# Jacoco简介

jacoco 是一个开源的覆盖率工具，它针对的开发语言是 java。其使用方法很灵活，可以嵌入到 ant、maven 中；可以作为 Eclipse 插件；可以作为 javaAgent 探针监控 java 程序等等。

很多第三方的工具提供了对 jacoco 的集成，如 sonar、jenkins 等等。

jacoco 包含了多种尺度的覆盖率计数器，包含指令级覆盖（Instructions,C0coverage）、分支覆盖（Branches,C1coverage）、圈复杂度（CyclomaticComplexity）、行覆盖（Lines）、方法覆盖（non-abstract methods）、类覆盖（classes），其含义如下：

行覆盖率：度量被测程序的每行代码是否被执行，判断标准行中是否至少有一个指令被执行。

类覆盖率：度量计算 class 类文件是否被执行。

分支覆盖率：度量 if 和 switch 语句的分支覆盖情况，计算一个方法里面的总分支数，确定执行和不执行的分支数量。

方法覆盖率：度量被测程序的方法执行情况，是否执行取决于方法中是否有至少一个指令被执行。

指令覆盖：计数单元是单个 java 二进制代码指令，指令覆盖率提供了代码是否被执行的信息，度量完全独立源码格式。

圈复杂度：在（线性）组合中，计算在一个方法里面所有可能路径的最小数目，缺失的复杂度同样表示测试案例没有完全覆盖到这个模块。

综合以上分析：此工具的扫描结果是完全覆盖UT和API测试的

# Maven集成jacoco插件

关于此插件的版本号可以在maven中央仓库查询，评审择优

<plugin>  
 <groupId>org.jacoco</groupId>  
 <artifactId>jacoco-maven-plugin</artifactId>  
 <version>0.7.7.201606060606</version>  
 <configuration>  
 <destFile>target/coverage-reports/jacoco-unit.exec</destFile>  
 <dataFile>target/coverage-reports/jacoco-unit.exec</dataFile>  
 </configuration>  
 <executions>  
 <execution>  
 <id>jacoco-initialize</id>  
 <goals>  
 <goal>prepare-agent</goal>  
 </goals>  
 </execution>  
 <execution>  
 <id>jacoco-site</id>  
 <phase>test</phase>  
 <goals>  
 <goal>report</goal>  
 </goals>  
 </execution>  
 </executions>  
</plugin>

