**TestNG Listener**

Listener is defined as interface that **modifies the default TestNG's behavior**. As the name suggests Listeners "listen" to the event defined in the selenium script and behave accordingly. It is used in selenium by implementing Listeners Interface. It allows customizing TestNG reports or logs.

In TestNG, a listener is represented by the marker interface org.testng.ITestNGListener. TestNG provides you with many more interfaces that extend org.testng.ITestNGListener. Each interface defines one aspect of TestNG. In order to extend TestNG behavior one needs to implement the TestNG-provided listener interface and then integrate it with TestNG.

Below are some of the listeners that TestNG provides:

* IExecutionListener
* IAnnotationTransformer
* ISuiteListener
* **ITestListener**
* IConfigurationListener
* IMethodInterceptor
* IInvokedMethodListener
* IHookable
* IReporter

**IExecutionListener**

IExecutionListener is a listener that monitors the beginning and end of a TestNG run. It has two methods, onExecutionStart() and onExecutionFinish(). Method onExecutionStart() is called before the TestNG starts running the suites and onExecutionFinish() is called after TestNG is done running all the test suites.

Test Class <-> Class implementing IExecutionListener <-> testing.xml having listener class

Below is the LoginTest class having two test methods:

**package** TestNG;

**import** org.testng.annotations.AfterMethod;

**import** org.testng.annotations.BeforeMethod;

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

**public** **class** LoginTest {

@BeforeMethod

**public** **void** open\_browser\_and\_navigate() {

System.***out***.println("Inside before method");

// 1) Launch browser

// 2) Navigate URL

}

@AfterMethod

**public** **void** exit\_browser() {

// selenium code to cext browser

System.***out***.println("inside exit browser");

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

}

@Test()

**public** **void** Test\_Login() {

System.***out***.println("Inside Logintest");

}

@Test()

**public** **void** call\_conference\_test() {

System.***out***.println("Inside call conference");

}

}

Below ExecutionListener1 class implements IExecutionListener interface

**package** TestNG;

**import** org.testng.IExecutionListener;

**public** **class** ExecutionListener1 **implements** IExecutionListener {

@Override

**public** **void** onExecutionStart() {

System.***out***.println("ExecutionListener1 class...onExecutionStart");

}

@Override

**public** **void** onExecutionFinish() {

System.***out***.println("ExecutionListener1 class...onExecutionFinish");

}

}

Below is testing.xml

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

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

<suite name=*"Suite1"* thread-count=*"5"*>

<listeners>

<listener class-name=*"TestNG.ExecutionListener1"*></listener>

</listeners>

<test name=*"Test1"*>

<classes>

<class name=*"TestNG.LoginTest"*></class>

</classes>

</test>

</suite>

IAnnotationTransformer

Annotations are static in nature by design, so any change in the values requires recompilation of source files. Since TestNG relies heavily on annotations, it would be nice if one can override its behavior at runtime. This is exactly what TestNG allows you to do using its annotation transformation framework.

IAnnotationTransformer is a TestNG listener which allows you to modify **TestNG (@Test) annotation** and configure it further.

TestAnnotationTransformerExample is the class having test method ‘Cat’

**package** TestNG;

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

**public** **class** TestAnnotationTransformerExample {

@Test(dataProviderClass=DataProviderFactory.**class**,dataProvider="dp1")

**public** **void** Cat(**int** id, String Name) {

System.***out***.println("Inside Cat method and id = " + id);

System.***out***.println("Inside Cat method and name = " + Name);

}

}

DataProviderFactory class has @DataProvider dp1, dp2 and dp3

**package** TestNG;

**import** org.testng.annotations.DataProvider;

**public** **class** DataProviderFactory {

@DataProvider()

**public** Object[][] dp1() {

Object o1[][] = **new** Object[3][2];

o1[0][0] = 1;

o1[0][1] = "Jigar";

o1[1][0] = 2;

o1[1][1] = "Mehta";

o1[2][0] = 3;

o1[2][1] = "Learning";

**return** o1;

}

@DataProvider

**public** Object[][] dp2() {

Object o1[][] = **new** Object[3][2];

o1[0][0] = 11;

o1[0][1] = "TestNG";

o1[1][0] = 12;

o1[1][1] = "is";

o1[2][0] = 13;

o1[2][1] = "easy";

**return** o1;

}

@DataProvider()

**public** Object[][] dp3() {

Object o1[][] = **new** Object[3][2];

o1[0][0] = 21;

o1[0][1] = "How's";

o1[1][0] = 22;

o1[1][1] = "Your";

o1[2][0] = 23;

o1[2][1] = "Day";

**return** o1;

}

}

When below testing.xml is run the output shows data from dp1

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

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

<suite name=*"Suite1"* thread-count=*"5"*>

<!-- <listeners>

<listener class-name="TestNG.TestAnnotationTransformerListener2"></listener>

</listeners> -->

<test name=*"Test1"*>

<classes>

<class name=*"TestNG.TestAnnotationTransformerExample"*></class>

</classes>

</test>

</suite>

Output:

[RemoteTestNG] detected TestNG version 6.14.2

Inside Cat method and id = 1

Inside Cat method and name = Jigar

Inside Cat method and id = 2

Inside Cat method and name = Mehta

Inside Cat method and id = 3

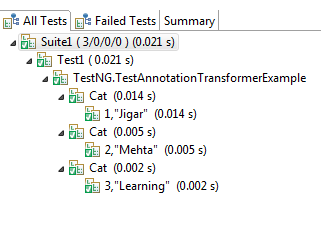
Inside Cat method and name = Learning

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

Suite1

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

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



Below two classes implements IAnnotationTransformer

**package** TestNG;

**import** java.lang.reflect.Constructor;

**import** java.lang.reflect.Method;

**import** org.testng.IAnnotationTransformer;

**import** org.testng.annotations.ITestAnnotation;

**public** **class** TestAnnotationTransformerListener1 **implements** IAnnotationTransformer {

@Override

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

**if** (testMethod.getName().equals("Cat")) {

System.***out***.println("Data provider for " + testMethod.getName());

System.***out***.println("Current DataProvider class = " + annotation.getDataProviderClass());

System.***out***.println("Current DataProvider = " + annotation.getDataProvider());

annotation.setDataProviderClass(DataProviderFactory.**class**);

annotation.setDataProvider("dp2");

}

}

}

**package** TestNG;

**import** java.lang.reflect.Constructor;

**import** java.lang.reflect.Method;

**import** org.testng.IAnnotationTransformer;

**import** org.testng.annotations.ITestAnnotation;

**public** **class** TestAnnotationTransformerListener2 **implements** IAnnotationTransformer {

@Override

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

**if** (testMethod.getName().equals("Cat")) {

System.***out***.println("Data provider for " + testMethod.getName());

System.***out***.println("Current DataProvider class = " + annotation.getDataProviderClass());

System.***out***.println("Current DataProvider = " + annotation.getDataProvider());

annotation.setDataProviderClass(DataProviderFactory.**class**);

annotation.setDataProvider("dp3");

}

}

}

Observer the change in TestNg.xml. <listeners> tag is added and Dataprovider class and Dataproviders for method ‘Cat’ in TestAnnotationTransformerExample class changed to dp2 using TestNg.xml.

