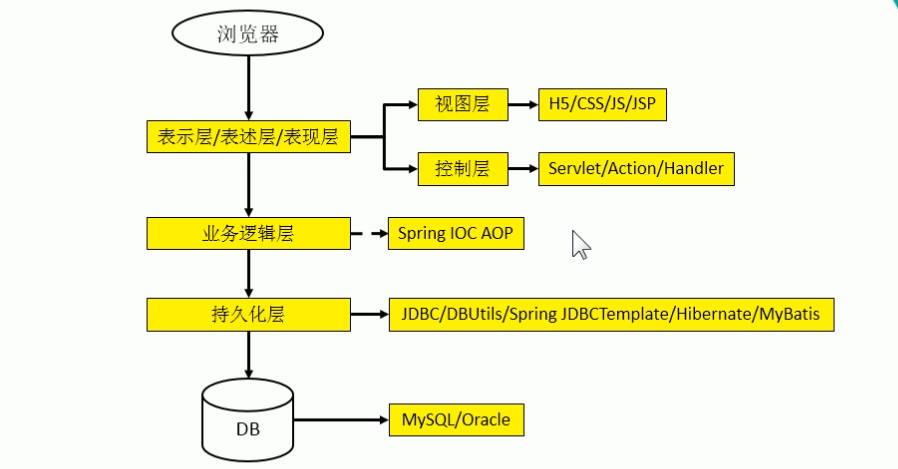
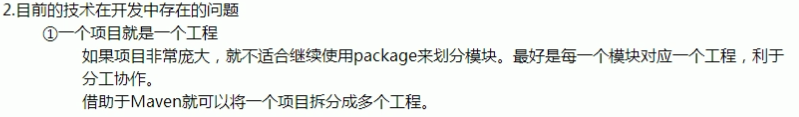
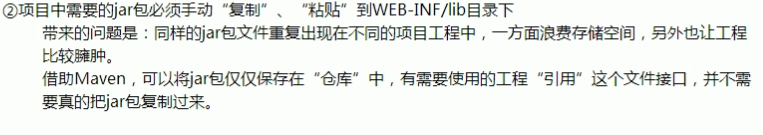
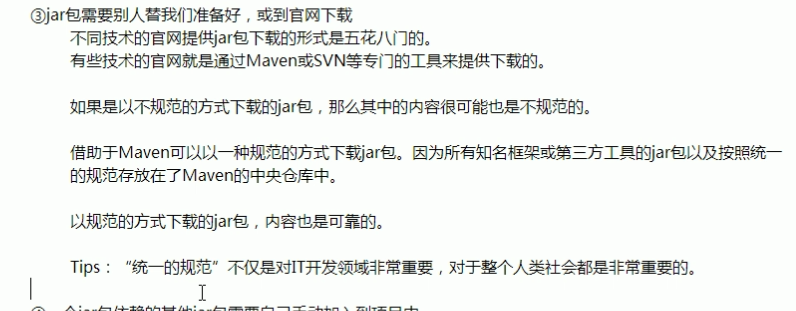
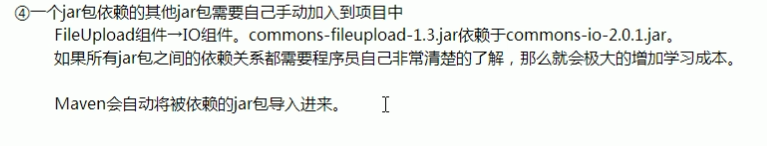
现在掌握的技术：

