Introduction to TestNG:

In Selenium using java there are two frame works available JUNIT , TestNG

TestNG Testing Framework: Initially developers use TestNG for unit testing. Now it is used for all the testings.

TestNG is a testing framework designed to simplify a broad range testing needs, from unit testing to system testing. TestNG is an open source framework where NG stands for Next Generation

TestNG inspired from Junit(Java Platform) and Nunit(.Net platform), but introduced some new functionalities that make it more powerful and easier to use.

Advantages of TestNG:

1.TestNG Annotations are easy to create the test cases

2.Test cases can be Grouped and Prioritized more easily and supports parameterization(global)

3.Supports data driven testing (passing multiple values ) using Dataproviders.

4.Execute Multiple programs / classes using XML.

5.Generate HTML reports

6.Parallel test execution is possible and readily supports integration with other tool and plugins foe ex: Eclipse, IDE, build tools like Ant, Maven

Note: using TestNG we can create test case , group test cases, prioritize test case, execute test cases, and generate Test Reports.

Installing TestNG in Eclipse:

In Eclipse IDE

>help menu

>Install new Software

>click add

>Enter name as TestNG

>Enter URL as <http://beust.com/eclipse/>

>Select TestNG

>Next>Next>Accept the Agreement>Finish

First TestNG Test Case:

Test Case Name : Verify the title of the page

Test steps

1. Launch the browser
2. Navigate to gamil.com

Verification point:

Capture the page title and compare with expected

Expected=

Actual =

Status =

Verifying Page Title

**package** SeleniumTests;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.testng.Assert;

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

**public** **class** VerifyPageTitle {

@Test

**public** **void** verifyTitle(){

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

WebDriver driver = **null**;

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

String pageTitle = driver.getTitle();

Assert.*assertEquals*(pageTitle, "Gmail");

}

}

Note:

1.main method is not used for TestNG programs.

2.TestNG program contains only methods that contain @Test Annotations.

3. If we don’t write @Test Annotation then the method is not going to be executed.

Multiple Test Cases in a Program:

**package** SeleniumTests;

**import** org.testng.Assert;

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

**public** **class** Multipletestcases {

@Test

**public** **void** Multipletest(){

Assert.*assertEquals*("Gmail", "Gmail");

}

@Test

**public** **void** abcd(){

Assert.*assertEquals*("Yahoo", "Yahoo");

}

@Test

**public** **void** xyz(){

Assert.*assertEquals*("abcde", "abcde");

}

}

**As per the program**

MultipleTest

Abcd

Xyz

**Execution Flow**

Abcd

Multile test

Xyz

**NOTE:**  TestNG test cases are executed in alphabetical order. If you want to control the execution flow then use priority attribute.

**package** SeleniumTests;

**import** org.testng.Assert;

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

**public** **class** Multipletestcases {

@Test (priority = 1)

**public** **void** Multipletest(){

Assert.*assertEquals*("Gmail", "Gmail");

}

@Test (priority = 2)

**public** **void** abcd(){

Assert.*assertEquals*("Yahoo", "Yahoo");

}

@Test (priority = 3)

**public** **void** xyz(){

Assert.*assertEquals*("abcde", "abcde");

}

}

Result: PASSED:

Multipletest

PASSED: abcd

PASSED: xyz

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

Default test

Tests run: 3, Failures: 0, Skips: 0

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

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

Default suite

Total tests run: 3, Failures: 0, Skips: 0

dependsOnMethods attribute: Instead of priority attribute we can use dependsOnMethods attribute

**package** SeleniumTests;

**import** org.testng.Assert;

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

**public** **class** verifypagetitle1 {

@Test

**public** **void** login(){

Assert.*assertEquals*("xyz", "xyz");

}

@Test (dependsOnMethods = {"advancedSearch"})

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

Assert.*assertEquals*("abcder", "abcder");

}

@Test (dependsOnMethods = {"login"})

**public** **void** search(){

Assert.*assertEquals*("abcde", "abcde");

}

@Test (dependsOnMethods = {"search"})

**public** **void** advancedSearch(){

Assert.*assertEquals*("abcd", "xyz");

}

}

Result: PASSED: login

PASSED: search

FAILED: advancedSearch

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

Default test

Tests run: 4, Failures: 1, Skips: 1

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

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

Default suite

Total tests run: 4, Failures: 1, Skips: 1

If we use priority attribute then all test cases will be execute with no skips.

If we use dependsOnMethod attribute then it will skip the test case whenever dependsOnMethod Test case fails, if you want to execute the test case forcebly then use alwaysRun attribute.

If there is no functionality dependency then use priority attribute.

If there is any functionality dependency then use dependsOnMethod attribute.

