**TEST NG – Test Driven Development**

**TDD – Test Driven Development.**

**Inspired by J-Unit**

**Features:**

**1. Simple Annotations - STCM**

**2. Invocation Count**

**3. Priority**

**4. Ignore**

**5. Grouping**

**6. Dependency**

**7. Time Out**

**8. Expected Exception Test**

**9. Parallel Test**

10. Cross Browser Test

**11. Report**

**12. Failed Test**

13. Parameterized Test [Single set of Data]

14. Data Provided [ Multi set of Data]

15. IRetry Analyser

Installation Procedure:

Help-> Eclipse Marketplace 🡪 TestNG🡪 Install

Or

Try this

<https://www.lambdatest.com/blog/how-to-install-testng-in-eclipse-step-by-step-guide/>

Or

Search in google TestNG download 🡪

<https://testng.org/testng-eclipse-update-site>

Help 🡪 Install New software 🡪 Select Add 🡪 paste the link, give the header as TESTNG🡪 next 🡪 finish.

Or

Download JAR 🡪 build path 🡪 configure build path

<https://jar-download.com/artifacts/org.testng/testng>

Pom.xml

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.4.0</version>

<scope>test</scope>

</dependency>

**TestNG Features**

1. Simple Annotation – STCM:

Based on your requirements, we can access the test methods

Example with flow to understand the simple annotation

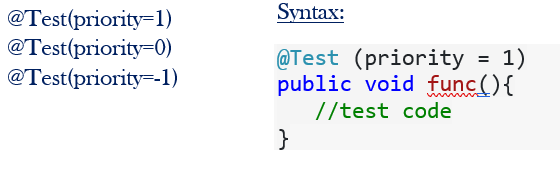


1. Priority

* Schedule the test cases.
* One method is allowed to have only one priority in TestNG.
* Priority cannot pass through the XML files.
* Priority order goes to -ve to 0 to +ve

-5 -4 -3 -2 -1 0 +1 +2 +3 +4 +5

If we didn’t provide the priority, @Test will consider as Zero



1. Invocation Count

Number of times a test method should be invoked or executed before executing any other test methods

@Test(invocationCount = 2)

1. Ignore Test

When the test case that is not ready to test is bypassed.

* @Ignore
* @Test (enabled = false)
* <exclude name=”method name”> in xml file

1. Dependence On Method

It allows a test method to depend on a single or a group of test methods.

@Test(dependsOnMethods = "required methodName")

Example: UG, PG, Phd

Report 🡪 emailable-report.html

1. TimeOut Test

* It may take longer execution time than expected. In such scenarios, we may need to mark the said test as fail and then move to the next test in the suite.
* **Timeout Configuration**

TestNG allows user to configure a time period to wait for a test to complete execute.

**🡪 @Test(timeOut=1000) 🡪** After 1Sec it will timeOut.

Timeout can be configured at two levels:

**Suite level** –Thread.sleep()

**Test level** – This will be applicable for the said test method and will override the timeout period if configured at the suite level

**Example:**

@Test(timeOut = 1000)

public void method1() throws InterruptedException {

Thread.sleep(4000);

System.out.println("Time test method one");

}

Note:

Process Executed🡪 If given time is Higher than thread value.

**Internal thread timeOut Exception** 🡪 If given timeOut is lower than thread.

1. Expected Exception

We can pass our test case even we get an expected exception in our program.

Syntax:

@Test(expectedExceptions = ArithmeticException.class)

@Test(expectedExceptions = ArithmeticException.class, IOException.class)

1. Grouping Test

* **Doesn’t Exist in Junit Framework**
* Special Feature in TestNG
* Allow you to perform grouping of different test methods

GROUPS

GROUPS OF GROUP

<groups>

<run>

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

</run>

</groups>

<classes>

<class name=*"package.class"*>

</class></classes>

GROUP OF GROUP

<groups>

<define name=*"Entertainment"*>

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

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

</define>

<define name=*"Education"*>

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

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

</define>

<run>

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

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

</run>

</groups>

1. Parallel Test

To run multiple tests at the same time

Parallel default is false.

<test name=*"test"* parallel=*"methods"* thread-count=*"4"*>

1. Cross Browser Test

At the end of the deployment, web application can be opened in any browser by the end user.

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

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

<suite name="TestSuite" thread-count="2" parallel="tests" >

<test name="ChromeTest">

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

<classes>

<class name="test.CrossBrowserTestingScript">

</class>

</classes>

</test>

<test name="FirefoxTest">

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

<classes>

<class name="test.CrossBrowserTestingScript">

</class>

</classes>

</test>

</suite>

15)IRetry Analyzer

* We can see some random failure during an automated test run.
* These failures might not necessarily because of product bugs.

Failures :

Browser is not responded

Server issue

Unexpected delay

Trnasform [ IAnnotationTransformer – Interface]

ITestAnnotation: Refer all the test case

Annotation.setRetryAnalyzer(classname.class);

VALIDATION[it will stop execute] [ASSERT -Class]

**Hard Assert** – it will stop execute if the test is failed

Assert.*assertEquals*(actual, expected); //test pass

Assert.*assertNotEquals*(actual, expected); // test pass

Assert.*assertSame*(actual, expected);

Assert.*assertNull*(actual);

Assert.*assertNotNull*(object);

Assert.*assertTrue*(condition);

Assert.*assertFalse*(condition);

**Soft Assert** [VERIFICATION- to check]– it will run complete even exception throws

SoftAssert s=new SoftAssert();

s.assertEquals(actual String, expected String);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sample:

**private** **void** uName() {

String expected="Sashwin";

String actual="Sashwin";

Assert.*assertEquals*(actual, expected);

}

IRetryAnalyzer

**int** count=1;

**int** limit=5;

@Override

**public** **boolean** retry(ITestResult result) {

**if**(count <= limit) {

count++;

**return** **true**;

}

**return** **false**;}

IAnnotationTransformer

@Override

**public** **void** transform(ITestAnnotation annotation, Class testClass, Constructor testConstructor, Method testMethod) {

annotation.setRetryAnalyzer(retry\_analyse.**class**);

}

<listeners>

<listener class-name=*"com.testng\_pgm.Transformer\_Class"*></listener>

</listeners>