1、javac -d . com/ljq/Hello.java 编译java程序（-d指定文件目录，.代表当前目录）

2、jar -cvfm hello.jar main/resources/MENIFEST.MF(jar命令打包)

3、

使用 maven 命令生成项目（idea 和 eclipse 生成项目最终也是依赖 maven 插件生成的）不同的项目用不同的插件生成

```
mvn archetype:generate -DgroupId=enjoy -DartifactId=simple -DarchetypeArtifactId=maven-archetype-quickstart -Dversion=1.0

mvn archetype:generate -DgroupId=enjoy -DartifactId=simple-web -DarchetypeArtifactId=maven-archetype-webapp -Dversion=1.0
```

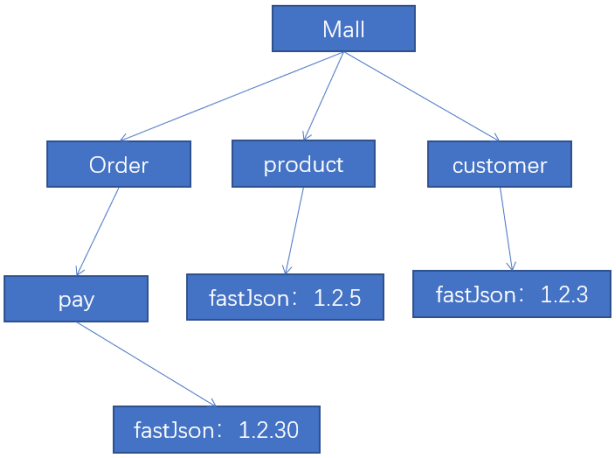
4、exclusions排除冲突，execution指定插件运行目标

5、每个命令都与插件想关联。

6、执行插件mvn com.enjoy:enjoy-plugin:1.0:*log* -Dlog.name=ljq（可以通过-D传参）。***log为插件目标，一个插件中可以有多个AbstractMojo接口的实现，每个实现就是不同的目标，所以执行插件的时候要指定目标。如图：***

7、jar包冲突其实底层是jvm在作祟，因为classLoader对同一个类只能加载一次，当引入多个相同的jar时里边的class类会冲突，jvm加载时会不知所措。

