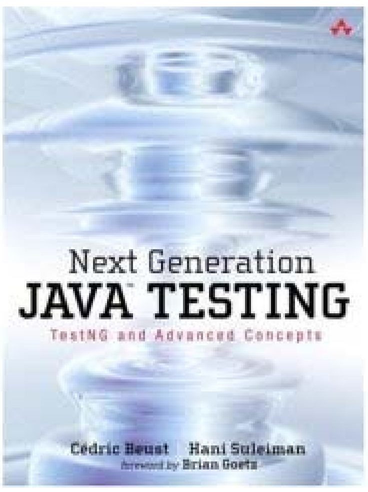
Welcome Download Documentation Migrating from JUnit JavaDoc Selenium

Eclipse IDEA Maven Ant Miscellaneous Book

# TestNG

## Now available



Click for more details.

*Cédric Beust (cedric at beust.com)*
*Current version: 6.7*
*Created: April 27th, 2004*
*Last Modified: July 14th, 2012*

TestNG的灵感来自JUnit和NUnit，其功能强大且易于使用。

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.

功能：
● 注解
● 线程池
● 多线程安全
● 方便的测试配置
● 数据驱动测试
● 参数支持
● 强大的执行模式
● 方法依赖测试

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

其被设计用来进行 单元、功能、集成测试 等

I started TestNG out of frustration for some JUnit deficiencies which I have documented on my weblog here and here Reading these entries might give you a better idea of the goal I am trying to achieve with TestNG.  You can also check out a quick overview of the main features and an article describing a very concrete example where the combined use of several TestNG's features provides for a very intuitive and maintainable testing design.

Here is a very simple test:

```
package example1;

import org.testng.annotations.*;

public class SimpleTest {

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

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

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

}
```

The method `setUp()` will be invoked after the test class has been built and before any test method is run.  In this example, we will be running the group fast, so `aFastTest()` will be invoked while `aSlowTest()` will be skipped.

Things to note:

- No need to extend a class or implement an interface.
- Even though the example above uses the JUnit conventions, our methods can be called any name you like, it's the annotations that tell TestNG what they are.
- A test method can belong to one or several groups.

Once you have compiled your test class into the `build` directory, you can invoke your test with the command line, an ant task (shown below) or an XML file:

```
<project default="test">

 <path id="cp">
   <pathelement location="lib/testng-testng-5.13.1.jar"/>
   <pathelement location="build"/>
 </path>

 <taskdef name="testng" classpathref="cp"
          classname="org.testng.TestNGAntTask" />

 <target name="test">
   <testng classpathref="cp" groups="fast">
     <classfileset dir="build" includes="example1/*.class"/>
   </testng>
 </target>

</project>
```

Use ant to invoke it:

```
c:> ant
Buildfile: build.xml

test:
[testng] Fast test
[testng] ===============================================
[testng] Suite for Command line test
[testng] Total tests run: 1, Failures: 0, Skips: 0
[testng] ===============================================


BUILD SUCCESSFUL
Total time: 4 seconds
```

Then you can browse the result of your tests:

```
start test-output\index.html (on Windows)
```

## Requirements

TestNG requires JDK 5 or higher.

## Mailing-lists

- The users mailing-list can be found on Google Groups.
- If you are interested in working on TestNG itself, join the developer mailing-list.
- If you are only interested in hearing about new versions of TestNG, you can join the low volume TestNG announcement mailing-list.

## Locations of the projects

If you are interested in contributing to TestNG or one of the IDE plug-ins, you will find them in the following locations:

- TestNG
- Eclipse plug-in
- IDEA IntelliJ plug-in
- NetBeans plug-in

## Bug reports

If you think you found a bug, here is how to report it:

- Create a small project that will allow us to reproduce this bug. In most cases, one or two Java source files and a `testng.xml` file should be sufficient. Then you can either zip it and email it to the testng-dev mailing-list or make it available on an open source hosting site, such as github or Google code and email `testng-dev` so we know about it. Please make sure that this project is self contained so that we can build it right away (remove the dependencies on external or proprietary frameworks, etc...).
- If the bug you observed is on the Eclipse plug-in, make sure your sample project contains the `.project` and `.classpath` files.
- File a bug.

For more information, you can either download TestNG, read the manual or browse the links at thetop.

## License

Apache 2.0