<https://my.oschina.net/wangmengjun/blog/974067>

Maven工程配置代码覆盖工具Jacoco

[收藏](javascript:void(0);)

[王孟君](https://my.oschina.net/wangmengjun/home)

* 发表于 8个月前

* 阅读 2045

* 收藏 192

* 点赞 4

* [评论 3](https://my.oschina.net/wangmengjun/blog/974067#comment-list)

[**新睿云服务器60天免费使用，快来体验！>>>**](https://www.xrcloud.net/ruicloud/#/vps)  https://my.oschina.net/img/hot3.png

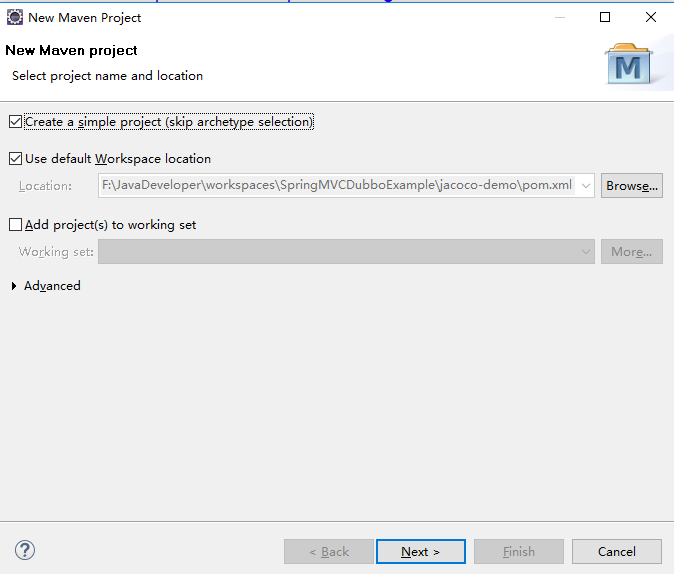
本篇博文我们将给出示例理解**如何在Maven工程中配置Jacoco**和**如何使用Jacoco查看代码覆盖报告**~

Jacoco是一个开源的Java代码覆盖率工具，Jacoco可以嵌入到Ant 、Maven中，并提供了EclEmma Eclipse插件,也可以使用JavaAgent技术监控Java程序。很多第三方的工具提供了对Jacoco的集成，如sonar、Jenkins等。

Maven工程

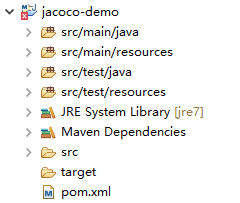
创建Maven工程

打开Eclipse，**File->New->Project->Maven Project**，新建一个Maven工程~



点击“Next”按钮，然后填写**groupId**和**artifactId**信息后点击"Finish"按钮即可~

**groupId -->  com.xxx.tutorial  
artifactId --> jacoco-demo**



配置Jacoco

添加maven-complier-plugin

<plugin>

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

<artifactId>maven-compiler-plugin</artifactId>

<version>3.6.1</version>

<configuration>

<skipMain>true</skipMain>

<skip>true</skip>

<source>1.7</source>

<target>1.7</target>

</configuration>

</plugin>

添加jacoco-maven-plugin

<plugin>

<groupId>org.jacoco</groupId>

<artifactId>jacoco-maven-plugin</artifactId>

<version>${jacoco.version}</version>

<executions>

<execution>

<id>prepare-agent</id>

<goals>

<goal>prepare-agent</goal>

</goals>

</execution>

<execution>

<id>report</id>

<phase>prepare-package</phase>

<goals>

<goal>report</goal>

</goals>

</execution>

<execution>

<id>post-unit-test</id>

<phase>test</phase>

<goals>

<goal>report</goal>

</goals>

<configuration>

<dataFile>target/jacoco.exec</dataFile>

<outputDirectory>target/jacoco-ut</outputDirectory>

</configuration>

</execution>

</executions>

<configuration>

<systemPropertyVariables>

<jacoco-agent.destfile>target/jacoco.exec</jacoco-agent.destfile>

</systemPropertyVariables>

</configuration>

</plugin>

在这里，我们将单元测试结果的输出目录确定为**target/jacoco-ut**目录下~

完整的pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.xxx.tutorial</groupId>

<artifactId>jacoco-demo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<properties>

<jacoco.version>0.7.5.201505241946</jacoco.version>

<junit.version>4.12</junit.version>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>${junit.version}</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