生成的jacoco的可执行文件是target/coverage-reports/下的jacoco-unit.exec

生成的本地覆盖率报告路路径：target/site/jacoco/index.html

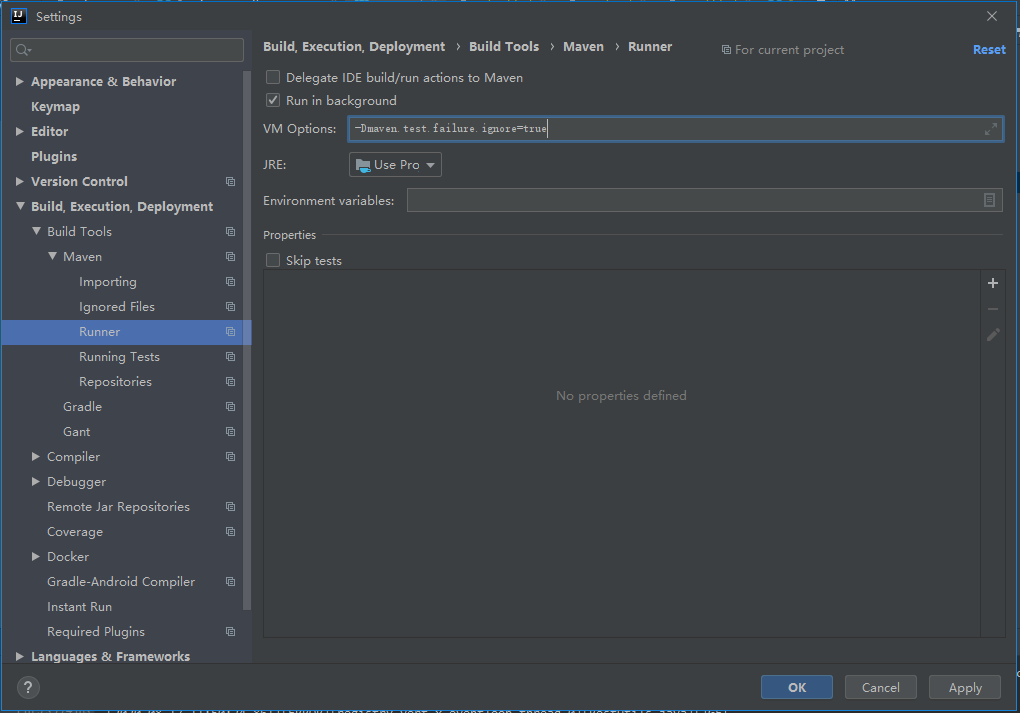
注意：

有时候我们用IntelliJ IDEA 的maven视窗执行mvn test，有可能会因为某个测试方法的失败，从而导致不能生成jacoco的可执行文件

其实我们也可以用maven命令去执行：

mvn clean test org.jacoco:jacoco-maven-plugin:0.7.7.201606060606:prepare-agent install -Dmaven.test.failure.ignore=true

