TestNG Vs JUnit – What's the Difference?

TestNG and **JUnit** are two very popular terms spoken amongst the testers simply because they are used very heavily in the community. They are the most popular testing frameworks for Java applications. Both TestNG and Junit are Testing frameworks used for Unit Testing.

Let's start with the foundation of these frameworks i.e Unit Testing.

What is Unit Testing?

- Before starting with the differences between TestNG and JUnit, we should understand what is meant by unit testing. A unit test is a way of testing a unit – the smallest piece of code that can be logically isolated in a system. A unit is the smallest testable part of an application like functions, classes, or procedures.
- The word "unit" is used very popularly in two major fields: mathematics and budgeting. In mathematics, the unit means "1*" like I traveled a unit distance from my home, etc. Similarly, in budgeting, we say "unit" when we want to refer to a single element of which many are available. For example, I bought eighty umbrellas for $3 per unit. So literally, the term unit refers to singularity.
- Unit testing is done by putting the values by the tester and can be associated with the functions of a program or a single module, etc. The goal of unit testing is to segregate each part of the program and test that the individual parts are working correctly. Unit testing helps in the faster development of software, easing out the debugging process and making the code reusable.

What is JUnit?

A widely used Java framework, JUnit, is handy for running and writing tests. JUnit is used for unit testing in Java and gets its name from the combination of Java + unit testing.

- JUnit is an open-source framework, which is used for writing and running tests.
- Provides annotations to identify test methods.
- Provides assertions for testing expected results.
- It Provides test runners for running tests.
- JUnit is non-supportive of parallel test execution.
- JUnit aids the development of test-driven programming in a big way.

What is TestNG?

TestNG is another Java framework wherein tests have to be organized in classes. TestNG takes care of the limitations of JUnit through extra

functionalities and advanced annotations, especially those that do not find support in JUnit.

- It is an open-source automated TestNG framework. In TestNG, NG stands for **“Next Generation”**.
- TestNG reuses similar test class instances (just like Junit) for all kinds of test methods.
- TestNG offers a lot of flexibility in the manner in which it passes parameters into different unit tests.
- It permits many threads to run at the same time and simultaneously.
- Another feature that keeps this framework apart is its ability to put test methods in Java groups.
- In addition, The TestNG framework performs unit testing, end-to-end testing, integration testing, etc.

TestNg Vs JUnit:

Annotations In JUnit and TestNG

Here are some similar Annotations which work similar in both the JUnit and TestNG but their declaration is different –

Purpose	JU
Execute Before Each Test Method	@Be
Execute After Each Test Method	@A
Ignore Test	@ig

Similar Annotations (TestNG Vs JUnit)

Additional TestNG Annotations, these advanced annotations are not found in Junit –

- **@BeforeSuite** – It will run only once before all tests in the suite are executed.
- **@AfterSuite** – A method with this Annotation will run once after the execution of all tests in the suite is complete.
- **@BeforeTest** – This will be executed before the first @Test annotated method. It can be executed multiple times before the Test case.
- **@AfterTest** – A method with this Annotation will be executed when all the @Test annotated methods complete the execution of those classes inside the tag in the TestNG.xml file.
- **@BeforeGroups** – This method will run before the first test run of that specific group.

- **@AfterGroups** -This method will run after all test methods of that group complete their execution.
- **@DataProvider** – This annotation enables us to run a test method multiple times by passing different data sets.

JUnit:

```
@BeforeClass

public static void MethodName() {

   //code

}
```

TestNG:

```
@BeforeClass

public void MethodName() {

    //code

}
```

Test case Management in TestNG and JUnit frameworks

Managing test case execution is a lot easier in TestNG in comparison to JUnit. In TestNG, QA engineers can group tests, ignore tests, parameterize tests, and write dependent tests in an efficient manner.

1. **Group Test in JUnit and TestNG** – Group test means performing the tasks on a group of tests altogether. It is a new feature and comes with TestNG only. With group tests, the tester can create multiple groups with each group containing multiple tests in them. Additionally, the tester can then execute the test in a single group rather than various tests.
2. **Ignore Test in JUnit and TestNG** – The ignore test feature means whether the framework should ignore the test during execution or not. Both TestNG and JUnit contain this feature, but the declaration is different.

TestNG:

```
@Test(enabled=false)

public void TestWithException()

{

        //code

}
```

JUnit:

```
@Ignore

public void method1()

{

  //code

}
```

3. **Parameterized Test in JUnit and TestNG** – Parameterized tests mean running a test case by feeding different values at each execution. These tests are beneficial in data-driven testing and reduce the code length to a good extent. Both JUnit and TestNG contain the benefit of parameterized testing but in a slightly different manner.

