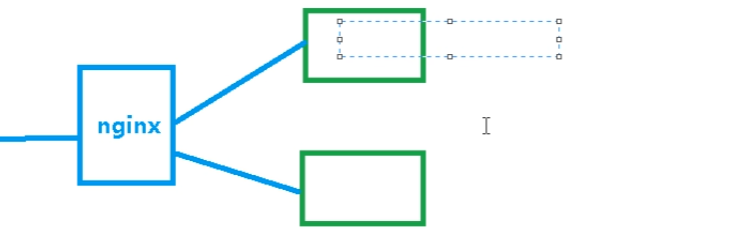


负载均衡 分摊到不同的服务器

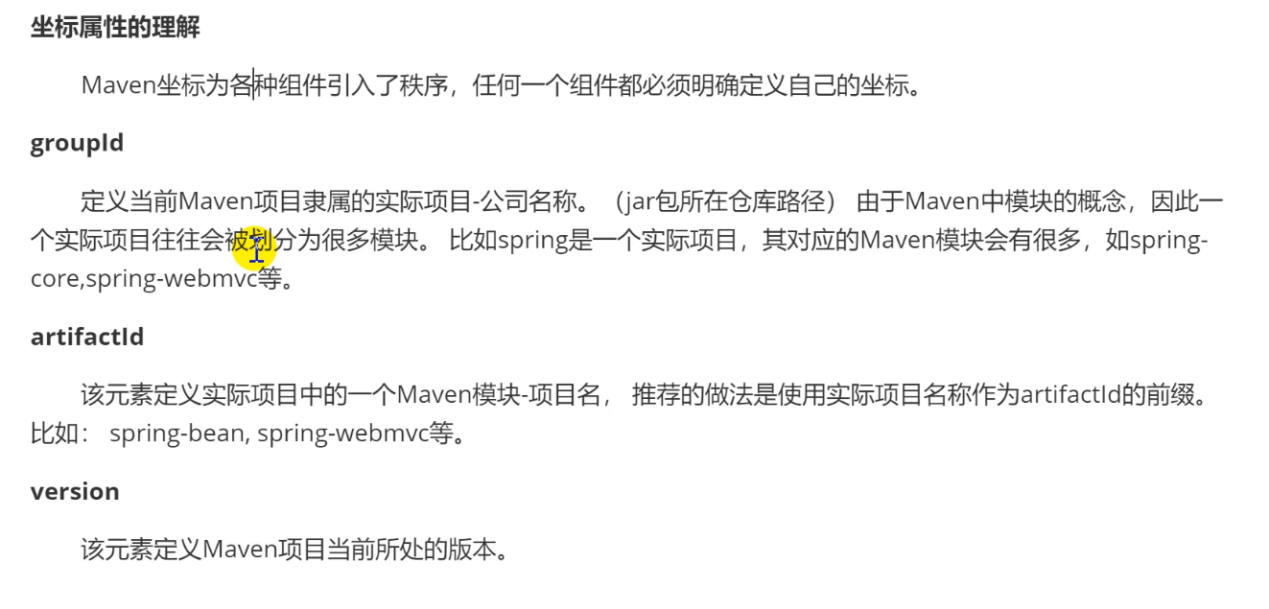


Maven jar 包的调味包

依赖自动构建 部署 xml dependency

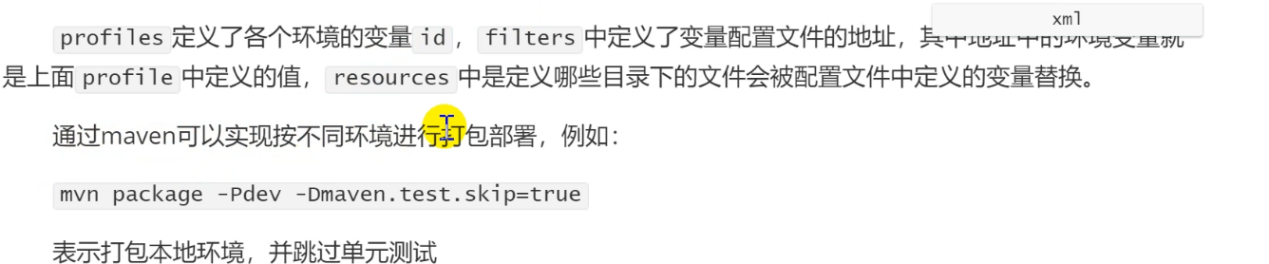
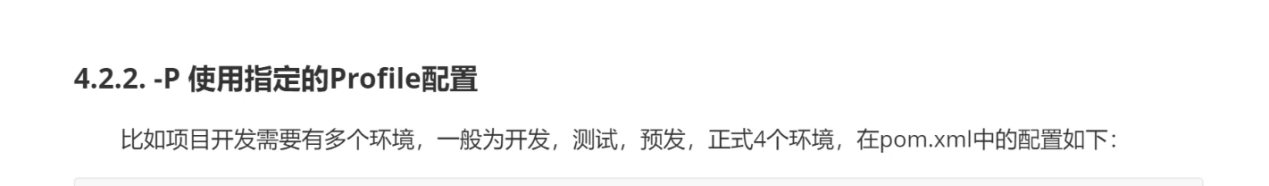
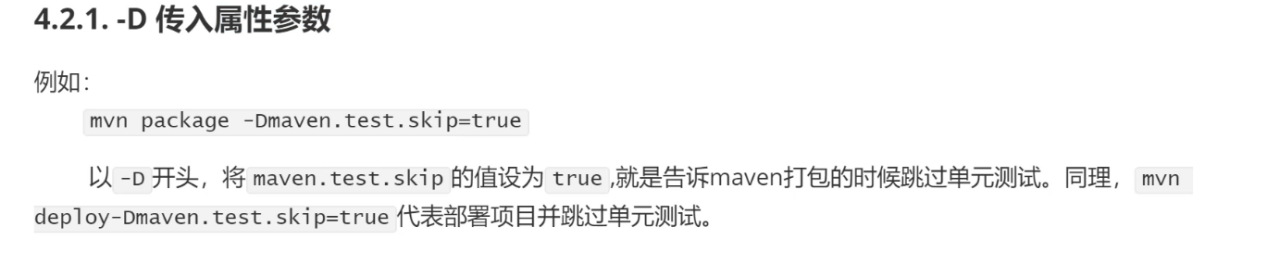
更改本地仓库位置

使用阿里镜像





Jar包 核心配置文件都放在pom.xml 中

将maven 仓库变成阿里巴巴镜像

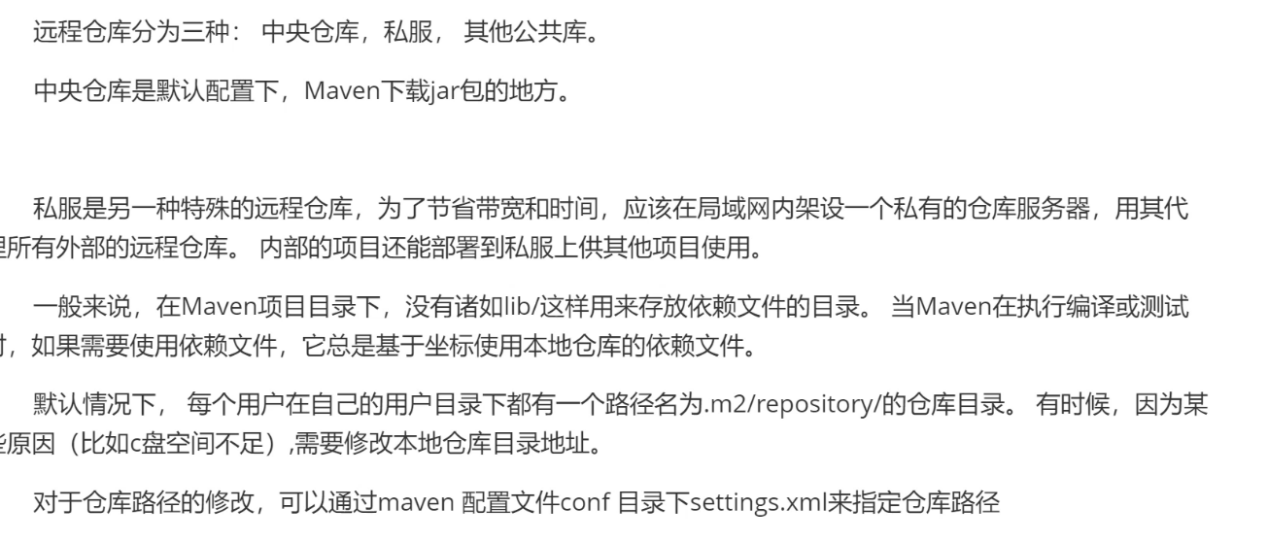
# Maven web 项目 更改配置

Sdk 是jdk 的一个子集

pom 中tomcat 插件<plugin>  
 <groupId>org.apache.tomcat.maven</groupId>  
 <artifactId>tomcat7-maven-plugin</artifactId>  
 <version>2.1</version>  
 <configuration>  
 <port>8081</port> <!-- 启动端口 默认:8080 -->  
 <path>/MavenWeb</path> <!-- 项目的站点名，即对外访问路径 -->  
 <uriEncoding>UTF-8</uriEncoding> <!-- 字符集编码 默认：ISO-8859-1 -->  
 <server>tomcat7</server> <!-- 服务器名称 -->  
 </configuration>  
</plugin>

