root.Testcase1">

<methods> // Here which method name are mentioned inside the “Include” tag, those will be executed and which are inside the “Exclude” tag, will be skiped.

<include name="testcase1"></include>

<include name="testcase2"></include>

<exclude name="testcase3"></exclude>

</methods>

</class>

</classes>

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

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

**N.B:** When customize in “.XML” file, have to run the code from the “.XML” file as well.

**Advantage:** If we use this feature, we will not have to comment or delete code from the main class file which is much easier and professional.

**dependsOnMethods**

**Scenario:** Suppose we have 3 test method and method 3 is dependable on method 2 and method 2 depends on method 1. In these cases we can use the “dependsOnMethods” title after the @Test annotation. i.e: **@Test(dependsOnMethods = "testcase1")**

**Code:**

@Test

**public** **void** testcase1() {

String title = driver.getTitle();

System.***out***.println("TestCase 1 runs: " + title);

Assert.*assertEquals*(title, "Selenium");

}

**// This Method depends on "testcase1"**

@Test(dependsOnMethods = "testcase1")

**public** **void** testcase2() {

String title2 = driver.getTitle();

System.***out***.println("TestCase 2 runs: " + title2);

}

**// This Method depends on "testcase2"**

@Test(dependsOnMethods = "testcase2")

**public** **void** testcase3() {

driver.findElement(By.*linkText*("Downloads")).click();

String text = driver.findElement(By.*xpath*("//body/div[1]/main[1]/div[10]/div[1]/div[1]")).getText();

System.***out***.println("Copy text are: " + text);

}

**Here run these codes from the Class file**

**groups**

**Scenario:** Suppose in our project we have “Smoke, Functional and Regression” test those we can mention as group name.

The syntax is **@Test(groups = {"Smoke", "Functional", "Regression"})**

**Class Code:**

@Test(groups = {"Smoke", "Functional", "Regression"})

**public** **void** testcase1() {

String title = driver.getTitle();

System.***out***.println("TestCase 1 runs: " + title);

Assert.*assertEquals*(title, "Selenium");

}

@Test(groups = {"Functional", "Regression"})

**public** **void** testcase2() {

String title2 = driver.getTitle();

System.***out***.println("TestCase 2 runs: " + title2);

}

@Test(groups = "Regression")

**public** **void** testcase3() {

driver.findElement(By.*linkText*("Downloads")).click();

String text = driver.findElement(By.*xpath*("//body/div[1]/main[1]/div[10]/div[1]/div[1]")).getText();

System.***out***.println("Copy text are: " + text);

}

**We have to run this code from the “.XML” file:**

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

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

<suite name="Suite">

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

<groups>

<run>

**// Here we are running those groups which has “Smoke” & “Functional”**

<include name="Smoke"></include>

<include name="Functional"></include>

</run>

</groups>

<classes>

<class name="root.Testcase1"></class>

</classes>

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

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

**This way we can Include and Exclude the group of code.**

**alwaysRun**

**Scenario:** If we need to allow any methods to run always, have to use “**@Test(alwaysRun = true”** like we need it for the abobe “groups” code.

@BeforeClass(alwaysRun = **true**)

**public** **void** beforeClass() {

driver = **new** ChromeDriver();

driver.get("https://www.selenium.dev/");

driver.manage().window().maximize();

driver.manage().deleteAllCookies();

}

**dependsOnGroups**

**Scenario:** We can also prioritize method’s execution base on their group name similar to the **“dependsOnMethods”**

**Code For Class:**

@Test(groups = "Smoke")

**public** **void** testcase1() {

String title = driver.getTitle();

System.***out***.println("TestCase 1 runs: " + title);

Assert.*assertEquals*(title, "Selenium");

}

@Test(dependsOnGroups = "Smoke", groups = "Functional")

**public** **void** testcase2() {

String title2 = driver.getTitle();

System.***out***.println("TestCase 2 runs: " + title2);

}

@Test(dependsOnGroups = {"Smoke", "Functional"}, groups = "Regression")

**public** **void** testcase3() {

driver.findElement(By.*linkText*("Downloads")).click();

String text = driver.findElement(By.*xpath*("//body/div[1]/main[1]/div[10]/div[1]/div[1]")).getText();

System.***out***.println("Copy text are: " + text);

}

**Code For “.XML”:**

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

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

<suite name="Suite">

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

<groups>

<run>

<include name="Functional"></include>

</run>

</groups>

<classes>

<class name="root.Testcase1"></class>

</classes>

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

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