# 配置

## coverage配置

1. [jmockit-]coverage-**output**: one or more comma-separated values between html, html-nocp ("nocp" stands for "no call points"), serial, and serial-append, which select the kind of output to be generated at the end of the test run. The default if none is specified is to generate the basic HTML report (html-nocp).   
   The "html" and "html-nocp" values are mutually exclusive, just like "serial" and "serial-append". However, it *is* valid to have one of each pair specified at the same time. In such a case, at the end of the test run both kinds of output will be written.   
   The presence of "serial" or "serial-append" causes a *serialized data file* of name "coverage.ser" to be generated; in the case of "serial-append", coverage data gathered by the current test run will be *appended* to the contents of a previously existing data file (if said file doesn't exist, it has the same effect as "serial").
2. [jmockit-]coverage-**outputDir**: absolute or relative path to the output directory, to be used for writing any "coverage.ser" or "index.html" files (plus the remaining ".html" files of the HTML report, in automatically created sub-directories). By default, the current working directory of the running JVM is used, with all ".html" files of the HTML report generated inside a "coverage-report" sub-directory.
3. [jmockit-]coverage-**srcDirs**: comma-separated list of Java source directories to be searched when generating an HTML report. (This is not relevant for the serialized data file.) Each directory is specified by an absolute or relative path. If no such directory is specified, all "src" directories under the current working directory are searched.
4. [jmockit-]coverage-**classes**: Either an OS-like regular expression (with the typical "\*" and "?" wildcards), or a[java.util.regex](http://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html)-conformable regular expression. The given expression will be used to select the classes (by fully qualified name) from production code which should be considered for coverage. By default, all classes in production code loaded during the test run and which are not inside jar files are considered.   
   For example, "some.package.\*" selects all classes under some.package or any sub-package.   
   As a special case, if the property is specified as "loaded", then all classes will be considered, but only those which get loaded by the JVM during the test run; classes that are part of the codebase but never get loaded are left out. This is very useful when the test run includes only a few tests, targeting only a subset of the codebase.

