

TestNG Document

March 2016

Week 12

Day 077 - 289

Date 17 - 03 - 2016

17

Thursday

→ a java class which contains at least one testNG annotⁿ is called TestNG class.

Configurⁿ Annotⁿ: annotated methods are called configuration method.

A testNG test can be configured by

① BeforeXXX XXX: Suite, Test, Class, Method

② AfterXXX (to) Groups

↳ These annotⁿ will also be honoured (inherited) when placed on a superclass of a testNG class.

↳ useful eg: to centralize test set up for multiple test classes in a common superclass

↳ TestNG guarantees that ① Before methods are executed in inheritance order (highest superclass first, then going down the inheritance chain) and ② After method in reverse order.

Notes: Attributes for Above annotⁿ

1) alwaysRun → used for All BeforeXXX except BeforeGroups
true/false → true → method will run regardless of what groups it belongs to
↳ default
if used for AfterXXX true: method will be run even if one or more methods invoked previously failed or was skipped

18

Friday

March 2016

Week 12

Day 078 • 288

Date 18 • 03 • 2016

	M	T	W	T	F	S	S
M		1	2	3	4	5	6
A	7	8	9	10	11	12	13
R	14	15	16	17	18	19	20
Tu	21	22	23	24	25	26	27
	28	29	30	31			

2. dependsOnGroups - the list of groups this method depends on.

3. dependsOnMethods - methods

4. enabled

5. groups

6. inheritGroups - if set true, this method will belong to groups specified in the @Test annotation at the class level.

@DataProvider: returns object[]
marks a method as supplying data for a test method

name → name of data provider, if its not supplied by default it would be set to methodName.

parallel → if set true tests generated using DPs are run in parallel. Default value false.

@Factory: Marks a method a factory method that returns objects that will be used by TestNG as test classes. Method must return object[]

@Listeners → defines listeners on a test class.

@Parameters → describe how to pass p^{rs} to a test method

value

M	T	W	T	F	S	S
			1	2	3	
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

March 2016

Week 12

Day 079 • 287

Date 19 • 03 • 2016

19
Saturday

② Test → Marks a class or a method as part of Test
Attributes :-

1. alwaysRun → if set true, method will always run even if it depends on a method that failed.
2. dataProvider → name of DP
3. dataProviderClass → The class where to look for DP
→ if not specified, will be checked in current class or one of its base classes.
if this attribute is specified, data provider method needs to be static on the specified class.
4. dependsOnGroups → both accepts reg^r & exprⁿ
5. dependsOnMethods → both accepts reg^r & exprⁿ
6. description
7. enabled
8. expectedExceptions → list of excep^{ns} test method is expected to throw, if no excepⁿ or diff^r than one on the list is thrown, this test will be marked a failure.
9. groups
10. priority → in millisec (100000) for 3min
11. timeOut → The max no. of milliseconds this test should take
→ work in both II & non-parallel mode.
12. invocationCount → the no. of times this method shud be invoked.
→ test will run 4 times even if it was passed in 1st invocation.
better use IRetryAnalyzer to run failed T.C.

20
Sunday

March 2016
Week 12
Day 080 + 280
Date 01/03/2016

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

17. invocationTimeout: max time in ms
for all invocⁿ (cumulative)
↓
ignored if invocⁿ count is
not given

18. threadPoolSize: size of thread pool for this method
↓
method will be invoked from
multiple threads as
specified by invocⁿ count
ignored in Invoⁿ Count is not given

19. singleThreaded → can only be used at class
level
↓
if set true, all methods on this
test class are guaranteed to
run in the same thread, even if
tests are currently being run with
parallel = "methods"
↓
earlier called sequential now
deprecated

20. SuccessPercentage
min %age of success expected from this
method to mark it as passed

@Test(invocationCount = 4, successPercentage = 98,
invocationTimeout = 180000, priority = 3)
↓
3min

F	S								
4	5	6	7	8	9	10	11	12	13
18	19	20	21	22	23	24	25	26	27

March 2016
 Week 13
 Day 081 • 285
 Date 21 • 03 • 2016

21
 Monday

Invoking testing.xml:

By classes

```
<suite name="suite2">
  <test name="test2">
    <classes>
      <class name="xyz"/>
    </classes>
  </test>
```

without java

→ instead of class name we can give pkg name as well.

