## Multiple Tests

There will be situations when you want to put number of tests under a single test class and like to run all in single shot.  With the help of TestNG ‘**@Test**‘ annotations we can execute multiple tests in single TestNG file.

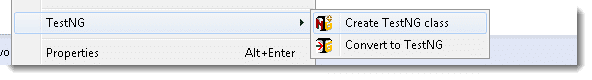
Take an example of four different tests under one testng class and print the test sequence on the console.

**How to do it…**

1) Press **Ctrl+N** , select “**TestNG Class**” under **TestNG**category and click **Next**.

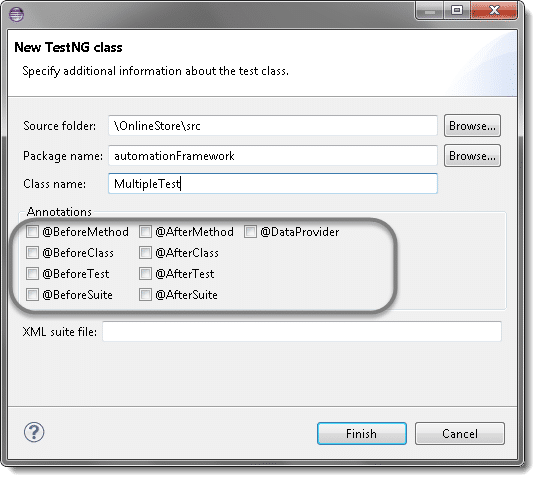
**Or**

Right click on Test Case folder, go to **TestNG**and select “**TestNG Class**“.



2) If your project is set up and you have selected the Test Case folder before creating TestNG class then the source folder and the package name will be pre-populated on the form. Set class name as ‘**TestNG**‘.

Leave rest of the settings untouched, do not check for “**@BeforeMethod**”, “**@AfterMethod**” for now and click **Finish**. That’s it.



3) By default a new class will have only one @Test method. Add two more methods by yourself and put your code accordingly in methods. Code will look like:

package automationFramework;

import org.openqa.selenium.WebDriver;

import org.testng.annotations.Test;

public class MultipleTest {

public WebDriver driver;

@Test

public void One() {

System.out.println("This is the Test Case number One");

}

@Test

public void Two() {

System.out.println("This is the Test Case number Two");

}

@Test

public void Three() {

System.out.println("This is the Test Case number Three");

}

@Test

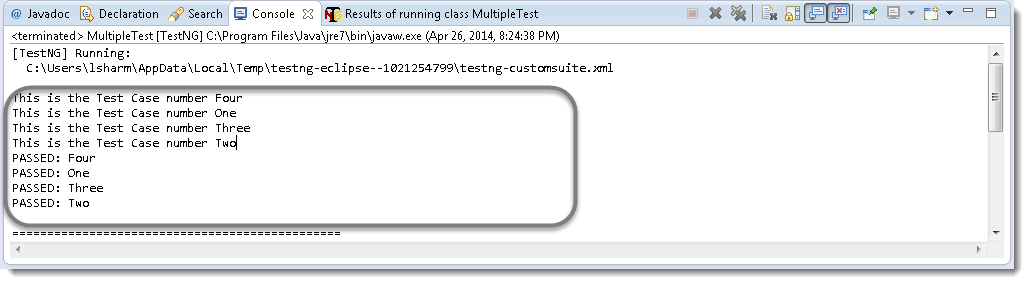
public void Four() {

System.out.println("This is the Test Case number Four");

}

}

This will enable you to execute all four tests with just one testng class. Take a look on the output.

**Attention:**By default, methods annotated by @Test are executed alphabetically. Take a look over the next topic to see how to prioritize @Test.

## Sequencing & Prioritizing

You need to use the ‘**priority**‘ parameter, if you want the methods to be executed in your order. **Parameters** are keywords that modify the annotation’s function.

Let’s take the same above example and execute all @Test methods in right order. Simply assign priority to all @Test methods starting from 0(Zero).

package automationFramework;

import org.openqa.selenium.WebDriver;

import org.testng.annotations.Test;

public class MultipleTest {

public WebDriver driver;

@Test(priority = 0)

public void One() {

System.out.println("This is the Test Case number One");

}

@Test(priority = 1)

public void Two() {

System.out.println("This is the Test Case number Two");

}

@Test(priority = 2)

public void Three() {

System.out.println("This is the Test Case number Three");

}

@Test(priority = 3)

public void Four() {

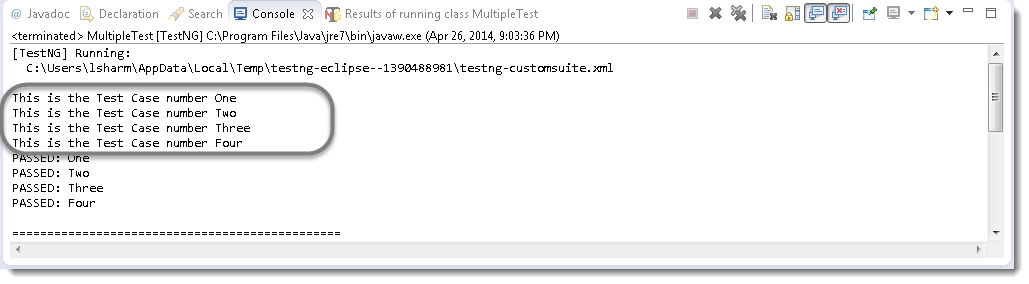
System.out.println("This is the Test Case number Four");

}

}

**Note:**TestNG will execute the @Test annotation with the lowest priority value up to the largest.

Output of the above:



## 

## Skipping a Test Case

Think of a situation where you are required to skip one or more @Test from your testng class. In testng, you can easily able to handle this situation by setting the ‘enabled’ parameter to ‘false’ for e.g.:

@Test(enabled = false)

To use two or more parameters in a single annotation, separate them with a comma:

@Test(priority = 3, enabled = false)

Again take the same example and set the value false for the third test.

package automationFramework;

import org.openqa.selenium.WebDriver;

import org.testng.annotations.Test;

public class MultipleTest {

public WebDriver driver;

@Test(priority = 0)

public void One() {

System.out.println("This is the Test Case number One");

}

@Test(priority = 1)

public void Two() {

System.out.println("This is the Test Case number Two");

}

@Test(priority = 2, enabled = false)

public void Three() {

System.out.println("This is the Test Case number Three");

}

@Test(priority = 3)

public void Four() {

System.out.println("This is the Test Case number Four");

}

}

Output of the above example:

