1 : **What Is the usage of testng.xml file?**  
  
**Answer** : In selenium WebDriver software testing tool, We are using testng.xml file to configure our whole test suite In single file. Few of the tasks which we can specify In testng.xml file are as bellow.

* We can define software testing test suite using set of test cases to run them from single place.
* Can Include or exclude test methods from software web application's test execution.
* Can specify a group to Include or exclude.
* Can pass parameter to use In test case of software web application.
* Can specify group dependencies.
* Can configure parallel test execution for software web application.
* Can define listeners.

**2** : **How to pass parameter with testng.xml file to use It In test case?**  
  
**Answer** : We can define parameter In testng.xml file using syntax like bellow.

<parameter name="browser" value="FFX" />

Here, name attribute defines parameter name and value defines value of that parameter. Then we can use that parameter In selenium webdriver software automation test case using bellow given syntax.

@Parameters ({"browser"})

**Selenium WebDriver Parallel Tests Execution Using TestNG - @Parameters**

Browser compatibility software testing Is most Important thing for any software web application and generally you have to perform **browser**compatibility testing before 1 or 2 days of final release of software web application. In such a sort time period, you have to verify each Important functionality In every browsers suggested by client. If you will go for **running your all webdriver tests**In each browsers one by one then It will take too much time to complete your software tests and you may not complete It before release. In such situation, **Running your tests In all required browsers at same time** will helps you to save your time efforts. So question Is - Can we **run our tests parallel** In all required **browsers**using **webdriver** software automation testing tool? Answer Is yes.

Before learning how to run webdriver test parallel in multiple browsers, You must have knowledge about how to run test In webdriver using TestNg framework. You will find links of TestNG tutorial post with examples on [**THIS PAGE**](http://www.software-testing-tutorials-automation.com/2014/05/selenium-webdriver-tutorials-part-two.html). I have described testng configuration with detailed explanation on those links so read them carefully one by one. You can read webdriver related more tutorials on [**THIS LINK**](http://www.software-testing-tutorials-automation.com/2014/01/learn-selenium-webdriver-online-free.html).

**Parallelism In TestNG**

You can configure your testng.xml file In such a way to run your test suite or tests or methods In separate browsers Is known as parallelism In TestNG. Interviewer can ask you this question. Now let us look at example of parallel test execution In webdriver using testng. Bellow given example will run the test In Mozilla Firefox and Google chrome browser parallel.

Created Test\_Parallel.java class file for testing application and configured testng.xml file to run tests parallel as shown In bellow given example. In testng.xml file, parallel="tests" Inside <suite> tag will Instruct TestNG to consider bunch of methods of each <test> tag as separate thread. Means If you wants to run your software test In 2 different browsers then you need to create two <test> blocks for each browser Inside testng.xml file and Inside each <test> tag block, define one more tag "parameter" with same name(In both <test> block) but with different values. In bellow given testng.xml file, each <test> block has "parameter" tag with same name = browser but different values(FFX and CRM).

In test case, I have used @Parameters annotation to pass parameter In method. Values of this parameter will be feed by testng.xml file. and then If condition will check that value to decide which driver to use for test. This way, example testng.xml file will feed two values(FFX and CRM) In parameter so It will open Firefox and Google chrome browsers and run test In both browsers. (For running test In google chrome, You need to download chromedriver.exe. [**VIEW THIS POST**](http://www.software-testing-tutorials-automation.com/2013/09/running-selenium-webdriver-test-in.html) to know more how to run webdriver test In google chrome browser).

Run bellow given test In your eclipse with testng to see how It runs your test In 2 browsers at same time.

Test\_Parallel.java

package Testng\_Pack;

import org.junit.Assert;

public class Test\_Parallel {

private WebDriver driver=null;

@BeforeClass

**//parameter value will retrieved from testng.xml file's <parameter> tag.**

@Parameters ({"**browser**"})

public void setup(String **browser**){**//Method will pass value of parameter.**

if (**browser**.equals("**FFX**")) {**//If value Is FFX then webdriver will open Firefox Browser.**

System.out.println("Test Starts Running In Firefox Browser.");

driver = new FirefoxDriver();

}else if (**browser**.equals("**CRM**")){**//If value Is CRM then webdriver will open chrome Browser.**

System.out.println("Test Starts Running In Google chrome.");

System.setProperty("webdriver.chrome.driver",

"D:\\chromedriver\_win32\_2.3\\chromedriver.exe");

driver = new ChromeDriver();

}

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

driver.get("http://only-testing-blog.blogspot.in/2014/05/login.html");

}

**//Both bellow given tests will be executed In both browsers.**

@Test

public void verify\_title(){

String title = driver.getTitle();

Assert.assertEquals("Only Testing: LogIn", title);

System.out.println("Title Is Fine.");

}

@Test

public void verify\_message(){

driver.findElement(By.xpath("//input[@name='userid']")).sendKeys("UID1");

driver.findElement(By.xpath("//input[@type='password']")).sendKeys("pass1");

driver.findElement(By.xpath("//input[@value='Login']")).click();

String alert = driver.switchTo().alert().getText();

driver.switchTo().alert().accept();

Assert.assertEquals("UserId Is : UID1 Password Is : pass1", alert);

System.out.println("Alert Is Fine.");

}

@AfterClass

public void closebrowser(){

driver.quit();

}

}

testng.xml

<suite name="webDriver" **parallel="tests"**>

<test name="Test In FireFox" >

**<parameter name="browser" value="FFX" />**

<classes>

<class name="Testng\_Pack.Test\_Parallel" />

</classes>

</test>

<test name="Test In Google Chrome" >

**<parameter name="browser" value="CRM"></parameter>**

<classes>

<class name="Testng\_Pack.Test\_Parallel"></class>

</classes>

</test>

</suite>

If you wants to run your software test In one more browser then you need to create new <test> tag block in testng.xml file and set another parameter value and then put one more if else condition in class file to check and run your test.

This way you can your test parallel In multiple browsers at same time to reduce your time efforts.

**3** : **I have a test case with two @Test methods. I wants to exclude one @Test method from execution. Can I do It? How?**  
  
**Answer** : Yes you need to specify @Test method exclusion In testng.xml file as bellow.

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

<suite name="Test Exclusion Suite">

<test name="Exclusion Test" >

<classes>

<class name="Your Test Class Name">

<methods>

<**exclude name="Your Test Method Name To Exclude"**/>

</methods>

</class>

</classes>

</test>

</suite>

You need to provide @Test method name In exclude tag to exclude It from execution.  
  
You can [**VIEW DETAILED EXAMPLE**](http://www.software-testing-tutorials-automation.com/2014/03/including-only-selected-test-methods-in.html) on how to exclude specific @Test method from execution.

# Include/Exclude Only Selected Test Methods In Selenium WebDriver-TestNg Test Suite Using testng.xml

If you are using **selenium webdriver** with **TestNg**framework then you can easily **run your selected test methods** from selected classes. Supposing you have a two classes in your package and first class have three test methods and second class have five test methods. Now you wants to run only one test method from first class and two test methods from second class then you can configure it very easily in **testng.xml file**. In sort, you can **include or exclude selected test methods** from execution. Let me show you how to do it with simple **webdriver**test configured by **testng.xml** file.

**Configure TestNGOne Project In Eclipse**

1. First of all, Configure TestNGOne project in eclipse as described in [**PREVIOUS POST**](http://www.software-testing-tutorials-automation.com/2014/03/configure-testngxml-in-eclipse-for.html).
2. Add testmethodtwo() bellow the testmethodone() as shown bellow in ClassOne.java and ClassTwo.java file of all three packages.

public class ClassTwo extends TestNGOnePack.BaseClassOne{

@Test

public void testmethodone() {

driver.navigate().to("http://only-testing-blog.blogspot.in/2014/01/textbox.html");

String title = driver.getTitle();

System.out.print("\nCurrent page title is : "+title);

String Workdir = System.getProperty("user.dir");

String Classpackname = this.getClass().getName();

System.out.print("\n'"+Workdir+" -> "+Classpackname+" ->testmethodone' has been executed successfully");

}

**//Add testmethodtwo() here**

@Test

public void testmethodtwo() {

driver.findElement(By.xpath("//input[@value='female']"));

String Workdir = System.getProperty("user.dir");

String Classpackname = this.getClass().getName();

System.out.print("\n'"+Workdir+" -> "+Classpackname+" ->testmethodtwo' has been executed successfully");

}

}

Now your TestNGOne project's structure will looks like bellow.

**Configure testng.xml file to Include and run selected webdriver test methods from few classes**

Now supposing, I wants to run only

* testmethodone() method from TestNGOnePack -> ClassOne.java
* testmethodone() and testmethodtwo() methods from TestNGTwoPack -> ClassTwo.java
* testmethodone() and testmethodtwo() methods from TestNGThreePack -> ClassOne.java
* testmethodone() and testmethodtwo() methods from TestNGThreePack -> ClassTwo.java

To perform above action, I have to configure my testng.xml file as bellow.

<suite name="Suite One">

<test name="Test One" >

<classes>

<class name="TestNGOnePack.ClassOne" />

<methods>

<include name="testmethodone" />

</methods>

<class name="TestNGTwoPack.ClassTwo" />

</classes>

<packages>

<package name="TestNGThreePack" />

</packages>

</test>

</suite>

In above given testng.xml file, methods and include tags are new to learn. You can read about TestNg Framework's <suite>, <test>, <classes> and <class> tags in [**THIS POST**](http://www.software-testing-tutorials-automation.com/2014/03/creating-and-running-test-suit-using.html) and <packages>, <package> tags in [**THIS POST**](http://www.software-testing-tutorials-automation.com/2014/03/configure-testngxml-in-eclipse-for.html). **<methods> tag defines the group of methods and <include> tag defines which method you wants to include in execution**. Now execute this testng.xml file and verify the result report.

If you see in above result, Only test testmethodone() is executed from TestNGOnePack -> ClassOne.java. This way we can include any specific method in our test suite to execute from any class.  
  
**Configure testng.xml file to exclude specific test method from class**  
Sameway, To exclude specific test method from class, We can use <exclude> tag inside <methods> tag as bellow.

<suite name="Suite One">

<test name="Test One" >

<classes>

<class name="TestNGOnePack.ClassOne" />

<methods>

<include name="testmethodone" />

</methods>

<class name="TestNGTwoPack.ClassTwo" />

<methods>

<exclude name="testmethodone" />

</methods>

</classes>

<packages>

<package name="TestNGThreePack" />

</packages>

</test>

</suite>

When you run above given testng.xml file, it will exclude testmethodone() test method of TestNGTwoPack ->ClassTwo.jav from execution and will execute only ClassTwo.java file as shown in bellow given test result report.

This way we can configure our testng.xml file to include or exclude any specific test method from execution.

**4** : **Tell me syntax to skip @Test method from execution.**  
  
**Answer** : You can use bellow given syntax Inside @Test method to skip It from test execution.

throw new SkipException("Test Check\_Checkbox Is Skipped");

It will throw skip exception and @Test method will be sipped Immediately from execution. You can [**VIEW FULL EXAMPLE**](http://www.software-testing-tutorials-automation.com/2014/05/how-to-skip-webdriver-test-in-testng.html) on how to skip @Test method from execution.

**5** : **Arrange bellow give testng.xml tags from parent to child.**  
 **<test>**  
**<suite>**  
**<class>**  
**</methods>**  
**</classes>**  
  
**Answer** : Parent to child arrangement for above testng tags Is as bellow.  
  
<suite>  
<test>  
</classes>  
<class>  
</methods>

**6** : **How to set priority of @Test method? What Is Its usage?**

**Answer** : In your software web application's test case, you can set priority for TestNG @Test annotated methods as bellow.

