**Interview Questions on TestNG:**

**What is TestNG?**

TestNG is a testing framework inspired from JUnit and NUnit, which is used to run selenium test cases. It has introduced some new functionalities that made TestNG more powerful than JUnit and NUnit. Some of these features are

* Supports annotations
* Supports parallel execution
* Supports parameterizing the test cases with @Parameters annotation
* It allows data driven testing using @DataProvider annotation
* Introduced TestNG groups – You can group the tests into different groups and run whichever group you want.
* Supports dependent test methods, parallel testing etc.
* It generates detailed TestNG reports

**How to run tests with TestNG?**

First you need to install TestNG.

Import org.testng.annotations.Test

Include the annotation @Test before the method.

Right click on the class and run as TestNG.

**What is TestNG xml and why is it used for?**

TestNG xml is a configuration file in TestNG. It is used to define test suites and tests. It provides different options to include packages, classes and independed test methods in our tests. It also allows us to configure multiple tests in a single test suite.

**How to exclude a specific method from running in TestNG?**

If you want to exclude a specific test from execution you can delete it from the TestNG xml and run it. But if you want to exclude a specific method from running you can use exclude tag.

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

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

<suite name=*"Suite"*>

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

<classes>

<class name=*"TestNGExamples.RadioButtons"*>

<methods>

<exclude name=*"RadioButton2"* />

<exclude name=*"RadioButton3"* />

</methods>

</class>

<class name=*"TestNGExamples.CheckBoxes"*/>

<class name=*"TestNGExamples.TextArea"*/>

<class name=*"TestNGExamples.TextBox"*/>

</classes>

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

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

If you have more number of methods to exclude, you can use regular expression to exclude those methods.

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

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

<suite name=*"Suite"*>

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

<classes>

<class name=*"TestNGExamples.RadioButtons"*>

<methods>

<exclude name=*"RadioButton.\*"* />

</methods>

</class>

<class name=*"TestNGExamples.CheckBoxes"*/>

<class name=*"TestNGExamples.TextArea"*/>

<class name=*"TestNGExamples.TextBox"*/>

</classes>

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

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

The above will exclude all the methods starting with RadioButton.

**How to run tests at package level?**

You can use packages main tag with package sub tags to run tests at package level.

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

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

<suite name=*"Suite"*>

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

<packages>

<package name=*"TestNGExamples"*/>

</packages>

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

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

**What are the different test annotations we have in TestNG?**

@BeforeTest – This method will be executed before the tests in a test folder

@AfterTest – This method will be executed after the tests in a test folder

@BeforeSuite – This method will be executed before the test suite

@AfterSuite – This method will be executed after the test suite

@BeforeMethod – This method will be executed before each method in a test folder

@AfterMethod – This method will be executed after each method in a test folder’

@BeforeClass – This method will be executed before any method with in a class

@AfterClass – This method will be executed after all the methods with in a class

@BeforeTest and @AfterTest will be run before and after a test. For example if we these in TextBox.java and when we runt he TestNG xml, @BeforeTest method will run before all the methods in TextBox.java and @AfterTest method will run after all the methods in TextBox.java.

These can be used if you have to clear the database tables before running tests etc.

@BeforeSuite and @AfterSuite will run before and after test suite. For example if we have these in any program when we run the TestNG xml,@BeforeSuite method will run before all the methods in all tests under that suite and @AfterSuite will run after all the methods in all tests under the Suite.

These can be used if you want to set the environment variables. Let’s say your environments change for your execution and in these cases you can use this.

@BeforeMethod and @AfterMethod will run before and after each method in a test folder.

If you have to do some validations before and after each method like checking if the server is up or not etc., then we can use these annotations.

@BeforeClass and @AfterClass will run before and after each class.

**How can you create groups in TestNG?**

In each and every test case, the methods that you want to run as part of smoke testing, mention them they belong to a group.

@Test(groups={“smoke”})

@Test(groups={“smoke”,”regression”})

Once we name the methods belong to smoke testing, change the TestNG xml to execute only those methods belong to smoke. After <test> tag add groups section in TestNG xml.

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

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

<suite name=*"Suite"*>

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

<groups>

<run>

<include name=*"smoke"*/>

</run>

</groups>

<classes>

<class name=*"TestNGExamples.RadioButtons"*/>

<class name=*"TestNGExamples.CheckBoxes"*/>

<class name=*"TestNGExamples.TextArea"*/>

<class name=*"TestNGExamples.TextBox"*/>

</classes>

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

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

**How to exclude test cases with a group name?**

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

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

<suite name=*"Suite"*>

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

<groups>

<run>

<exclude name=*"smoke"*/>

</run>

</groups>

<classes>

<class name=*"TestNGExamples.RadioButtons"*/>

<class name=*"TestNGExamples.CheckBoxes"*/>

<class name=*"TestNGExamples.TextArea"*/>

<class name=*"TestNGExamples.TextBox"*/>

</classes>

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

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

**How do you create dependency of tests on other tests?**

@Test(dependsOnMethods={“methodname”})

**How to skip a particular test in TestNG?**

@Test(enabled=false).

**How to set timeout for a particular test in TestNG?**

@Test(timeOut=4000) – for 40 Sec.

**How do you pass Global parameters in TestNG?**

Global Parameters like URLs, environment variables etc., can be defined at test level and suite level.

IF you define them at test level, the parameters can be accessed at test level only. If you define them at suite level, the parameters can be accessed for the entire suite.

For defining the parameters, first in TestNG, you have to use the parameter tag and define the name and value pair.

**<parameter name=*"URL"* value=*"file:///D:/Selenium%20Course/Java/index.html"*/>**

In the test above the method define the parameter annotation.

