**TEST NG:**

It is an open source automated testing framework inspired from ***JUnit with*** some new functionality that make it more powerful and easier to use.

**INSTALLATION OF TESTNG:**

**STEP 1:** Launch the Eclipse IDE and from Help menu, click “**Install New**

**Software**”

**STEP 2:** You will see a dialog window, click “**Add**” button.

**STEP 3:** Type name as you wish, lets take “**TestNG**” and type

“**http://beust.com/eclipse/**” as location. Click OK

**STEP 4:** You come back to the previous window but this time you must see TestNG

option in the available software list. Just Click TestNG and press “**Next**”

button.

**STEP 5:** Click “**I accept the terms of the license agreement**” then click **Finish**.

**STEP 6:** You may or may not encounter a Security warning, if in case you do just

click **"OK".**

**STEP 7:** Click **Next** again on the succeeding dialog box until it prompts you to

Restart the Eclipse so click **"YES"** to restart the eclipse.

**STEP 8:** After restart, verify if TestNG was indeed successfully installed. Right click

on you project and see if **TestNG**is displayed in the opened menu.

( **OR )**

**STEP 1 :** Click ECLIPSE MARKET PLACE in from Help menu.

**STEP 2:** Search TestNG by using find option.

**STEP 3:** Click install to install the TESTNG.

**STEP 3:** You may or may not encounter a Security warning, if in case you do just

click **"OK".**

**BENEFITS OF TESTNG:**

* It gives the ability to produce HTML REPORTS of execution
* ***Annotations***made testers life easy
* Test cases can be ***Grouped & Prioritized*** more easily
* ***Parallel***testing is possible
* Generates ***Logs(reports)***
* Data ***Parameterization***is possible

**ANNOTATIONS :**

* It identifies the methods it is interested in by looking up annotations. Hence

method names are not restricted to any pattern or format

* We can pass additional parameters to annotations.
* Annotations are strongly typed, so the compiler will flag any mistakes right away.
* Test classes no longer need to extend anything (such as Test Case, for JUnit 3).

**TYPES OF ANNOTATIONS :**

* @BeforeSuite
* @BeforeTest
* @Beforeclass
* @Beforemethod
* @Test
* @AfterMethod
* @AfterClass
* @AfterTest
* @AfterSuite
* @DataProvider
* @Parameters
* @BeforeGroups
* @AfterGroups

**1.@BeforeSuite:** This method will be run before all tests in this suite have run.

**2. @BeforeTest:** This method will be run before all tests in this suite have run.

**3.@BeforeClass:** This method will be run before the first test method.

**4.@BeforeMethod:** This method will be run before each test method.

**5.@Test:** This method is a part of a test case.

**6.@AfterMethod:** This method will be run after each test method.

**7.@AfterClass:** This method will be run after the first test method.

**8.AfterTest:** This method will be run after all tests in this suite have run.

**9.AfterSuite:** This method will be run after all tests in this suite have run.

**10.DataProvider:** This method supplying data for a test method and 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 data

provider needs to use a dataProvider name equals to the name of this

annotation**.(eg: @Test(dataProvider = "datas")**

**11.@parameters:** This method describes how to pass parameters to a @Test

method.

**12.@BeforeGroups:** This method will run before the first test method that belongs

to any of this groups is invoked.

**13.@AfterGroups:** This method will run after the last test method that belongs to any of

this groups is invoked.

**SAMPLE PROGRAM FOR TESTNG USING ANNOTATIONS:**

**public** **class** Testng {

@Test(enabled=**false**)//enabled function is used to disable this method

**public** **void** m1()

{

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

}

@BeforeTest

**public** **void** m2()

{

System.***out***.println("This is my before test method");

}

@AfterTest

**public** **void** m3()

{

System.***out***.println("This is my after test method");

}

@BeforeClass

**public** **void** m4()

{

System.***out***.println("This is my before class method");

}

@AfterClass

**public** **void** m5()

{

System.***out***.println("This is my after class method");

}

@Test(priority=1) //priority is used to run the test case

based on priority

**public** **void** m6()

{

System.***out***.println("this is my second program");

}

@BeforeSuite

**public** **void** m7()

{

System.***out***.println("this is my before suite");

}

@AfterSuite

**public** **void** m8()

{

System.***out***.println("this is my after suite");

}

@BeforeMethod

**public** **void** m9()

{

System.***out***.println("this is my bfr method");

}

@AfterMethod

**public** **void** m10()

{

System.***out***.println("this is my afr method");

}

@Test(priority=2)

**public** **void** M11()

{

System.***out***.println("this is my third prgrm");

}

}

**OUTPUT:**

[RemoteTestNG] detected TestNG version 6.9.9

[TestNG] Running:

C:\Users\dell\AppData\Local\Temp\testng-eclipse--1610774242\testng-customsuite.xml

this is my before suite

This is my before test method

This is my before class method

this is my bfr method

this is my second program

this is my afr method

this is my bfr method

this is my third prgrm

this is my afr method

This is my after class method

This is my after test method

PASSED: m6

PASSED: M11

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

Default test

Tests run: 2, Failures: 0, Skips: 0

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

this is my after suite

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

Default suite

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

**SAMPLE PROGRAM FOR DATAPROVIDER:**

**public** **class** demotstng {

@Test(dataProvider= "datas")