<artifactId>maven-compiler-plugin</artifactId>

<version>3.6.1</version>

<configuration>

<skipMain>true</skipMain>

<skip>true</skip>

<source>1.7</source>

<target>1.7</target>

</configuration>

</plugin>

<plugin>

<groupId>org.jacoco</groupId>

<artifactId>jacoco-maven-plugin</artifactId>

<version>${jacoco.version}</version>

<executions>

<execution>

<id>prepare-agent</id>

<goals>

<goal>prepare-agent</goal>

</goals>

</execution>

<execution>

<id>report</id>

<phase>prepare-package</phase>

<goals>

<goal>report</goal>

</goals>

</execution>

<execution>

<id>post-unit-test</id>

<phase>test</phase>

<goals>

<goal>report</goal>

</goals>

<configuration>

<dataFile>target/jacoco.exec</dataFile>

<outputDirectory>target/jacoco-ut</outputDirectory>

</configuration>

</execution>

</executions>

<configuration>

<systemPropertyVariables>

<jacoco-agent.destfile>target/jacoco.exec</jacoco-agent.destfile>

</systemPropertyVariables>

</configuration>

</plugin>

</plugins>

</build>

</project>

编写代码

Calculator.java

package com.xxx.tutorial;

/\*\*

\*

\* @author wangmengjun

\*

\*/

public class Calculator {

public int add(int a, int b) {

return a + b;

}

public int sub(int a, int b) {

return a - b;

}

}

Calculator\_Test.java

package com.xxx.tutorial;

import org.junit.Assert;

import org.junit.Test;

/\*\*

\*

\* @author wangmengjun

\*

\*/

public class Calculator\_Test {

private Calculator instance = new Calculator();

@Test

public void testAdd() {

int a = 10;

int b = 20;

int expected = 30;

Assert.assertEquals(expected, instance.add(a, b));

}

@Test

public void testSub() {

int a = 10;

int b = 20;

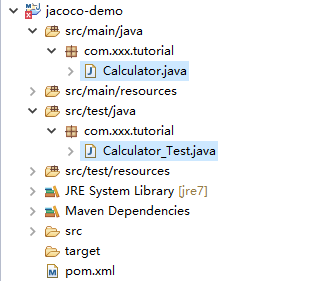
int expected = -10;

Assert.assertEquals(expected, instance.sub(a, b));

