<http://blog.csdn.net/u011858405/article/details/78974192>

**Maven+TravisCI持续集成和Coveralls测试覆盖率检测**

原创 2018年01月04日 18:48:52

* 标签：
* [maven](http://so.csdn.net/so/search/s.do?q=maven&t=blog) /
* [持续集成](http://so.csdn.net/so/search/s.do?q=%E6%8C%81%E7%BB%AD%E9%9B%86%E6%88%90&t=blog)
* 78

之前挖了不少开源项目的坑 ,准备慢慢填起来 .第一步当然从项目管理开始 ,把之前的maven项目弄得更加规整.

**Maven**

还是延续之前的parent-pom/child-jar的结构   
\* parent中只留配置 : dependencies/plugins 都替换成 management节点   
\* 补充scm/issue等节点信息 ,供其他工具集成   
\* 增加source/javadoc等plugin ,方便第三方使用者使用.   
\* 增加代码扫描工具(最好是通过IDEA的编写时工具检测更好)

<?xml version="1.0" encoding="UTF-8"?>

<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>site.koalazoo.cutekoala</groupId>

<artifactId>cute-koala</artifactId>

<version>0.1.0</version>

<packaging>pom</packaging>

<name>Cute Koala</name>

<url>https://github.com/Anddd7/cute-koala</url>

<description>Manage your Java Application easiler.</description>

<licenses>

<license>

<name>Apache License, Version 2.0</name>

<url>http://www.apache.org/licenses/LICENSE-2.0.txt</url>

<distribution>repo</distribution>

</license>

</licenses>

<issueManagement>

<system>GitHub Issues</system>

<url>https://github.com/Anddd7/cute-koala/issues</url>

</issueManagement>

<inceptionYear>2017</inceptionYear>

<scm>

<connection>scm:git:https://github.com/Anddd7/cute-koala.git</connection>

<developerConnection>scm:git:git@github.com:Anddd7/cute-koala.git</developerConnection>

<url>https://github.com/Anddd7/cute-koala</url>

</scm>

<developers>

<developer>

<id>and777</id>

<name>Eddy Liao</name>

<email>liaoad\_space@sina.com</email>

<roles>

<role>owner</role>

<role>developer</role>

</roles>

<timezone>+8</timezone>

</developer>

</developers>

<modules>

<module>koala-core</module>

<module>koala-test</module>

</modules>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.build.jdk.v>1.8</project.build.jdk.v>

<junit.v>4.12</junit.v>

<slf4j-log4j12.v>1.7.5</slf4j-log4j12.v>

<lombok.v>1.16.18</lombok.v>

<fastjson.v>1.2.33</fastjson.v>

<mysql-jdbc.v>6.0.6</mysql-jdbc.v>

<guava.v>22.0</guava.v>

<beetl.v>2.7.16</beetl.v>

<snakeyaml.v>1.18</snakeyaml.v>

</properties>

<dependencyManagement>

<dependencies>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-log4j12</artifactId>

<version>${slf4j-log4j12.v}</version>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

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

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<version>${lombok.v}</version>

<scope>provided</scope>

</dependency>

</dependencies>

</dependencyManagement>

<build>

<plugins>

<plugin>

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

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

<version>3.6.1</version>

<configuration>

<source>${project.build.jdk.v}</source>

<target>${project.build.jdk.v}</target>

</configuration>

</plugin>

</plugins>

<pluginManagement>

<plugins>

<plugin>

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

<artifactId>maven-source-plugin</artifactId>

<version>2.1.2</version>

<executions>

<execution>

<id>attach-sources</id>

<goals>

<goal>jar-no-fork</goal>

</goals>

</execution>

</executions>

</plugin>

<plugin>

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

<artifactId>maven-javadoc-plugin</artifactId>

<version>2.10.4</version>

<configuration>

<quiet>true</quiet>

<notimestamp>true</notimestamp>

<additionalparam>-Xdoclint:-html</additionalparam>

<encoding>${project.build.sourceEncoding}</encoding>

<docencoding>${project.build.sourceEncoding}</docencoding>

<charset>${project.build.sourceEncoding}</charset>

<additionalparam>-XDignore.symbol.file</additionalparam>

<additionalparam>-Xdoclint:-html</additionalparam>

<linksource>true</linksource>

</configuration>

<executions>

<execution>

<id>attach-javadocs</id>

<goals>

<goal>jar</goal>

</goals>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.projectlombok</groupId>