@Test(priority=0)

Using priority, We can control @Test method execution manner as per our requirement. That means @Test method with priority = 0 will be executed 1st and @Test method with priority = 1 will be executed 2nd and so on. [**VIEW PRACTICAL EXAMPLE**](http://www.software-testing-tutorials-automation.com/2014/11/set-priority-for-selenium-webdriver.html) to know how to use It.

**7** : **Tell me any 5 assertions of TestNG which we can use In selenium webdriver software testing tool.**

**Answer** : There are many different assertions available In TestNG but generally I am using bellow given assertions In my test cases.

1. assertEquals [**VIEW EXAMPLE**](http://www.software-testing-tutorials-automation.com/2014/03/assertassertequals-testng-with-selenium.html)
2. assertNotEquals [**VIEW EXAMPLE**](http://www.software-testing-tutorials-automation.com/2014/03/example-of-assertnotequals-in-selenium.html)
3. assertTrue [**VIEW EXAMPLE**](http://www.software-testing-tutorials-automation.com/2014/03/selenium-webdriver-asserttrue-assertion.html)
4. assertFalse [**VIEW EXAMPLE**](http://www.software-testing-tutorials-automation.com/2014/03/example-of-assertassertfalse-assertion.html)
5. assertNull [**VIEW EXAMPLE**](http://www.software-testing-tutorials-automation.com/2014/03/selenium-webdriver-assertion-assertnull.html)
6. assertNotNull [**VIEW EXAMPLE**](http://www.software-testing-tutorials-automation.com/2014/04/testng-assertion-assertnotnull-with.html)

**8** : **Can you tell me usage of TestNG Soft Assertion In selenium webdriver software testing tool?**

**Answer** : Using TestNG soft assertion, We can continue our test execution even if assertion fails. That means on failure of soft assertion, remaining part of @Test method will be executed and assertion failure will be reported at the end of @Test method.

**9** : **Which time unit we provide In time test? minutes? seconds? milliseconds? or hours? Give Example.**

**Answer** : Time unit we provide on @Test method level or test suite level Is In milliseconds.You can [**VIEW FULL EXAMPLE**](http://www.software-testing-tutorials-automation.com/2014/12/testng-timeout-test-for-selenium.html) to know how to set time out.

# TestNG Timeout Test For Selenium WebDriver

Sometimes you will have a case where few of the **@Test methods** are taking very very long time to complete the execution or get stuck due to the some reason. In this case, Supposing you wants to **mark such @Test methods as failed** and go for executing next @Test method. **TestNG**has very good feature using which you can **set time period to wait for a test** to completely execute.

To get this feature, You needs to **set timeout on test suite level** In testng.xml file. Let's try to Implement It practically with example of selenium webdriver as bellow. Create bellow given test case In eclipse. In this test case, I have two @Test methods. In first @Test method I have used 5 seconds wait time (Thread.sleep(5000);) and In second @Test method I have used 1 second wait time (Thread.sleep(1000);) Intentionally.

[**VIEW MORE RELATED TESTNG TUTORIALS**](http://www.software-testing-tutorials-automation.com/2014/10/selenium-webdriver-advanced-tutorials.html)

**timeoutTest.java**

package Testing\_Pack;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

public class timeoutTest {

WebDriver driver;

@BeforeTest

public void setup() throws Exception {

System.out.println("In @BeforeTest Of Test\_One.");

driver =new FirefoxDriver();

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

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

driver.get("http://only-testing-blog.blogspot.in/2014/04/calc.html");

}

@Test

public void timeoutTestOne() throws InterruptedException {

System.out.println("Executing timeoutTestOne.");

