**TestNG** – NG stand for Next generation

Java – Programming Language

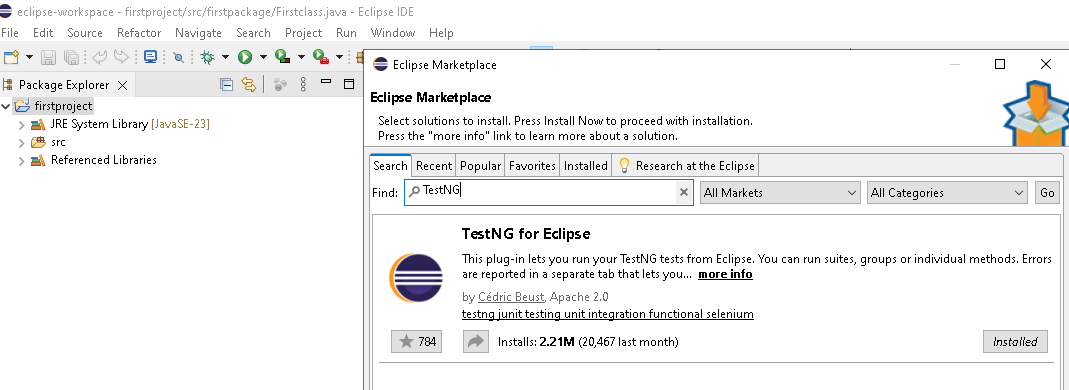
Selenium – web application automation framework

TestNG -> Testing Framework used for unit, functional, integration, end to end & so on, it have inbuilt progress for testing result reporting, execution , test data (its like cognizant craft framework)

Java + Selenium + testing ->combined used for real time business challenges

**Hw to install TestNG**

Open elipse - Go to help - Ellipse Marketplace – search Test NG & click install



**Test annotation(@Test)**

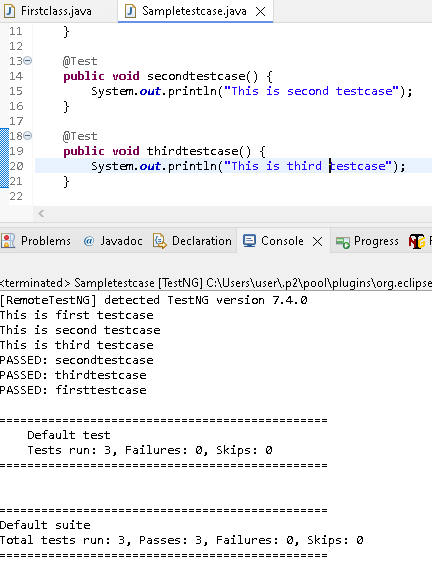
**Write normal method in elipse without main method “public** **static** **void** main(String[] args)” **& add @test, then method will execute thru testing framework and result will have pass , fail, skips**

**public** **static** **void** main(String[] args) if present – it will run as java application only, not thru testing

@ - annotation

@test – consider each method as test case

Test NG execute Method order as per method alphabets name only

****

**package** testNG;

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

**public** **class** Sampletestcase {

@Test

**public** **void** firsttestcase() {

System.***out***.println("This is first testcase");

}

@Test

**public** **void** secondtestcase() {

System.***out***.println("This is second testcase");

}

@Test

**public** **void** thirdtestcase() {

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

}

**priority parameter in @Test Annotation –** execute method/TC based on the priority number

@test – consider each method as test case

Test NG execute Method order as per method alphabets name only

User want to execute the method order by using priority parameter

@Test (priority = 3) - >it will execute third

**public** **void** firsttestcase() {

System.***out***.println("This is first testcase");

}

@Test (priority = 1) - >it will execute first

**public** **void** secondtestcase() {

System.***out***.println("This is second testcase");

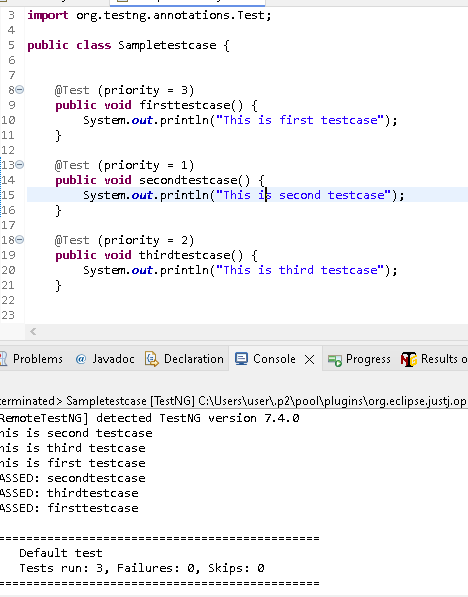
}

@Test (priority = 2) - >it will execute second

**public** **void** thirdtestcase() {

System.***out***.println("This is third testcase");

}



**Enabled parameter in @Test Annotation** – if we give Enabled parameter as false then method wont excute/skipped

@Test (priority = 3)

**public** **void** firsttestcase() {

System.***out***.println("This is first testcase");

}

@Test (priority = 1)