# Pom中jetty插件

<plugin>  
 <groupId>org.eclipse.jetty</groupId>  
 <artifactId>jetty-maven-plugin</artifactId>  
 <version>9.4.11.v20180605</version>  
  
 <!-- 热部署，每10秒扫描一次 -->  
  
 <!-- 可指定当前项目的站点名 -->  
 <configuration>  
 <httpConnector>  
 <port>8081</port>  
 </httpConnector>  
 <scanIntervalSeconds>10</scanIntervalSeconds>  
 <webAppConfig>  
 <contextPath>/test</contextPath>  
 </webAppConfig>  
 </configuration>  
</plugin>





<https://mvnrepository.com/>

去maven 仓库寻找jar 包

# Maven 多模块项目



Maven parent 下创建quick strart service项目 dao项目

Controller 是一个webapp 项目

设置模块之间的依赖关系

Dao 新建userDao

添加maven\_dao 的依赖

Controller 添加service 的依赖



引入依赖 才能使用对应类中的方法

Controller 要引入servlet 依赖 去maven 官网 仓库获取



Controller 里面创建一个java 包

所有项目配置都install 一下

# 项目打包

添加java 文件夹 标记为源文件夹

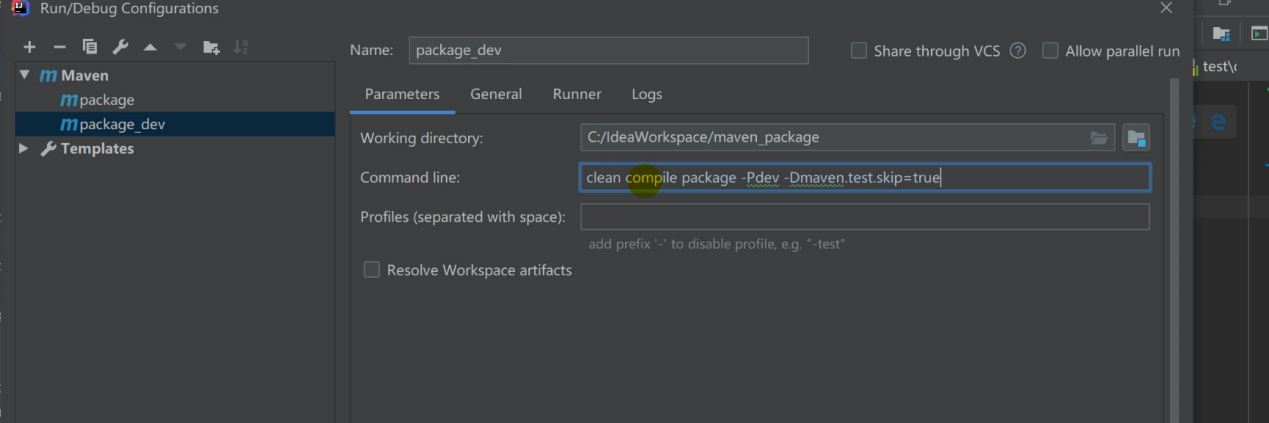
与上面的文件夹名称保持一致

放入pom 中



Activation activatebydefault 默认打包环境

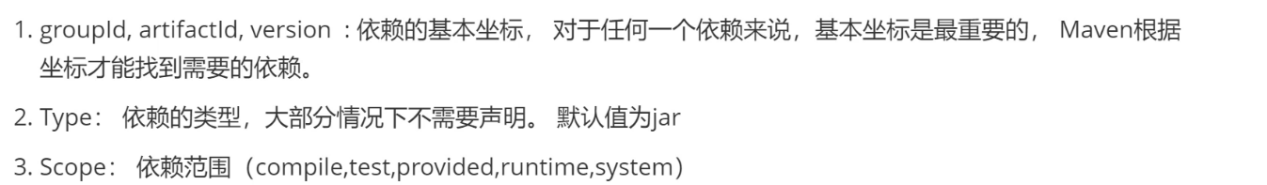
资源文件放入pom.xml的build 中