}

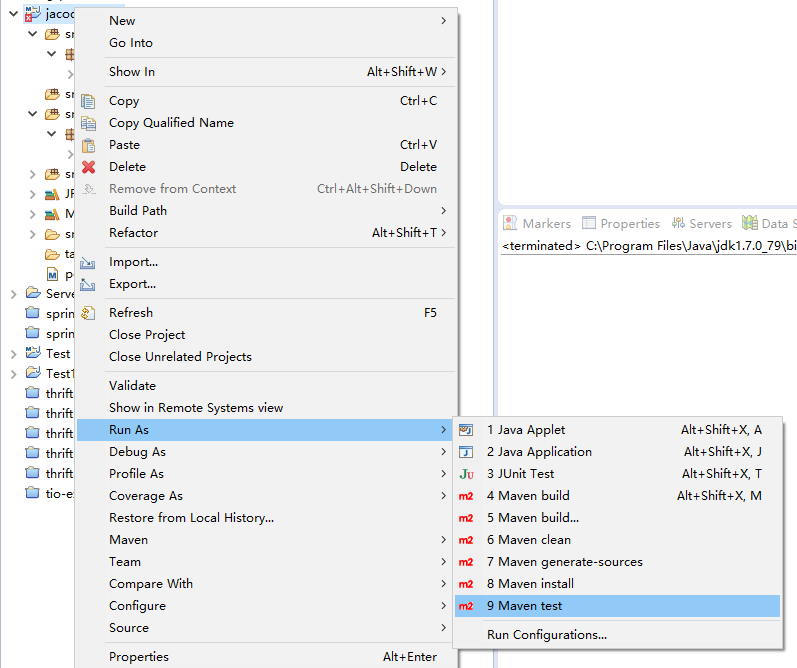
}

代码结构如下：

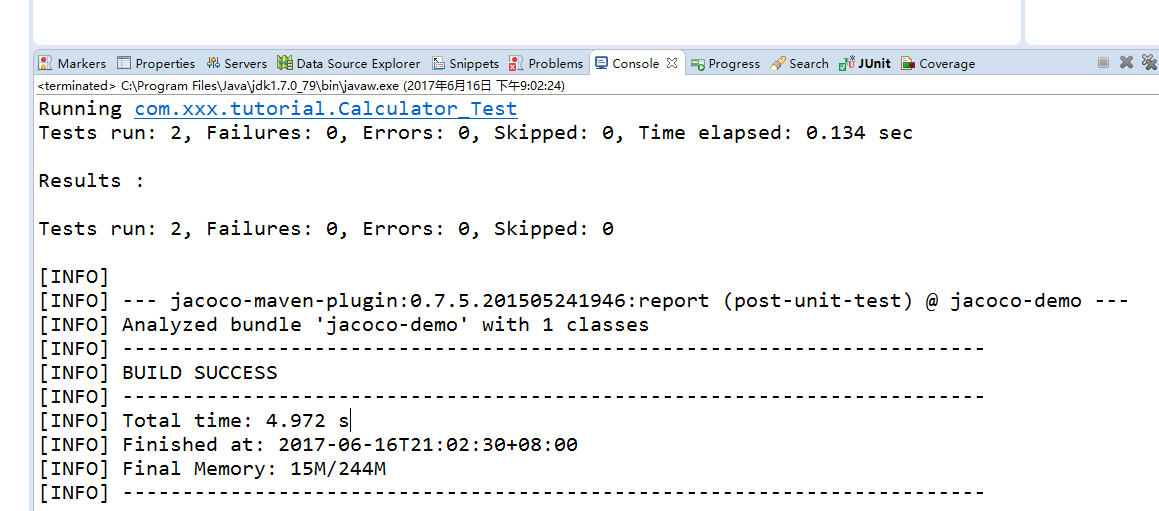


运行并查看Jacoco报告

运行Maven test



执行**Maven test**， 控制台输出如下结果：



[INFO] Scanning for projects...

[INFO]

[INFO] ------------------------------------------------------------------------

[INFO] Building jacoco-demo 0.0.1-SNAPSHOT

[INFO] ------------------------------------------------------------------------

[INFO]

[INFO] --- jacoco-maven-plugin:0.7.5.201505241946:prepare-agent (prepare-agent) @ jacoco-demo ---

[INFO] argLine set to -javaagent:D:\\java\_tools\\Reponsitories\\Maven\\org\\jacoco\\org.jacoco.agent\\0.7.5.201505241946\\org.jacoco.agent-0.7.5.201505241946-runtime.jar=destfile=F:\\JavaDeveloper\\workspaces\\SpringMVCDubboExample\\jacoco-demo\\target\\jacoco.exec

[INFO]

[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ jacoco-demo ---

[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!

[INFO] Copying 0 resource

[INFO]

[INFO] --- maven-compiler-plugin:3.6.1:compile (default-compile) @ jacoco-demo ---

[INFO] Not compiling main sources

[INFO]

[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ jacoco-demo ---

[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!

[INFO] Copying 0 resource

[INFO]

[INFO] --- maven-compiler-plugin:3.6.1:testCompile (default-testCompile) @ jacoco-demo ---

[INFO] Not compiling test sources

[INFO]

[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ jacoco-demo ---

[INFO] Surefire report directory: F:\JavaDeveloper\workspaces\SpringMVCDubboExample\jacoco-demo\target\surefire-reports

-------------------------------------------------------

T E S T S

-------------------------------------------------------

Running com.xxx.tutorial.Calculator\_Test

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.134 sec

Results :

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0

[INFO]

[INFO] --- jacoco-maven-plugin:0.7.5.201505241946:report (post-unit-test) @ jacoco-demo ---