By Packages

```
<test name="test2">
  <packages>
    <package name="abc.xyz"/>
  </packages>
</test>
```

only testng class will be included in the given pkg.

→ **<groups>** can be put at suite or test level.

```
<run>
  <include name="regression"/>
  <exclude name="broken"/>
</run>
```

→ To get clean report

</groups> → classes or packages has to be given after this.

16

SATURDAY

JANUARY

016-350 • WK 03

Handling pop Window (HTML page or page opened in new Tab)

package testcases;

import java.util.Iterator;

import java.util.Set;

public class HandlingPopUpWindow

{

public void run(String args[]) throws InterruptedException

{

WebDriver driver = new FirefoxDriver();

// First get All the windows opened by driver instance

// window1

driver.get("http://www.seleniumhq.org/");

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

String homePage = driver.getWindowHandle();

System.out.println("Home window ID: " + homePage);

// window2

driver.findElement(By.xpath("//*[text()='Get User ID']")).click();

// Get the IDs of all opened windows

Set<String> set = driver.getWindowHandles();

System.out.println("No. of opened windows: " + set.size());

// To go through (iterate) each element of set.

2016

is a Testing framework developed in lines of JUnit & NUnit

however it introduces some new functionalities that make it more powerful & easier to use.

JUnit short comings:

- Can not do dependency Tg
- Poor config control (setUp/tearDown)
- Intrusive: Forces you to extend class & name your methods a certain way.
- Static Programming Model: Forces you to recompile unnecessarily

NG → stands for Next Generation

TestNG is not a JUnit extension, it is inspired by JUnit

Creator of TestNG - Cedric Beust

TestNG Features:

- Support annotations
- Uses more Java & OO features
- Support tg integrated class
e.g by default no need to create a new test class instance for every test methods
- Separate compile time test code from runtime config/data
- Flexible runtime config
"front-end", "fast", "slow", "database"
- introduces "test groups"

05

SATURDAY
MARCH

1	2	3	4	5	6	7	8	9	10	11	12
13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31					
W	T	Th	F	S	S	M	T	W	T	F	S

- Supports dependent test methods
 - Parallel Tg
 - Load Tg
 - Partial Failure

→ Flexible plug-in API

→ Support for multithreading Tg

→ JDK 2.5 or above (min Java 5 is required)



JDK should be installed
on your machine to use TestNG



As Java Annotations
were introduced
with JDK 5.0

⇒ When a Java class contains at least
one `@TestNG` annotation — it becomes `TestNGClass`

TestNG can be invoked in several ways:

→ With a `testng.xml` file

→ With API

→ From the command line

→ POJO class:

Plain Old Java Object

→ similar to java bean which contains G.F.S
but java bean follow some rules

→ properties should be private

→ no-argument constructor etc

→ In JUnit's test methods were prefixed by word "test"
`testSum() { }`

TestNG Annot^{ns} $12 + 3 \Rightarrow 15$

① **Factory**: marks a method as a factory that returns object that will be used by TestNG as test cases. The method must return object [].

② **Listeners**: defines listeners on a test class.

③ **Parameters**: describes how to pass parameters to a @Test method.

④ **Test**: Marks a class or a method as a part of the test.

Benefits of testNG:

→ Test classes no need to extend anything (such as TestCase for JUnit3)

→ we can pass additional parameters to annot^{ns}

08

TUESDAY
MARCH

068-298 • WK 11

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
M	T	W	T	F	S	S	M	T	W	T	F	S	S															

TestNG Execⁿ Procedure

```
public class TestingAnnotation {
```

```
    @Test
    p.v. TC1() { sop("In TC1"); }
```

```
    @Test
    p.v. TC2() { sop("In TC2"); }
```

```
    @BeforeMethod
    p.v. beforeMeth() { sop("In before meth"); }
```

```
    @AfterMethod
    p.v. afterMeth() { sop("In after meth"); }
```

```
    @BeforeClass
    p.v. beforeClass() { sop("In before class"); }
```

```
    @AfterClass
    p.v. afterClass() { sop("In after class"); }
```

```
    @BeforeTest
    p.v. beforeTest() { sop("in before test"); }
```

```
    @AfterTest
    p.v. afterTest() { sop("in after test"); }
```

```
    @BeforeSuite
    p.v. beforeSuit() { sop("in before suit"); }
```

```
    @AfterSuite
    p.v. afterSuit() { sop("in after suit"); }
```

```
testng.xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
    <suit name="Suite1">
        <test name="test1">
            <classes>
                <class name="TestingAnnotation"/>
            </classes>
        </test>
    </suite>
```