8、maven的依赖传递解决措施如图：



Mall项目应该使用哪个fastjson包呢？

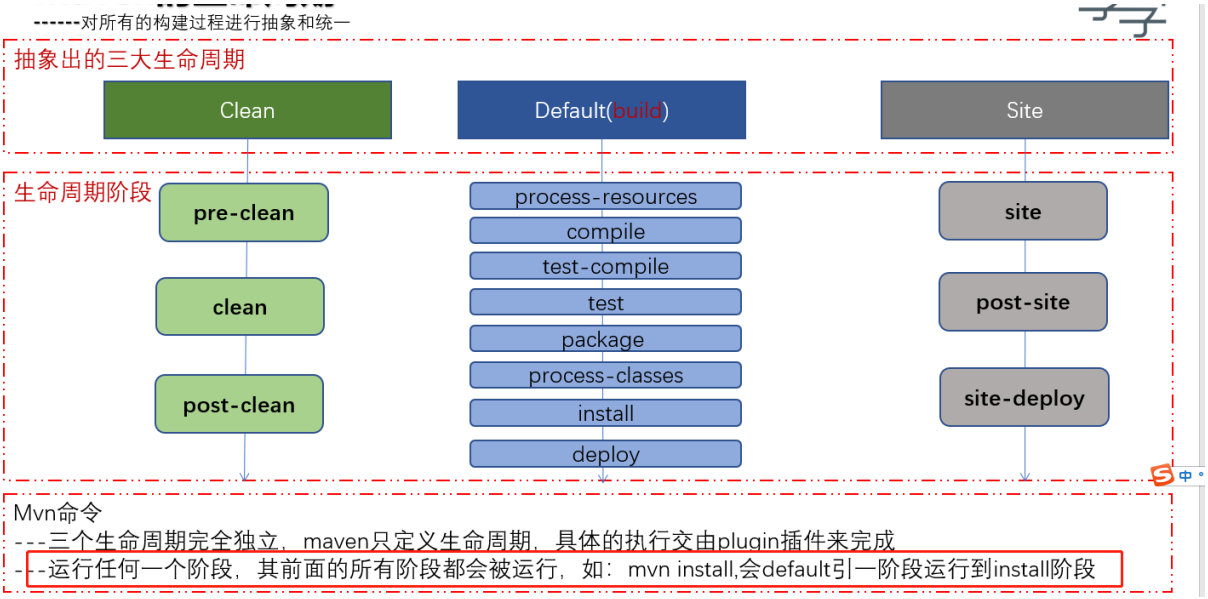
1、路径最短原则：product和customer里的引用路径较短

2、同路径长度下，谁先声明谁优先：看product依赖和customer依赖在pom中声明位置，谁先用谁的

3、使用exclusions排除，声明不要product或者customer不要fastjson依赖。

9、mvn dependency:tree会将pom最终决策出来的依赖拉出来，树形显示

10、maven的生命周期。执行某个命令会默认执行它前面的阶段，如图：

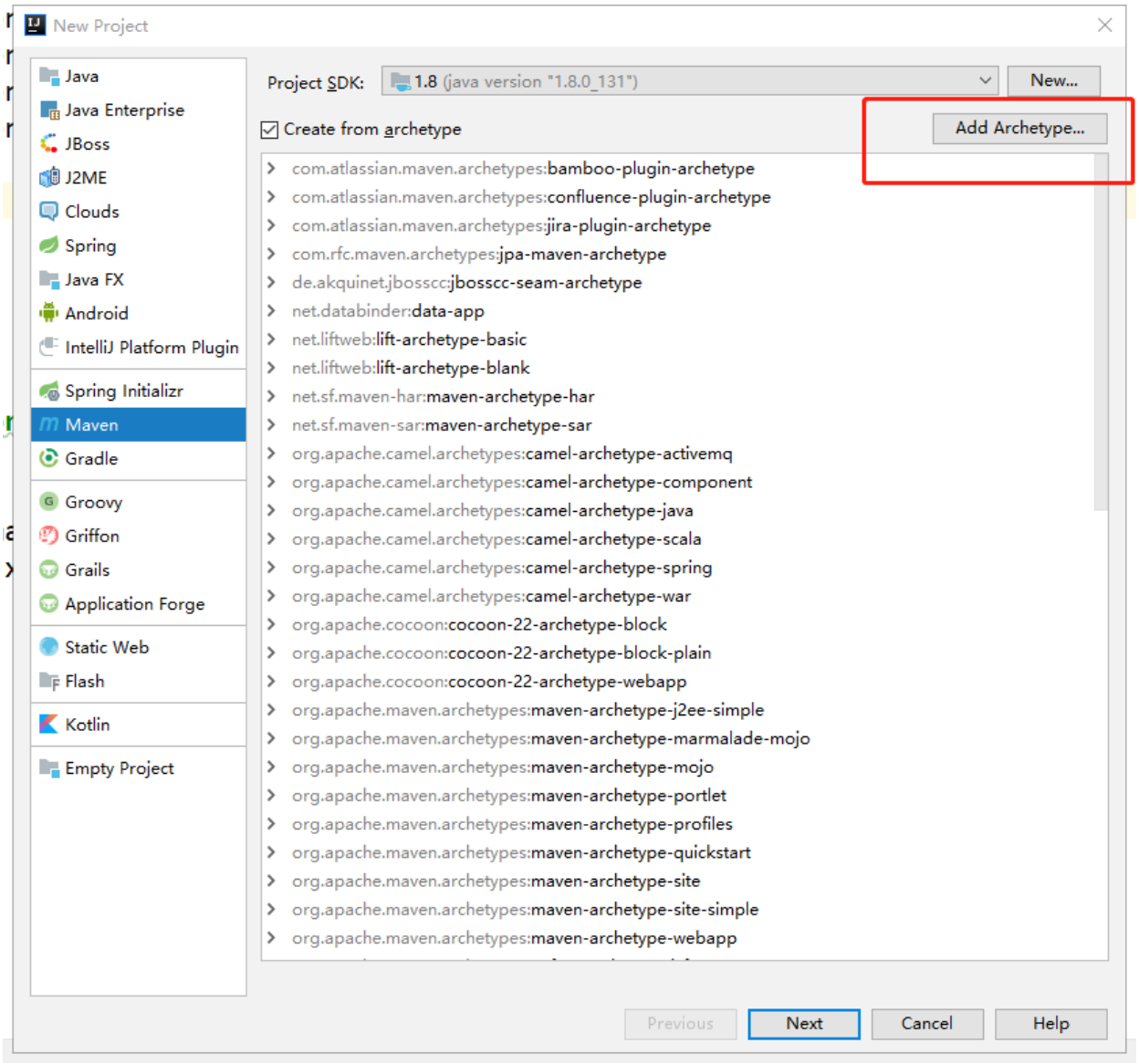


11、

在多继承的时候它只能import其他pom里边的dependencyManagement里边的依赖，继承以后再子pom的dependencies中还是需要手动引入依赖(和子pom的dependencyManagement没啥区别)

