

100 TESTNG QUESTIONS AND ANSWERS

Basic Concepts

Q: What is TestNG?

A: TestNG is a testing framework inspired by JUnit and NUnit but introduces new functionalities to make testing easier and more powerful.

Q: How do you install TestNG in Eclipse?

A: Go to Eclipse -> Help -> Eclipse Marketplace -> Search for TestNG -> Install.

Q: What are the main advantages of TestNG over JUnit?

A: Better annotations, support for parallel testing, flexible test configuration, and better reporting.

Q: Explain the @Test annotation.

A: The @Test annotation is used to mark a method as a test method in TestNG.

Q: How do you disable a test case in TestNG?

A: Use the enabled attribute in the @Test annotation: @Test(enabled = false).

Annotations

Q: List some commonly used TestNG annotations.

A: @BeforeSuite, @AfterSuite, @BeforeTest, @AfterTest, @BeforeClass, @AfterClass, @BeforeMethod, @AfterMethod, @Test.

Q: Explain the @BeforeMethod and @AfterMethod annotations.

A: @BeforeMethod runs before each test method, and @AfterMethod runs after each test method.

100 TESTNG QUESTIONS AND ANSWERS

Q: Explain the @BeforeSuite, @AfterSuite, @BeforeTest, @AfterTest, @BeforeClass, @AfterClass, @Test Annotations

A:

- @BeforeSuite: Executes before any suite of tests in TestNG.
 - @AfterSuite: Executes after all tests in a suite in TestNG.
 - @BeforeTest: Executes before any test method in TestNG.
 - @AfterTest: Executes after all test methods in TestNG.
 - @BeforeClass: Executes before the first test method in a class in TestNG.
 - @AfterClass: Executes after all test methods in a class in TestNG.
 - @Test: Denotes a test method in TestNG.
-

Q: What is the use of @DataProvider in TestNG?

A: @DataProvider is used to provide a set of data to a test method.

Q: How do you create a dependent test in TestNG?

A: Use the dependsOnMethods attribute in the

@Test annotation: @Test(dependsOnMethods = {"methodName"}).

Q: What is the purpose of @Factory annotation?

A: @Factory is used to create instances of test classes dynamically.

Configuration

Q: How do you configure TestNG to run with a testng.xml file?

A: Create a testng.xml file with suite and test tags, then run it as a TestNG suite in Eclipse.

Q: Explain the structure of a testng.xml file.

A: It includes <suite> and <test> tags, where <suite> can contain multiple <test> tags, and each <test> can contain multiple <classes> and <methods> tags.

Stay tuned with Sonu Madheshiya on LinkedIn <https://www.linkedin.com/in/sonumadheshiya/> for more interesting content on automation testing

100 TESTNG QUESTIONS AND ANSWERS

Q: How can you include or exclude groups in TestNG?

A: Use the include and exclude tags within the groups tag in the testng.xml file.

Q: What is the use of the priority attribute in TestNG?

A: The priority attribute is used to define the order in which the test methods should be executed.

Q: How do you run tests in parallel using TestNG?

A: Configure the parallel attribute in the testng.xml file with values like tests, classes, or methods.

Data Providers

Q: How do you pass parameters to test methods using @DataProvider?

A: Define a method annotated with @DataProvider that returns Object[][], then use dataProvider attribute in @Test annotation.

Q: Can @DataProvider be in a different class?

A: Yes, by specifying the class and method name in the dataProviderClass attribute of the @Test annotation.

Q: What is the return type of a @DataProvider method?

A: The return type is Object[][].

Q: How do you handle complex data types in @DataProvider?

A: Return a 2D array or collection of the desired complex data type.

100 TESTING QUESTIONS AND ANSWERS

Q: How can you use a `@DataProvider` for multiple test methods?

A: Use the same `@DataProvider` method in multiple `@Test` methods by specifying the `dataProvider` attribute.

Assertions

Q: What is the purpose of assertions in TestNG?

A: Assertions are used to validate the expected results of a test.

Q: List some common TestNG assertion methods.

A: `assertEquals`, `assertTrue`, `assertFalse`, `assertNull`, `assertNotNull`, `assertSame`, `assertNotSame`.

Q: Explain `assertEquals`, `assertTrue`, `assertFalse`, `assertNull`, `assertNotNull`, `assertSame`, `assertNotSame`.