O/p:- {
 in before suit
 in before test
 in before class
 in before method
 in test case 1
 in after method
 in before method
 in test case 2
 in after method
 in after class
 in after test
 in after suite

Java class to be tested

```
public class MessageUtil {
    private String msg;
    public MessageUtil(String msg)
    {
        this.msg = msg;
    }
    public String printMsg1 () {
        sop(msg);
        return msg;
    }
}
```

10 THURSDAY
MARCH
070-098 • WK 11

Java test (TC) class:

p. c. sampleTest {

MessageUtil mu = new MessageUtil ("Hello Deepak")

(pmsg)

@Test

p.v. testPrintMsg () {

Assert.assertEquals ("Hello Deepak", mu.printMsg())

(pmsg)

}

testng.xml

<class name = "sampleTest" />

O/p: Hello Deepak

TestNG SuiteTest

Attributes of <suite>

- 1) name → mandatory
- 2) verbose → The level or verbosity for this run
- 3) parallel → whether testing should run different threads to run this suite.
- 4) thread-count → No. of threads to use, if parallel mode is enabled
- 5) type of annotations → being used in ur test
- 6) timeout → default timeout to be used on all test methods found in this test.

Test enabled = false ⇒ In this case method will not be shown testing report ⇒ No details
 throw skipException ⇒ method will be displayed in report as **11 Skipped**
 is actually Ignore or Bypass.

```

<?xml version="1.0" ?>
<suite name="Suite1">
  <test name="test1">
    <classes>
      <class name="A" />
      <class name="B" />
    </classes>
  </test>
  <test name="test2">
    </test>
  </classes>
  <class name="abc">
    <methods>
      <include name="A" />
      <exclude name="B" />
    </methods>
  </class>
</suite>
  
```

exclude a part of test method from execution sequence/suite/class

2 ways to skip a test method :-

- 1) Test (enabled = false)
 Fully public void m1() {}
- 2) Test → By throwing skipException
 Partially public void m1() {

will be skipped completely

some condition match then skip test method

```

{
  if (title.equals("xyz"))
    throw new skipException("test m1 is skipped")
  else {}
}
  
```

remaining code of test method will not execute

12

SATURDAY

MARCH

072-294 • WK 11

To ignore a test

① Test (enabled = false)

dev

if

code is not ready

T.C. will fail

So skip or bypass

or disable

Testing Group Test

↳ don't exist in ~~JUnit~~ JUnit

↳ <groups> tag can be found either under <test> or <suite> tag in testing.xml.

↳ Groups specified in the <suite> tag apply to all the <test> tags underneath.

A test method may belong to multiple groups.

① Test method() { }

② Test (groups = { "checkintest" }) method() { }

③ Test (groups = { "checkintest", "functest" })

testing.xml

<suite>

<test>

<groups>

<run>

<include name = "checkintest" />

</run>

</groups>

<classes>

<class name = "GroupTestEx" />

</classes>

</test>

</suite>

only method
method
will be
called

13 SUNDAY

2018

Group of Groups → metagroup

<test>

<groups>

<define name = "all">

<include name = "functest" /> <exclude name = "test" />

<include name = "checkintest" />

</define>

<run>

<include name = "all" />

</run>

</groups>

<classes>

</classes>

</test>

App can be excluded
by <exclude name = "test" />

Test NG Excepⁿ Test

We can test whether a code throws a desired excepⁿ or not

Test class to be tested

class ABC {

p.s.v. msg() { int a = 1/0; }

}

Test Class

P.C xyz {

Test (expected Excepⁿ = Arithmetic Excepⁿ class)

p.v. testmsg() {

ABC.msg();

