Fun Doo
TESTERS

Everything About TestNG

Part 1

Like

Swipe for more

Share

Why TestNG?

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.

Example

```java
package com.fundootesters;

import org.testng.annotations.*;

public class Example {

 @BeforeClass
 public void setUp() {
   // code that will be invoked when this test is instantiated
 }

 @Test(groups = { "smoke" })
 public void aFastTest() {
   System.out.println("smoke test");
 }

 @Test(groups = { "regression" })
 public void aSlowTest() {
    System.out.println("regression test");
 }
```

Annotations

- **@BeforeSuite:** The annotated method will be run before all tests in this suite have run.
- **@AfterSuite:** The annotated method will be run after all tests in this suite have run.
- **@BeforeTest:** The annotated method will be run before any test method belonging to the classes inside the <test> tag is run.
- **@AfterTest:** The annotated method will be run after all the test methods belonging to the classes inside the <test> tag have run.
- **@BeforeGroups:** The list of groups that this configuration method will run before. This method is guaranteed to run shortly before the first test method that belongs to any of these groups is invoked.
- **@AfterGroups:** The list of groups that this configuration method will run after. This method is guaranteed to run shortly after the last test method that belongs to any of these groups is invoked.
- **@BeforeClass:** The annotated method will be run before the first test method in the current class is invoked.
- **@AfterClass:** The annotated method will be run after all the test methods in the current class have been run.
- **@BeforeMethod:** The annotated method will be run before each test method.
- **@AfterMethod:** The annotated method will be run after each test method.

TestNG provides initialization and cleanup at the method and class level. While @BeforeClass and @AfterClass remain the same at the class level, the method level annotations are @BeforeMethod and @AfterMethod:

```java
@BeforeClass
public void initialize() {
    numbers = new ArrayList<>();
}

@AfterClass
public void tearDown() {
    numbers = null;
}

@BeforeMethod
public void runBeforeEachTest() {
    numbers.add(1);
    numbers.add(2);
    numbers.add(3);
}

@AfterMethod
public void runAfterEachTest() {
    numbers.clear();
}
```

TestNG also offers, @BeforeSuite, @AfterSuite, @BeforeGroup, and @AfterGroup annotations, for configurations at suite and group levels

```
@BeforeGroups("positive_tests")
public void runBeforeEachGroup() {
    numbers.add(1);
    numbers.add(2);
    numbers.add(3);
}

@AfterGroups("negative_tests")
public void runAfterEachGroup() {
    numbers.clear();
}
```

Fun Doo
TESTERS

If you find it useful
Support us by a little ❤

Keep learning with fun

Follow "**Fun Doo Testers**" for
Tutorials and fun

@Fundootesters