**public** **void** secondtestcase() {

System.***out***.println("This is second testcase");

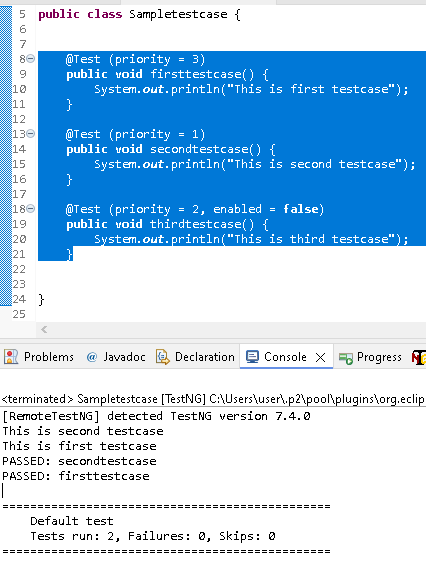
}

@Test (priority = 2, enabled = **false**) ->This method wont execute/skipped

**public** **void** thirdtestcase() {

System.***out***.println("This is third testcase");

}



**dependsOnMethods parameter in @Test Annotation** – Particular Method/TC will execute based on the dependsOnMethods executed.

In below eg , all method wont execute because secondtestcase method depend on firsttestcase method & firsttestcase we skipped. Similarly thirdtestcase method depend on secondtestcase method

@Test (enabled = **false**)

**public** **void** firsttestcase() {

System.***out***.println("This is first testcase");

}

@Test (dependsOnMethods = "firsttestcase")

**public** **void** secondtestcase() {

System.***out***.println("This is second testcase");

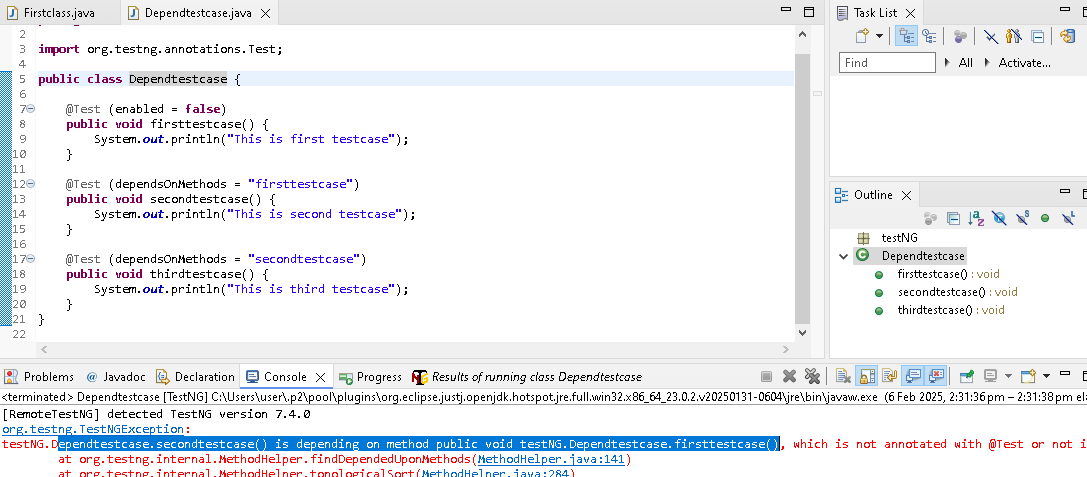
}

@Test (dependsOnMethods = "secondtestcase")

**public** **void** thirdtestcase() {

System.***out***.println("This is third testcase");

}



Another eg ,

Note: Here priority not assigned, based on dependsOnMethods parameter each method will execute in the order

@Test ()

**public** **void** firsttestcase() {

System.***out***.println("This is first testcase");

}

@Test (dependsOnMethods = "firsttestcase")

**public** **void** secondtestcase() {

System.***out***.println("This is second testcase");

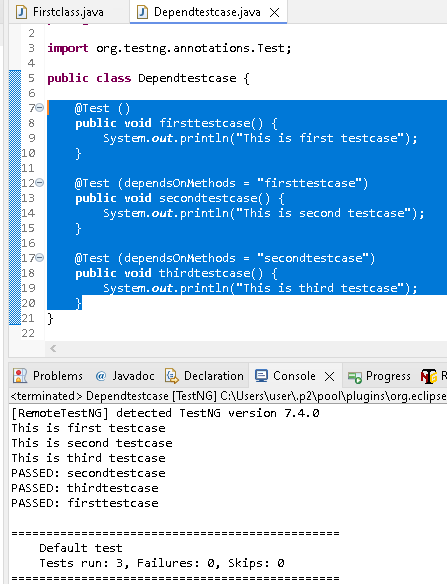
}

@Test (dependsOnMethods = "secondtestcase")

**public** **void** thirdtestcase() {

System.***out***.println("This is third testcase");

}



---------------------------------------------------------------------------------------------------------------

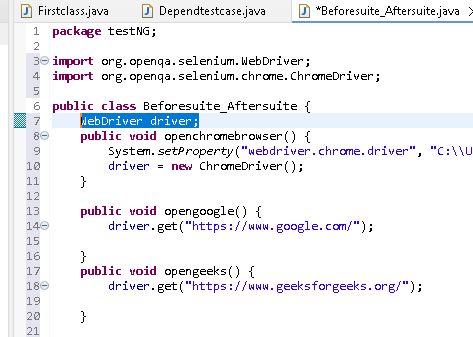
**@Beforesuite & Aftersuite in TestNG framework**

If we define any variables inside method, then its a local variables to that particular method in a class . And other methods in the same class can’t use those variables

If we define any variables in a class level, then all methods in the same class can use those variables

Eg below: driver variable





@BeforeSuite – It will execute first , then only Method/testcases will execute one by one.