}

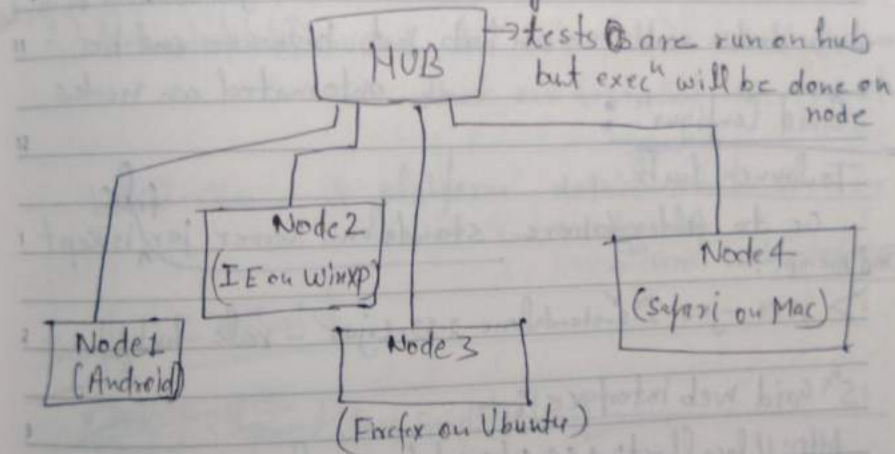
[illegible]

MARCH

30

2 versions Grid 2 \rightarrow latest

→ 5th Grid uses Hub-node concept. Grid 4 → old



When to use G^m Grid

- ② To run your test against diff^t browsers, operating systems, and machines all at the same time.

of
↓
To ensure your app is fully compatible with wide range of OS combis. Use of your test suites

- ↓ compatible with
(2) save time in execution of your test suites through

→ Hub & nodes should be in same Network.
Same LAN or WiFi

31

THURSDAY

MARCH

081-275 • WK 14

1	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	28
29													
M	T	W	T	F	S	S	M	T	W	T	F	S	S

Hub: The central point where tests are kept/loaded.
 ↳ There should be only one Hub in a Grid.

Nodes s^m instances that will execute the tests that you loaded on the hub.
 ↳ Tests will run on hub, but browser will be automated on nodes.

Grid Configurⁿ:

To launch hub

Go to folder where standalone server jar ^{file} is kept.
 Cmd prompt:

> java -jar s-s-standalone-2.53.1.jar -role hub

s^m grid web interface:-

http://localhost:4444/grid/console

Hub

A

IP: 192.168.1.3

node

B

192.168.1.4

To launch node

java -jar s-s-standalone-2.53.1.jar -role node ^{webdriver}

-hub http://192.168.1.3:4444/grid/register

↓
Hub IP

↓
Hub port

→ Go to s^m grid web interface & refresh the page.
 you will see node details.
 http://localhost:4444/grid/console

FRIDAY

APRIL

01

Testscript design for Grid

two objects required:-

1) DesiredCapabilities: \Rightarrow to set B^r of OS that will automatic

2) RemoteWebDriver: to set which node (M/C) that our test will run against.

browserName & platform detail can be seen in
 \Downarrow
 S^m Grid web interface.

Sample T.C. on Grid

```

public class GridStarter {
    public static RemoteWebDriver driver;
    { ps vm (-)
    {

```

```

DesiredCapabilities cap = new DC().firefox();

```

```

cap.setPlatform(Platform.WIN8);

```

```

cap.setBrowserName("firefox");

```

```

dr = new RemoteWebDriver(new URL("http://192.168.1.4:5555/wd/hub"),
                           cap);

```

\downarrow hostIP \downarrow hostPort

```

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

```

```

dr.findElement(ByName("q")).sendKeys("Russia");

```

```

dr.findElement(ByName("btnG")).click();

```

```

}
}

```

2016

1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16 17 18 19 20 21 22
M 9 10 11 12 13 14 15 16 17 18 19 20 21 22
T 23 24 25 26 27 28 29 30 31
W M T W T F S S
Y *men error d C*

MONDAY

APRIL

04

WK 15 • 095-271

Common Error & Cause:

④ Couldn't register this node: the hub is down or not responding
» console.

REASON

② Can't reg^v this node: Error sending the registrⁿ request
↳ console

reason: Hub IP/port is not ^{given} correctly while registering a node.

(3) Unreachable Browser Excepⁿ

reasons if node is not running, node ip in script is incorrect.

Configuring / Setting up Hub & nodes :-

05

TUESDAY

APRIL

090-270 • WK 15

1 2 3 4 5 6 7 8 9 10 11 12
14 15 16 17 18 19 20 21 22 23 24 25 26
28 29 30 31
M T W T F S S M T W T F S S

Navigate on cmd prompt till folder where standalone server (jar) is kept.