[INFO] Analyzed bundle 'jacoco-demo' with 1 classes

[INFO] ------------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

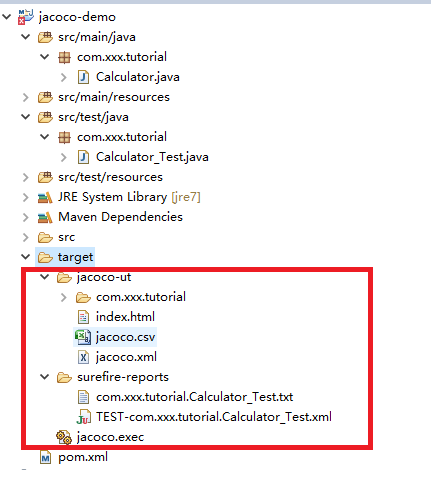
[INFO] Total time: 4.972 s

[INFO] Finished at: 2017-06-16T21:02:30+08:00

[INFO] Final Memory: 15M/244M

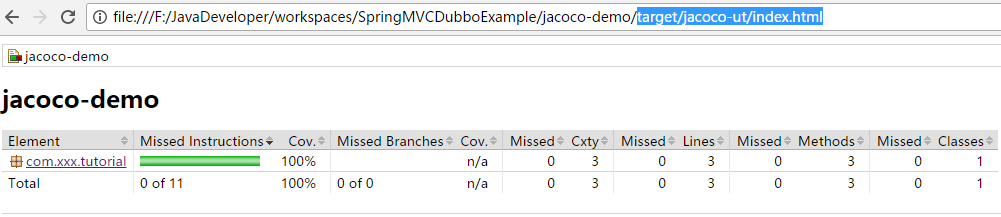
[INFO] ------------------------------------------------------------------------

我们可以**看到target目录下，已经生成了Jacoco的单元测试结果报告**~

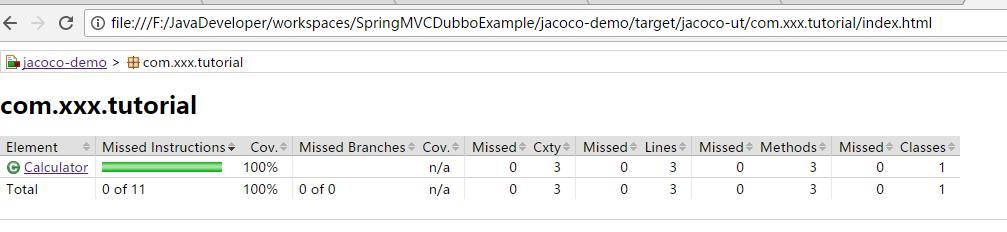


查看Jacoco报告

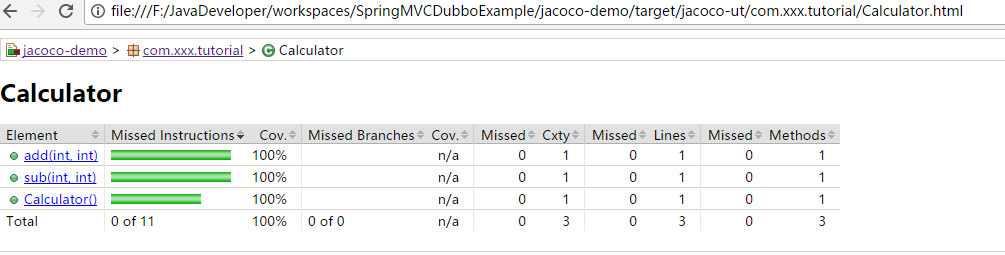
打开浏览器，在URL栏输入**<工程地址>/target/jacoco-ut/index.html**，如：



点击"**com.xxx.tutorial**"链接，查看这个com.xxx.tutorial包下的类。

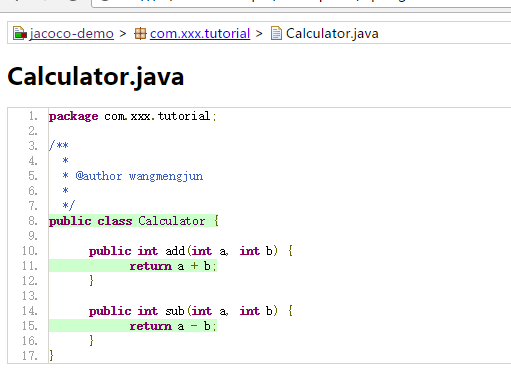


再点击"**Calculator**"链接，展示Calculator类的方法信息~



再点击任何方法的连接，将会出现该类代码覆盖的情况：

绿色的表示覆盖到的，如果没有覆盖则会用红色背景表示



至此，

在Maven工程中配置Jacoco插件，运行并查看执行报告结果的示例就完成了~

另外，如果Eclipse工程中安装了**EclEmma**插件，执行测试类，

https://static.oschina.net/uploads/space/2017/0616/212049_KyRp_2911530.png

也能得到相应的结果，如：