Opening browser will be common for all method, so kept under BeforeSuite

@AfterSuite – It will execute at the last in the TestNG

Closing browser kept under after Suite

**public** **class** Beforesuite\_Aftersuite {

WebDriver driver;

@BeforeSuite

**public** **void** openchromebrowser() {

System.*setProperty*("webdriver.chrome.driver", "C:\\Users\\user\\Downloads\\chromedriver.exe");

driver = **new** ChromeDriver();

}

@Test()

**public** **void** opengoogle() {

driver.get("https://www.google.com/");

}

@Test()

**public** **void** opengeeks() {

driver.get("https://www.geeksforgeeks.org/");

}

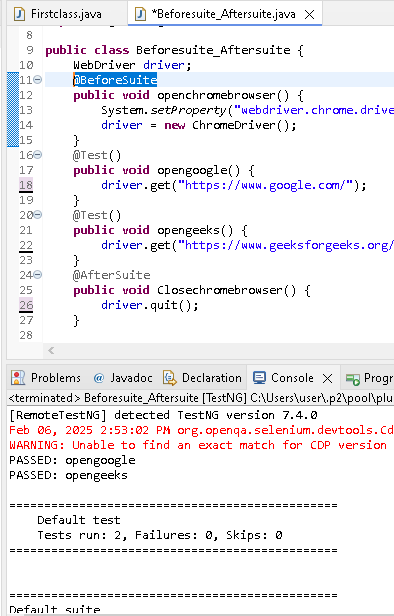
@AfterSuite

**public** **void** Closechromebrowser() {

driver.quit();

}

}



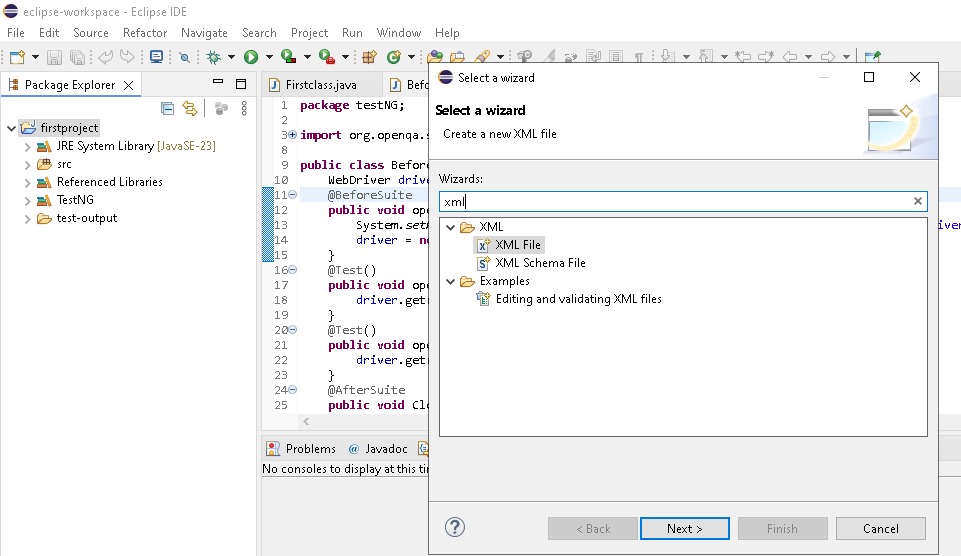
**How to write testsuite in testing with TestNG.xml**

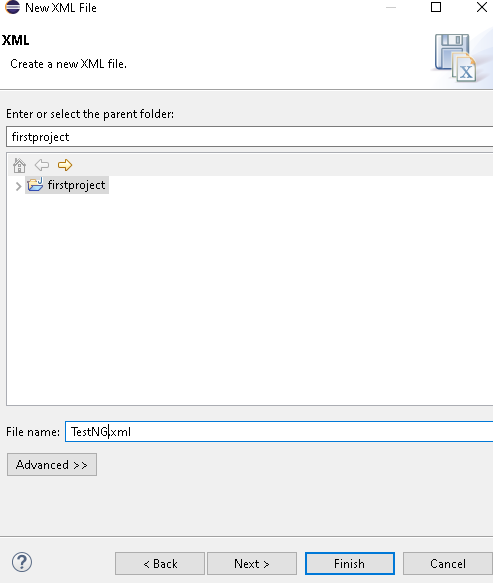
Using TestNG.xml we can execute all class and one go..in real time TestNG project,each class consider as 1 test cases.