To assign specific port

Starting HUB:-

1) java -jar sss-2.53.1.jar -role hub or -port XXXX
↓ ↓
standalone default port 4444

Connecting NODES:-

2) java -jar sss.jar -role webdriver -hub http://localhost:4444/grid/register -port 5556 URL

Connecting chrome NODE:-

~~java -jar sss.jar~~
3) java -Dwebdriver.chrome.driver=D:\dkv\chromedriver.exe -jar sss.jar -role webdriver -hub URL -browserName=chrome, platform=WINDOWS -port 5557

Connecting IE NODE:-

4) java -Dwebdriver.ie.driver=D:\dkv\IEDriver.exe -jar sss.jar -role webdriver -hub URL http://192.168.1.127:4444/grid/register -browserName=ie, platform=WINDOWS -port 5558

Firefox Node Set Up:-

5) java -jar sss.jar -port 5559 -role node -hub URL -browserName=chrome, ~~version=ANY~~ firefox, maxInstances=1 platform=ANY, seleniumProtocol=WebDriver

WEDNESDAY

APRIL

06

will there be three instances node

Scenario: Launching firefox, chrome, IE on node mlc
 Launching ff, ch, ~~op~~ IE on hub mlc.

Set up all drivers with a single node

We should set up each browser node independently.
 so that if one node goes down, it dznt bring other nodes down.

IE/chrome/firefox selenium Node

```
java -Dwebdriver.ie.driver=C:/IED.exe -Dwebdriver.chrome.driver=C:/chromedriver.exe -jar sss.jar -port 5655
  -role node -hub URL
  -browser "browserName=firefox, maxInstances=10, platform=ANY, SeleniumProtocol=WebDriver"
  -browser "browserName=internet explorer, version=11, platform=WINDOWS, maxInstances=10"
  -browser "browserName=chrome, version=ANY, maxInstances=10, platform=WINDOWS"
```

To connect all nodes, provide ip address of VM/machine where the node are configured

```
Webdriver = new RemoteWebDriver (new URL ("http://ipAddress: ssss/wd/hub"), capability);
```

1 2 3 4 5 6 7 8 9 10 11 12 13 14
F 15 16 17 18 19 20 21 22 23 24 25 26 27 28
E 29
B M T W T F S S M T W T F S S

WEDNESDAY

JANUARY

20

Reading Excel using Apache POI

```
import java.io.File;
```

```
import java.io.FileInputStream;
```

```
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
```

```
import org.apache.poi.xssf.usermodel.XSSFSheet;
```

```
public class ReadExcel_POI
```

```
{
```

```
    public static void main(String[] args) throws Exception
```

4 lines of code to read

```
{
    1 File src = new File("E:\\testData.xlsx");
```

```
    2 FileInputStream fis = new FileInputStream(src);
```

// For xlsx file: XSSFWorkbook & XSSFSheet are classes of Apache POI

```
    3 XSSFWorkbook wb = new XSSFWorkbook(fis);
```

```
    4 XSSFSheet sheet1 = wb.getSheetAt(0);
```

```
    // XSSFSheet sheet1 = wb.getSheet("sheet1");
```

```
    // for xls file: HSSFWorkbook & HSSFSheet
```

```
    int rowCount = sheet1.getLastRowNum();
```

```
    System.out.println("total row count is: " + (rowCount + 1));
```

```
    for (int i = 0; i <= rowCount; i++)
```

```
    { for (int j = 0; j <= 1; j++)
```

row or cell no. minus 1

```
    { String data = sheet1.getRow(i).getCell(j).getStringCellValue();
```

0,0 => 1st cell (1st, 1st cell)

21

THURSDAY

JANUARY

021-345 • WK 04

1	2	3	4	5	6	7	8	9	10	11	12
14	15	16	17	18	19	20	21	22	23	24	25
28	29	30	31								
M	T	W	T	F	S	S	M	T	W	T	F

```

    sop("data from row" + i + " and column" + j + " is: " + data);
  }
}

```

```

// close the workbook at the end to avoid
// memory leaks
wb.close();
}
}

```

Data Driven Framework

```

package reusableLibrary;

```

```

public class ExcelDataConfig

```

```

{

```

```

    XSSFWorkbook wb;

```

```

    XSSFSheet sheet1;

```

constructor

```

    public ExcelDataConfig(String excelPath)
    {

```

```

        try {

```

```

            File src = new File(excelPath);

```

```

            FileInputStream fis = new FIS(src);

```

```

            wb = new XSSFWorkbook(fis);

