1、surefire插件介绍853769620

通常我们在开发完成之后，都会编写一些单元测试，但是随着系统日益增长，单元测试会很多很多很多，可能多大数百个，数千个，甚至是数万个。而我们每次如果要将开发好的代码跟别人开发的代码进行持续集成，或者是发布到测试环境联调测试，或者是直接上线，每次都是需要将所有的单元测试都运行一次的，因为这是保障系统质量非常重要的行为。如果所有单元测试都能测试通过，那么这个系统的质量也可以得到保障。

微服务这一块的架构，集成测试

假如说有某个哥儿们，写了一段代码，偷懒，嫌弃麻烦，就没运行所有的单元测试，就告诉你说，o了，哥儿们，我要上线了，你觉得你能相信他吗？

不能完全依靠对每个工程师的信任，说他每次修改代码肯定会去运行单元测试

持续集成，自动化集成测试

在大公司里面，都有持续集成的服务器，jenkins，hudson

那么实际上，持续集成，说的就是每个工程师当天写完代码之后，都必须将代码合并到一个公共的持续集成的代码分支上面去，一提交，要触发jenkins持续集成服务器，要立即自动化执行一次构建，自动化既有surefire这种插件，去运行所有的单元测试，确保你的代码没有把系统给改坏。

势必说要用到maven的自动化运行单元测试的插件，surefire

而我们如果手动进行测试的话，那么是无法想象的，此时就需要用maven来自动化运行单元测试。

maven中默认内置了surefire插件来运行单元测试，与最新流行的junit单元测试框架整合非常好。一般是在default生命周期的test阶段，会运行surefire插件的test goal，然后执行src/test/java下面的所有单元测试的。

而surefire插件会根据一定的规则在sre/test/java下面找单元测试类，具体规则如下：

\*\*/Test\*.java

\*\*/\*Test.java

\*\*/\*TestCase.java

通常比较流行的是用\*Test.java格式类命名单元测试的类

2、跳过单元测试

如果某次你的确不想要执行单元测试就打包，那么可以跳过单元测试，但是在实际开发过程中，这是绝对不被允许的。

用mvn package -DskipTests即可

3、指定想要运行的测试类

mvn test -Dtest=\*\*Test，直接指定你要运行的类即可

通常见于开发人员自己刚写了某个单元测试类，然后就想自己运行一下看看效果。可以用逗号隔开，也可以用\*匹配多个类。

4、自定义包含与排除测试类

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.5</version>

<configuration>

<includes>

<include>\*\*/\*Tests.java</include>

</includes>

<excludes>

<exclude>\*\*/\*TempTest.java</exclude>

</excludes>

</configuration>

</plugin>

5、测试报告

单元测试覆盖率是什么意思

比如说，你总共写了100行代码

然后你写了3个单元测试方法，结果这3个单元测试的方法会执行你的100行代码中的80行，另外有20行代码是无论运行哪个单元测试都不会执行的

此时单元测试覆盖率就是80 / 100，80%

一般单元测试覆盖率，要求的是，两种，方法覆盖率，行覆盖率

比如你一共写了10个方法，100行代码

方法覆盖率要达到90%，那么要求就是至少对10个方法中的9个方法都要运行

代码覆盖率要求70%，那么要求就是至少对70行代码都要执行到

分支覆盖率，如果你的代码中有if else，或者是try catch，相当于是你的一段代码根据不同的情况，可能会有2个分支，分支里面还可以再嵌套if else，再分出来2个分支

if

if else

else

if else

2 \* 2 = 4

你的单元测试可以覆盖几个分支，1/4，25%；2 / 4，50%分支覆盖率

surefire插件默认就会在target/surefire-reports目录下生成错误报告，就是如果有单元测试运行失败的话，就会有。

同时可以引入cobertura插件，可以看到测试覆盖率的报告

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>cobertura-maven-plugin</artifactId>

<version>2.5.1</version>

</plugin>

为什么不配置executions去指定这个插件的goal绑定到哪儿？

为什么不配置configuration特殊的参数

第一个问题，其实每个插件你只要定义它的坐标，默认情况下，它自己内部就带了要把哪些goal绑定到哪个phase上去

第二个问题，默认如果没有特殊的要求，用默认的参数就够了