Fast ‘n’ Learning - TestNG Interview Questions

What is TestNG?

TestNG (NG for Next Generation) is a testing framework inspired from JUnit and NUnit but introducing some new functionalities that make it more powerful and easier to use.

It can be integrated with selenium or any other automation tool to provide multiple capabilities like assertions, reporting, parallel test execution etc.

Why TestNG Framework Is Used?

TestNG source code is controlled by the Apache software license and it is freely available for the download. The most common use of TestNG framework is with the Selenium WebDriver to produce browser based test automation.

TestNG framework supports intuitive HTML report style to share with stakeholders. The report includes product health indicators such as pass/fail count, the age of defect etc. This TestNG report fills the gap cropped up by the inability of WebDriver to generate the report. On the top, TestNG offers exceptional handling support that allows a program to run without exiting abruptly.

What are the Essential TestNG Framework Features?

TestNG framework introduces some new features that make it more powerful and easier to use, such as:

- Annotations: An annotation is a form of metadata in java which can be applied to various elements of java source code so that any tool, debugger or application program can use these annotations. Classes, methods, variables, parameters and packages may be annotated.
- Provides arbitrarily big thread pools to run tests in their own thread, one thread per test class, etc.
- Ensure that your code is thread-safe.
- Flexible test configuration.
- Supports data-driven testing (with @DataProvider).
- Support for parameters.
- Powerful execution model (Independent of TestSuite).
- Easy integration with a variety of tools and plug-ins (Eclipse, Maven, Jenkins etc.

List out the advantages of TestNG over Junit?

- Advantages of JUnit over TestNG includes
- Compare to JUnit annotations, TestNG are easy to understand
- Unlike JUnit, TestNG does not require to declare @BeforeClass and @AfterClass
- Method name constraint is not there in TestNG
- TestNG allows you the grouping of test cases easily which is not possible in JUnit
- TestNG supports following three additional setup: @Before/AfterSuite, @Before/AfterTest and @Before/AfterGroup
- TestNG does not need to extend any class
- In TestNG, it is possible to run selenium test cases in parallel
- Based on group TestNG allows you to execute the test cases
- TestNG allows you to determine the dependent test cases; each test case is autonomous to other test case
- It allows to assign priority to test methods
- It has support for parameterizing test cases using @Parameters annotation
- It allows data driven testing using @DataProvider annotation
- It has different assertions that helps in checking the expected and actual results
- Detailed (HTML) reports

Explain what are the basic steps required in writing TestNG tests?

- The basic steps required in writing TestNG includes
- Write down the business logic of your test and insert TestNG annotations in your code
- In a build.xml or testing.xml, add the information about your test
- Run TestNG

List out various ways in which TestNG can be invoked?

- TestNG can be invoked in different ways like
- Using Eclipse
- From the command line
- Using IntelliJ's IDEA
- With ant

Explain what is testng.xml file used for?

File testing.xml captures your entire testing in XML. This file makes it easy to define all your test suites and their parameters in one file, which you can verify in your code repository or e-mail to coworkers. It also makes easy to pull out subsets of your tests or split several runtime configurations.

In TestNG how can you disable a test?

To disable the test case you don't want, you can use annotations @Test(enabled = false).

Explain what is Time-Out test in TestNG?

The Time-Out test in TestNG is nothing but time allotted to perform testing for a method. If the test fails to finish in that specific time limit, TestNG will abandon further testing and mark it as a failed.

Mention what does the "suite test" does in TestNG?

"Suite Test" is helps when you have to run few test together, "Suite Test" bundle this test together. XML file is used to run the suite test.

Explain what is parametric testing?

Parameterized testing allows developers to execute the same test over and over again using different values. In two different ways TestNG allows you to pass parameters directly to your test methods.

With testing.xml

With Data Providers

Explain what does @Test(invocationCount=?) and (threadPoolSize=?) indicates?

- @Test (threadPoolSize=?): The threadPoolSize attributes tell TestNG to form a thread pool to run the test method through multiple threads. With threadpool, the running time of the test method reduces greatly.
- @Test(invocationCount=?): The invocationcount tells how many times TestNG should run this test method

Explain what is Group Test in TestNG?

It is a new feature included in TestNG, it allows you to dispatch methods into proper portions and perform grouping of test methods. With group test, you can not only declare methods that belong to groups, but you can also specify groups that contain other groups. Groups are determined in your testing.xml file using the group test.

Explain in what ways does TestNG allows you to specify dependencies?

TestNG allows you to specify dependencies in two ways

- Using attributes dependsOnMethods in @Test annotations
- Using attributes dependsOnGroups in @Test annotations

What are the annotations available in TestNG?

@BeforeTest
@AfterTest
@BeforeClass
@AfterClass
@BeforeMethod
@AfterMethod
@BeforeSuite
@AfterSuite
@BeforeGroups
@AfterGroups
@Test

Can you arrange the below testng.xml tags from parent to child?

<test>

<suite>

<class>

<methods>

<classes>

The correct order of the TestNG tags are as follows

<suite>

<test>

<classes>

<class>

<methods>

What is the importance of testng.xml file?

In a Selenium TestNG project, we use testng.xml file to configure the complete test suite in a single file. Some of the features are as follows.

