Comparative Analysis of JUnit and TestNG framework

Manasi Patil[1], Mona Deshmukh[2]

Student, Dept. of MCA, VES Institute of Technology, Maharashtra, India[1]
Professor, Dept. of MCA, VES Institute of Technology, Maharashtra, India[2]

Abstract - *Testing is an important phase in SDLC, Testing can be manual or automated. Nowadays Automation testing is widely used to find defects and to ensure the correctness and completeness of the software. Open source framework can be used for automation testing such as Robotframework, Junit, Spock, NUnit, TestNG, Jasmin, Mocha etc. This paper compares JUnit and TestNG based on their features and functionalities.*

***Key Words*: JUnit, TestNG, Automation Framework, Automation Testing, Selenium TestNG, Selenium JUnit**

1. INTRODUCTION

Testing a software manually is tedious work, and takes lot of time and efforts. Automation saves lot of time and money, also it increases the test coverage and improves accuracy. It helps developers as well as testers. Choosing the right automation framework is crucial, so that it helps different kind of testing such as unit, functional, regression, volume or data driven testing. While choosing the right framework, it is important to take care of some factors such as framework should be reusable, and code should be maintainable, proper organisation and readability of the code, time optimization in terms of writing Tests and framework should have reporting mechanism. All frameworks are different and covers different key factors which are compared below.

1.1 Framework

Framework is a collection of rule or guidelines to interact with components which are interlinked. These components are responsible for the creation and execution of automated Tests and reporting for results after Test execution.

2. JUnit

It is popular open source java based light weight framework designed by Kent Beck, Erich Gamma and it is hosted at Github. It is test driven development (TDD) used for writing and running Tests. It uses annotations for identifying Tests, which are written as methods. It is a part of XUnit, Unit testing family. The current version used is Junit 5.

3. TestNG

It is an open source framework where NG stands for Next Generation in TestNG. This framework is inspired by JUnit and NUnit, but it overcomes the JUnit flaws. TestNG is user friendly and it covers unit, integration and functional testing. It is flexible than Junit and supports parametrization, parallel execution and data driven testing.

The Table – 1 compares different functionalities of TestNG and JUnit framework.

Table -1: Functionality TestNG vs JUnit

Functionality	TestNG	JUnit
Support Annotations	Yes	Yes
Support Test Suit Initialization	Yes	Yes
Support for Tests groups	Yes	No
Support Parametrized Test	Yes	Limited support
Support Exception Test	Yes	Yes
Support Dependency Test	Yes	No
Support Ignore or Disable Test	Yes	Yes
Support Parallel Testing	Yes	No
Support Maven	Yes	Yes
Support Data driven approach or dynamic input	Yes with Data provider	No
Support for Tests Reports	Yes	No
IDE support	Yes	Yes

4. Annotations

In simple language annotations are metadata for the code, i.e. data about the data. Annotations stats with '@' followed by annotation name. They are introduced java JDK 5.0 onwards. They are applied before the method. Complier checks for annotations and consider the following method as overridden method. Complier checks for existence of the same method in the parent class otherwise throws a compiler error. Annotations does not directly affect the program execution but provide compile time instructions to the complier.

The Table – 2 gives the list of different annotations of TestNG and JUnit framework for similar functionalities.

Table - 2: Annotations Support TestNG vs JUnit

Annotations	TestNG Annotations	JUnit Annotations
For test annotations	@Test	@Test
Applied to methods to run before all tests in a Test Suite	@BeforeSuite	-
Applied to methods to run after all tests in a Test Suite	@AfterSuite	-
Applied to methods to run before the test	@BeforeTest	-
Applied to methods to run after the test	@AfterTest	-
Applied to methods to run before the first Tests from the same group	@BeforeGroups	-
Applied to methods to run after the first Tests from the same group	@AfterGroups	
Applied to methods to run before the first method in a class is invoked	@BeforeClass	@BeforeClass
Applied to methods to run after the first method in a class is invoked	@AfterClass	@AfterClass
Applied to the methods to run before each test method	@BeforeMethod	@Before
Applied to the methods to run after each test method	@AfterMethod	@After
To ignore a test method	@ignore	@Test(enabled = false)
Timeout for test method	@Test(timeout = 1000)	@Test(timeout = 1000)

5. Annotations Syntax

Following are JUnit code snippets for above annotations:

```
public class JUnitAnnotations {

@BeforeClass
   public static void BeforeClass_Method() {
     System.out.println("Annotation @BeforeClass - before all test cases ");
}
@Before
   public void Before_Method() {
      System.out.println("Annotation @Before - before each test cases ");
   }
 @AfterClass
    public static void AfterClass_Method() {
     System.out.println("Annotation @AfterClass - after all test cases");
   }
 @After
    public void After_Method() {
      System.out.println("Annotation @After - after each test cases");
   }
  @Test
     public void Test_Method(int a, int b){
      int add = a+b;
      assertEquals (add,10,"Assertion is correct");
   }
  @Ignore
     public void Ignore_Method() {
      System.out.println("Annotation @Ignore - this method is ignored");
   }
  @Test(timeout = 1000)
     Public void Timeout_Method() {
      System.out.println("Annotation @Test(timeout) - enforce timeout in
test case");
    }
   @Test(expected = ArithmeticException.class)
   public void Expected_Method() {
      System.out.println("Annotation @Test(expected) - test for specified
exception during its execution");
      int i = 10/0;
   } }
```

Following are TestNG code snippets for above annotations:

```
public class TestNGAnnotations {

@BeforeGroups("Regression")
    public void BeforeGroups_Method() {
     System.out.println("Annotation @BeforeGroups - before test executed
from same group");
}
@AfterGroups("Regression")
   public void AfterGroups_Method() {
    System.out.println("Annotation @AfterGroups - after test executed from
same group");
}
@BeforeClass
   public void BeforeClass_Method() {
    System.out.println("Annotation @BeforeClass - before all test cases ");
}
@AfterClass
   public static void AfterClass_Method() {
      System.out.println("Annotation @AfterClass - after all test cases");
}
@BeforeMethod
    public void Before_Method() {
      System.out.println("Annotation @Before - before each test cases ");
}
@AfterMethod
   public void After_Method() {
      System.out.println("Annotation @After - after each test cases");
}
@Test(groups = "Regression")
   public void Test_Method1(){
      System.out.println("@Test - Included in the Regression group");
}
@Test
    public void Test_Method2() {
     System.out.println("@Test - Not included in any group");
}
@Test(timeout = 1000)
    public void Timeout_Method() {
      System.out.println("Annotation @Test(timeout) - enforce timeout in
test case");
}
@Test(enabled=false)
   public void Ignore_Method() {
      System.out.println("This method is ignored");
   }

@Test(expected = ArithmeticException.class)
   public void Expected_Method() {
      System.out.println("Annotation @Test(expected) - test for specified
exception during its execution");
      int i = 10/0;
   }
}
```

3. CONCLUSIONS

As per the above study both the frameworks are good for automation testing. Though JUnit is most popular and widely used testing framework, it does not offer much functionalities as compared to TestNG such as grouping, parallelism, dataProvider/ parametrization, reporting. TestNG provides better flexibility than JUnit.

REFERENCES

[1] http://toolsqa.com/selenium-webdriver/testng-tutorial

[2] https://en.wikipedia.org/wiki/JUnit

[3] http://toolsqa.com/java/junit-framework/junit-introduction

[4] http://www.java2novice.com/junit-examples/junit-annotations

[5] https://www.tutorialspoint.com/testng/testng_basic_annotations.htm