<artifactId>lombok-maven-plugin</artifactId>

<version>1.16.18.1</version>

<executions>

<execution>

<phase>generate-sources</phase>

<goals>

<goal>delombok</goal>

</goals>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.jacoco</groupId>

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

<version>0.7.6.201602180812</version>

<executions>

<execution>

<id>prepare-agent</id>

<goals>

<goal>prepare-agent</goal>

</goals>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.eluder.coveralls</groupId>

<artifactId>coveralls-maven-plugin</artifactId>

<version>4.3.0</version>

</plugin>

</plugins>

</pluginManagement>

</build>

</project>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55
* 56
* 57
* 58
* 59
* 60
* 61
* 62
* 63
* 64
* 65
* 66
* 67
* 68
* 69
* 70
* 71
* 72
* 73
* 74
* 75
* 76
* 77
* 78
* 79
* 80
* 81
* 82
* 83
* 84
* 85
* 86
* 87
* 88
* 89
* 90
* 91
* 92
* 93
* 94
* 95
* 96
* 97
* 98
* 99
* 100
* 101
* 102
* 103
* 104
* 105
* 106
* 107
* 108
* 109
* 110
* 111
* 112
* 113
* 114
* 115
* 116
* 117
* 118
* 119
* 120
* 121
* 122
* 123
* 124
* 125
* 126
* 127
* 128
* 129
* 130
* 131
* 132
* 133
* 134
* 135
* 136
* 137
* 138
* 139
* 140
* 141
* 142
* 143
* 144
* 145
* 146
* 147
* 148
* 149
* 150
* 151
* 152
* 153
* 154
* 155
* 156
* 157
* 158
* 159
* 160
* 161
* 162
* 163
* 164
* 165
* 166
* 167
* 168
* 169
* 170
* 171
* 172
* 173
* 174
* 175
* 176
* 177
* 178
* 179
* 180
* 181
* 182
* 183
* 184
* 185

**持续集成**

之前尝试在阿里云上搭Jenkins ,因为内存爆炸(免费机) ,以失败告终.在添加SCM节点时 ,搜索了一下集成的工具 ,发现了 [Travis](https://travis-ci.org/" \t "_blank).Travis是一个持续集成的平台 ,自带与GitHub的集成 ,能够检测代码提交并触发继承脚本.

* 进入官网使用GitHub登录 ,然后选择想要继承的repo
* 在repo下新建.travis.yml文件 ,写入自动集成触发的脚本
* push代码到github ,然后就可以到travis首页查看编译情况

[coveralls](https://coveralls.io/)是一个测试报告展示平台 ,能够集成多种测试报告 .也是在使用travis的时候检索到的 ,通过travis持续集成并生成测试报告 ,然后上传到coveralls .

* 使用GitHub账号登录 ,选择repo
* 复制repoToken

repoToken是你访问测试报告的凭证 ,因此不能明文上传到github上 ,但是我们又需要在CI中使用这个repoToken.

 加密Token 

* 使用travis的对称加密工具(需要ruby gem工具):travis encrypt COVERALLS\_TOKEN=xxxxxx
* 复制加密后的token到.travis.yml文件中
* 使用变量名访问

 在travis脚本中使用token和测试报告生成工具

 生成并上传测试报告

Java项目官方推荐了一个plugin使用 ,token不能使用明文 ,所以不能直接把token配置在pom中.而是在travis集成时 ,把解密后的token拿来使用.

<build>

<plugins>

<!-- 直接引用parent中配置好的plugin -->

<plugin>

<groupId>org.jacoco</groupId>

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

</plugin>

<plugin>

<groupId>org.eluder.coveralls</groupId>

<artifactId>coveralls-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13

language: java

sudo: false

script: mvn clean verify

after\_success:

- mvn clean test

## 生成测试报告

- mvn jacoco:report

## 获取travis解密的token ,并调用coveralls-plugin上传报告

- mvn coveralls:report -DrepoToken="${COVERALLS\_TOKEN}"

## 加密后的token字串

env:

global:

- secure: "..............................."

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14

**最后**

就可以在主页上看到持续集成和测试报告的结果 ,还可以导入非常有逼格的budge