使用命令来打包 跳过测试文件？



Resource 写道pom.xml build 中





# 需要将公司中的maven setting.xml

拷贝到自己的电脑中

# Maven模块之间可以继承

模块之间互相引入 就可以使用jar 包了

父项目的编译命令会将子模块的也进行编译

# 达成jar包才能用 pom 形式的jar 中的类引用不了

Install 发布到本地

Deploy 发布到私服

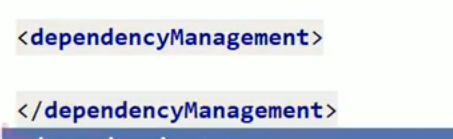
变动之后可以打成1.1 version 不同的版本

可以在maven 这边看依赖的jar 包

Snapshot 和release 版本的区别

Release 在push 到私服的时候必须修改版本号

Dependency management作用

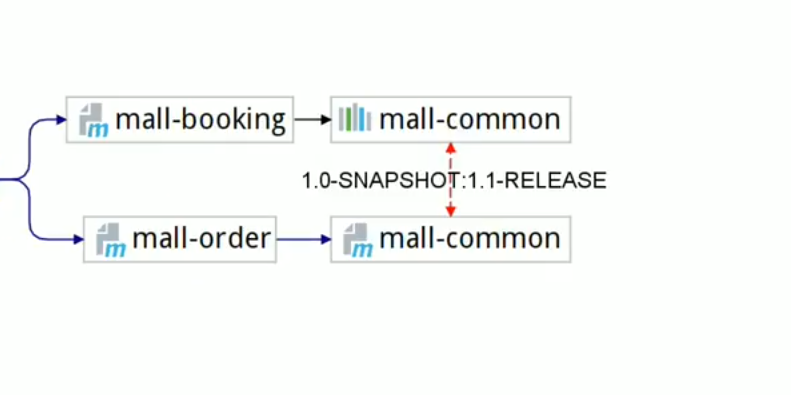


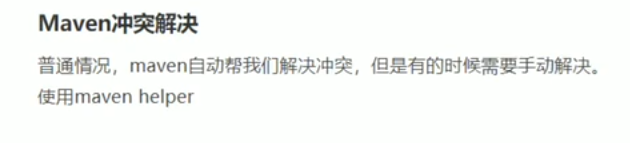
[maven](https://so.csdn.net/so/search?q=maven&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/weixin_39153210/article/details/_blank)中使用dependencyManagement 来提供一种管理依赖版本号的方式；一般是在最顶层父工程中使用该标签。  
****dependencyManagement 只是声明依赖，并不实际引入，因此需要在子项目中引入****

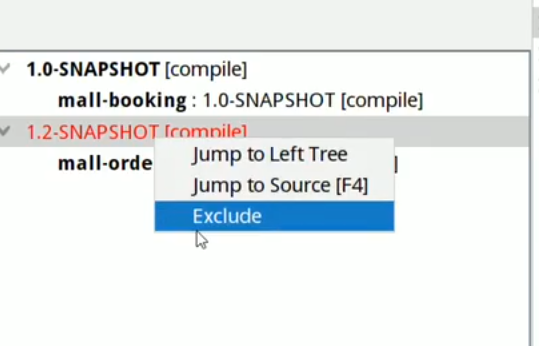
****子模块中需要引入，且就不需要 再指定版本号码了****

# 一个模块引入两个不同的模块，这两个模块含有同一个版本，称为版本冲突

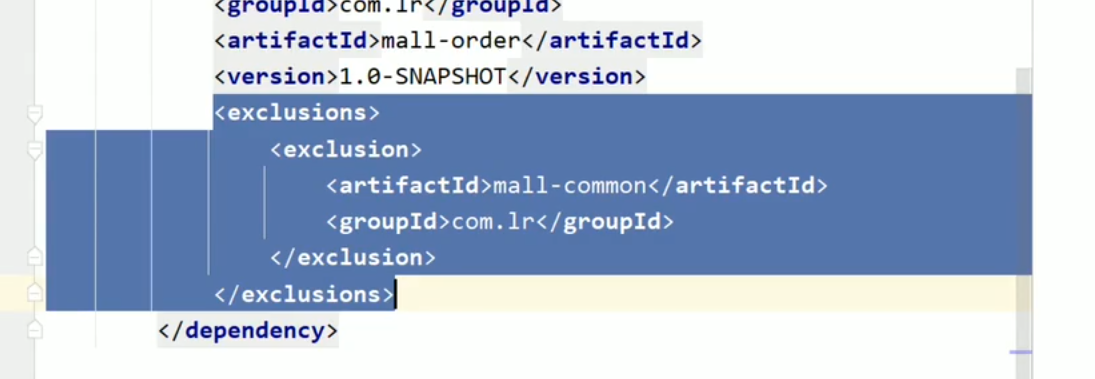
****会生成图****

谁放前 就用谁 maven 自动解决冲突





Exclude 排除解决冲突



排除

