TESTNG - OVERVIEW

Testing is the process of checking the functionality of an application to ensure it works as per requirements.Unit testing comes into picture at the developer level where adequate measures are taken to test every single entity *classormethod* to ensure the final product meets the requirements.

JUnit has driven developers to understand the usefulness of tests, especially of unit tests, when compared to any other testing framework. Leveraging a rather simple, pragmatic, and strict architecture, JUnit has been able to "infect" great number of developers. Do take a look at our tutorial on JUnit to have a good understanding of its features. JUnit, at the same time, has some shortcomings as well, which are listed below:

- Initially designed to enable unit testing only, now used for all kinds of testing.
- Cannot do dependency testing.
- Poor configuration control *setUp*/*tearDown*.
- Intrusive *forcesyoutoextendclassesandnameyourmethodsacertainway*.
- Static programming model *forcesyoutorecompileunnecessarily*.
- The management of different suites of tests in complex projects can be very tricky.

What is TestNG?

Definition of TestNG as per its documentation is as follows:

TestNG is a testing framework inspired from JUnit and NUnit, but introducing some new functionalities that make it more powerful and easier to use.

TestNG is an open source automated testing framework; where **NG** means **N**ext **G**eneration. TestNG is similar to JUnit *especiallyJUnit*4, but it is not a JUnit extension. It is inspired by JUnit. It is designed to be better than JUnit, especially when testing integrated classes. The creator of TestNG is *Cedric Beust*.

Eliminating most of the limitations of the older framework, TestNG gives the developer the ability to write more flexible and powerful tests. As it heavily borrows from Java Annotations *introducedwithJDK*5.0 to define tests, it can also show you how to use this new feature of the Java language in a real production environment.

TestNG Features

- Supports annotations.
- TestNG uses more Java and OO features.
- Supports testing integrated classes *e. g.* , *bydefault*, *noneedtocreateanewtestclassinstanceforeverytestmethod*.
- Separates compile-time test code from run-time configuration/data info.
- Flexible runtime configuration.
- Introduces ‘test groups’. Once you have compiled your tests, you can just ask TestNG to run all the "front-end" tests, or "fast", "slow", "database" tests, etc.
- Supports Dependent test methods, parallel testing, load testing, and partial failure.
- Flexible plug-in API.
- Support for multi threaded testing.