Hence no code change or compilation is required

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

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

<suite name=*"Suite1"* thread-count=*"5"*>

<listeners>

<listener

class-name=*"TestNG.TestAnnotationTransformerListener1"*></listener>

</listeners>

<test name=*"Test1"*>

<classes>

<class name=*"TestNG.TestAnnotationTransformerExample"*></class>

</classes>

</test>

</suite>

Output:

[RemoteTestNG] detected TestNG version 6.14.2

Data provider for Cat

Current DataProvider class = class TestNG.DataProviderFactory

Current DataProvider = dp1

Inside Cat method and id = 11

Inside Cat method and name = TestNG

Inside Cat method and id = 12

Inside Cat method and name = is

Inside Cat method and id = 13

Inside Cat method and name = easy

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

Suite1

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

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

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

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

<suite name=*"Suite1"* thread-count=*"5"*>

<listeners>

<listener

class-name=*"TestNG.TestAnnotationTransformerListener2"*></listener>

</listeners>

<test name=*"Test1"*>

<classes>

<class name=*"TestNG.TestAnnotationTransformerExample"*></class>

</classes>

</test>

</suite>

Output

[RemoteTestNG] detected TestNG version 6.14.2

Data provider for Cat

Current DataProvider class = class TestNG.DataProviderFactory

Current DataProvider = dp1

Inside Cat method and id = 21

Inside Cat method and name = How's

Inside Cat method and id = 22

Inside Cat method and name = Your

Inside Cat method and id = 23

Inside Cat method and name = Day

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

Suite1

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

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

**IAnnotationTransformer**

**package** TestNG;

**import** java.lang.reflect.Constructor;

**import** java.lang.reflect.Method;

**import** java.util.ArrayList;

**import** org.testng.IAnnotationTransformer2;

**import** org.testng.annotations.IConfigurationAnnotation;

**import** org.testng.annotations.IDataProviderAnnotation;

**import** org.testng.annotations.IFactoryAnnotation;

**import** org.testng.annotations.ITestAnnotation;

**public** **class** IAnnotationTransformerListenerExample **implements** IAnnotationTransformer2 {

@Override

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

// **TODO** Auto-generated method stub

}

@Override

**public** **void** transform(IConfigurationAnnotation annotation, Class testClass, Constructor testConstructor,

Method testMethod) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** transform(IDataProviderAnnotation annotation, Method method) {

**if** (annotation.getName().equals("DataProvider1")) {

System.***out***.println("Class = " + annotation.getClass());

System.***out***.println("Name = " + annotation.getName());

System.***out***.println("Indices = " + annotation.getIndices());

System.***out***.println("Is parallel " + annotation.isParallel());

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

ArrayList<Integer> L1 = **new** ArrayList<Integer>();

L1.add(Integer.*valueOf*(0));

L1.add(Integer.*valueOf*(1));

annotation.setIndices(L1);

// annotation.setName("DataProvider2"); // This is not working and giving error

annotation.setParallel(**false**);

System.***out***.println("After setting ");

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

System.***out***.println("Class = " + annotation.getClass());

System.***out***.println("Name = " + annotation.getName());

System.***out***.println("Indices = " + annotation.getIndices());

System.***out***.println("Is parallel " + annotation.isParallel());

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

}

}

@Override

**public** **void** transform(IFactoryAnnotation annotation, Method method) {

// **TODO** Auto-generated method stub

}

}

**package** TestNG;

**import** org.testng.annotations.DataProvider;

**public** **class** DataProviderFactory {

@DataProvider(name="DataProvider1",indices= {1,2},parallel=**true**)

**public** Object[][] dp1() {

Object o1[][] = **new** Object[3][2];

o1[0][0] = 1;

o1[0][1] = "Jigar";

o1[1][0] = 2;

o1[1][1] = "Mehta";

o1[2][0] = 3;

o1[2][1] = "Learning";

**return** o1;

}

@DataProvider(name="DataProvider2")

**public** Object[][] dp2() {

Object o1[][] = **new** Object[3][2];

o1[0][0] = 11;

o1[0][1] = "TestNG";

o1[1][0] = 12;

o1[1][1] = "is";

o1[2][0] = 13;

o1[2][1] = "easy";

**return** o1;

}

@DataProvider(name="DataProvider3")

**public** Object[][] dp3() {

Object o1[][] = **new** Object[3][2];

o1[0][0] = 21;

o1[0][1] = "How's";

o1[1][0] = 22;

o1[1][1] = "Your";

o1[2][0] = 23;

o1[2][1] = "Day";

**return** o1;

}

@DataProvider(name="get\_Constructor\_Parameter")

**public** Object[][] get\_Constructor\_Parameter() {

**return** **new** Object[][] {{"a"},{"b"},{"c"}};

}

}

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

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

<suite name=*"Suite1"* thread-count=*"5"*>

<listeners>

<listener class-name=*"TestNG.IAnnotationTransformerListenerExample"*></listener>

</listeners>

<test name=*"Test1"*>

<classes>

<class name=*"TestNG.LogoutTest"*>

<methods>

<include name=*"Test\_Logout1"*></include>

</methods>

</class>

</classes>

</test>

</suite>

Output:

[RemoteTestNG] detected TestNG version 6.14.2

Class = class org.testng.internal.annotations.DataProviderAnnotation

Name = DataProvider1

Indices = [1, 2]

Is parallel true

After setting

Class = class org.testng.internal.annotations.DataProviderAnnotation

Name = DataProvider1

Indices = [0, 1]

Is parallel false

id = 1

Password = Jigar

Inside Test\_Logout1

id = 2

Password = Mehta

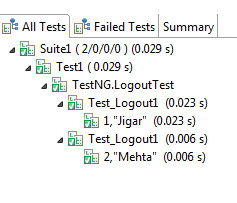
Inside Test\_Logout1

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

Suite1

Total tests run: 2, Failures: 0, Skips: 0

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



**ISuiteListener**

We also have a listener for the suite called ISuiteListener. It has two methods, onStart and onFinish. Method onStart is invoked before TestNG starts running the suite and onFinish is invoked after TestNG has run the suite.

The listener is called for each suite, if the parent suite contains child suites then the child suites are first run before running the parent suite. This is done so that the results for parent suite can reflect the combined results of the child suites.