**//Wait for 5 seconds.**

Thread.sleep(5000);

driver.findElement(By.xpath("//input[@id='2']")).click();

driver.findElement(By.xpath("//input[@id='plus']")).click();

driver.findElement(By.xpath("//input[@id='6']")).click();

driver.findElement(By.xpath("//input[@id='equals']")).click();

String Result = driver.findElement(By.xpath("//input[@id='Resultbox']")).getAttribute("value");

System.out.println("Result of timeoutTestOne = "+Result);

}

@Test

public void timeoutTestTwo() throws InterruptedException {

System.out.println("Executing timeoutTestTwo.");

**//Wait for 1 second.**

Thread.sleep(1000);

driver.findElement(By.xpath("//input[@id='Resultbox']")).clear();

driver.findElement(By.xpath("//input[@id='3']")).click();

driver.findElement(By.xpath("//input[@id='plus']")).click();

driver.findElement(By.xpath("//input[@id='7']")).click();

driver.findElement(By.xpath("//input[@id='equals']")).click();

String Result = driver.findElement(By.xpath("//input[@id='Resultbox']")).getAttribute("value");

System.out.println("Result of timeoutTestTwo = "+Result);

}

}

Create testng.xml file as bellow. You can see that I have used 3000 milliseconds timeout (time-out="3000") on suite level that means If any @Test method will take more than 3000 milliseconds to complete execution then Immediately TestNG will mark that @Test method as failed and will go to execute next @Test method. That means, Particular method can take maximum 3 seconds to complete the execution otherwise It will be marked as failed. Based on that, timeoutTestOne method will fail and timeoutTestTwo will pass when we execute testng.xml file.

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

<suite name="Timeout Test Suite" **time-out="3000"** verbose="1">

<test name="Timeout Test" >

<classes>

<class name="Testing\_Pack.timeoutTest"/>

</classes>

</test>

</suite>

Execution result of above test case will looks like bellow.

Also you can set timeout on Individual @Test method as bellow.

@Test(**timeOut=3000**)

public void timeoutTestTwo() throws InterruptedException {

//Test code

}

# Executing Selenium WebDriver Test Classes Parallel Using TestNG

Sometimes you need to execute multiple test cases at same time to save test execution time.**TestNG** has very good feature to **execute test cases/classes In parallel** to each other means you can execute two different test cases In **two different windows of same browser simultaneously**. If you remember, We have already learnt how to execute two different tests In two different browsers(Example : Firefox and Google Chrome) simultaneously In [**THIS POST**](http://www.software-testing-tutorials-automation.com/2014/05/selenium-webdriver-parallel-tests.html).

Now let us try to Implement **test classes parallel execution** practically. Supposing I have two calc test classes as Test\_One.java to sum values and Test\_Two.java to subtract values as bellow. Each having two different @Test methods.

1. Test\_One.java

package Testing\_Pack;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

public class Test\_One {

WebDriver driver;

WebElementdragElementFrom;

@BeforeTest

public void setup() throws Exception {

System.out.println("In @BeforeTest Of Test\_One.");

driver =new FirefoxDriver();

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

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

driver.get("http://only-testing-blog.blogspot.in/2014/04/calc.html");

}

@Test(priority=1)

public void testCaseOne\_Test\_One() {

System.out.println("Executing testCaseOne\_Test\_One.");

driver.findElement(By.xpath("//input[@id='2']")).click();

driver.findElement(By.xpath("//input[@id='plus']")).click();

driver.findElement(By.xpath("//input[@id='6']")).click();

driver.findElement(By.xpath("//input[@id='equals']")).click();

String Result = driver.findElement(By.xpath("//input[@id='Resultbox']")).getAttribute("value");

System.out.println("Result of testCaseOne\_Test\_One = "+Result);

}

@Test(priority=2)

public void testCaseTwo\_Test\_One() {

System.out.println("Executing testCaseTwo\_Test\_One.");

driver.findElement(By.xpath("//input[@id='Resultbox']")).clear();

driver.findElement(By.xpath("//input[@id='3']")).click();

driver.findElement(By.xpath("//input[@id='plus']")).click();

driver.findElement(By.xpath("//input[@id='7']")).click();

driver.findElement(By.xpath("//input[@id='equals']")).click();