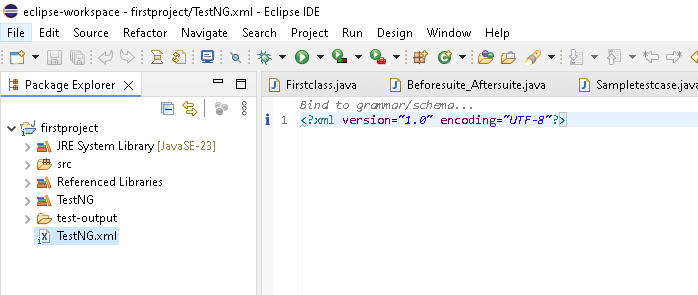
Create TestNG.xml by below step

Rightclick the project, then New -> project -> others

In below window type XML & proceed







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

<suite name=*"test Suite"*> any name for suite

<test name=*"test Cases"*> any name for test

<classes>

<class name = *"testNG.Sampletestcase"*> </class> ->Package.Classname

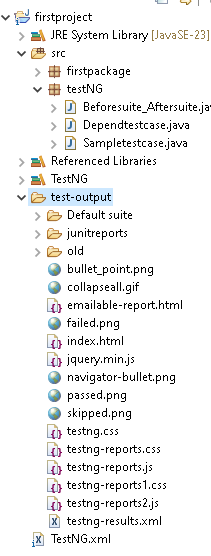
<class name = *"testNG.Dependtestcase"*> </class>