A:

- **`assertEquals`:** Verifies that two values are equal, throwing an `AssertionError` if they are not.
 - **`assertTrue`:** Verifies that a condition is true, throwing an `AssertionError` if it is not.
 - **`assertFalse`:** Verifies that a condition is false, throwing an `AssertionError` if it is not.
 - **`assertNull`:** Verifies that an object is null, throwing an `AssertionError` if it is not.
 - **`assertNotNull`:** Verifies that an object is not null, throwing an `AssertionError` if it is null.
 - **`assertSame`:** Verifies that two references point to the same object, throwing an `AssertionError` if they do not.
 - **`assertNotSame`:** Verifies that two references do not point to the same object, throwing an `AssertionError` if they do.
-

Q: How do you verify a condition without stopping the test execution?

A: Use the soft `Assert` class which allows you to collect all errors and report them at the end.

Q: What is the difference between `assert` and `verify` in TestNG?

A: `Assert` stops test execution if the condition fails, while `verify` (using soft assertions) collects failures and continues execution.

100 TESTNG QUESTIONS AND ANSWERS

Q: How do you use soft assertions in TestNG?

A: Create an instance of SoftAssert, call assert methods on it, and finally call `assertAll()` to report all collected errors.

Listeners and Reports

Q: What are TestNG Listeners?

A: Listeners are interfaces that allow you to modify the behavior of TestNG and listen to events like test start, finish, pass, fail, etc.

Q: Name some TestNG listeners.

A: `ITestListener`, `ISuiteListener`, `IInvokedMethodListener`, `IAnnotationTransformer`.

Q: How do you implement a TestNG listener?

A: Implement the desired listener interface and override its methods.

Q: How do you add a listener to your TestNG tests?

A: Use the `@Listeners` annotation or specify it in the `testng.xml` file.

Q: What is `IReporter` in TestNG?

A: `IReporter` is an interface used to generate custom reports by implementing the `generateReport` method.

TestNG Suite

Q: How do you create a TestNG suite?

A: Create a `testng.xml` file with `<suite>` and `<test>` tags and specify the classes and methods to be included.

100 TESTNG QUESTIONS AND ANSWERS

Q: What is the significance of <suite> and <test> tags in testng.xml?

A: <suite> defines a group of tests, and <test> defines a single test which can include multiple classes and methods.

Q: How do you run a subset of tests in a suite?

A: Use the include and exclude tags within the <methods> tag in the testng.xml file.

Q: Can you define multiple suites in a single testng.xml file?

A: No, a single testng.xml file can define only one <suite> tag.

Q: How do you parameterize a suite in TestNG?

A: Use the <parameter> tag within the <suite> or <test> tags in the testng.xml file.

Dependency Testing

Q: What is dependency testing in TestNG?

A: Dependency testing allows you to specify dependencies between test methods, so a test method can be skipped if its dependent method fails.

Q: How do you create a test dependency in TestNG?

A: Use the dependsOnMethods or dependsOnGroups attribute in the @Test annotation.

Q: What happens if a dependent test method fails?

A: The dependent test method will be skipped.

100 TESTNG QUESTIONS AND ANSWERS

Q: Can a test method depend on multiple methods?

A: Yes, list the methods in the dependsOnMethods attribute separated by commas.

Q: How do you group dependent tests in TestNG?

A: Use the dependsOnGroups attribute to specify group dependencies.

TestNG Parameters

Q: How do you pass parameters to test methods in TestNG?

A: Use the @Parameters annotation and define parameters in the testng.xml file.

Q: How do you handle optional parameters in TestNG?

A: Use the @Optional annotation to specify a default value for the parameter.

Q: How do you pass parameters to a @DataProvider method?

A: Define parameters in the testng.xml file and retrieve them using the @Parameters annotation in the method that calls the @DataProvider.

Q: Can you pass parameters to a constructor in TestNG?

A: Yes, use the @Parameters annotation with the constructor.

Q: How do you pass multiple parameters to a test method?

A: Define multiple <parameter> tags in the testng.xml file and list the parameters in the @Parameters annotation.

Parallel Testing

Q: What is parallel testing in TestNG?

A: Parallel testing allows you to run multiple tests simultaneously to reduce execution time.

