一、TestNG基本注解

|  |  |
| --- | --- |
| **注解** | **描述** |
| **@BeforeSuite** | 注解的方法将只运行一次，运行所有测试前此套件中。 |
| **@AfterSuite** | 注解的方法将只运行一次此套件中的所有测试都运行之后。 |
| **@BeforeClass** | 注解的方法将只运行一次先行先试在当前类中的方法调用。 |
| **@AfterClass** | 注解的方法将只运行一次后已经运行在当前类中的所有测试方法。 |
| **@BeforeTest** | 注解的方法将被运行之前的任何测试方法属于内部类的 <test>标签的运行。 |
| **@AfterTest** | 注解的方法将被运行后，所有的测试方法，属于内部类的<test>标签的运行。 |
| **@BeforeGroups** | 组的列表，这种配置方法将之前运行。此方法是保证在运行属于任何这些组第一个测试方法，该方法被调用。 |
| **@AfterGroups** | 组的名单，这种配置方法后，将运行。此方法是保证运行后不久，最后的测试方法，该方法属于任何这些组被调用。 |
| **@BeforeMethod** | 注解的方法将每个测试方法之前运行。 |
| **@AfterMethod** | 被注释的方法将被运行后，每个测试方法。 |
| **@DataProvider** | 标志着一个方法，提供数据的一个测试方法。注解的方法必须返回一个Object[] []，其中每个对象[]的测试方法的参数列表中可以分配。 该@Test 方法，希望从这个DataProvider的接收数据，需要使用一个dataProvider名称等于这个注解的名字。 |
| **@Factory** | 作为一个工厂，返回TestNG的测试类的对象将被用于标记的方法。该方法必须返回Object[]。 |
| **@Listeners** | 定义一个测试类的监听器。 |
| **@Parameters** | 介绍如何将参数传递给@Test方法。 |
| **@Test** | 标记一个类或方法作为测试的一部分。 |

属性：

    alwaysRun    对于每个bufore方法(beforeSuite, beforeTest, beforeTestClass 和 beforeTestMethod, 但是不包括 beforeGroups):

                如果设置为true，被配置的方法将总是运行而不管它属于哪个组。

                对于after方法(afterSuite, afterClass, ...): 如果设置为true，被配置的方法甚至在一个或多个先调用的方法失败或被忽略时也将运行。

    dependsOnGroups        这个方法依赖的组列表

    dependsOnMethods    这个方法依赖的方法列表

    enabled            这个类的方法是否激活

    groups            这个类或方法所属的分组列表

    inheritGroups        如果设置为true，这个方法被属于在类级别被@Test annotation指定的组

@DataProvider    标记一个方法用于为测试方法提供数据。

                被注释的方法必须返回Object[][], 其中每个Object[]可以指派为这个测试方法的参数列表。

                从这个DataProvider接收数据@Test方法需要使用一个和当前注释相同名称的dataProvider名称

    name         这个DataProvider的名称

@Factory    标记方法作为一个返回对象的工厂，这些对象将被TestNG用于作为测试类。这个方法必须返回Object[]

@Parameters    描述如何传递参数给@Test方法

    value    用于填充这个方法的参数的变量列表

@Test        标记一个类或方法作为测试的一部分

    alwaysRun     如果设置为true，这个测试方法将总是运行，甚至当它依赖的方法失败时。

    dataProvider     这个测试方法的data provider的名称

    dataProviderClass     用于查找data provider的类。

           如果不指定，将在当前测试方法所在的类或者它的基类上查找data provider。

           如果这个属性被指定, 则data provider方法需要是指定类的static方法。

    dependsOnGroups     当前方法依赖的组列表

    dependsOnMethods     当前方法依赖的方法列表

    description     当前方法的描述

    enabled     当前类的方法/方法是否被激活

    expectedExceptions     测试方法期望抛出的异常列表。如果没有异常或者抛出的不是列表中的任何一个，当前方法都将标记为失败.

    groups     当前类/方法所属的组列表

    invocationCount     当前方法被调用的次数

    successPercentage     当前方法期望的成功率

    sequential     如果设置为true，当前测试类上的所有方法保证按照顺序运行。甚至测试们在parallel="true"的情况下.

         这个属性只能用于类级别，如果用于方法级别将被忽略。

    timeOut     当前方法容许花费的最大时间，单位毫秒。

    threadPoolSize     当前方法的线程池大小。方法将被多线程调用，次数由invocationCount参数指定

        注意：如果invocationCount没有指定则这个属性将被忽略

注：

    上面是TestNG中用到的annotation列表，从中我们可以看到TestNG提供的一些特性

1. before方法和after方法    带来了足够丰富的测试生命周期控制

2. dependsOnGroups/dependsOnMethods 提供了依赖检查机制，并可以严格控制执行顺序

3. DataProvider 使得对同一个方法的测试覆盖变的非常轻松，非常适合进行边界测试，只要给出多种测试数据就可以针对一个测试方法进行覆盖

4. expectedExceptions 使得异常测试变的非常轻松

5. invocationCount/threadPoolSize 终于可以简单的直接进行多线程测试了，这个绝对是junit的超级弱项，回想junit中那个万恶的System.exist(0)...

6. timeOut 终于不用死等然后手工强行关闭测试，TestNG想的太周到了

二、测试套件

测试套件是用于测试软件程序的行为或一组行为的测试用例的集合。 在TestNG中，我们无法在测试源代码中定义一个套件，但它可以由一个XML文件表示，因为套件是执行的功能。 它还允许灵活配置要运行的测试。 套件可以包含一个或多个测试，并由<suite>标记定义。

<suite>是testng.xml的根标记。 它描述了一个测试套件，它又由几个<test>部分组成。

下表列出了<suite>接受的所有定义的合法属性。

|  |  |
| --- | --- |
| **属性** | **描述** |
| name | 套件的名称，这是一个强制属性。 |
| verbose | 运行的级别或详细程度。 |
| parallel | TestNG是否运行不同的线程来运行这个套件。 |
| thread-count | 如果启用并行模式(忽略其他方式)，则要使用的线程数。 |
| annotations | 在测试中使用的注释类型。 |
| time-out | 在本测试中的所有测试方法上使用的默认超时。 |

**1、**TestNG可以通过suite标签中的parallel属性，让测试方法运行在不同的线程中。

parallel的取值有三个：

①tests：TestNG 会在相同的线程中运行相同<test>标记下的所有方法，但是每个<test>下的方法会运行在不同的线程下。

应用：可以将非线程安全的类放在同一个<test>标签下，并且使其可以利用TestNG多线程的特性的同时，让这些类运行在相同的线程中。（不太理解）

②methods：TestNG 会在不同的线程中运行测试方法，除非那些互相依赖的方法。那些相互依赖的方法会运行在同一个线程中，并且遵照其执行顺序。

③classes：TestNG 会在相同线程中相同类中的运行所有的方法，但是每个类都会用不同的线程运行。

属性 thread-count 允许你为当前的执行指定可以运行的线程数量。