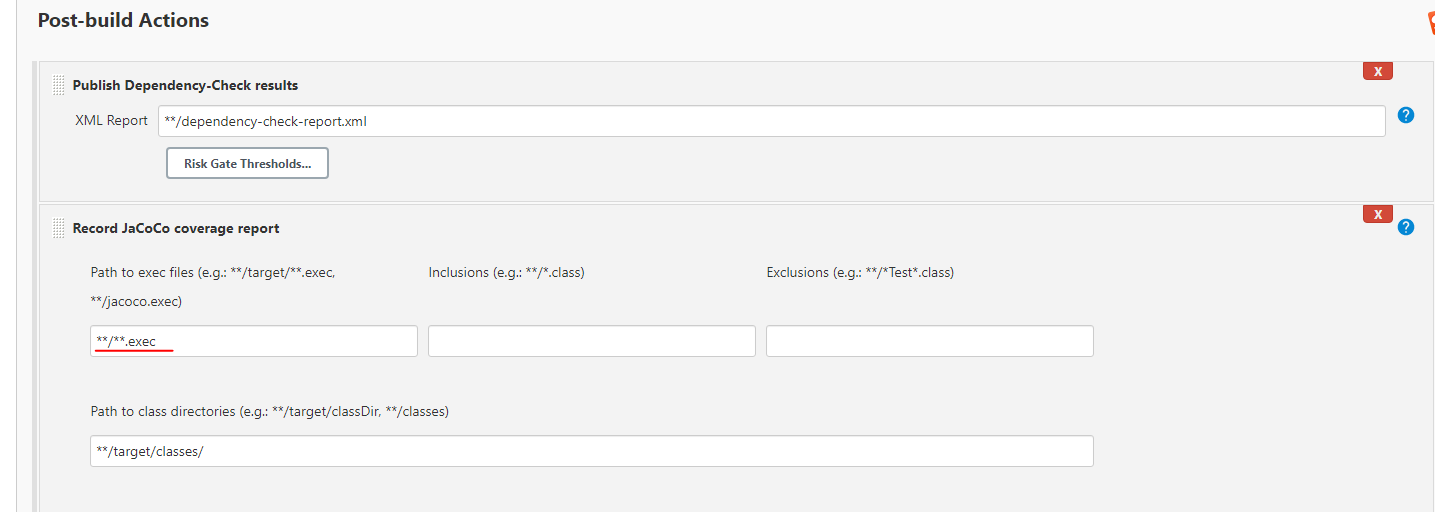
也可以在idea的VM option中设置忽略执行失败的单元测试：



# Jenkins集成jacoco插件

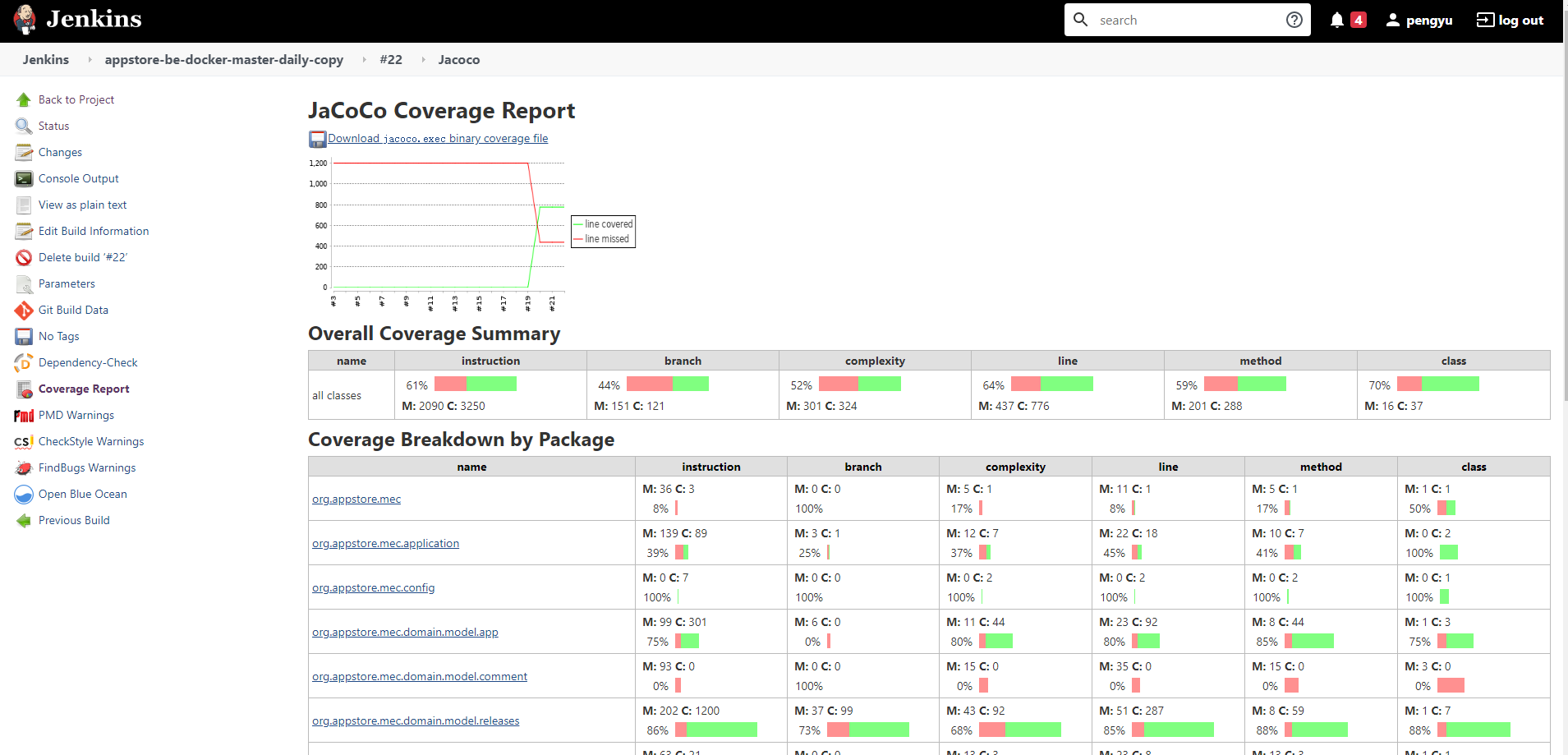
关于jenkins如何集成jacoco可google

jenkins集成jacoco会使用到上面maven集成jacoco部分生成的可执行文件，下图中标红的部分



# Jenkins实例应用集成执行展示





# SonarQube集成jacoco插件

参见：

[使用Jenkins流水线集成SonarQube做代码质量分析](https://blog.csdn.net/nklinsirui/article/details/90518618)

除了可以在本地查看JaCoCo生成的代码测试覆盖率报告外，还可以将JaCoCo收集到的代码测试覆盖率信息（target/jacoco.exec)文件送给SonarQube来生成和展示代码测试覆盖率报告。

命令示例：

mvn sonar:sonar \

-s settings.xml \

-Dsonar.host.url=http://192.168.87.154:9000 \

-Dsonar.login=7c4b5baf2e96a565d27fe062c11baa05b5eb2f2a

注意：由于算法不尽相同，本地JaCoCo测试覆盖率报告(target/site/jacoco/index.html)和SonarQube上的代码测试覆盖率报告（Coverage)的测试覆盖率可能有细微差异。

使用SonarQube的QualityGate功能检查代码测试覆盖率是否符合要求。

# 总结

无论是jenkins或者是SonqrQube都会用到maven集成jacoco部分中的可执行文件，SonarQube（本身没有可以观察覆盖率的插件）虽然可以集成jacoco但是感觉还是多此一举。

利用jenkins环境做线上代码覆盖率检查，maven插件做线下代码覆盖率检查就OK.