```

```

        }

```

```

        catch (Exception e)

```

```

        {
            sop(e.getMessage());
        }

```

```

    public String

```

```

        getData(int sheetNumber, int row, int col)
    {

```

```

        Sheet1 = wb.getSheetAt(sheetNumber);

```

```

        String data = sheet1.getRow(row).getCell(col).getStringCellValue();

```

```

        return data;
    }
}

```

2016

```

16 public int getRowCount (int sheetIndex)
17 {
18     int row = wb.getSheetAt (sheetIndex).getLastRowNum();
19     row = row + 1;
20     return row;
21 }
    
```

package cdfAndExcel; ← TestNG Framework

```

21 package cdfAndExcel;
22 import org.testng.Assert;
23     .testing.annotations.DataProvider;
24     .testing.annotations.Test;
25 import reusableLibrary.ExcelDataConfig;
26     .WebDriverConfig;
    
```

```

27 public class WordpressLoginExcel
28 {
29     @Test (dataProvider = "wordpressData")
    
```

This test will depend on a DP named "1"

p.v. loginToWordpress (String username, String Pwd) throws Eⁿ

```

30 {
31     WebDriverConfig wdc = new WebDriverConfig ();
32     wdc.openInFirefox ("http://demo-site-center/wordpress/wp-login.php");
33     wdc.fdriver.findElement (By.id ("wp-login")).sendKeys (username);
34     wdc.fdriver.findElement (By.id ("wp-pass")).sendKeys (pwd);
35     wdc.fdriver.findElement (By.xpath ("//*[@Id='wp-submit']")).click();
36     Thread.sleep (5000);
37     Assert.assertEquals (wdc.fdriver.getTitle().contains ("Dashboard"),
38         "not able to login - Invalid Credential");
39 } sop ("Login Successful");
    
```


23

SATURDAY

JANUARY

023-343 • WK 04

return ExcelReader.getExcelData()

② DataProvider (name = "wordpressData")

public Object [][] passData ()

{ ExcelDataConfig edc = new EDC("C:\\input\\data\\data.xlsx");

int row = edc.getRowCount();

Object [][] data = new Object [row][2];

for (int i = 0; i < row; i++)

{ data[i][0] = edc.getData(0, i, 0);

data[i][1] = edc.getData(0, i, 1);

}

return data;

username	PWD
deepak1	Test@123
dkv2	"
dkv3	"

⇒ First DP will be executed then Test will run as many times as the no. of rows in input data excel.

For 10 rows

① Sheet.getPhysicalNumberOfRows() ⇒ give No. of rows

② Sheet.getLastRowNum() ⇒ gives the index of last row

getStringCellValue() → return String: used for text

getNumericCellValue() → return double: used for numbers

getBooleanCellValue()

Including 1st & last row

For xlsx column index must be in $0 \leq \text{index} \leq 1048576$

cannot be negative → otherwise
Illegal
Argument
Exceptⁿ

Writing to excel using Apache Poi

```

import java.io.*;
import org.apache.poi.xssf.usermodel.*;

public class WriteExcel_POI
{
    public static void main(String args[]) throws Excn
    {
        File src = new File("E:\\test data\\Test Data.xlsx");
        FileInputStream fis = new FIS(src);
        XSSFWorkbook wb = new XSSFWorkbook(fis);
        XSSFSheet sheet1 = wb.getSheetAt(0);
        {
            sheet1.getRow(0).createCell(2).setCellValue("Pass");
            sheet2.getRow(1).createCell(2).setCellValue("Fail");
        }
        FileOutputStream fos = new FileOutputStream(src);
        wb.write(fos);
        wb.close();
        sop("It's Done!!!");
    }
}

XSSFSheet sheet = wb.createSheet("sampleSheet");
→ int data = (int) wb.getSheetAt(0).getRow(0).getCell(2).getNumericValue()

```


Actions → class

Action → interface

M	1	2	3	4	5	6	7	8	9	10	11	12	13
A	14	15	16	17	18	19	20	21	22	23	24	25	26
R	28	29	30	31									
	M	T	W	T	F	S	S	M	T	W	T	F	S

MONDAY

FEBRUARY

22

WK 09 • 053/313

used to perform Advanced User Interactions:

→ In WebDriver, Handling K.B & Mouse Events is done using API.