</classes>

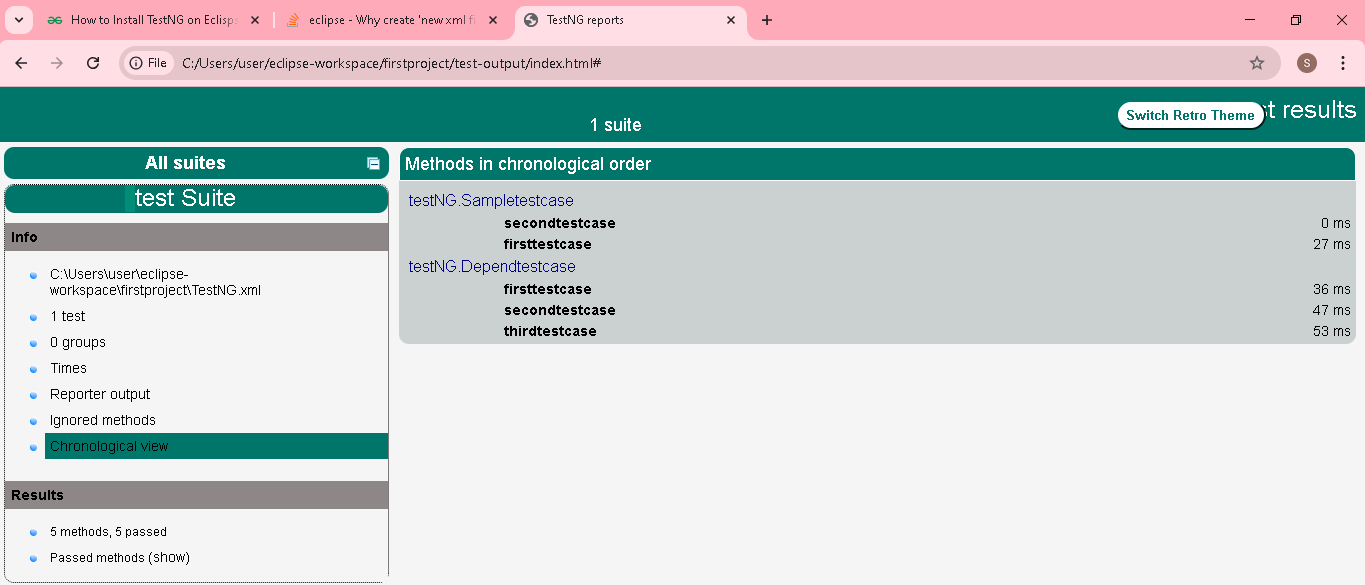
</test>

</suite>

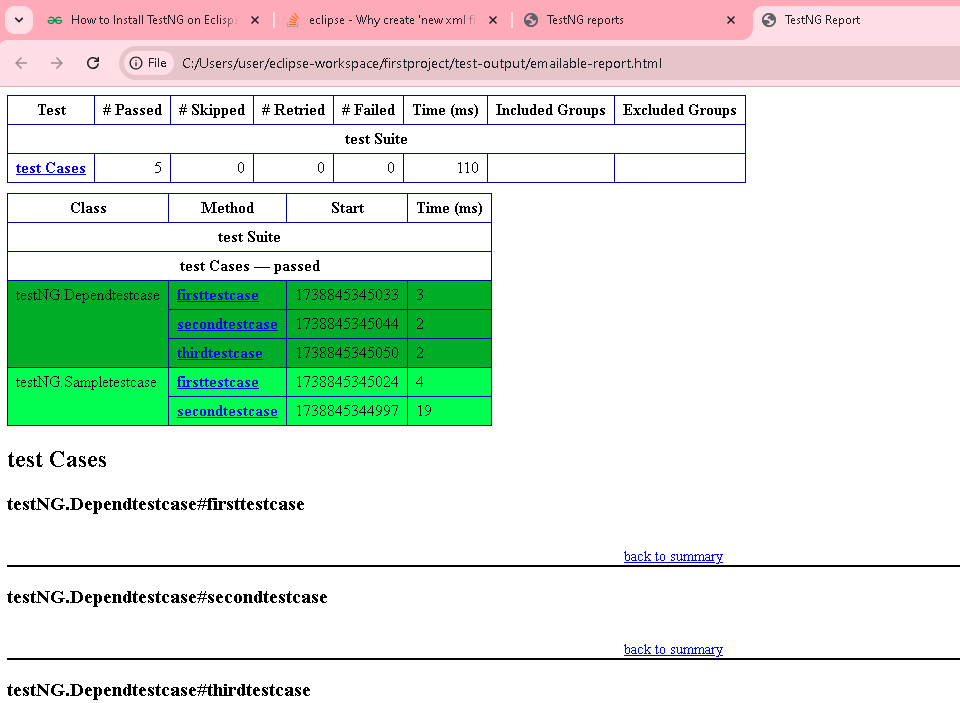
Rightclick the project & refresh then below folder(“test.output”) will create & contain reporting in simplified format(Pass, fail) & html detailed format



Right click the index.html & open with webbrowser & looks like below



Right click the emailble-report.html & open with webbrowser & looks like below



**Group our testcases in TestNG**

In Test annotation, we need to use the groups parameter. Need to assign the group name for each method/TC in classes.

Once done, in .xml files we can call those specific method/TC using the groups name

@Test (groups = {"Apple"})

**public** **void** apple1() {

System.***out***.println("Apple1 Testing");

}

@Test (groups = {"Apple"})

**public** **void** apple2() {

System.***out***.println("Apple2 Testing");

}

@Test (groups = {"Moto"})

**public** **void** Moto1() {

System.***out***.println("Moto1 Testing");

}

@Test (groups = {"Moto"})

**public** **void** Moto2() {

System.***out***.println("Moto2 Testing");

}

@Test (groups = {"Vivo"})

**public** **void** Vivo1() {

System.***out***.println("Vivo1 Testing");

}

@Test (groups = {"Vivo"})

**public** **void** Vivo2() {

System.***out***.println("Vivo2 Testing");

}

It will executed only apple & moto methods as per include names in groups

<groups>

<run>

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

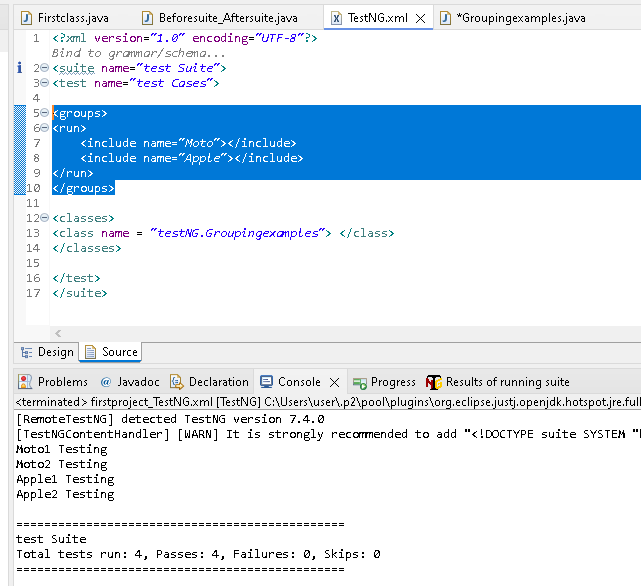
<include name=*"Apple"*></include>

</run>

</groups>

We can use exclude also, it will exclude methods in a class. Eg some time we need to execute all method except one..in that we can this.

<Exclude name=*"Apple"*></include>



**Parameterization in TestNg using @Parameters Annotation**

Parameterization – Passing the test data from TestNG.xml to methods.

In Testng.xml , we need to use parameter tag with name & values. As below

<parameter name=*"Username"* value=*"Suku"*></parameter>

In methods we need to use @Parameters Annotation to receive the values from xml

In methods need to add string variable(String name) in order to receive the value