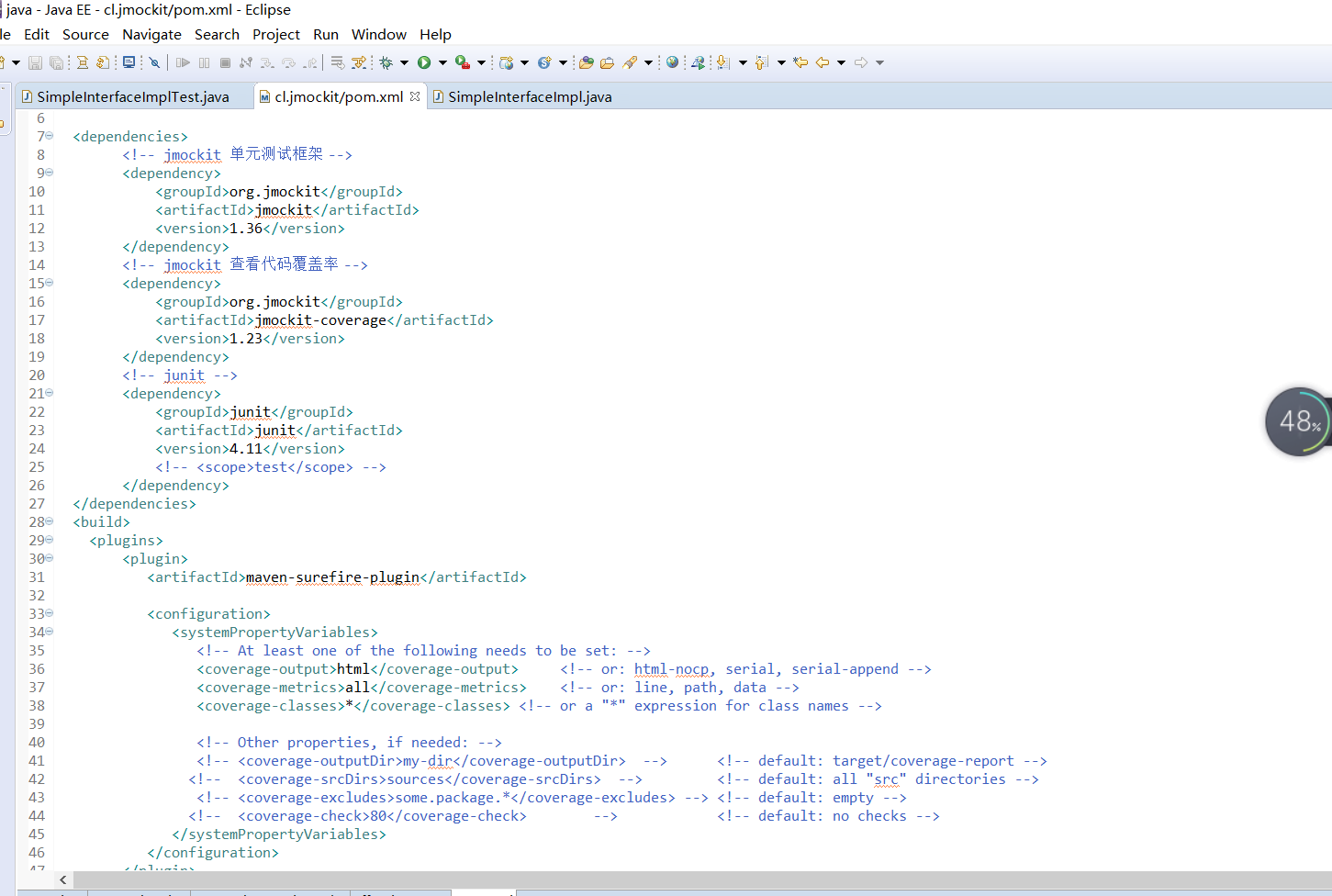


1. [jmockit-]coverage-**excludes**: The same as the previous property, but for class names which should be excluded from consideration when instrumenting classes for coverage. This property can be used together with coverage-classes or on its own. By default, no classes between those selected for coverage are excluded from consideration.
2. [jmockit-]coverage-**metrics**: one or more comma-separated words between line (the default), path, data, and all, which select the specific set of code coverage **metrics** to gather coverage information for.
3. [jmockit-]coverage-**check**: one or more semicolon-separated rules specifying *minimum coverage* checks to be performed at the end of a test run. By default, no such checks are performed. For details, see the [Checking minimum coverage](http://jmockit.github.io/tutorial/CodeCoverage.html#checking) section.

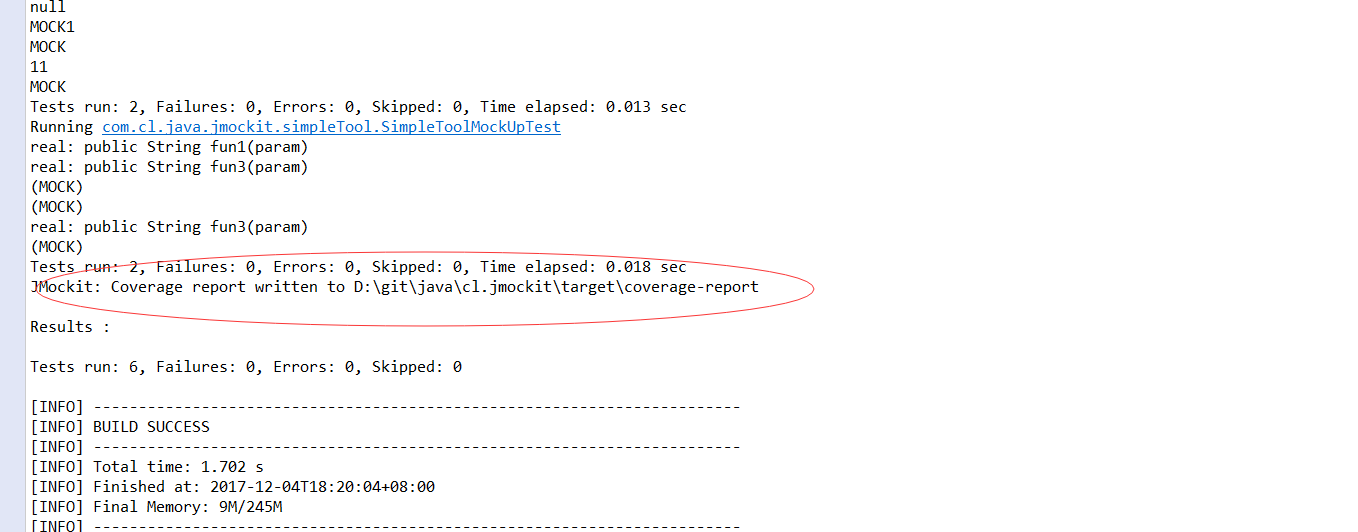
## [Aggregated reports f](http://jmockit.github.io/tutorial/CodeCoverage.html#merging)