→ Drag & Drop → Keyboard Events

→ right click, double click

→ Resizing a web element (e.g. text area etc) → use moveByOffset

```
import org.openqa.selenium.interactions.Actions;
```

```
• Action;
```

```
Actions action = new Actions(driver);
```

```
action.contextClick(eleele).perform(); // performs right click on ele.
```

→ opens contextual menu

→ To perform multiple operations/actions or a sequence of ops^{ns} on the same element we use build() method.

Send Text in Cap^l Letters

→ for CAPS

```
action.moveToElement(ele).click().keyDown(ele, Keys.SHIFT)
```

```
• sendKeys(ele, "deepak").build().perform();
```

↓
compiles multiple ops^{ns} into a single step.

→ To right click on a link & open appⁿ in a New Window

```
action.contextClick(ele).sendKeys("w").build().perform();
```

→ To fetch Tooltip of a control in S^m

unsupported command.

```
action.clickAndHold(ele).perform();
```

→ or • moveToElement(ele)

↓
[Exit] (click to exit app)

```
String tooltip = ele.getText();
```

23

TUESDAY
FEBRUARY

054-312 • WK 09

→ Drag & Drop.

```
action.dragAndDrop(draggable, droppable).perform();
```

→ To refresh an appⁿ using Action class

```
action.keyDown(Keys.CONTROL).sendKeys(Keys.F5).perform();
```

→ we need to use `perform()` to execute the action.
if there is only one action/appⁿ then no need to use `build()`.

→ To click an ele

```
action.moveToElement(ele).click().perform();
```

`action.click(ele)` → clicks in the middle of ele.

↳ `keyDown(modifierKey)`: `Keys.ALT`, `Keys.SHIFT`, `Keys.CONTROL`
↳ performs a key press & ~~release~~ hold.

Subsequent interactions may assume its kept pressed

↳ `keyUp(u)` → performs a key release

↳ `moveByOffset(x-offset, y-offset)`

↳ moves the mouse posⁿ from its current posⁿ by the given offset

→ if `x` has (-)ve value ⇒ move mouse to left side

`y` has (-)ve ⇒ move mouse it up.

↳ `moveToElement(e)` → moves the mouse to the middle of element

2016

→ `release()` → it releases the left mouse button at the current mouse location.

→ `sendKeys(onElement, charSequence)`

→ receives keystrokes
→ usually text field

Any string value representing the sequence of keystrokes to be sent

source → 3 parameters.
↳ `dragAndDropBy(x-offset, y-offset);`

↳ To drag by a given offset & drop

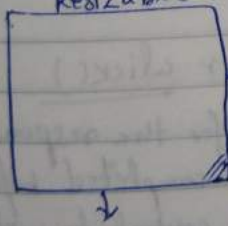
↳ when you open any context menu by right click & want to select an option/ele so first locate that option then `click()` on that contextOptions ele `click();`

→ To resize a web element

int values in both

`action.clickAndHold(eleToResize).moveByOffset(50, 50).release().build().perform();`

Resizable



→ With cursor, grab the right or bottom border & drag to the desired height or width.

if this ele is present in a frame, put an explicit for frame

`wait.until(EC.frameToBeAvailableAndSwitchToIt(By.css("demo-frame")));`

25

THURSDAY
FEBRUARY
088-310 • WK 09

1 2 3 4 5 6 7 8 9
11 12 13 14 15 16 17 18 19 20 21 22 23
25 26 27 28 29 30 31
M T W T F S S M

M	14	15
A	28	29
R	M	1

```
{ click()
{ click(etc)
```

- { click
- { click (etc)

- Click And Hold (C)

- ✓ click And Model (cbe)

context click()

- context click (ele)

doubleClick()

doubleClick (ele)

Key Down (Key) \rightarrow Key press

KeyDown (ele, Key) \rightarrow Key press

same way

Key VF

after focusing on
an ele.

$\{ \text{moveToElement}(\text{ele}) \rightarrow \text{move the mouse at the middle of ele}$
 $\{ \text{moveToElement}(\text{ele}, \text{x-offset}, \text{y-offset})$

```
moveToElement(ele, x-offset, y-offset)
```

↳ move the mouse to an offset

release() → current mouse locⁿ from top-left corner of ele

• $\{ \text{release}(\text{ele}) \rightarrow \text{in the middle of ele}$

```
5 sendKeys(charSeq)
```

$\text{sendKeys}(ele, \text{charseq}) \Rightarrow \text{equiv}^t \text{ to } \Rightarrow \text{action.wlick}(ele).\text{sendKeys}(\text{keys})$

Diff b/w

① Action's click() WebDriver click()

do not wait for
action to be completed

↓ wait for the response to be completed before passing control to next statement.