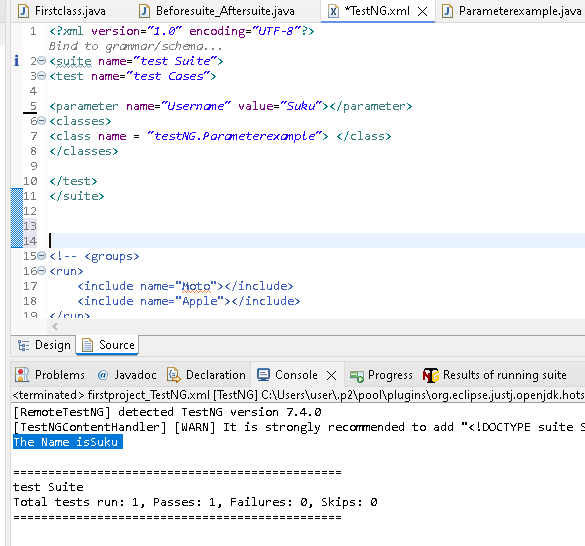
@Test

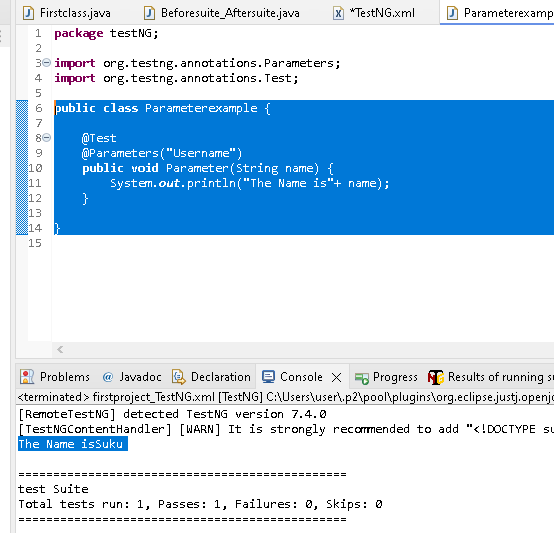
@Parameters("Username")

**public** **void** Parameter(String name) {

System.***out***.println("The Name is"+ name);

}





**Parallel testing in TestnG**

It means we can execute multiple test cases/methods at same time in different browser, no need to wait the first test case need to complete & followed by next one

Just add parallel =*"methods"* thread-count=*"2” – it means execute 2 diff methods in 2 browser parallels/same time*

<suite name=*"test Suite"* parallel=*"methods"* thread-count=*"2"*>

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

<suite name=*"test Suite"* parallel=*"methods"* thread-count=*"2"*>

<test name=*"test Cases"* >

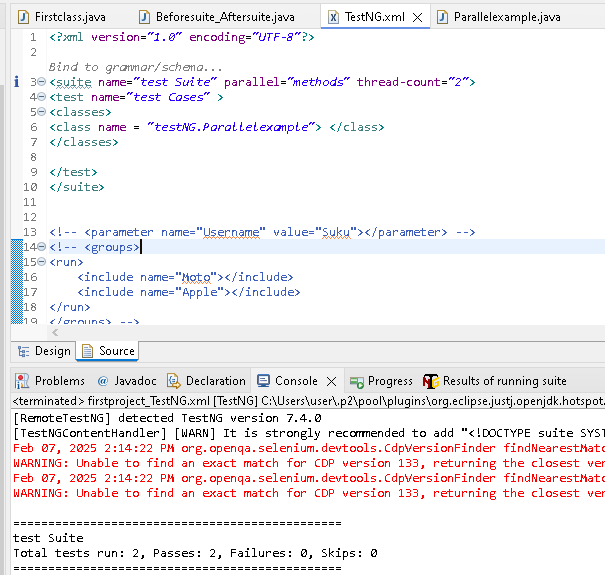
<classes>

<class name = *"testNG.Parallelexample"*> </class>

</classes>

</test>

</suite>





**Assertions in testing**

It used for developer who perform unit testing using testng.

What is unit testing? After product developed, developer who developed the product will perform testing to ensure product are working(high level) is called unit testing.

With assert, we can easily test features (compare 2 text, number, objects, Boolean atc) with simple 1 liner. If we use if statement need to write 4 line of code

*assertEquals, assertnotEquals, asserttrue etc*

Assert.*assertEquals*(Actual, expected);

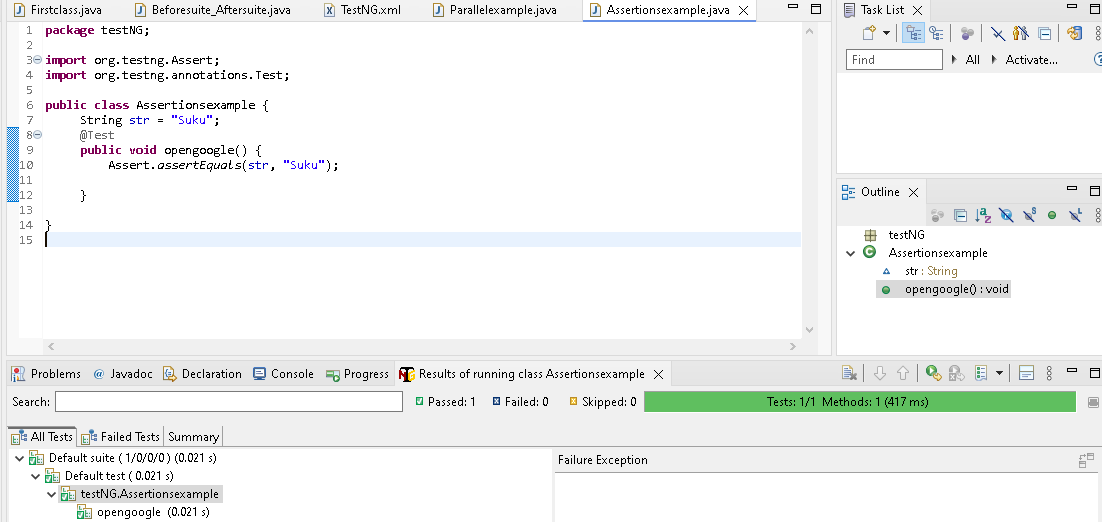
String str = "Suku";