**Files used:** LogoutTest.java, ISuiteListener.java, testng2.xml, testng3.xml, testng4.xml

**package** TestNG;

**import** org.testng.annotations.AfterSuite;

**import** org.testng.annotations.BeforeSuite;

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

**public** **class** LogoutTest {

@BeforeSuite

**public** **void** before\_suite() {

System.***out***.println("I am in @BeforeSuite");

}

@AfterSuite

**public** **void** after\_suite() {

System.***out***.println("I am in @AfterSuite");

}

@Test(dataProvider = "DataProvider1", dataProviderClass = DataProviderFactory.**class**, suiteName = "hellosuite", testName = "Mishika", description = "description", priority = 3)

**public** **void** Test\_Logout1(**int** id, String password) {

System.***out***.println("id = " + id);

System.***out***.println("Password = " + password);

System.***out***.println("Inside Test\_Logout1");

}

@Test(dataProvider = "DataProvider1", dataProviderClass = DataProviderFactory.**class**)

**public** **void** Test\_Logout2(**int** id, String password) {

System.***out***.println("id = " + id);

System.***out***.println("Password = " + password);

System.***out***.println("Inside Test\_Logout2");

}

}

**package** TestNG;

**import** java.util.Collection;

**import** java.util.Iterator;

**import** java.util.List;

**import** java.util.Map;

**import** java.util.Set;

**import** org.testng.IInvokedMethod;

**import** org.testng.ISuite;

**import** org.testng.ISuiteListener;

**import** org.testng.ITestNGMethod;

**import** org.testng.xml.XmlSuite;

**import** org.testng.xml.XmlTest;

**public** **class** ISuiteListenerExample **implements** ISuiteListener {

@Override

**public** **void** onStart(ISuite suite) {

System.***out***.println("onStart ISuite");

// the total number of methods found in this suite.

// The presence of factories or data providers might cause the actual number of test methods run be bigger than this list.

List<ITestNGMethod> all\_methods = suite.getAllMethods();

**for** (**int** i = 0; i < all\_methods.size(); i++) {

System.***out***.println("Method name = " + all\_methods.get(i).getMethodName());

}

// prints the name of this suite.

System.***out***.println("Suite name = " + suite.getName());

// All the methods that were not included in this test run.

Collection<ITestNGMethod> excludedmethod\_list = suite.getExcludedMethods();

System.***out***.println("excludedmethod\_list size = " + excludedmethod\_list.size());

**for** (Iterator iterator = excludedmethod\_list.iterator(); iterator.hasNext();) {

ITestNGMethod iTestNGMethod = (ITestNGMethod) iterator.next();

System.***out***.println("Excluded method = " + iTestNGMethod.getMethodName());

}

suite.setAttribute("junit", "hello...");

Set<String> attributes\_name = suite.getAttributeNames(); // this only works when an attribute is set using

// setAttribute method

System.***out***.println("attributes\_name size = " + attributes\_name.size());

System.***out***.println("junit = " + suite.getAttribute("junit"));

Iterator<String> attributes\_name\_ite = attributes\_name.iterator();

**while** (attributes\_name\_ite.hasNext()) {

String string = (String) attributes\_name\_ite.next();

System.***out***.println("Attribute name = " + string);

}

// The representation of the current XML suite file.

XmlSuite xmlsuite = suite.getXmlSuite();

XmlSuite Parent\_suite = xmlsuite.getParentSuite();

System.***out***.println("Parent suite = " + Parent\_suite);

**try** {

**if** (Parent\_suite.getName().equals("Main-Suite")) {

List<XmlSuite> child\_suitefiles = Parent\_suite.getChildSuites();

System.***out***.println("child\_suitefiles.size() = " + child\_suitefiles.size());

**for** (**int** i = 0; i < child\_suitefiles.size(); i++) {

System.***out***.println("child\_Suitefiles = " + child\_suitefiles.get(i).toString());

}

}

} **catch** (Exception e) {

System.***out***.println("Exception ");

}

// Returns the tests.

List<XmlTest> tests = xmlsuite.getTests();

System.***out***.println("Tests size = " + tests.size());

**for** (**int** i = 0; i < tests.size(); i++) {

System.***out***.println("Test name = " + tests.get(i).getName());

System.***out***.println("Test expression = " + tests.get(i).getExpression());

System.***out***.println("Test index = " + tests.get(i).getIndex());

System.***out***.println("Test parameter Gender = " + tests.get(i).getParameter("Gender"));

System.***out***.println("Test parameter Status = " + tests.get(i).getParameter("Status"));

System.***out***.println("Test threadcount = " + tests.get(i).getThreadCount());

System.***out***.println("Test verbose = " + tests.get(i).getVerbose());

System.***out***.println("Test AllowReturValues = " + tests.get(i).getAllowReturnValues());

}

Map<String, String> xmlsuite\_map = xmlsuite.getAllParameters();

Set<String> keys = xmlsuite\_map.keySet();

System.***out***.println("Key length = " + keys.size());

System.***out***.println("Below are the keys: ");

**for** (Iterator iterator = keys.iterator(); iterator.hasNext();) {

String string = (String) iterator.next();

System.***out***.println("key = " + string);

}

// Fetch values for parameter mentioned in testng.xml

System.***out***.println("Status = " + xmlsuite\_map.getOrDefault("Status", "not present"));

System.***out***.println("Status1 = " + xmlsuite\_map.getOrDefault("Status1", "not present"));

System.***out***.println("Course = " + xmlsuite\_map.getOrDefault("Course", "not present"));

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

}

@Override

**public** **void** onFinish(ISuite suite) {

System.***out***.println("onFinish ISuite");

List<IInvokedMethod> all\_invoked\_methods = suite.getAllInvokedMethods();

System.***out***.println("All invoked methods size = " + all\_invoked\_methods.size());

**for** (**int** i = 0; i < all\_invoked\_methods.size(); i++) {

System.***out***.println("All methods = " + all\_invoked\_methods.get(i));

}

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

}

}

testng2.xml

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

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

<suite name=*"Main-Suite"* thread-count=*"5"* parallel=*"none"* verbose=*"2"*>

<listeners>

<listener class-name=*"TestNG.ISuiteListenerExample"*>

</listener>

</listeners>

<parameter name=*"Status"* value=*"Maintestng"*></parameter>

<suite-files>

<suite-file path=*"testng3.xml"*></suite-file>

<suite-file path=*"testng4.xml"*></suite-file>

</suite-files>

</suite>

testng3.xml

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

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

<suite name=*"Suite3"* parallel=*"true"* junit=*"false"*>

<test name=*"Test1"* enabled=*"true"* junit=*"false"* verbose=*"4"*>

<parameter name=*"Gender"* value=*"Male"*></parameter>

<parameter name=*"Age"* value=*"30"*></parameter>

<parameter name=*"Status"* value=*"30"*></parameter>

<classes>

<class name=*"TestNG.LogoutTest"*>

<methods>

<include name=*"Test\_Logout1"*></include>

</methods>

</class>

</classes>

</test>

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

testng4.xml

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

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

<suite name=*"Suite4"* verbose=*"1"*>

<test name=*"Test2"* enabled=*"true"* junit=*"false"* verbose=*"4"*>

<parameter name=*"Gender"* value=*"Female"*></parameter>

<parameter name=*"Age"* value=*"300"*></parameter>

<parameter name=*"Status"* value=*"300"*></parameter>

<classes>

<class name=*"TestNG.LogoutTest"*>

<methods>

<include name=*"Test\_Logout2"*></include>

</methods>

</class>

</classes>

</test>

</suite>

Output:

[RemoteTestNG] detected TestNG version 6.14.2

[TestNG] [WARN] Ignoring duplicate listener : TestNG.ISuiteListenerExample

[TestNG] [WARN] Ignoring duplicate listener : TestNG.ISuiteListenerExample

...

... TestNG 6.14.2 by Cédric Beust (cedric@beust.com)

...

onStart ISuite

Method name = Test\_Logout1

Suite name = Suite3

excludedmethod\_list size = 1

Excluded method = Test\_Logout2

attributes\_name size = 1

junit = hello...

Attribute name = junit

Parent suite = [Suite: "Main-Suite" ]

child\_suitefiles.size() = 2

child\_Suitefiles = [Suite: "Suite3" org.testng.xml.XmlTest@58d1841 ]

child\_Suitefiles = [Suite: "Suite4" org.testng.xml.XmlTest@73f2e422 ]

Tests size = 1

Test name = Test1

Test expression = null

Test index = 0

Test parameter Gender = Male

Test parameter Status = 30

Test threadcount = 5

Test verbose = 4

Test AllowReturValues = false

Key length = 3

Below are the keys:

key = Status

key = Gender

key = Age

Status = 30

Status1 = not present

Course = not present

I am in @BeforeSuite

id = 2

Password = Mehta

Inside Test\_Logout1

id = 3

Password = Learning

Inside Test\_Logout1

===== Invoked methods

LogoutTest.before\_suite()[pri:0, instance:TestNG.LogoutTest@394df057] 961409111

LogoutTest.Test\_Logout1(int, java.lang.String)[pri:3, instance:TestNG.LogoutTest@394df057]2 Mehta 961409111

LogoutTest.Test\_Logout1(int, java.lang.String)[pri:3, instance:TestNG.LogoutTest@394df057]3 Learning 961409111

=====

PASSED: Test\_Logout1(2, "Mehta")

description

PASSED: Test\_Logout1(3, "Learning")

description

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

Test1

Tests run: 2, Failures: 0, Skips: 0

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

I am in @AfterSuite

onFinish ISuite

All invoked methods size = 4

All methods = LogoutTest.before\_suite()[pri:0, instance:TestNG.LogoutTest@394df057] 961409111

All methods = LogoutTest.Test\_Logout1(int, java.lang.String)[pri:3, instance:TestNG.LogoutTest@394df057]2 Mehta 961409111

All methods = LogoutTest.Test\_Logout1(int, java.lang.String)[pri:3, instance:TestNG.LogoutTest@394df057]3 Learning 961409111

All methods = LogoutTest.after\_suite()[pri:0, instance:TestNG.LogoutTest@394df057] 961409111

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

Suite3

Total tests run: 2, Failures: 0, Skips: 0

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

onStart ISuite

Method name = Test\_Logout2

Suite name = Suite4

excludedmethod\_list size = 1

Excluded method = Test\_Logout1

attributes\_name size = 1

junit = hello...

Attribute name = junit

Parent suite = [Suite: "Main-Suite" ]

child\_suitefiles.size() = 2

child\_Suitefiles = [Suite: "Suite3" org.testng.xml.XmlTest@58d1841 ]

child\_Suitefiles = [Suite: "Suite4" org.testng.xml.XmlTest@73f2e422 ]

Tests size = 1

Test name = Test2

Test expression = null

Test index = 0

Test parameter Gender = Female

Test parameter Status = 300

Test threadcount = 5

Test verbose = 4

Test AllowReturValues = false

Key length = 3

Below are the keys:

key = Status

key = Gender

key = Age

Status = 300

Status1 = not present

Course = not present

I am in @BeforeSuite

id = 2

Password = Mehta

Inside Test\_Logout2

id = 3

Password = Learning

Inside Test\_Logout2

===== Invoked methods

LogoutTest.before\_suite()[pri:0, instance:TestNG.LogoutTest@68e5eea7] 1759899303

LogoutTest.Test\_Logout2(int, java.lang.String)[pri:0, instance:TestNG.LogoutTest@68e5eea7]2 Mehta 1759899303

LogoutTest.Test\_Logout2(int, java.lang.String)[pri:0, instance:TestNG.LogoutTest@68e5eea7]3 Learning 1759899303

=====

PASSED: Test\_Logout2(2, "Mehta")

PASSED: Test\_Logout2(3, "Learning")

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

Test2

Tests run: 2, Failures: 0, Skips: 0

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

I am in @AfterSuite

onFinish ISuite

All invoked methods size = 4

All methods = LogoutTest.before\_suite()[pri:0, instance:TestNG.LogoutTest@68e5eea7] 1759899303

All methods = LogoutTest.Test\_Logout2(int, java.lang.String)[pri:0, instance:TestNG.LogoutTest@68e5eea7]2 Mehta 1759899303

All methods = LogoutTest.Test\_Logout2(int, java.lang.String)[pri:0, instance:TestNG.LogoutTest@68e5eea7]3 Learning 1759899303

All methods = LogoutTest.after\_suite()[pri:0, instance:TestNG.LogoutTest@68e5eea7] 1759899303

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

Suite4

Total tests run: 2, Failures: 0, Skips: 0

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

onStart ISuite

Suite name = Main-Suite

excludedmethod\_list size = 0

attributes\_name size = 1

junit = hello...

Attribute name = junit

Parent suite = null

Exception

Tests size = 0

Key length = 1

Below are the keys:

key = Status

Status = Maintestng

Status1 = not present

Course = not present

onFinish ISuite

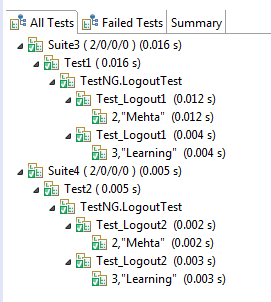
All invoked methods size = 0

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

Main-Suite

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

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



**ITestListener**

ITestListener is the listener for test running. You can either implement ITestListener or extend the TestNG provided implementation TestListenerAdapter as it has many convenient methods and we don’t have to re-invent the wheel.