String Result = driver.findElement(By.xpath("//input[@id='Resultbox']")).getAttribute("value");

System.out.println("Result of testCaseTwo\_Test\_One = "+Result);

}

}

2. Test\_Two.java

package Testing\_Pack;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

public class Test\_Two {

WebDriver driver;

WebElementdragElementFrom;

@BeforeTest

public void setup() throws Exception {

System.out.println("In @BeforeTest Of Test\_Two.");

driver =new FirefoxDriver();

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

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

driver.get("http://only-testing-blog.blogspot.in/2014/04/calc.html");

}

@Test(priority=1)

public void testCaseOne\_Test\_Two() {

System.out.println("Executing testCaseOne\_Test\_Two.");

driver.findElement(By.xpath("//input[@id='2']")).click();

driver.findElement(By.xpath("//input[@id='minus']")).click();

driver.findElement(By.xpath("//input[@id='6']")).click();

driver.findElement(By.xpath("//input[@id='equals']")).click();

String Result = driver.findElement(By.xpath("//input[@id='Resultbox']")).getAttribute("value");

System.out.println("Result of testCaseOne\_Test\_Two = "+Result);

}

@Test(priority=2)

public void testCaseTwo\_Test\_Two() {

System.out.println("Executing testCaseTwo\_Test\_Two.");

driver.findElement(By.xpath("//input[@id='Resultbox']")).clear();

driver.findElement(By.xpath("//input[@id='7']")).click();

driver.findElement(By.xpath("//input[@id='minus']")).click();

driver.findElement(By.xpath("//input[@id='3']")).click();

driver.findElement(By.xpath("//input[@id='equals']")).click();

String Result = driver.findElement(By.xpath("//input[@id='Resultbox']")).getAttribute("value");

System.out.println("Result of testCaseTwo\_Test\_Two = "+Result);

}

}

I wants to execute above given **selenium webdriver test classes simultaneously**. For that, I need to prepare testng.xml file as bellow.

You can [**VIEW LINK LIST OF TESTNG ADVANCED TUTORIALS**](http://www.software-testing-tutorials-automation.com/2014/10/selenium-webdriver-advanced-tutorials.html).

**testng.xml**

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

<suite name="Parallel Class Suite" **parallel="classes" thread-count="2"**>

<test name="Parallel Class Test" >

<classes>

<class name="Testing\_Pack.Test\_One"/>

<class name="Testing\_Pack.Test\_Two"/>

</classes>

</test>

</suite>

In above testng.xml file, I have used **parallel="classes"** that means testng will execute both test classes(Test\_One.java and Test\_Two.java) In two different thread/browser window. **thread-count="2"** parameter describes maximum number of threads allowed In execution at same time. Run above example and observe console result at the end of execution. It will looks like bellow.

As per console result, Both test classes executed parallel In two different threads.

**10. What is the difference between @BeforeMethod and @BeforeClass ?**

Ans- @BeforeMethod- this will execute before every @Test method.

@BeforeClass- this will execute before every class.

**11.What are the TestNG annotations? What is the Execution order**

**Ans:**

**Find below the annotation of TestNG with the execution order**

* @BeforeSuite
* @BeforeTest
* @BeforeClass
* @BeforeMethod
* @Test
* @AfterMethod
* @AfterClass
* @AfterTest
* @AfterSuite

**12.There are 2 test available in my class with @test i have 1 @beforeclass how will be the execution**

**Ans:**

* First @test and @beforeclass is ‘TestNg’ annotations
* As per this scenario first @beforeclass method code is executed