JavaScriptExecutor

(Takes Screenshot) driver
(JavaScript) driver

↓
interface

MONDAY

FEBRUARY

29
↓
Downcasting

→ provides mechanism to execute javascript through selenium driver.

→ It provides executeScript & executeAsyncScript methods to run javascript in the context of currently selected frame or window.

↓
injecting javascript into browser using selenium

↓
or AUT
(App Under Test)

Why we use it?

To enhance the capabilities of existing scripts by performing javascript injection into our AUT.

import org.openqa.selenium.JavaScriptExecutor;

Syntax

JavaScriptExecutor js = (JavaScriptExecutor) driver;
js.executeScript(script, arguments)

↓
Javascript
to execute.

↓
the arguments to
script (optional)

1) To click a button

js.executeScript("arguments[0].click();", element);

2) To generate alert popup window in 5th

js.es("alert('Hello World');");

3) To Refresh Browser

js.es("history.go(0)");

2016

27

SATURDAY
FEBRUARY

055-308 • WK 09

Casting $\begin{cases} \rightarrow \text{upcasting} \\ \rightarrow \text{downcasting} \end{cases}$

→ we are downcasting ~~where~~ object to JSEx (obj)
Where have you used JavaScript Ex?

- while working with frames
 - with ck-editor
 - To scroll the page
 - To highlight an ele
 - when ele is not clickable using normal method.
- To type text in text box if field "Id" is given
Id = "fname"

js.es("document.getElementById('fname').value = 'Deepak'")

How click an an Invisibl = / disabled ele. (Mukesh Atwari)

↓ make is forcefully visible / enabled

→ selenium do not hv capability to change the visibility of an ele.

by javascript

↓
the full control of DOM

① string is =

js.es("document.getElementById('fname').style.height = 'auto'; arguments[0].style.visibility = 'visible'")

② js.es(js, ele);

③ ele.click(); → otherwise Ele Not Visible except

28 SUNDAY

→ so javascript is much powerful which takes the complete access of Dom.

2016

01

TUESDAY

MARCH

001-305 • WAK 10

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
M	T	W	T	F	S	S	M	T	W	T	F	S	S															

4) To get title of webpage

```
String s = js.es("return document.title;").to String();
```

_____ .URL _____ // Fetch URL of current page

_____ .domain _____ // Fetch domain name

www.xyz.com

5) perform scroll (0, -500) → for scroll up.

```
js.es("window.scrollTo(0, 50)"); // vertical scroll down by 50 pixels
```

```
js.es("window.scrollTo(0, document.body.scrollHeight)");
```

To ✓ // scrolling till the bottom.

```
js.es("arguments[0].scrollIntoView(true);", element);
```

6) Navigate to a new page:

```
js.es("window.location = 'http://www.facebook.com/ufitHelp'");
```

(can be removed) → SWC

7) Highlight in S^h

↳ Highlight is the most prominent feature used in Object SPY in VFT/QTP

↳ Highlight plays crucial role in debugging test script

↳ It highlights the web element on page while execution

↳ can be done by changing backgroundColor/border color of element

```
js.es("arguments[0].style.border = '4px groove green';", ele);
```

2018 ⇒ Janscript[®] inject javascript into our app^h which will change the css properties of ele at runtime.

arguments[0].style.border → injects css ~~prop~~ style tag into the ele & making its border 4px wide green line with groove look.

ele: → ele on which border would be drawn.

executeScript(script, argument)

↓
 Return type
 {
 boolean
 Long/Double for decimal
 String
 List
 WebElement
 null
 }

↓ must be {
 a WebElement
 a number
 a boolean
 optional {
 a String or
 a List of any
 "combin" of above.
 otherwise Exceptⁿ
 (maybe empty)
}

4 Set Attribute for an web element

js.es("arguments[0].setAttribute('attr', '10')", element);

5 Set value for an ele (hint text)

js.es("arguments[0].value = arguments[1]", ele, "value").
 script arguments

6 can Add an ele in DOM.

arguments[0] → parameter
 ele → value for param.

js.es("arguments[0].setAttribute('style', 'border: solid 2px red')", useName)