@Test

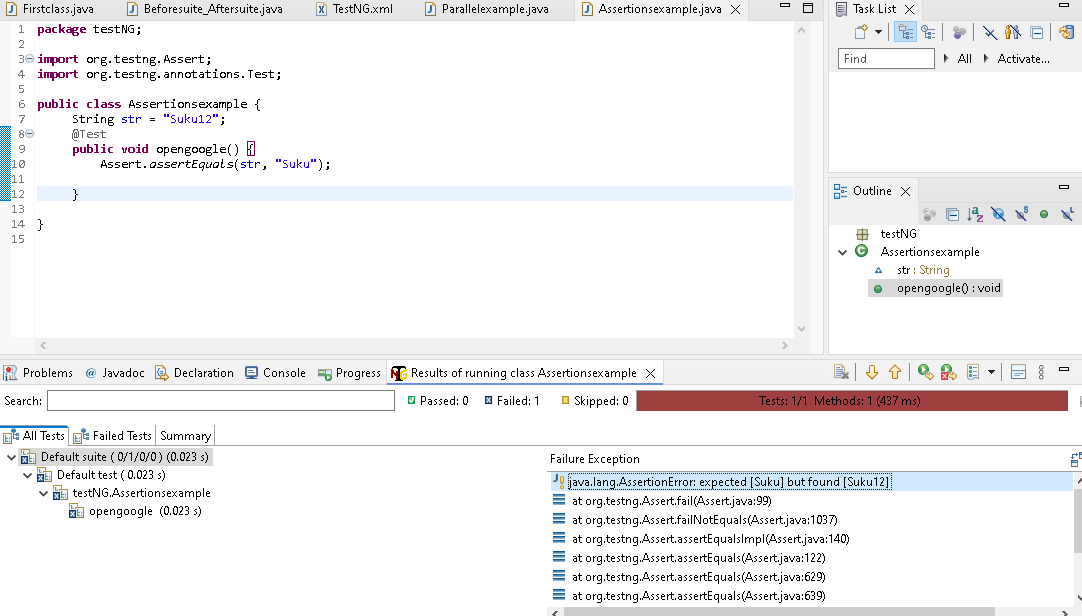
**public** **void** opengoogle() {

Assert.*assertEquals*(str, "Suku");

