不谈技术，先来聊点八卦。在看过的几个项目中一般都有这么几个jar包：commons-logging、log4j。这其实有点像接口与实现类的关系，开发人员只调用commons-logging的方法，具体的任务是由log4j实现的。但你随便点开一个它门中的某个包你会发现它们的前缀都一样。

org.apache.logging.log4j

org.apache.commons.logging

统统都是Apache开发的，乍一看是不是很多此一举，既然都是你们做的，好端端的为什么要隔成两个jar包，又不是不同的项目。当初Apache开发完log4j时，极力向sun公司推荐将log4j做为jdk1.4的一部分，但jdk发布在即，sun公司拒绝了。

后来，由于sun自带的日志java.util.logging性能太烂，大家基本上还是在用Apache的log4j，但不管怎么说还是有少部分人在用Java自带的日志的。这会导致什么情况呢，你写了一个项目用的log4j，你觉得不赖然后开源了。某个人看到你的项目他也觉得不赖，拉到本地准备运行，可他就是不想用log4j，强行把日志框架换成了java.util.logging，然后boom，项目崩了。为了解决这个局面，Apache又搞出了commons-logging，这个jar包为log4j与java.util.logging提供了统一接口，使用此接口编程就可以不管项目到底使用哪一种日志。

而slf4j（simple logging facade for Java）和commons-logging类似，但更强大，因为它为所有日志都提供了一个统一接口，而且创新了打日志的方式。

SLF4J: Class path contains multiple SLF4J bindings.

SLF4J: Found binding in [jar:file:/D:/Maven/repo/m2/ch/qos/logback/logback-classic/1.2.3/logback-classic-1.2.3.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: Found binding in [jar:file:/D:/Maven/repo/m2/org/slf4j/slf4j-log4j12/1.7.16/slf4j-log4j12-1.7.16.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: See http://www.slf4j.org/codes.html#multiple\_bindings for an explanation.

SLF4J: Actual binding is of type [ch.qos.logback.classic.util.ContextSelectorStaticBinder]

pom.xml，exclusion冲突的jar包