4. **Dependent Tests in JUnit and TestNG** – The dependent test feature refers to providing a dependency of a test method onto another test method. JUnit currently does not support dependent tests while TestNG does. The main reason for this drawback is that the development of TestNG happened for a wider variety of test types, unlike JUnit, which supports only unit testing.

```
@Test (dependsOnMethods = { "Launch" })
```

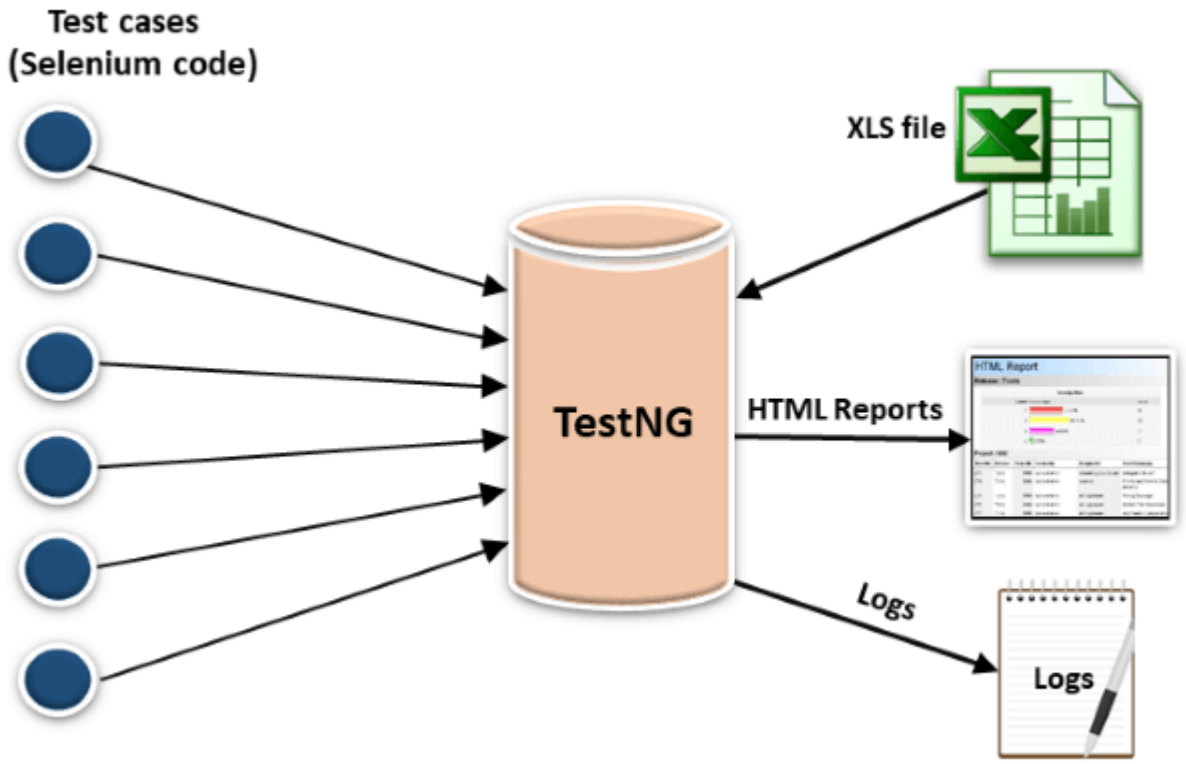
```
public void login() {

    System.out.println("Login to the portal");

}
```

Here, the test method login() is dependent on the Launch() method.

Advantages of TestNG over JUnit



- In TestNG, annotations are easier to understand than Junit.
- It produces the HTML reports for implementation.
- It also generates the Logs.
- TestNG enables you to group the test cases easily which is not possible in JUnit.
- TestNG supports three additional levels such as @Before/After suite, @Before/AfterTest, and Before/AfterGroup.
- It does not extend any class. Therefore, the TestNG framework allows you to define the test cases where each test case is independent of other test cases.
- Parallel execution of test cases, i.e., running multiple test cases is only possible in the TestNG framework.

Conclusion

Conclusively, TestNG, and JUnit is popular frameworks used in the testing world. TestNG was born after JUnit, with the main focus being the current tester's requirements and overcoming the anomalies of JUnit. It is, therefore, quite implicitly proven fact that TestNG would be more powerful in TestNG Vs JUnit. Additionally, TestNG can perform a bunch of testing, such as unit testing, integration testing, end-to-end testing, etc. Whereas, JUnit is used only for unit testing.