100 TESTNG QUESTIONS AND ANSWERS

Q: How do you enable parallel testing in TestNG?

A: Set the parallel attribute in the testng.xml file to tests, classes, or methods.

Q: What is the threadCount attribute in TestNG?

A: threadCount specifies the number of threads to be used for parallel execution.

Q: Can you run methods in parallel within the same class?

A: Yes, set parallel to methods in the testng.xml file.

Q: How do you handle thread safety in parallel testing?

A: Ensure that shared resources are synchronized or use ThreadLocal variables.

Advanced Topics

Q: What is the use of the @Factory annotation in TestNG?

A: @Factory is used to create instances of test classes dynamically, allowing for different test data setups.

Q: Explain the use of @Listeners annotation.

A:

Q: What is the @Parameters annotation used for?

A: @Parameters is used to pass parameter values to test methods through the testng.xml file.

Q: Explain the @Optional annotation.

A: @Optional is used to specify a default value for a parameter in case it is not provided in the testng.xml file.

Stay tuned with Sonu Madheshiya on LinkedIn <https://www.linkedin.com/in/sonumadheshiya/> for more interesting content on automation testing

100 TESTNG QUESTIONS AND ANSWERS

Q: How can you create a custom annotation in TestNG?

A: Create a new annotation by defining it with the `@interface` keyword and then use it in combination with a custom implementation of `IAnnotationTransformer` to modify test behavior.

Q: What is the use of the `@BeforeSuite` annotation?

A: `@BeforeSuite` is used to define a method that should run before all tests in the suite are executed.

Q: Explain the use of the `@AfterSuite` annotation.

A: `@AfterSuite` is used to define a method that should run after all tests in the suite have completed.

TestNG XML Configuration

Q: How do you include or exclude tests in `testng.xml`?

A: Use the `<include>` and `<exclude>` tags within the `<methods>` tag to specify which tests to include or exclude.

Q: What is the purpose of the `<listeners>` tag in `testng.xml`?

A: The `<listeners>` tag is used to specify listener classes that should be applied to the test suite.

Q: How do you define groups in `testng.xml`?

A: Use the `<groups>` tag to define and configure groups, and specify included or excluded groups within `<run>` tag.

Q: Explain the use of `<parameter>` tag in `testng.xml`.

A: The `<parameter>` tag is used to pass parameters to test methods defined in the `@Parameters` annotation.

100 TESTNG QUESTIONS AND ANSWERS

Q: How do you set a time-out for a test suite in testng.xml?

A: Use the time-out attribute in the <suite> or <test> tag to specify a maximum time for test execution.

TestNG Assertions and Verifications

Q: How do you use the assertEquals method in TestNG?

A: Use assertEquals(actual, expected) to compare the actual and expected values.

Q: What is the difference between assertTrue and assertFalse?

A: assertTrue checks if a condition is true, while assertFalse checks if a condition is false.

Q: How do you use assertNotNull in TestNG?

A: Use assertNotNull(object) to verify that an object is not null.

Q: Explain the use of assertSame in TestNG.

A: assertSame verifies that two object references point to the same object.

Q: How do you use the fail method in TestNG?

A: Use fail(message) to mark a test as failed with a specific message.

TestNG Listeners and Reporting

Q: What is ITestContext in TestNG?

A: ITestContext provides information about the test run and allows interaction with the test context.

Q: How do you use ITestListener in TestNG?

A: Implement the ITestListener interface and override its methods to perform actions on test events.

100 TESTNG QUESTIONS AND ANSWERS

Q: What is the use of IAnnotationTransformer in TestNG?

A: IAnnotationTransformer allows you to modify test annotations at runtime before they are executed.

Q: How do you generate HTML reports in TestNG?

A: TestNG generates HTML reports automatically in the test-output folder after test execution.

Q: What is the EmailableReporter in TestNG?

A: EmailableReporter generates a simple HTML report that can be emailed to stakeholders.

Parallel Execution

Q: How do you configure methods to run in parallel?

A: Set parallel="methods" in the <suite> tag of the testng.xml file and specify thread-count.

Q: How do you configure tests to run in parallel?

A: Set parallel="tests" in the <suite> tag of the testng.xml file and specify thread-count.

Q: What is the impact of parallel execution on test data?