ITestListener has methods on following events:

* onStart is invoked after the test class is instantiated and before any configuration method is called
* onTestSuccess is invoked on success of a test
* onTestFailure is invoked on failure of a test
* onTestSkipped is invoked whenever a test is skipped
* onTestFailedButWithinSuccessPercentage is invoked each time a method fails but is within the success percentage requested.
* onFinish is invoked after all the tests have run and all their Configuration methods have been called.

Classes used: LoginTest.java, ITestListenerExample.java, testng1.xml

**package** TestNG;

**import** org.testng.Assert;

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

**public** **class** LoginTest {

@Test()

**public** **void** LoginTest1() {

System.***out***.println("Inside Logintest1");

}

@Test

**public** **void** LoginTest2() {

System.***out***.println("Inside Logintest2");

Assert.*assertEquals*("X", "Y");

}

@Test(dependsOnMethods = { "LoginTest2" })

**public** **void** LoginTest3() {

System.***out***.println("Inside Logintest3");

}

**private** **int** i = 0;

@Test(invocationCount = 2, successPercentage = 50)

**public** **void** LoginTest4() {

System.***out***.println("Inside Logintest4");

System.***out***.println("i = " + i);

i++;

**if** (i == 2) {

Assert.*assertEquals*(i, 10);

}

}

}

**package** TestNG;

**import** org.testng.ITestContext;

**import** org.testng.ITestListener;

**import** org.testng.ITestResult;

**public** **class** ITestListenerExample **implements** ITestListener {

@Override

**public** **void** onTestStart(ITestResult result) {

System.***out***.println("Method name = " + result.getName() + " onTestStart");

}

@Override

**public** **void** onTestSuccess(ITestResult result) {

System.***out***.println("Method name = " + result.getName() + " onTestSuccess");

}

@Override

**public** **void** onTestFailure(ITestResult result) {

System.***out***.println("Method name = " + result.getName() + " onTestFailure");

}

@Override

**public** **void** onTestSkipped(ITestResult result) {

System.***out***.println("Method name = " + result.getName() + " onTestSkipped");

}

@Override

**public** **void** onTestFailedButWithinSuccessPercentage(ITestResult result) {

System.***out***.println("Method name = " + result.getName() + " onTestFailedButWithinSuccessPercentage");

}

@Override

**public** **void** onStart(ITestContext context) {

System.***out***.println("Context name = " + context.getName() + " onStart");

}

@Override

**public** **void** onFinish(ITestContext context) {

System.***out***.println("Context name = " + context.getName() + " onFinish");

}

}

Testnt1.xml

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

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

<suite name=*"Suite1"* parallel=*"methods"*>

<listeners>

<listener class-name=*"TestNG.ITestListenerExample"*></listener>

</listeners>

<test name=*"Test\_12"*>

<classes>

<class name=*"TestNG.LoginTest"*>

<methods>

<include name=*"LoginTest1"*></include>

</methods>

</class>

</classes>

</test>

<test name=*"Test\_13"*>

<classes>

<class name=*"TestNG.LoginTest"*>

<methods>

<include name=*"LoginTest2"*></include>

<include name=*"LoginTest3"*></include>

<include name=*"LoginTest4"*></include>

</methods>

</class>

</classes>

</test>

</suite>

**IConfigurationListener**

IIConfigurationListener is the listener interface for events related to configuration methods e.g. @BeforSuite, @AfterSuite etc

LoginTest.java, IConfigurationListener2Example.java, testing.xml

**Explanation:**

Failed @Before Suite hence all the remaining methods fails. onConfigurationFailure and onConfigurationSkip methods are invoked.

**package** TestNG;

**import** org.testng.Assert;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.BeforeSuite;

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

**public** **class** LoginTest {

@BeforeSuite()

**public** **void** beforesuite() {

System.***out***.println("Beforesuite method");

Assert.*assertEquals*("X", "Y");

}

@BeforeMethod

**public** **void** beforMethod() {

}

@Test()

**public** **void** LoginTest1() {

System.***out***.println("Inside Logintest1");

}

@Test

**public** **void** LoginTest2() {

System.***out***.println("Inside Logintest2");

Assert.*assertEquals*("X", "Y");

}

@Test(dependsOnMethods = { "LoginTest2" })

**public** **void** LoginTest3() {

System.***out***.println("Inside Logintest3");

}

**private** **int** i = 0;

@Test(invocationCount = 2, successPercentage = 50)

**public** **void** LoginTest4() {

System.***out***.println("Inside Logintest4");

System.***out***.println("i = " + i);

i++;

**if** (i == 2) {

Assert.*assertEquals*(i, 10);

}

}

}

**package** TestNG;

**import** org.testng.IConfigurationListener2;

**import** org.testng.ITestResult;

**public** **class** IConfigurationListener2Example **implements** IConfigurationListener2{

@Override

**public** **void** onConfigurationSuccess(ITestResult itr) {

System.***out***.println("inside onConfigurationSuccess");

}

@Override

**public** **void** onConfigurationFailure(ITestResult itr) {

System.***out***.println("inside onConfigurationFailure");

}

@Override

**public** **void** onConfigurationSkip(ITestResult itr) {

System.***out***.println("inside onConfigurationSkip");

}

@Override

**public** **void** beforeConfiguration(ITestResult tr) {

System.***out***.println("inside beforeConfiguration");

}

}

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

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

<suite name=*"Suite1"* parallel=*"methods"*>

<listeners>

<listener class-name=*"TestNG.IConfigurationListener2Example"*></listener>

</listeners>

<test name=*"Test\_12"*>

<classes>

<class name=*"TestNG.LoginTest"*>

<methods>

<include name=*"LoginTest1"*></include>

</methods>

</class>

</classes>

</test>

<test name=*"Test\_13"*>

<classes>

<class name=*"TestNG.LoginTest"*>

<methods>

<include name=*"LoginTest2"*></include>

<include name=*"LoginTest3"*></include>

<include name=*"LoginTest4"*></include>

</methods>

</class>

</classes>

</test>

</suite>

**Output:**

[RemoteTestNG] detected TestNG version 6.14.2

inside beforeConfiguration

Beforesuite method

inside onConfigurationFailure

inside onConfigurationSkip

inside onConfigurationSkip

inside onConfigurationSkip

inside onConfigurationSkip

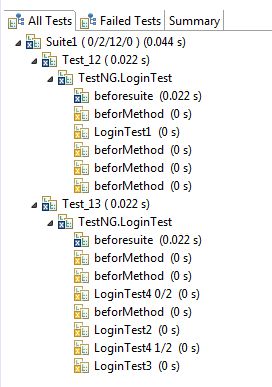
===============================================

