Bug life cycle

Test case

Test plan

Manual Testing ----

Web , Mobile-99, Desktop (Windows) -1

UI Testing -- Selenium, cypress, smart bear, tosca, playwright, Katalon

API Testing (Micro services) – post man, Rest assured api , karate

Database testing: reading data from tables , DML , writing complex sql quires

Reports in your application ,

ETL Testing – Data warehouse , applying transformation as per mapping , validate warehouse or data marts

Mobile Testing: IOS, Android, Cloud execution (Sauce labs)

Performance testing: Load runner, JMeter

Devops : CI /CD pipeline , docker

Security testing : penetration testing

Globalisation / localization

Domain Knowledge: Banking, Insurance, Retail,

Travel, Health care ,

Advanced topics: chatbot, IVR, ML testing (Face reg , Finger print)

Specialist: ATM Testing, Modem testing, Telcom, 5 g

Topics:

Groups

Disable a test

Exception Test

Time Test

Parameterization

* Using testing.xml file
* Optional parameters
* Data provider

Dependency test

Parallelism

Hard Assertion vs Soft Assertion

Reporting in Test ng

Listeners

Connect to DB

POM

BDD (Cucumber)

Testing Paths

1 -3 -- Junior tester , Test Engineer

3 -5 -- Test Associate , Test Analyst

5-8 --- Sr Test Engineer

8-13 -- Test Lead (People Management/ Technical Test lead (framework dev)

13+ -- Test Manager / Agile Test Manager / Test Delivery Manager -- Management Roles

Test Architect

16 + --TCOE – Testing Centre of Excellence

VP -- Vice President

SVP – Senior VP

Executive Director

Manging Director

Country Head ( India , SG , Australia)

Region Head ( Asia pacific)

Team – squad

2 – Front end developers web – React js

2 – Front end developers mobile – 1 ios , 1 andriod

2 – Back-end developers ( Java , API)

2 – Testers ( Manual / Automation )

1 – BA / PO

1 – Scrum master--- (1 -3)

1 – tester manager

1 – Project manager --quality, client

1 – Release manger

1 – Middleware - Kafka , rabbit , json ,

Group :

**public** **class** ExcludeGroups {

@Test(groups="group1")

**public** **void** testMethodone() {

System.***out***.println("test method one belongs to include group");

}

@Test(groups="group1")

**public** **void** testMethodtwo() {

System.***out***.println("test method two belongs to include group");

}

@Test(groups={"group1","group2"})

**public** **void** testMethodthree() {

System.***out***.println("test method three belongs to 2 groups");

}

@Test(groups={"regression","group2"})

**public** **void** testMethodFour() {

System.***out***.println("test method four belongs to 2 groups");

}

}

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

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

<suite name="ExcludeGroup Suite">

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

<groups>

<run>

<include name="group1"></include>

<exclude name="group2"></exclude>

</run>

</groups>

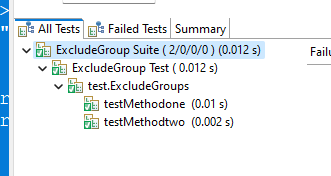
<classes>

<class name="test.ExcludeGroups" />

</classes>

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

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



Disabling a test

**package** test;

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

**public** **class** DisabledTest {

@Test(enabled=**true**)

**public** **void** testMethodone() {

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

}

@Test(enabled=**false**)

**public** **void** testMethodtwo() {

System.***out***.println("test method two");

}

@Test(enabled=**true**)

**public** **void** testMethodthree() {

System.***out***.println("test method three");

}

}

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

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

<suite name="DisableSuite">

<test thread-count="5" name="DisableTest">

<classes>

<class name="test.DisabledTest"/>

</classes>

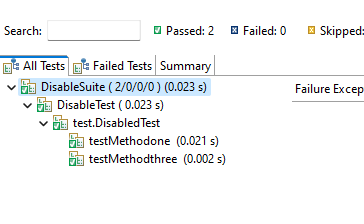
</test> <!-- DisableTest -->

</suite> <!-- DisableSuite -->

Summary :

If enabled is false , then test is ignored from execution

By default enabled id true



Exception Test:

**package** test;

**import** java.io.IOException;

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

**public** **class** ExceptionTest {

@Test(expectedExceptions=IOException.**class**)

**public** **void** testone() **throws** Exception{

**throw** **new** IOException();

}

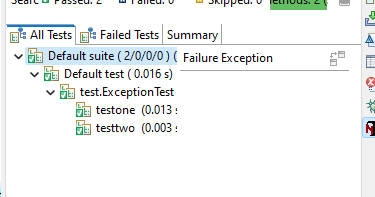
@Test(expectedExceptions={IOException.**class**,NullPointerException.**class**})