**public** **void** login(String fname,String lname,String mstatus,String phn,String name,String email )

{

WebDriver dm=**new** FirefoxDriver();

dm.get("http://demoqa.com/registration/");

dm.findElement(By.*id*("name\_3\_firstname")).sendKeys(fname);

dm.findElement(By.*id*("name\_3\_lastname")).sendKeys(lname);

List<WebElement> radios=dm.findElements(By.*xpath*(".//input[@type='radio']"));

**for**(WebElement s:radios)

{

String val=s.getAttribute("value");

**if**(val.equalsIgnoreCase("single"))

**break**;

}

dm.findElement(By.*id*("phone\_9")).sendKeys("phn");

dm.findElement(By.*id*("username")).sendKeys("name");

dm.findElement(By.*id*("email\_1")).sendKeys("email");

}

@DataProvider(name="datas")

**public** Object[][] data()

{

**return** **new** Object[][] {{"vdg","s","single","9002673462","sjhg","viogu@hsfy"},

{"jdgu","d","married","9829628259","dutd","vysvgf@gm.com"}}

**TESTNG PARAMETERS:**

* Test parameters allows us to automatically run a test case multiple times with different input and validation values.
* TestNG pass parameters directly to test methods with a help of testng.xml.

**HOW TO CREATE TESTNG.XML??**

**STEP 1:** Right click the class name.

**STEP 2:** Click TestNG->Convert to TestNG

**STEP 3:** Click Finish

**SAMPLE PROGRAM FOR PASSING PARAMETERS TO TESTNG :**

**TESTNG PROGRAM:**

**public** **class** Newtours {

WebDriver driver;

@BeforeTest

**public** **void** setup()

{

driver=**new** FirefoxDriver();

driver.get("http://newtours.demoaut.com/");

}

@Test

@Parameters({"user","pass"})

**public** **void** login(String Uname,String pwd)

{

driver.findElement(By.*name*("userName")).sendKeys(Uname);

driver.findElement(By.*name*("password")).sendKeys(pwd);

driver.findElement(By.*name*("login")).click();

}

@Test

**public** **void** logout()

{

driver.findElement(By.*linkText*("SIGN-OFF"));

}

}

TestNG.xml:

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

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

<suite name=*"Suite"*>

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

<parameter name=*"user"* value=*"123"*/>

<parameter name=*"pass"* value=*"123"*/>

<classes>

<class name=*"chrome.Newtours"*/> // packagename.classname.

</classes>

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

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

**TESTCASE GROUPING:**

‘**Groups**‘ is one more annotation of TestNG which can be used in the execution of multiple tests.

EXAMPLE:

Let’s say you have hundred tests of class vehicle and in it ten method of car, ten method of scooter and so on. You probably like to run all the scooter tests together in a batch. And you want all to be in a single test suite. With the help of grouping you can easily overcome this situation.

We can group the methods by using the syntax  (groups = { ” Group Name” })

**SAMPLE PROGRAM:**

**public** **class** Newtours {

WebDriver driver;

@BeforeTest(groups= {"lo"})

**public** **void** setup()

{

driver=**new** FirefoxDriver();

driver.get("http://newtours.demoaut.com/");

}

@Test(groups= {"lo"})

@Parameters({"user","pass"})

**public** **void** login(String Uname,String pwd)

{

driver.findElement(By.*name*("userName")).sendKeys(Uname);

driver.findElement(By.*name*("password")).sendKeys(pwd);

driver.findElement(By.*name*("login")).click();

}

@Test

**public** **void** logout()

{

driver.findElement(By.*linkText*("SIGN-OFF"));

}

//taking screen shot

@AfterMethod

**public** **void** takescreenshot(ITestResult result) **throws** IOException

{

**if**(result.isSuccess()) {

File ss=((TakesScreenshot)driver).getScreenshotAs(OutputType.***FILE***);

FileUtils.*copyFile*(ss,**new** File("c:\\Test\\"+ result.getName()+".png"));

}

}

@AfterTest

**public** **void** teardown()

{

driver.close();

}

}

**TESTNG.XML:**

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

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

<suite name=*"Suite"*>

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

<parameter name=*"user"* value=*"123"*/>

<parameter name=*"pass"* value=*"123"*/>

<groups>

<run>

<include name=*"lo"*/>

</run>

</groups>

<classes>

<class name=*"chrome.Newtours"*/>

</classes>

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

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

**NOTE:**

If more than one group, we have to add like the following example to testng xml,

<group>

<define name="All">

<include name="car" />

<include name="scooty" />

</define>

<run>

<include name ="All"/>

</run></groups>

**OUTPUT:**

[RemoteTestNG] detected TestNG version 6.9.9

[TestNG] Running:

C:\Users\dell\eclipse-workspace\chrome\testng.xml

log4j:WARN No appenders could be found for logger (org.apache.http.client.protocol.RequestAddCookies).

log4j:WARN Please initialize the log4j system properly.

log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.

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

Suite

Total tests run: 1, Failures: 0, Skips: 0

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

**DEPENDENT TEST:**

It is to share some data and state between method or we can invoke methods in a Test case in a particular order.

TestNG allows you to specify dependencies either with:

* Using attributes *dependsOnMethods* in @Test annotations OR
* Using attributes *dependsOnGroups* in @Test annotations.