Suite1

Total tests run: 5, Failures: 0, Skips: 5

Configuration Failures: 1, Skips: 4

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



Example of when all one test case fails and all other passes

LoginTest.java, IConfigurationListener2Example.java, testing.xml

**package** TestNG;

**import** org.testng.Assert;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.BeforeSuite;

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

**public** **class** LoginTest {

@BeforeSuite()

**public** **void** beforesuite() {

System.***out***.println("Beforesuite method");

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

}

@BeforeMethod

**public** **void** beforMethod() {

}

@Test()

**public** **void** LoginTest1() {

System.***out***.println("Inside Logintest1");

}

@Test

**public** **void** LoginTest2() {

System.***out***.println("Inside Logintest2");

Assert.*assertEquals*("X", "Y");

}

@Test(dependsOnMethods = { "LoginTest2" })

**public** **void** LoginTest3() {

System.***out***.println("Inside Logintest3");

}

**private** **int** i = 0;

@Test(invocationCount = 2, successPercentage = 50)

**public** **void** LoginTest4() {

System.***out***.println("Inside Logintest4");

System.***out***.println("i = " + i);

i++;

**if** (i == 2) {

Assert.*assertEquals*(i, 10);

}

}

}

**package** TestNG;

**import** org.testng.IConfigurationListener2;

**import** org.testng.ITestResult;

**public** **class** IConfigurationListener2Example **implements** IConfigurationListener2{

@Override

**public** **void** onConfigurationSuccess(ITestResult itr) {

System.***out***.println("inside onConfigurationSuccess = " + itr.getName());

}

@Override

**public** **void** onConfigurationFailure(ITestResult itr) {

System.***out***.println("inside onConfigurationFailure" + itr.getName());

}

@Override

**public** **void** onConfigurationSkip(ITestResult itr) {

System.***out***.println("inside onConfigurationSkip" + itr.getName());

}

@Override

**public** **void** beforeConfiguration(ITestResult tr) {

System.***out***.println("inside beforeConfiguration" + tr.getName());

}

}

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

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

<suite name=*"Suite1"* parallel=*"methods"*>

<listeners>

<listener class-name=*"TestNG.IConfigurationListener2Example"*></listener>

</listeners>

<test name=*"Test\_12"*>

<classes>

<class name=*"TestNG.LoginTest"*>

<methods>

<include name=*"LoginTest1"*></include>

</methods>

</class>

</classes>

</test>

<test name=*"Test\_13"*>

<classes>

<class name=*"TestNG.LoginTest"*>

<methods>

<include name=*"LoginTest2"*></include>

<include name=*"LoginTest3"*></include>

<include name=*"LoginTest4"*></include>

</methods>

</class>

</classes>

</test>

</suite>

Output:

[RemoteTestNG] detected TestNG version 6.14.2

inside beforeConfigurationbeforesuite

Beforesuite method

inside onConfigurationSuccess = beforesuite

inside beforeConfigurationbeforMethod

inside onConfigurationSuccess = beforMethod

Inside Logintest1

inside beforeConfigurationbeforMethod

inside onConfigurationSuccess = beforMethod

inside beforeConfigurationbeforMethod

inside onConfigurationSuccess = beforMethod

Inside Logintest2

Inside Logintest4

i = 0

inside beforeConfigurationbeforMethod

inside onConfigurationSuccess = beforMethod

Inside Logintest4

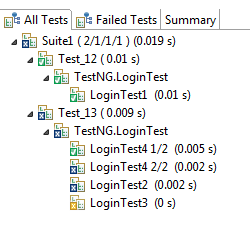
i = 1

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

Suite1

Total tests run: 5, Failures: 2, Skips: 1

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



**IMethodInterceptor**

This class is used to alter the list of test methods that TestNG is about to run.

An instance of this class will be invoked right before TestNG starts invoking test methods. Only methods that have no dependents and that don't depend on any other test methods will be passed in parameter. Implementers of this interface need to return a list of IMethodInstance that represents the list of test methods they want run. TestNG will run these methods in the same order found in the returned value.

Typically, the returned list will be just the methods passed in parameter but sorted differently, but it can actually have any size (it can be empty, it can be of the same size as the original list or it can contain more methods).

**package** TestNG;

**import** org.testng.Assert;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.BeforeSuite;

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

**public** **class** LoginTest {

@BeforeSuite()

**public** **void** beforesuite() {

System.***out***.println("Beforesuite method");

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

}

@BeforeMethod

**public** **void** beforMethod() {

}

@Test(description = "Test for Login 1", groups = "{Login group},{Logout group}")

**public** **void** LoginTest1() {

System.***out***.println("Inside Logintest1");

}

@Test

**public** **void** LoginTest2() {

System.***out***.println("Inside Logintest2");

Assert.*assertEquals*("X", "Y");

}

@Test(dependsOnMethods = { "LoginTest2" })

**public** **void** LoginTest3() {

System.***out***.println("Inside Logintest3");

}

**private** **int** i = 0;

@Test(invocationCount = 2, successPercentage = 50)

**public** **void** LoginTest4() {

System.***out***.println("Inside Logintest4");

System.***out***.println("i = " + i);

i++;

**if** (i == 2) {

Assert.*assertEquals*(i, 10);

}

}

}

**package** TestNG;

**import** java.util.List;

**import** java.util.ListIterator;

**import** org.testng.IMethodInstance;

**import** org.testng.IMethodInterceptor;

**import** org.testng.ITestContext;

**public** **class** IMethodInterceptorExample **implements** IMethodInterceptor {

@Override

**public** List<IMethodInstance> intercept(List<IMethodInstance> methods, ITestContext context) {

System.***out***.println("Method size = " + methods.size());

ListIterator<IMethodInstance> list\_methods = methods.listIterator();

**while** (list\_methods.hasNext()) {

IMethodInstance iMethodInstance = (IMethodInstance) list\_methods.next();

System.***out***.println("Method name = " + iMethodInstance.getMethod().getMethodName());

String[] group\_names = iMethodInstance.getMethod().getGroups();

**for** (**int** i = 0; i < group\_names.length; i++) {

System.***out***.println("Group name = " + group\_names[i].toString());

}

System.***out***.println("Description name = " + iMethodInstance.getMethod().getDescription());

String[] dependent\_methods = iMethodInstance.getMethod().getMethodsDependedUpon();

**for** (**int** i = 0; i < dependent\_methods.length; i++) {

System.***out***.println("Dependent method name = " + dependent\_methods[i].toString());

}

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

}

**return** methods;

}

}

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

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

<suite name=*"Suite1"* parallel=*"methods"*>

<listeners>

<listener class-name=*"TestNG.IMethodInterceptorExample"*></listener>

</listeners>

<test name=*"Test\_12"*>

<classes>

<class name=*"TestNG.LoginTest"*>

<methods>

<include name=*"LoginTest1"*></include>

</methods>

</class>

</classes>

</test>

<test name=*"Test\_13"*>

<classes>

<class name=*"TestNG.LoginTest"*>

<methods>

<include name=*"LoginTest2"*></include>

<include name=*"LoginTest3"*></include>

<include name=*"LoginTest4"*></include>

</methods>

</class>

</classes>

</test>

</suite>

Output:

[RemoteTestNG] detected TestNG version 6.14.2

Beforesuite method

Method size = 1

Method name = LoginTest1

Group name = {Login group},{Logout group}

Description name = Test for Login 1

Inside Logintest1

Method size = 3

Method name = LoginTest2

Description name = null

Method name = LoginTest3

Description name = null

Dependent method name = TestNG.LoginTest.LoginTest2

Method name = LoginTest4

Description name = null

Inside Logintest4

i = 0

Inside Logintest2

Inside Logintest4

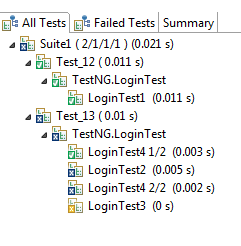
i = 1

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

Suite1

Total tests run: 5, Failures: 2, Skips: 1

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



In the below example we are using IMethodInterceptor and ISuiteListener

**package** TestNG;

**import** org.testng.Assert;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.BeforeSuite;

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

**public** **class** LoginTest {

@BeforeSuite()

**public** **void** beforesuite() {

System.***out***.println("Beforesuite method");

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

}

@BeforeMethod

**public** **void** beforMethod() {

}

@Test(description = "Test for Login 1", groups = "{Login group},{Logout group}")

**public** **void** LoginTest1() {

System.***out***.println("Inside Logintest1");

}

@Test

**public** **void** LoginTest2() {

System.***out***.println("Inside Logintest2");

Assert.*assertEquals*("X", "Y");

}

@Test()

**public** **void** LoginTest3() {

System.***out***.println("Inside Logintest3");

}

**private** **int** i = 0;

@Test(invocationCount = 2, successPercentage = 50)

**public** **void** LoginTest4() {

System.***out***.println("Inside Logintest4");

// System.out.println("i = " + i);

i++;

**if** (i == 2) {

Assert.*assertEquals*(i, 10);

}

}

}

**package** TestNG;

**import** java.util.Iterator;

**import** java.util.List;

**import** java.util.ListIterator;

**import** org.testng.IMethodInstance;

**import** org.testng.IMethodInterceptor;

**import** org.testng.ISuite;

**import** org.testng.ISuiteListener;

**import** org.testng.ITestContext;

**import** org.testng.xml.XmlSuite;

**import** org.testng.xml.XmlTest;

**public** **class** IMethodInterceptorExample **implements** IMethodInterceptor, ISuiteListener {

@Override

**public** List<IMethodInstance> intercept(List<IMethodInstance> methods, ITestContext context) {

System.***out***.println("Method size = " + methods.size());

ListIterator<IMethodInstance> list\_methods = methods.listIterator();

**while** (list\_methods.hasNext()) {

IMethodInstance iMethodInstance = (IMethodInstance) list\_methods.next();

System.***out***.println("Method name = " + iMethodInstance.getMethod().getMethodName());

System.***out***.println("XmlTest Name = " + iMethodInstance.getMethod().getXmlTest().getName());

System.***out***.println("iMethodInstance.getInstance() = " + iMethodInstance.getInstance());

String[] group\_names = iMethodInstance.getMethod().getGroups();

}

// methods.remove(0);

**for** (**int** i = 0; i < methods.size(); i++) {

System.***out***.println("methods.get(i) = " + methods.get(i));

**if** (methods.get(i).getMethod().getMethodName().equals("LoginTest2")) {

methods.remove(methods.get(i));

}

}

**return** methods;

}

@Override

**public** **void** onStart(ISuite suite) {

XmlSuite xml\_suite = suite.getXmlSuite();

System.***out***.println("inside onStart of ISuite");

List<XmlTest> all\_tests = xml\_suite.getTests();

**for** (**int** i = 0; i < all\_tests.size(); i++) {

System.***out***.println("all\_tests.get(i).getName() = " + all\_tests.get(i).getName());

}

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

}

@Override

**public** **void** onFinish(ISuite suite) {

// **TODO** Auto-generated method stub

}

}

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

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

<suite name=*"Suite1"* parallel=*"methods"*>

<listeners>

<listener class-name=*"TestNG.IMethodInterceptorExample"*></listener>

</listeners>

<test name=*"Test\_12"*>

<classes>

<class name=*"TestNG.LoginTest"*>

<methods>

<include name=*"LoginTest1"*></include>

</methods>

</class>

</classes>

</test>

<test name=*"Test\_13"*>

<classes>

<class name=*"TestNG.LoginTest"*>

<methods>

<include name=*"LoginTest2"*></include>

<include name=*"LoginTest3"*></include>

<include name=*"LoginTest4"*></include>

</methods>

</class>

</classes>

</test>

</suite>

Output:

[RemoteTestNG] detected TestNG version 6.14.2

inside onStart of ISuite

all\_tests.get(i).getName() = Test\_12

all\_tests.get(i).getName() = Test\_13

Beforesuite method

Method size = 1

Method name = LoginTest1

XmlTest Name = Test\_12

iMethodInstance.getInstance() = TestNG.LoginTest@1e66f1f5

methods.get(i) = [MethodInstance method=LoginTest.LoginTest1()[pri:0, instance:TestNG.LoginTest@1e66f1f5] instance=TestNG.LoginTest@1e66f1f5]

Inside Logintest1

Method size = 3

Method name = LoginTest2

XmlTest Name = Test\_13

iMethodInstance.getInstance() = TestNG.LoginTest@5c90e579

Method name = LoginTest3

XmlTest Name = Test\_13

iMethodInstance.getInstance() = TestNG.LoginTest@5c90e579

Method name = LoginTest4

XmlTest Name = Test\_13

iMethodInstance.getInstance() = TestNG.LoginTest@5c90e579

methods.get(i) = [MethodInstance method=LoginTest.LoginTest2()[pri:0, instance:TestNG.LoginTest@5c90e579] instance=TestNG.LoginTest@5c90e579]

methods.get(i) = [MethodInstance method=LoginTest.LoginTest4()[pri:0, instance:TestNG.LoginTest@5c90e579] instance=TestNG.LoginTest@5c90e579]

Inside Logintest4

Inside Logintest3

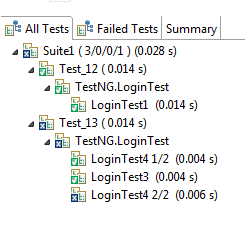
Inside Logintest4

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

Suite1

Total tests run: 4, Failures: 1, Skips: 0

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



**IInvokedMethodListener**

IInvokedMethodListener is listener that gets invoked before and after a method is invoked by TestNG. It will be invoked for all methods, both test and the configuration methods.