**public** **void** testtwo() **throws** Exception{

**throw** **new** Exception();

}

}



Summary: The test will fail when expected exception list does not match with exception thrown

Time Test :

**package** test;

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

**public** **class** TimeTest {

@Test

**public** **void** Testone() **throws** InterruptedException {

Thread.*sleep*(1000);

System.***out***.println("Test method one called");

}

@Test

**public** **void** Testtwo() **throws** InterruptedException {

Thread.*sleep*(400);

System.***out***.println("Test method two called");

}

}

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

<suite name="First Suite" verbose="1" >

<test name="First Test" >

<classes>

<class name="test.FirstTest" />

</classes>

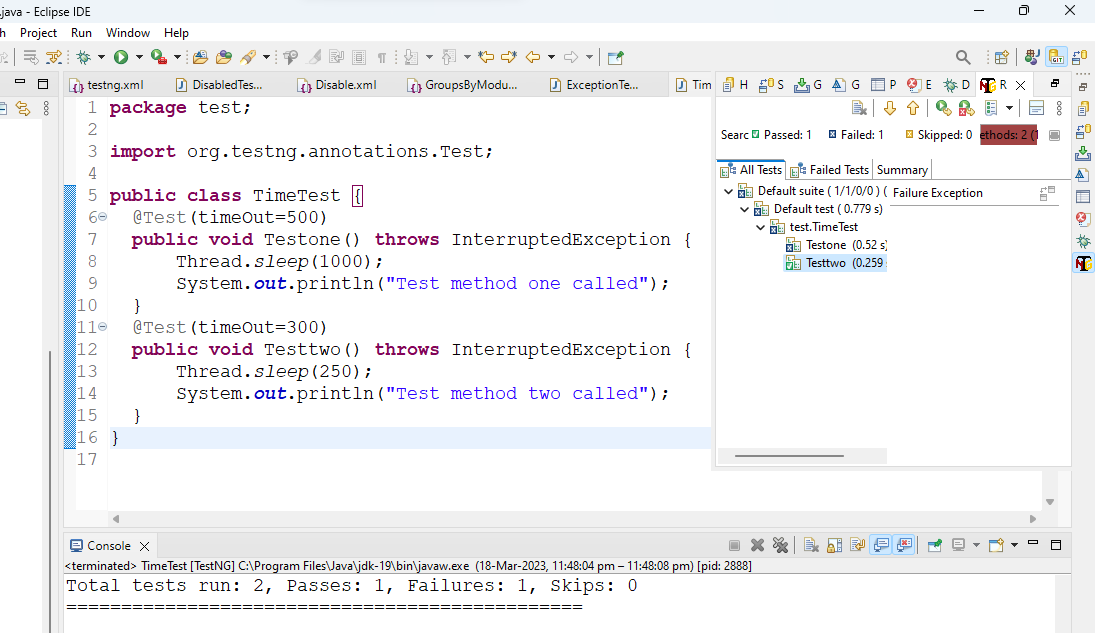
</test>

</suite>

Summary: First test has failed because test had taken more than 500 secs to executed .

Second test passed because it could complete execution within time outs

Timeout at Test level :



**package** test;

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

**public** **class** TimeTest {

@Test(timeOut=500)

**public** **void** Testone() **throws** InterruptedException {

Thread.*sleep*(1000);

System.***out***.println("Test method one called");

}

@Test(timeOut=300)

**public** **void** Testtwo() **throws** InterruptedException {

Thread.*sleep*(250);

System.***out***.println("Test method two called");

}

}

Summary :

First test failed because the test took more time to execute than the time specified in timeout attribute .

Parameterization

* Using testing.xml file
* Optional parameters
* Data provider

Using testing.xml : when we need to pass values to test method as string type , we have to comfigure .xml file with annotation called Paramerters

**package** test;

**import** org.testng.annotations.Parameters;

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

**public** **class** ParameterTest {

@Parameters("username")

@Test

**public** **void** Testone(String name) {

System.***out***.println("Test method one called");

System.***out***.println("Test method pne parameter is: "+name);

}

@Parameters("password")

@Test

**public** **void** Testtwo(String pwd) {

System.***out***.println("Test method two called");

System.***out***.println("Test method two parameter is: "+pwd);

}

}

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

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

<suite name="ParamSuite">

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

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

<test thread-count="5" name="ParamTest">

<classes>

<class name="test.ParameterTest"/>

</classes>

</test> <!-- ParamTest -->

</suite> <!-- ParamSuite -->