Test Cases:

Launch browser

Verify gmail home page

Verify yahoo home page

Close browser

TestNG Program:

**package** SeleniumTests;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.testng.Assert;

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

**public** **class** TestngProgram {

**public** WebDriver driver;

@Test (priority = 1)

**public** **void** launchBrowser(){

driver = **new** FirefoxDriver();

}

@Test (priority = 2)

**public** **void** verifyGmailpage(){

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

Assert.*assertEquals*("Gmail",driver.getTitle());

}

@Test (priority=3)

**public** **void** verifyYahoopage(){

driver.get("https://in.yahoo.com/");

Assert.*assertEquals*("Yahoo",driver.getTitle());

}

@Test (priority= 4)

**public** **void** closeBrowser(){

driver.close();

}

}

Result:

PASSED: launchBrowser

PASSED: verifyGmailpage

PASSED: verifyYahoopage

PASSED: closeBrowser

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

Default test

Tests run: 4, Failures: 0, Skips: 0

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

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

Default suite

Total tests run: 4, Failures: 0, Skips: 0

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

TestNg Annotations:

@Test : Makes Method as Test Case

@BeforeMethod: Pre condition for every test case in a class

@AfterMethod: Post condition for every test case in a class

@BeforeClass: Pre condition for all test cases in a class

@AfterClass: Post condition for all test cases in a class

@BeforeTest: Pre condition for all test cases in multiple class

@AfterTest: Post condition for every test case in a class

Parallel test execution using TestNG (using single machine or computer)

Parallel test execution using Selenium Grid (using multiple machines and computers)

package seleniumtests;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class annotations2 {

@BeforeClass

public void login(){

System.out.println("Login Succesful");

}

@AfterClass

public void logout(){

System.out.println("Logout Succesful");

}

@Test (priority = 1)

public void accountSummary(){

System.out.println("account summary Succesful");

}

@Test (priority = 2)

public void fundTransfer(){

System.out.println("fund Transfer Succesful");

}

@Test (priority = 3)

public void billPayment(){

System.out.println("bill payment Succesful");

}

}

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

**package** seleniumtests;

**import** org.testng.annotations.AfterClass;

**import** org.testng.annotations.AfterTest;

**import** org.testng.annotations.BeforeClass;

**import** org.testng.annotations.BeforeTest;

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

**public** **class** annotations {

@BeforeTest

**public** **void** login(){

System.***out***.println("Login Succesful");

}

@AfterTest

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

System.***out***.println("Logout Succesful");

}

@Test (priority = 1)

**public** **void** search(){

System.***out***.println("search Succesful");

}

@Test (priority = 2)

**public** **void** advancedSerach(){

System.***out***.println("advanced search Succesful");

}

@Test (priority = 3)

**public** **void** buyingProducts(){

System.***out***.println("buying products Succesful");

}

@Test (priority = 4)

**public** **void** testCase(){

System.***out***.println("Test Case in "+getClass().getSimpleName()

+"with thread id:"+Thread.*currentThread*().getId());

}

}

Multiple XML:

suite name = *"Ecommerce suite"*>

<test name = *"Sanity Test"*>

<classes>

<class name = *"seleniumtests.annotations"*/>

<class name = *"seleniumtests.annotations2"*/>

</classes>

</test></suite>

Parallel XML:

<suite name = *"Parallel Test suite"* parallel= *"Classes"* thread-count = *"2"*>

<test name = *"Sanity Test"*>

<classes>

<class name = *"seleniumtests.annotations"*/>

<class name = *"seleniumtests.annotations2"*/>

</classes>

</test>

</suite>

Parallel

Parallel = “methods”: TestNG will run all the methods in separate threads.

Parallel = “Classes”: TestNG will run all the methods in the same class in the same thread

Parallel = “tests”: TestNG will run all the methods in the same <test> tag in the same thread

Class 1

3 methods thread(1) 1

Class 2

3 methods thread (1) 2

Class 1

3 methods thread (3) 1

Class 2

3 methods thread (3) 2

Listeners: are the very important features of TestNG which allow you to customized logs or report of TESTNG

As the name says it listen to certain events and behaves accordingly

We can fully customize the logs using listeners.

Different ways to implement:

We can either extend TestlistnerAdapter class or we can implement interface Itestlistner which has certain methods which we have to implement

1. We can implement at class level
2. We can implement at suite level

**package** Listnersdemo;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.testng.annotations.Listeners;

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

@Listeners(Listnersdemo.TestNGListener.**class**)

**public** **class** FirstTestCase {

@Test

**public** **void** GoogleTitleVerify(){

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

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

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

System.***out***.println(driver.getTitle());

driver.quit();

}

}