TestNG:

TestNG is an open source automated testing framework; where NG of TestNG means Next Generation. TestNG is similar to JUnit, but its not a JUnit extension. Its inspired by JUnit. It is designed to be better than JUnit, especially when testing integrated classes.

TestNG is designed to cover all categories of tests: unit, functional, end-to-end, integration.

Requirements

TestNG requires JDK version 5 or higher.

What are the benefits of using annotations?

Following are some of the benefits of using annotations:

TestNG 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

What are the different ways in which TestNG can be invoked?

You can invoke TestNG in several different ways:

Using Eclipse

With ant OR Maven

From the command line

Maven dependency to add in POM.xml file

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>6.14.3</version>

<scope>test</scope>

</dependency>

How can you disable a test case in TestNG?

Annotation @Test(enabled = false) helps to disable the test case which you want to ignore.

Installation of TestNg frame work:

1. Launch Eclipse IDE & under Help menu click on “Install New Software” option.

2. Install dialog box will be appeared

3. In the install dialog box, enter URL as “ https://testng.org/testng-eclipse-update-site” in Work with text box & press keyboard enter key.

4. Under Name column TestNG check box will be displayed

5. Select the “TestNG” check box & click on Next button

6. On Install Details screen, make sure that TestNG is selected & then click on Next button

7. On Review Licenses screen read license agreement & select “I accept…”

8. Ensure that Installing Software dialog appear

9. One security warning message will appear during installation. Ensure that you click on Ok button if same security dialog is appeared.

10. Once the installation process is completed then Eclipse ask you to Restart, click on Yes button

11. Once the Eclipse is restarted, to make sure whether TestNG is installed or not. Just under Run menu click on “Run As” option & check new option is added called “TestNG Test”

Annotations:

@BeforeSuite: The annotated method will run before all tests in this suite have.

@AfterSuite: The annotated method will run after all tests in this suite have.

@BeforeTest: The annotated method will run before any test method belonging to the classes inside the <test> tag is run.

@AfterTest: The annotated method will run after all the test methods belonging to the classes inside the <test> tag have run.

@BeforeGroups: The list of groups that this configuration method will run before. This method is guaranteed to run shortly before the first test method that belongs to any of these groups is invoked.

@AfterGroups: The list of groups that this configuration method will run after. This method is guaranteed to run shortly after the last test method that belongs to any of these groups is invoked.

@BeforeClass: The annotated method will run before the first test method in the current class is invoked.

@AfterClass: The annotated method will be run after all the test methods in the current class have been run.

@BeforeMethod: The annotated method will be run before each test method.

@AfterMethod: The annotated method will be run after each test method.

@Test: This we need to use for test cases

@DataProvider: Marks a method as supplying data for a test method. The annotated method 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 DataProvider needs to use a dataProvider name equals to the name of this annotation.

name: The name of this data provider. If it's not supplied, the name of this data provider will automatically be set to the name of the method.

parallel: If set to true, tests generated using this data provider are run in parallel. Default value is false.

package Pk1;

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 TestNGExampleTest {

@Test

public void testCase1() {

System.out.println("in test case 1");

}

// test case 2

@Test

public void testCase2() {

System.out.println("in test case 2");

}

@BeforeMethod

public void beforeMethod() {

System.out.println("in beforeMethod");

}

@AfterMethod

public void afterMethod() {

System.out.println("in afterMethod");

}

@BeforeClass

public void beforeClass() {

System.out.println("in beforeClass");

}

@AfterClass

public void afterClass() {

System.out.println("in afterClass");

}

@BeforeTest

public void beforeTest() {

System.out.println("in beforeTest");

}

@AfterTest

public void afterTest() {

System.out.println("in afterTest");

}

@BeforeSuite

public void beforeSuite() {

System.out.println("in beforeSuite");

}

@AfterSuite

public void afterSuite() {

System.out.println("in afterSuite");

}

}

TestNG Parallel Execution:

It means that if you want to execute test cases Parallel then we can go for using this option. TestNG provides this option and saves time as test cases are executing Parallel.It is manily used to save time and for compatibility testing

Threads in TestNG: By default in TestNG it will have 5 threads and will be used for upto 5 methods. But if you have more methods you need to mention thread count in testng.xml file