A: Parallel execution may lead to data inconsistency if tests share the same data. Use thread-safe techniques to manage data.

Q: How do you handle synchronization issues in parallel tests?

A: Use synchronization mechanisms like synchronized blocks or locks to manage shared resources.

Q: How can you run test classes in parallel?

A: Set parallel="classes" in the <suite> tag of the testng.xml file and specify thread-count.

100 TESTNG QUESTIONS AND ANSWERS

Parameterization

Q: How do you pass parameters from testng.xml to a test method?

A: Define parameters in testng.xml using <parameter> tags and retrieve them in the test method using @Parameters annotation.

Q: How do you use @Optional with @Parameters?

A: @Optional provides a default value for a parameter if it is not specified in the testng.xml file.

Q: Can you use @Parameters with a @BeforeClass method?

A: Yes, @Parameters can be used with @BeforeClass to pass parameters before any test methods are executed.

Q: How do you use @Parameters with a @DataProvider?

A: @Parameters can be used to pass parameters to a @DataProvider method, which then supplies the data to test methods.

Q: How do you handle multiple parameters in TestNG?

A: Define multiple <parameter> tags in the testng.xml file and list the parameter names in the @Parameters annotation.

TestNG Integration

Q: How do you integrate TestNG with Maven?

A: Add the TestNG dependency to the pom.xml file and configure the surefire plugin to run TestNG tests.

Q: How do you integrate TestNG with Jenkins?

A: Use the TestNG plugin in Jenkins to run TestNG test suites and generate reports.

Stay tuned with Sonu Madheshiya on LinkedIn <https://www.linkedin.com/in/sonumadheshiya/> for more interesting content on automation testing

100 TESTNG QUESTIONS AND ANSWERS

Q: How do you integrate TestNG with Ant?

A: Add TestNG tasks to the build.xml file and configure the Ant script to run TestNG tests.

Q: How do you use TestNG with Selenium?

A: Use TestNG to manage and execute Selenium WebDriver test scripts, allowing for better test organization and reporting.

Q: How do you generate custom reports in TestNG?

A: Implement the IReporter interface to create custom HTML or XML reports based on test results.

TestNG Groups

Q: What is a test group in TestNG?

A: A test group is a way to categorize test methods so they can be included or excluded as a group during execution.

Q: How do you create a test group in TestNG?

A: Use the groups attribute in the @Test annotation to assign methods to a group.

Q: How do you include and exclude groups in testng.xml?

A: Use the <groups> tag with <run> and <include>/<exclude> tags to specify which groups to run or skip.

Q: Can you assign a method to multiple groups?

A: Yes, list multiple group names in the groups attribute of the @Test annotation separated by commas.

100 TESTNG QUESTIONS AND ANSWERS

Q: How do you use group dependencies in TestNG?

A: Use the dependsOnGroups attribute in the @Test annotation to specify group dependencies.

Exception Testing

Q: How do you handle expected exceptions in TestNG?

A: Use the expectedExceptions attribute in the @Test annotation to specify the exceptions that a test method is expected to throw.

Q: How do you test for multiple expected exceptions?

A: List multiple exception classes in the expectedExceptions attribute.

Q: Can you specify a message for expected exceptions?

A: No, expectedExceptions attribute only checks for the exception type, not the message.

Q: How do you verify an exception message in TestNG?

A: Use a try-catch block within the test method and assert the exception message using assertion methods.

Q: What is the use of expectedExceptionsMessageRegExp attribute?

A: expectedExceptionsMessageRegExp is used to specify a regular expression to match the exception message.

Miscellaneous

Q: How do you skip a test in TestNG?

A: Use the SkipException class to throw a SkipException and skip the test dynamically.

100 TESTNG QUESTIONS AND ANSWERS

Q: How do you retry failed tests in TestNG?

A: Implement the IRetryAnalyzer interface and override the retry method, then associate it with @Test using the retryAnalyzer attribute.

Q: How do you run TestNG tests from the command line?

A: Use the command: `java -cp <path-to-testng-jar>:<path-to-your-tests> org.testng.TestNG <path-to-testng.xml>`.

Stay tuned with Sonu Madheshiya on LinkedIn <https://www.linkedin.com/in/sonumadheshiya/> for more interesting content on automation testing