}

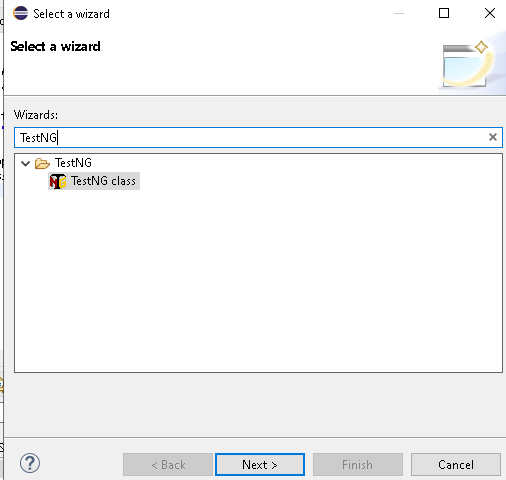


Passing String str = "Suku12"; validate failed result

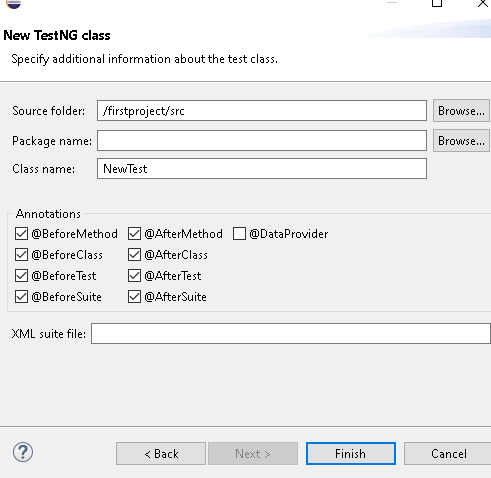


**All Annotation order of execution in TestNG**

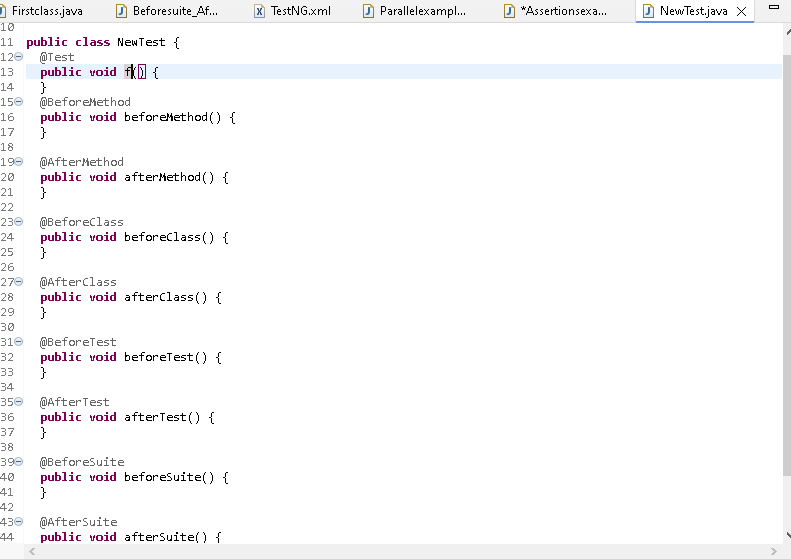
Create a testNG class, rightclick the project ->New ->Others ->Type testNg in a wizard



Refer the below diagram for list of annotations in testng



Select all annotation except Dataprovider Click finish



By default the @Test method name as F, Changed the f name to Test & execute to see the order of execution.

import org.testng.annotations.Test;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeSuite;

import org.testng.annotations.AfterSuite;

public class NewTest {

@Test

public void Test() {

System.out.println("Test");

}

@BeforeMethod

public void beforeMethod() {

System.out.println("beforeMethod");

}

@AfterMethod

public void afterMethod() {

System.out.println("afterMethod");

}

@BeforeClass

public void beforeClass() {

System.out.println("beforeClass");

}

@AfterClass

public void afterClass() {

System.out.println("afterClass");

}

@BeforeTest

public void beforeTest() {

System.out.println("beforeTest");

}

@AfterTest

public void afterTest() {

System.out.println("afterTest");

}

@BeforeSuite

public void beforeSuite() {

System.out.println("beforeSuite");

}

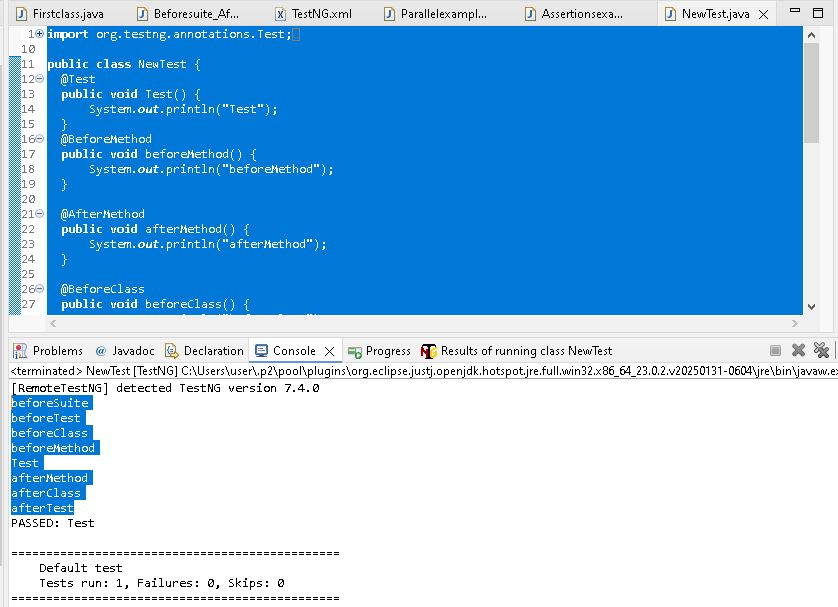
@AfterSuite

public void afterSuite() {

System.out.println("afterSuite");

}

}



Order of execution in testNG class be: (During interview they ask this kind of question)

beforeSuite

beforeTest

beforeClass

beforeMethod

Test

afterMethod

afterClass

afterTest

Add Test1 method with @Test Annotation & execute

beforeSuite

beforeTest

beforeClass

beforeMethod

Test

afterMethod

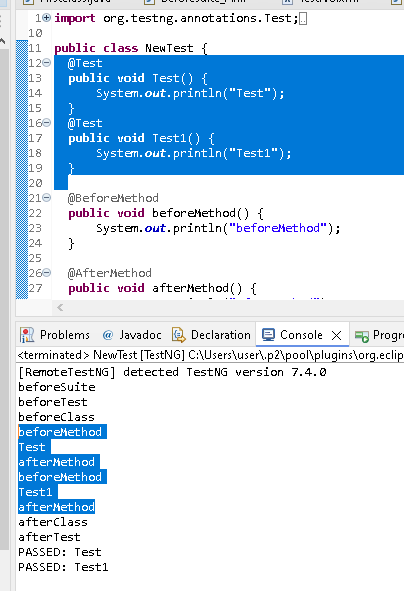
beforeMethod

Test1

afterMethod

afterClass

afterTest



**Listerners in testNG** ->It will listen to testsuite, class, method & what need to do base on the reult

Different lieterners -> TestListerner, classListerner, suiteListerner

By implementing the Listerners concept to , we can tell what i need to do if the testcase passed,

we can tell what i need to do if the testcase failed, eg(take screenshot)

**Just watched the youtube video**...