Why TestNG and its advantages

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

* Annotations.
* Run your tests in arbitrarily big thread pools with various policies available (all methods in their own thread, one thread per test class, etc...).
* Test that your code is multithread safe.
* Flexible test configuration.
* Support for data-driven testing (with @DataProvider).
* Support for parameters.
* Powerful execution model (no more TestSuite).
* Supported by a variety of tools and plug-ins (Eclipse, IDEA, Maven, etc...).
* Embeds BeanShell for further flexibility.
* Default JDK functions for runtime and logging (no dependencies).
* Dependent methods for application server testing.

TestNG is designed to cover all categories of tests:  unit, functional, end-to-end, integration, etc...

Documentation: <https://testng.org/doc/documentation-main.html#parallel-running>

Installation from Update site through Help / Install New Software in Eclipse : <https://testng.org/doc/download.html>

Running testcases in TestNG

Importance of TestNG.xml file in execution-concept of suite, test (or test folder/module) in xml file class, methods in xml file related to java class and methods

Include & Exclude to control test cases by name or regex

Execute from package level

@BeforeSuite,@AfterSuite, @BeforeTest,@AfterTest, @BeforeMethod,@AfterMethod etc

Helper attributes: dependsOnMethods,enabled,dependsOnMethods="login", expectedExceptions=ArithmeticException.**class**, invocationCount=10, timeOut=1000, verbose=”5”

Global data using @Parameters

Local data using @DataProviders

Listeners interface

Running tests in parallel