- testng.xml file allows to include or exclude the execution of test

methods and test groups
- It allows to pass parameters to the test cases
- Allows to add group dependencies
- Allows to add priorities to the test cases
- Allows to configure parallel execution of test cases
- Allows to parameterize the test cases

What is TestNG Assert and list out common TestNG Assertions?

TestNG Asserts help us to verify the condition of the test in the middle of the test run. Based on the TestNG Assertions, we will consider a successful test only if it is completed the test run without throwing any exception.

Some of the common assertions supported by TestNG are

- assertEqual (String actual, String expected)
- assertEqual (String actual, String expected, String message)
- assertEquals (boolean actual, boolean expected)
- assertTrue(condition)
- assertTrue (condition, message)
- assertFalse(condition)
- assertFalse (condition, message)

What is Soft Assert in TestNG?

Soft Assert collects errors during @Test. Soft Assert does not throw an exception when an assert fails and would continue with the next step after the assert statement.

If there is any exception and you want to throw it then you need to use assertAll()method as a last statement in the @Test and test suite again continue with next @Test as it is.

What is Hard Assert in TestNG?

Hard Assert throws an AssertException immediately when an assert statement fails and test suite continues with next @Test

How to set test case priority in TestNG?

We use priority attribute to the @Test annotations. In case priority is not set

then the test scripts execute in alphabetical order.

@Test(priority=1)

What is Parameterized testing in TestNG?

Parameterized tests allow developers to run the same test over and over again using different values.

There are two ways to set these parameters:

- using testng.xml
- using Data Providers

How can we create data driven framework using TestNG?

By using @DataProvider annotation, we can create a Data Driven Framework.

How to run a group of test cases using TestNG?

TestNG allows you to perform sophisticated groupings of test methods. Not only can you declare that methods belong to groups, but you can also specify groups that contain other groups. Then TestNG can be invoked and asked to include a certain set of groups (or regular expressions) while excluding another set. This gives you maximum flexibility in how you partition your tests and doesn't require you to recompile anything if you want to run two different sets of tests back to back.

Groups are specified in your testng.xml file and can be found either under the <test> or <suite> tag. Groups specified in the <suite> tag apply to all the <test>tags underneath.

```
@Test (groups = { "smokeTest", "functionalTest" })

public void loginTest(){
```

```
System.out.println("Logged in successfully");

}
```

How to create Group of Groups in TestNG?

Groups can also include other groups. These groups are called MetaGroups. For example, you might want to define a group all that includes smokeTest and functionalTest. Let's modify our testng.xml file as follows:

```
<groups>
   <define name="all">
        <include name="smokeTest"/>
        <include name="functionalTest"/>
   </define>
   <run>
       <include name="all" />
   </run>
</groups>
```

How to run test cases in parallel using TestNG?

We can use "parallel" attribute in testng.xml to accomplish parallel test execution in TestNG

The parallel attribute of suite tag can accept four values:

tests – All the test cases inside <test> tag of testng.xml file will run parallel
classes – All the test cases inside a java class will run parallel
methods – All the methods with @Test annotation will execute parallel
instances – Test cases in same instance will execute parallel but two methods of two different instances will run in different thread.

```
<suite name="fastneasylearning" parallel="methods">
```

How to exclude a particular test method from a test case execution?

By adding the exclude tag in the testng.xml

```
<classes>
  <class name="TestCaseName">
    <methods>
      <exclude name="TestMethodNameToExclude"/>
    </methods>
  </class>
</classes>
```

How to exclude a particular test group from a test case execution?

By adding the exclude tag in the testng.xml

```
<groups>
    <run>
```

```
        <exclude name="TestGroupNameToExclude"/>

    </run>

 </groups>
```

How to skip a @Test method from execution in TestNG?

By using throw new SkipException()

Once SkipException() thrown, remaining part of that test method will not be executed and control will go directly to next test method execution.

```
 throw new SkipException("Skipping - This is not ready for testing ");
```

How TestNG allows to state dependencies?

TestNG allows two ways to declare the dependencies.

- Using attributes dependsOnMethods in @Test annotations
- Using attributes dependsOnGroups in @Test annotations

What is the use of @Listener annotation in TestNG?

TestNG listeners are used to configure reports and logging. One of the most widely used listeners in TestNG is ITestListener interface. It has methods like onTestStart, onTestSuccess, onTestFailure, onTestSkipped etc. We should implement this interface creating a listener class of our own. Next we should add the listeners annotation (@Listeners) in the Class which was created.

26. How to write regular expression in testng.xml file to search @Test methods containing “smoke” keyword.

Regular expression to find @Test methods containing keyword “smoke” is as mentioned below.

```
<methods>
    <include name=".*smoke.*"/>
</methods>
```

What is the time unit we specify in test suites and test cases?

We specify the time unit in test suites and test cases is in milliseconds.

What does the test timeout mean in TestNG?

The maximum number of milliseconds a test case should take.

```
@Test(threadPoolSize = 3, invocationCount = 10,  timeOut = 10000)
public void testCase1(){
```

In this example, the function testCase1 will be invoked ten times from three different threads. Additionally, a time-out of ten seconds guarantees that none of the threads will block on this thread forever.