目前的技术在开发中存在的问题

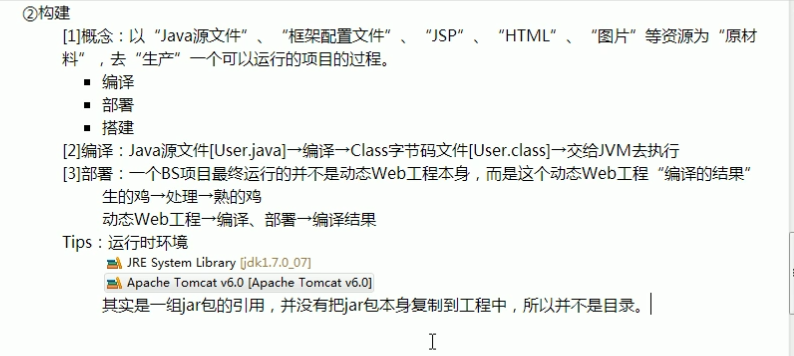




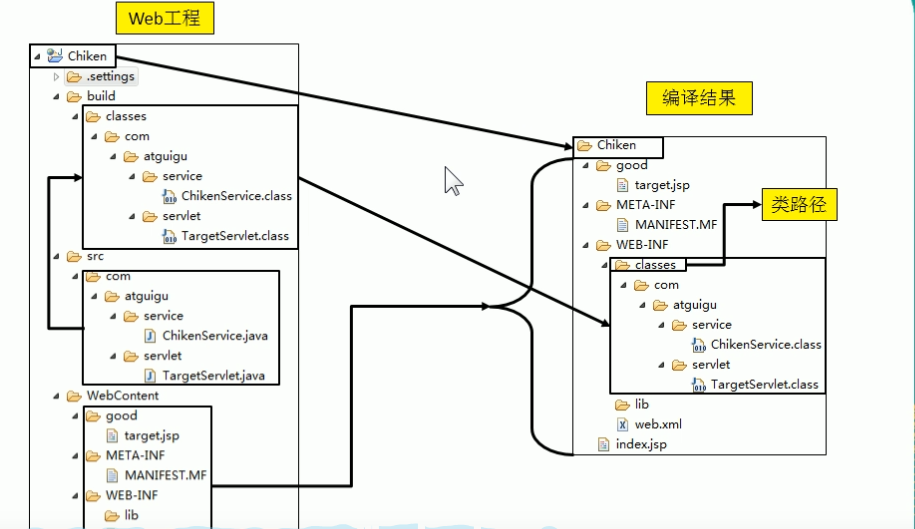


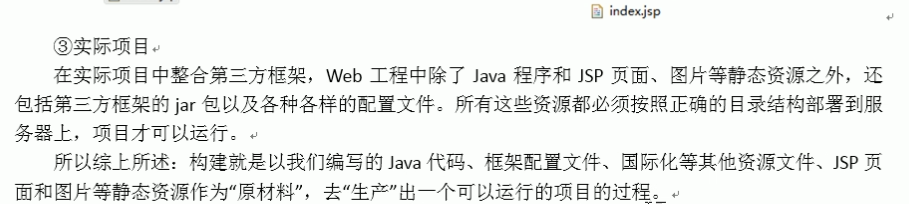


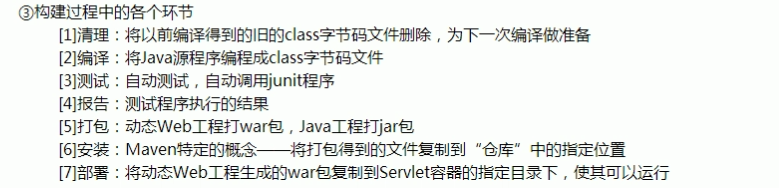
②构建：



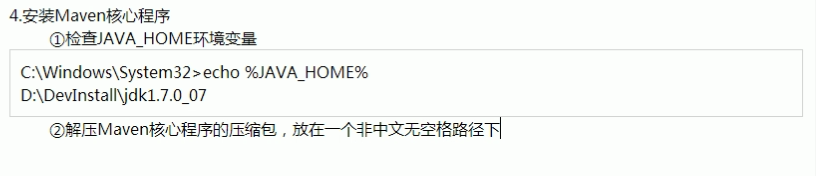
Web项目的编译结果：一个BS项目最终运行的并不是动态Web工程本身，而是这个动态Web工程“编译的结果”，开发过程中，所有的路径或配置文件中配置的类路径等都是以编译结果的目录结构为标准的。











