1

What is TestNG?

1

TestNG is a testing framework designed to simplify a broad range of testing needs, from unit testing to
integration testing

2

What are the advantages of TestNG?

2

- TestNG provides parallel execution of test methods
- It allows to define dependency of one test method over other method
- It allows to assign priority to test methods
- It allows grouping of test methods into test groups
- 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

3

What is the importance of testng.xml file?

3

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

4

What is the importance of testng.xml file?

4

- 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

5

What is Soft Assert in TestNG?

5

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.

6

What is Hard Assert in TestNG?

6

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

7

What is Parameterized testing in TestNG?

7

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

8

How TestNG allows to state dependencies?

8

TestNG allows two ways to declare the dependencies.
Using attributes dependsOnMethods in @Test annotations –
Using attributes dependsOnGroups in @Test annotations –

9

What are the different ways to produce reports for TestNG results?

9

TestNG offers two ways to produce a report.
- Listeners implement the interface org.testng.ITestListener and are notified in real time of when a test starts, passes, fails, etc…
- Reporters implement the interface org.testng.IReporterand are notified when all the suites have been run by TestNG. The IReporter instance receives a list of objects that describe the entire test run.

10

What is @DataProvider annotation?

10

@DataProvider: A test method that uses DataProvider will be executed the specific methods multiple number of times based on the data provided by the DataProvider. The test method will be executed using the same instance of the test class to which the test method belongs.

11

List out various ways in which TestNG can be invoked?

11

- Using Eclipse IDE
- Using ant build tool
- From the command line
- Using IntelliJ ' s IDEA

12

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

12

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

13

How to write regular expression In testng.xml file to search @Test methods containing " smoke " keyword.

13

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

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

14

How TestNG allows to state dependencies?

14

TestNG allows two ways to declare the dependencies.

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

15

What are the annotations available in TestNG?

15

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

16

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

16

The correct order of the TestNG tags are as follows

<suite>
<test>
<classes>
<class>
<methods>

17

How to create and run testng.xml ?

17

In TestNG framework, we need to create testng.xml file to create and handle multiple test classes. We do configure our test run, set test dependency, include or exclude any test, method, class or package and set priority etc in the xml file.

18

What is TestNG Assert and list out common TestNG Assertions?

18

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)

19

What is exception test in TestNG?

19

TestNG gives an option for tracing the Exception handling of code. You can verify whether a code throws the expected exception or not. The expected exception to validate while running the test case is mentioned using the expectedExceptions attribute value along with @Test annotation.

20

How to run a group of test cases using TestNG?

20

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");
}
```

21

How to create Group of Groups in TestNG?

21

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:

22

How to create Group of Groups in TestNG?

22

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>
```

23

What is timeOut in TestNG?

23

While running test cases, there can be a case when some test cases take much more time than expected. In such a case, we can mark the test case as a failed test case by using timeOut. TimeOut in TestNG allows you to configure the time period to wait for a test to get completely executed. It can be configured in two levels:

- At the suit level: It will be available to all the test methods.
- At each method level: It will be available to a particular test method.

The timeOut attribute can be specified as shown below:

```
@Test( timeOut = 700)
```

24

What is invocationCount in TestNG?

24

An invocationCount in TestNG is the number of times that we want to execute the same test.

25

How can we disable the test case from running?

25

We can disable the test case from running by using the enabled attribute. We can assign the false value to the enabled attribute

```
@Test(enabled=false)
public void testcase1()
{
```

26

What is the use of @Listener annotation in TestNG?

26

TestNG provides different kinds of listeners which can perform different actions whenever the event is triggered. The most widely used listener in TestNG is ITestListener interface. The ITestListener interface contains methods such as onTestSuccess, onTestfailure, onTestSkipped, etc.

Following are the scenarios that can be made:

- If the test case is failed, then what action should be performed by the listener.
- If the test case is passed, then what action should be performed by the listener.
- If the test case is skipped, then what action should be performed by the listener.

27

What is the use of @Factory annotation?

27

The @Factory annotation is useful when we want to run multiple test cases through a single test class. It is mainly used for the dynamic execution of test cases.

28

What is the difference between @Factory and @DataProvider annotation?

28

- @DataProvider: It is annotation used by TestNG to execute the test method multiple numbers of times based on the data provided by the DataProvider.
- @Factory: It is annotation used by the TestNG to execute the test methods present in the same test class using different instances of the respective class.

29

How to create Group of Groups in TestNG?

29

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.

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

30

How can we create data driven framework using TestNG?

30

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

```
@DataProvider(name="getData")
public Object[][] getData(){
//Object [][] data = new Object [rowCount][colCount];
Object [][] data = new Object [2][2];
data [0][0] = "FirstUid";
data [0][1] = "FirstPWD";
data[1][0] = "SecondUid";
data[1][1] = "SecondPWD";
return data;
}
```

31

How to run test cases in parallel using TestNG?

31

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="softwaretestingmaterial" parallel="methods">
```

32

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

32

By using throw new SkipException()
Once SkipException() thrown, remaining part of that test method will not be executed and control will goes directly to next test method execution.

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

33

What does the test timeout mean in TestNG?

33

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 timeout of ten seconds guarantees that none of the threads will block on this thread forever.

34

How to set test case priority in TestNG?

34

We use priority attribute to the @Test annotations. In case priority is not set then the test scripts execute in alphabetical order.
@Test(priority=0)
public void testCase1() {
system.out.println("Test Case 1");
}

35

What is parameterization in TestNG?

35

In TestNG, parameterization runs a test method multiple times with different values. Another name for this process is data-driven testing in TestNG. We can acquire Parameterization in TestNG in two ways:
Firstly, we can achieve it through the XML file.
Secondly, we can achieve it through the dataproviders in TestNG.

36

What are the optional parameters in TestNG?

36

Optional parameters work similarly to the default case in the parameterization in TestNG. We use the optional parameter when no other parameter gets defined for that test case method. Additionally, the @Optional annotation declares the optional parameter. We don ' t define the @Optional parameter above the test method definition but alongside where the method is declared. Subsequently, the following code snippet demonstrates the declaration of the optional parameters in TestNG:

```
{
@Test
@Parameters ("message")
public void OP( @Optional("Optional Parameter Selected") String message)
{
System.out.println(message);
}
}
```

37

Where is the emailable report generated and saved in TestNG?

37

Emailable reports generate under the project folder and test-output subfolder. This report is available as “ emailable-report.html ” by default.

38

Where is the index report generated and saved in TestNG?

38

The index report generates under the project folder and test-output subfolder. Moreover, this report is available as “ index.html ” by default.

39

. How to create an XML file in TestNG?

39

Go to the src folder -> click on file ->enter the name of the file(mostly written testing.xml)
We have a blank XML file. Here, we have to mention the project name and the classes to be executed along with the package name as shown below.

```
<Suite name = "Testing project">
<test name = "testing feature 1">
<classes>
<class name = "packagename.name of class1"/>
<class name = "packagename.name of class1"/>
<class name = "packagename.name of class1"/>
<class name = "packagename.name of class1"/>
</classes>
</test>
</Suite>
```