12、release(public)为稳定版本，snapshot为开发版本，格式：<version>1.0-snapshot</version>。

13、定义自己的项目骨架，打包到远程仓库。然后用idea创建项目的时候选择骨架所有人员就可以统一了，如图：

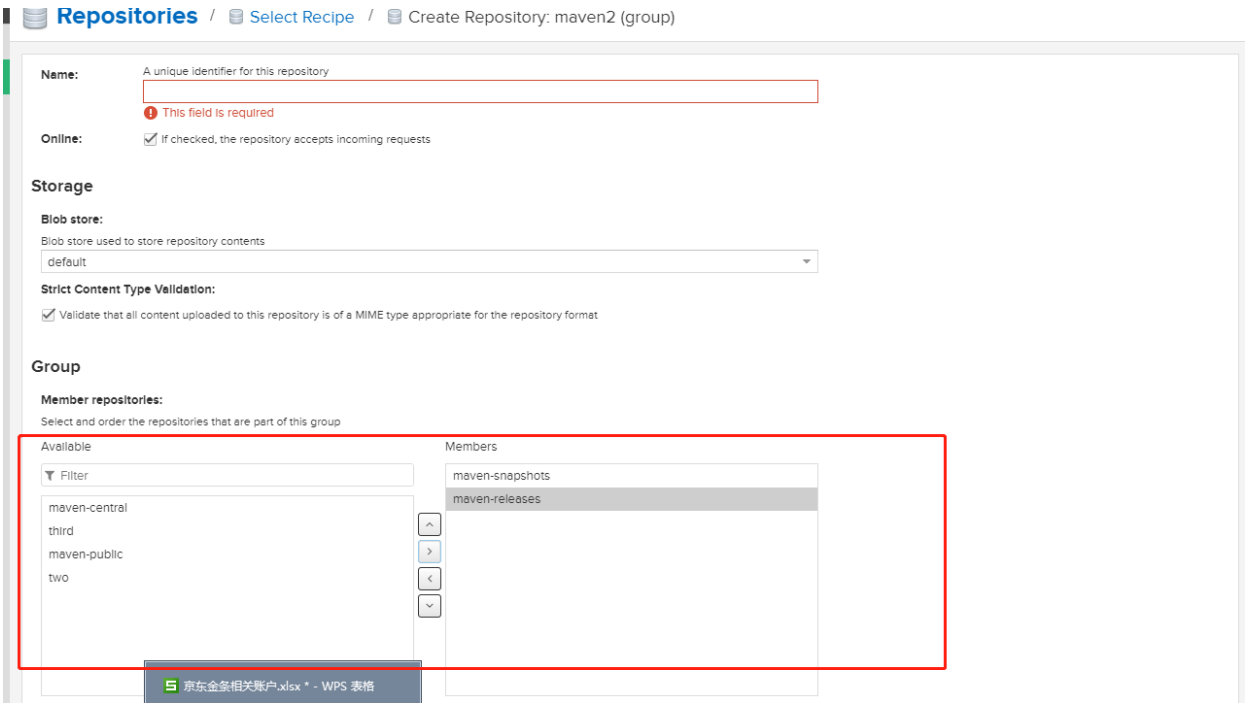


14、group、proxy、hosted的区别如图：

我们在nexus私服或其他maven库看仓库类型有hosted、proxy、group，有点懵比，今天查了一下

项目	具体说明
hosted	本地存储。像官方仓库一样提供本地私库功能
proxy	提供代理其它仓库的类型
group	组类型，能够组合多个仓库为一个地址提供服务

创建group，可以添加多个hosted、proxy，将它们组合起来对外服务：



创建proxy，需要添加远程仓库的url，比如阿里云demaven仓库地址：