

13-03-2017

TestNG: Test Next Generation is an ~~unit~~ testing (White Box Testing) framework. Basically used by developers to perform WBT. It is also used by selenium automation engineers to run multiple test classes and to generate execution result. Automation engineers call TestNG as unit testing tool.

Installing TestNG: In eclipse go to Help Menu >> Eclipse Marketplace >> Search for TestNG >> click Install button of TestNG. >> follow the default instructions till finish which will restart the eclipse.

→ Right Click on the Java Project >> Go to Build path >> Select add libraries >> Select testNG >> Click Next >> Click Finish.

//_

TestNG Class :- \rightarrow TestNG class is a java class which contains Test method

\hookrightarrow Any method developed using Test annotation (@Test) is called as Test method.

Ex:
package qsp;
import org.testng.Reporter;
import org.testng.annotations.Test;
public class DemoA {
 @Test
 public void testA() {
 Reporter.log("Executing testA", true);
 }
}

When we run the above class, it automatically generates the result class in html format. To see the result refresh java project, expand 'test-output' folder, right click on 'emailable-report.html', open with \gg select WebBrowser.

While developing testNG class, do not use ; default package, main method & S.O.P.

To run multiple testNG classes, we use TestNG suite.

To create it ; right click on java project ; go to TestNG, select the ~~new~~ to Convert to TestNG Suite \gg Click Finish.

Double click on xml file to view its content & remove the comments.

XML

Ex:

```
<suite name = "Suite" parallel = "none">
  <test name = "Test">
    <classes>
      <class name = "gsp.DemoA"/>
      <class name = "gsp.DemoB"/>
    </classes>
  </test>
</Suite>
```

To Execute it, Go to project; testing.xml >> right click >> Run as >> select testNG suite.

NOTE: a) In a same TestNG class, we can have multiple test methods, but they will be executed in alphabetical order.

b) To execute them in required order, we use priority, default priority is zero(0); It can be negative number also, but decimal numbers are not allowed. Variables are also not allowed.

c) It always executes in ascending order of priority, if two methods have same priority then those methods will be executed in alphabetical order.

To run the test method multiple times, we use invocation count, default value is 1(one). If it is zero or negative number, It will not execute the test method. Here also variable & decimal numbers are also not allowed.

Important Annotations of TestNG:

- 1> @BeforeSuite
- 2> @BeforeTest
- 3> @BeforeClass
- 4> @BeforeMethod
- 5> @Test
- 6> @AfterMethod
- 7> @AfterClass
- 8> @AfterTest
- 9> @AfterSuite

BaseTest.java

```
Ex: public class BaseTest {  
    @BeforeSuite  
    public void start() { Reporter.log("start", true); }  
    @AfterSuite  
    public void end() { Reporter.log("end", true); }  
    @BeforeTest  
    public void connectDB() { Reporter.log("ConnectDB", true); }  
    @AfterTest  
    public void disconnectDB() { Reporter.log("disconnectDB", true); }  
    @BeforeClass  
    public void openApp() { Reporter.log("openApp", true); }  
    @AfterClass  
    public void closeApp() { Reporter.log("closeApp", true); }  
    @BeforeMethod  
    public void login() { Reporter.log("login", true); }  
    @AfterMethod  
    public void logout() { Reporter.log("logout", true); }  
}
```


DemoA.java

```
public class DemoA extends BaseTest {  
    @Test (priority = 0, invocationCount = 2)  
    public void registerUser () {  
        Reporter.log ("registerUser", true);  
    }  
    @Test (priority = 1)  
    public void deleteAllUsers () {  
        Reporter.log ("deleteAllUsers", true);  
    }  
}
```

DemoB.java

```
public class DemoB extends BaseTest {  
    @Test (priority = 2)  
    public void testB () {  
        Reporter.log ("Executing testB", true);  
    }  
}
```

testng.xml

```
<suite name="Suite">  
    <test name = "Test1">  
        <classes>  
            <class name = "qsp.DemoA" />  
        </classes>  
    </test>  
    <test name = "Test2">  
        <classes>  
            <class name = "qsp.DemoB" />  
        </classes>  
    </test>  
</suite>
```