**package** TestNG;

**import** org.testng.Assert;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.BeforeSuite;

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

**public** **class** LoginTest {

@BeforeSuite()

**public** **void** beforesuite() {

System.***out***.println("Beforesuite method");

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

}

@BeforeMethod

**public** **void** beforMethod() {

System.***out***.println("beforMethod");

}

@Test(description = "Test for Login 1", groups = "{Login group},{Logout group}")

**public** **void** LoginTest1() {

System.***out***.println("Inside Logintest1");

}

@Test

**public** **void** LoginTest2() {

System.***out***.println("Inside Logintest2");

Assert.*assertEquals*("X", "Y");

}

@Test()

**public** **void** LoginTest3() {

System.***out***.println("Inside Logintest3");

}

**private** **int** i = 0;

@Test(invocationCount = 2, successPercentage = 50)

**public** **void** LoginTest4() {

System.***out***.println("Inside Logintest4");

// System.out.println("i = " + i);

i++;

**if** (i == 2) {

Assert.*assertEquals*(i, 10);

}

}

}

**package** TestNG;

**import** org.testng.IInvokedMethod;

**import** org.testng.IInvokedMethodListener;

**import** org.testng.ITestResult;

**public** **class** IInvokedMethodListenerExample **implements** IInvokedMethodListener{

@Override

**public** **void** beforeInvocation(IInvokedMethod method, ITestResult testResult) {

System.***out***.println("beforeInvocation");

}

@Override

**public** **void** afterInvocation(IInvokedMethod method, ITestResult testResult) {

System.***out***.println("afterInvocation");

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

}

}

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

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

<suite name=*"Suite1"* parallel=*"methods"*>

<listeners>

<listener class-name=*"TestNG.IInvokedMethodListenerExample"*></listener>

</listeners>

<test name=*"Test\_12"*>

<classes>

<class name=*"TestNG.LoginTest"*>

<methods>

<include name=*"LoginTest1"*></include>

</methods>

</class>

</classes>

</test>

<test name=*"Test\_13"*>

<classes>

<class name=*"TestNG.LoginTest"*>

<methods>

<include name=*"LoginTest2"*></include>

<include name=*"LoginTest3"*></include>

<include name=*"LoginTest4"*></include>

</methods>

</class>

</classes>

</test>

</suite>

Output:

[RemoteTestNG] detected TestNG version 6.14.2

beforeInvocation

Beforesuite method

afterInvocation

beforeInvocation

beforMethod

afterInvocation

beforeInvocation

Inside Logintest1

afterInvocation

beforeInvocation

beforMethod

afterInvocation

beforeInvocation

beforMethod

afterInvocation

beforeInvocation

Inside Logintest2

afterInvocation

beforeInvocation

beforMethod

afterInvocation

beforeInvocation

Inside Logintest4

afterInvocation

beforeInvocation

Inside Logintest3

afterInvocation

beforeInvocation

beforMethod

afterInvocation

beforeInvocation

Inside Logintest4

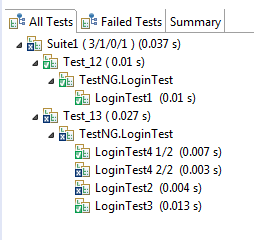
afterInvocation

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

Suite1

Total tests run: 5, Failures: 2, Skips: 0

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



Define Listeners at class level

**package** TestNG;

**import** org.testng.Assert;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.BeforeSuite;

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

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

@Listeners({TestNG.IInvokedMethodListenerExample.**class**})

**public** **class** LoginTest {

@BeforeSuite()

**public** **void** beforesuite() {

System.***out***.println("Beforesuite method");

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

}

@BeforeMethod

**public** **void** beforMethod() {

System.***out***.println("beforMethod");

}

@Test(description = "Test for Login 1", groups = "{Login group},{Logout group}")

**public** **void** LoginTest1() {

System.***out***.println("Inside Logintest1");

}

@Test

**public** **void** LoginTest2() {

System.***out***.println("Inside Logintest2");

Assert.*assertEquals*("X", "Y");

}

@Test()

**public** **void** LoginTest3() {

System.***out***.println("Inside Logintest3");

}

**private** **int** i = 0;

@Test(invocationCount = 2, successPercentage = 50)

**public** **void** LoginTest4() {

System.***out***.println("Inside Logintest4");

// System.out.println("i = " + i);

i++;

**if** (i == 2) {

Assert.*assertEquals*(i, 10);

}

}

}

**package** TestNG;

**import** org.testng.IInvokedMethod;

**import** org.testng.IInvokedMethodListener;

**import** org.testng.ITestResult;

**public** **class** IInvokedMethodListenerExample **implements** IInvokedMethodListener{

@Override

**public** **void** beforeInvocation(IInvokedMethod method, ITestResult testResult) {

System.***out***.println("beforeInvocation");

}

@Override

**public** **void** afterInvocation(IInvokedMethod method, ITestResult testResult) {

System.***out***.println("afterInvocation");

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

}

}

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

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

<suite name=*"Suite1"* parallel=*"methods"*>

<!-- <listeners>

<listener class-name="TestNG.IInvokedMethodListenerExample"></listener>

</listeners> -->

<test name=*"Test\_12"*>

<classes>

<class name=*"TestNG.LoginTest"*>

<methods>

<include name=*"LoginTest1"*></include>

</methods>

</class>

</classes>

</test>

<test name=*"Test\_13"*>

<classes>

<class name=*"TestNG.LoginTest"*>

<methods>

<include name=*"LoginTest2"*></include>

<include name=*"LoginTest3"*></include>

<include name=*"LoginTest4"*></include>

</methods>

</class>

</classes>

</test>

</suite>

Output:

[RemoteTestNG] detected TestNG version 6.14.2

beforeInvocation

Beforesuite method

afterInvocation

beforeInvocation

beforMethod

afterInvocation

beforeInvocation

Inside Logintest1

afterInvocation

beforeInvocation

beforMethod

afterInvocation

beforeInvocation

Inside Logintest2

afterInvocation

beforeInvocation

beforeInvocation

beforMethod

afterInvocation

beforeInvocation

beforMethod

afterInvocation

Inside Logintest3

afterInvocation

beforeInvocation

Inside Logintest4

afterInvocation

beforeInvocation

beforMethod

afterInvocation

beforeInvocation

Inside Logintest4

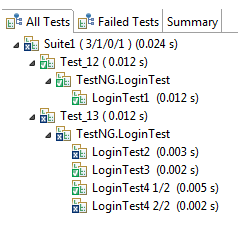
afterInvocation

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

Suite1

Total tests run: 5, Failures: 2, Skips: 0

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



* IHookable (This is homework)
* IReporter (This is homework)