**@Parameters({"URL"})**

@Test

**public** **void** TextBox1(**String urlname**) **throws** InterruptedException{

System.*setProperty*("webdriver.gecko.driver", "C:\\BrowserDrivers\\geckodriver.exe");

WebDriver driver = **new** FirefoxDriver();

**driver.get(urlname);**

//Thread.sleep(2000);

driver.findElement(By.*id*("firstname")).sendKeys("Subbu");

//Thread.sleep(2000);

driver.quit();

}

**How to pass multiple parameters in TestNG?**

TestNG xml:

<test thread-count=*"5"* name=*"TextBoxTest"*>

**<parameter name=*"URL"* value=*"file:///D:/Selenium%20Course/Java/index.html"*/>**

**<parameter name=*"USERNAME"* value=*"subbu"*/>**

<classes>

<class name=*"TestNGExamples.TextBox"*/>

</classes>

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

Test Method:

**@Parameters({"URL","USERNAME"})**

@Test

**public** **void** TextBox1(**String urlname, String uname**) **throws** InterruptedException{

System.*setProperty*("webdriver.gecko.driver", "C:\\BrowserDrivers\\geckodriver.exe");

WebDriver driver = **new** FirefoxDriver();

**driver.get(urlname);**

//Thread.sleep(2000);

**driver.findElement(By.*id*("firstname")).sendKeys(uname);**

//Thread.sleep(2000);

driver.quit();

}

**How to pass parameters to a specific method?**

IF you want to provide parameters to a specific method we use dataprovider annotations.

**package** testpackage;

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

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

**public** **class** hltest1 {

@Test

**public** **void** hlt1() {

System.***out***.println("hlt1");

}

@Test(dataProvider="getData")

**public** **void** hlt1a(String username, String password){

System.***out***.println("hlt1a");

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

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

}

@Test

**public** **void** hlt1b(){

System.***out***.println("hlt1b");

}

@Test

**public** **void** hlt1c(){

System.***out***.println("hlt1c");

}

@DataProvider

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

//First combination - firstun and firstpw

//Second combination - secondun and secondpw

//Third combination - thirdun and thirdpw

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

data[0][0] = "firstun";

data[0][1] = "firstpw";

data[1][0] = "secondun";

data[1][1] = "secondpw";

data[2][0] = "thirdun";

data[2][1] = "thirdpw";

**return** data;

}

}

**What is a listener in TestNG?**

A listener is an interface which listens to all the events like test start, test failure, test complete etc. You can use this listener to do some extra actions like taking screenshots when a test is failed, clean the database before running script etc.

**Explain about ITestListener?**

ITestListener is an interface which provides the following methods.

onTestStart – This will be executed on each test (method) starts

onTestSuccess – This will be executed on each test (method) is successful

onTestFailure – This will be executed when a test(method) fails

onTestSkipped – This will be executed when a test is skipped

onTestFailedButWithinSuccessPercentage

onStart – This will be executed when the test suite starts

onFinish – This will be executed when the test suite finishes execution

**How to implement listeners in TestNG?**

Define a new class extending the listener (for example iTestListener). Since listener is an interface, implement all the methods of the listener.

Modify the TestNG xml to make sure that the test we are going to run use the above listener. This can be done by using listeners tags.

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

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

<suite name=*"Suite"*>

<listeners>

<listener class-name=*"TestNGExamples.Listeners"*/>

</listeners>

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

<classes>

<class name=*"TestNGExamples.TextBoxNew"*/>

<class name=*"TestNGExamples1.RadioButtons"*/>

</classes>

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

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

**How to take screenshots in selenium?**

Selenium provides an interface “TakesScreenshot” which has a method “getScreenshotAs()” which will get screenshots into a file.

WebDriver driver = **new** FirefoxDriver();

File src = ((TakesScreenshot)driver).getScreenshotAs(OutputType.***FILE***);

FileUtils.*copyFile*(src, **new** File("D:\\TestNGScreenshots\\screenshot.png"));

**How to take screenshots for failed tests only in selenium?**

We have to use TestNG and listeners for taking screenshots only for failed methods. We will use the above method in Listeners onTestFailure() method and get the screenshots. But for this we have to pass the driver to the listeners onTestFailure() method. So we normally create a map with test method as key and the driver as the value.

We can get the method name in onTestFailure using the method getInstanceName() and we can get the driver from the map using this method name.

**What is properties file?**

We can use properties file to store any global data like paths, urls, usernames, passwords etc. Whenever we need this data, we create a properties class object and get this data.

Properties prop = new Properties();

FileInputStream fip = new FileInputStream("src/globaldata/data.properties");

prop.load(fip);

System.out.println(prop.getProperty("username"));

**What are the different assertions in TestNG?**

We have two types of assertions in TestNG. They are

1. Hard Assert
2. Soft Assert

Hard Assert fails the test case and aborts the remaining steps.

Soft assert fails the test case but doesn’t abort the remaining test steps. We have to use assertAll() at the end of the test case to make the test case fail. Otherwise it will pass the test case.

**What are the different assertion methods we have?**

The following are the different methods under Assert.

AssertEquals

AssertNotEquals

AssertTrue

AssertFalse

AssertNull

AssertNotNull

**How to set priority in TestNG?**

We use priority attribute in @Test annotation for setting priority.

Ex:

@Test(priority=1)

@Test(Priority=2)

**How to run tests in parallel in TestNG?**

Using parallel attribute in TestNG xml.

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

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

<suite name=*"BasicSuite"*>

<test parallel=*"methods"* thread-count=*"2"* name=*"ParallelTest"*>

<parameter name=*"URL"* value=*"file:///D:/Selenium%20Course/Java/index.html"*/>

<classes>

<class name=*"TestNGExamples.ParallelRunning"*/>

</classes>

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

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

**How to disable or ignore a test case in TestNG?**

@Test(enabled = false)