# 生成测试覆盖率报告

## 整个项目覆盖率报告



运行 mvn test



## 单个文件覆盖率报告

-ea

-javaagent:D:/m2/repository/org/jmockit/jmockit/1.36/jmockit-1.36.jar=coverage

-Dcoverage-output=html

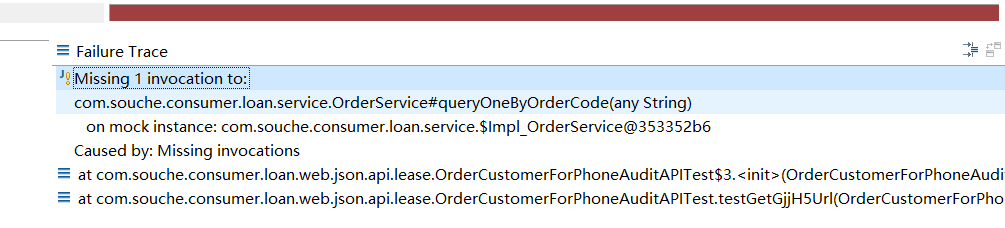
-Dcoverage-metrics=all

-Dcoverage-classes=\*

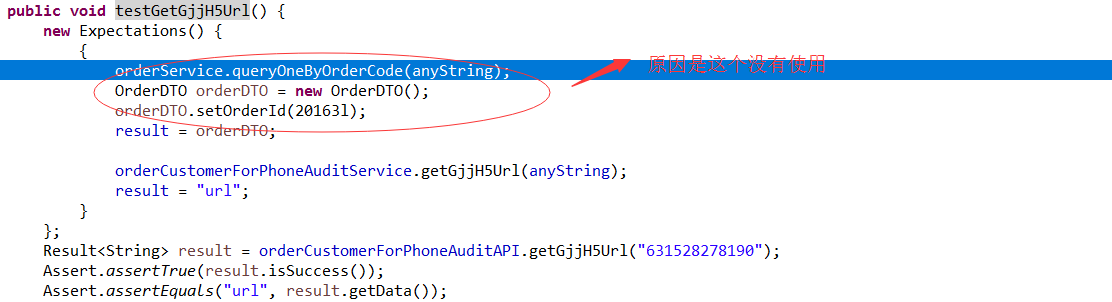
-Dcoverage-outputDir=target/coverage-report

# 问题

## Missing 1 invocation



原因



总结：

如果你在Expectations中，mock了某个类，但在调用中没有使用，会报这个错，代码能正式执行。

# 参考资料

1. <https://yq.aliyun.com/articles/47245> （JMockit学习笔记）
2. <http://blog.csdn.net/foreverling/article/details/51234149> （JMockit学习笔记）
3. <http://jmockit.github.io/tutorial/CodeCoverage.html>（官网）