After that @test method related codes are executed

**13.How to rerun the failed test cases?**

**Ans:**

* There may be many reasons for a Test case getting failed, may be due to element not found or time out exception or stale element exception etc. Normally in automation after executing scripts/tests, we will check for the results and if the test fails because of above reasons we will re-run then again.
* Instead of that we can ask testNG to execute the failed test cases again for X (we can define) number of times and check for the updated results.
* To achieve this we need to implement [TestNGIRetryAnalyzer](http://testng.org/javadoc/org/testng/IRetryAnalyzer.html). Below is the simple code snippet:

**###############################################################################**

import org.testng.Assert;

import org.testng.IRetryAnalyzer;

import org.testng.ITestResult;

import org.testng.annotations.Test;

public class Retry implements IRetryAnalyzer {

private intretryCount = 0;

private intmaxRetryCount = 1;

public boolean retry(ITestResult result) {

if (retryCount<maxRetryCount) {

retryCount++;

return true;

}

return false;

}

@Test(retryAnalyzer = Retry.class)

public void testGenX() {

Assert.assertEquals("james", "JamesFail"); // ListenerTest fails

}

@Test(retryAnalyzer = Retry.class)

public void testGenY() {

Assert.assertEquals("hello", "World"); // ListenerTest fails

}

}

**14.How to run same test case in 3 different browser at a time?**

**Ans:**

* Cross browser Testing is a technique to test web application with different web browsers.
* Selenium can support different type of browsers for automation.
* Selenium can be integrated with TestNG to perform Cross Browser Testing.
* From parameters in testing.xml we can pass browser name and in test case we can create WebDriver reference accordingly.

**###############################################################################**

**15.How to pass a value to a test?**

**Ans:**

We can pass the value with using parameterization technique

* Parameter
* Data provider

**Parameter:**

* Parameter used to same input in difference methods with using ‘Suite’ and ‘Test’ level
* If parameter is used in ‘Suite’ level then that parameter applicable for all test in the suite
* If parameter is used in Test level then that parameter is applicable for all classes in the test
* No for ‘Class’ and ‘Method’ level
* Multiple parameter can create with different user name

**Data Provider:**

* Data provider used in the class level only
* We need to declare the data provider in the appropriate method
* If data provider in different class then method should be ‘Static’

Parameterization:

In TestNG:

###############################################################

<!DOCTYPE suite SYSTEM "[http://testng.org/testng-1.0.dtd">](http://testng.org/testng-1.0.dtd%22%3E%3Csuite)

[<suite](http://testng.org/testng-1.0.dtd%22%3E%3Csuite) name="Parameterization Test Suite">

<test name="Testing Parameterization">

<parameter name="browser" value="Firefox"/>

<parameter name="username" value="testuser"/>

<parameter name="password" value="testpassword"/>

<classes>

<class name="com.parameterization.TestParameters" />

</classes>

</test>

</suite>

##################################################################

In Automation Script:

#################################################################################

package com.parameterization;

import org.testng.annotations.Parameters;

import org.testng.annotations.Test;

public class TestParameters {

@Parameters({ "browser" })

@Test

public void testCaseOne(String browser) {

System.out.println("browser passed as :- " + browser);

}

@Parameters({ "username", "password" })

@Test

public void testCaseTwo(String username, String password) {

System.out.println("Parameter for User Name passed as :- " + username);

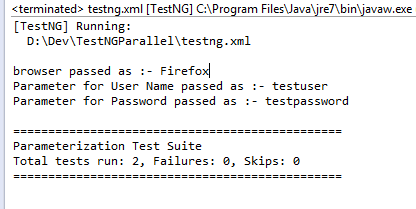
System.out.println("Parameter for Password passed as :- " + password);

}

}

Output:

############################################################################



############################################################################

**16.What are the annotations available in junit**

**Ans:**

@Test,@Before,@After,@BeforeClass,@AfterClass,@Ignore,@Runwith

**17.How to execute the particular test script in first without set the 'Priority'**

**Ans: We can give the alphabetic order in the method based on that it will run**

**18.In testNg how you will run parallel test?**

**19.Explain thread concept in TestNG?**