我这里用的是3.3.9

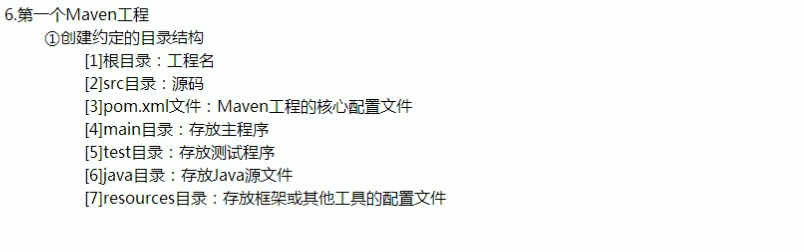




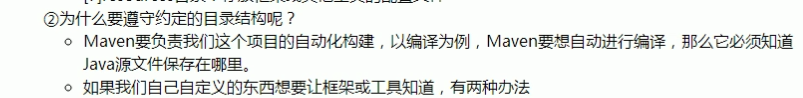
Maven的核心概念：



6.第一个Maven工程



为什么要遵守约定的目录结构呢：约定>配置>编码





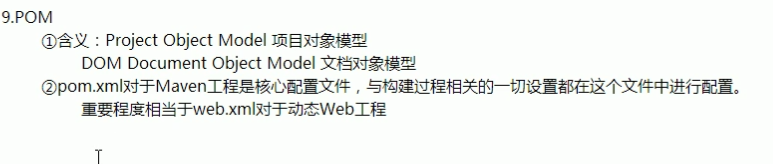
常用Maven命令：

Maven仓库问题：我这里使用的是idea就不改了。





9.POM：

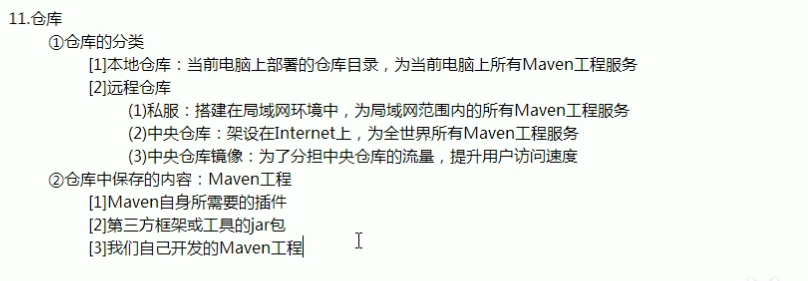


gar就是maven的坐标查找功能：

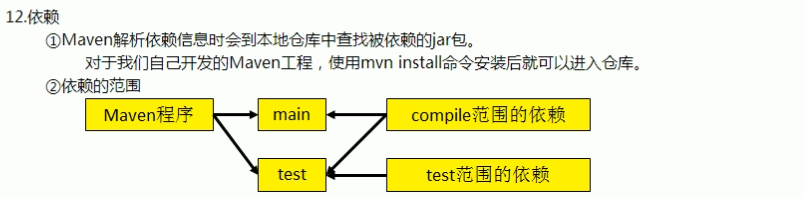


例如spring的pom文件：





依赖:

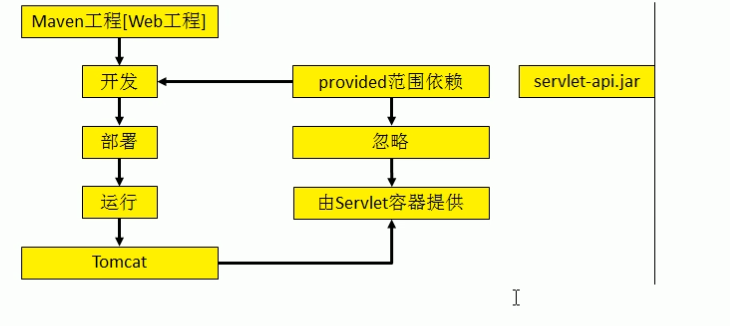
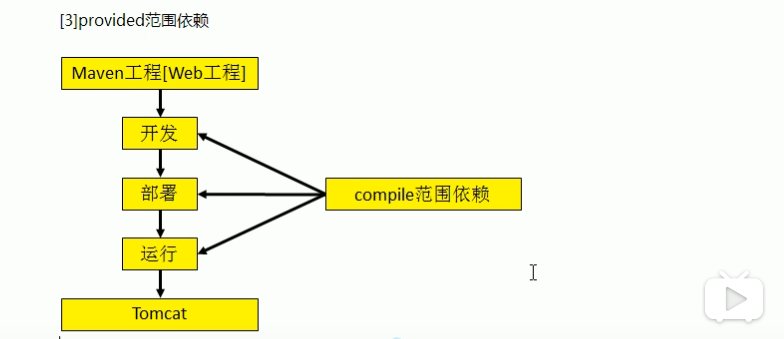




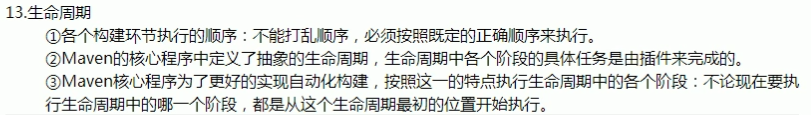


provided范围依赖:这个依赖就是编译程序的时候需要的jar包，服务器也有，所以不需要打包过去，只有开发过程中需要的，就是provided依赖。

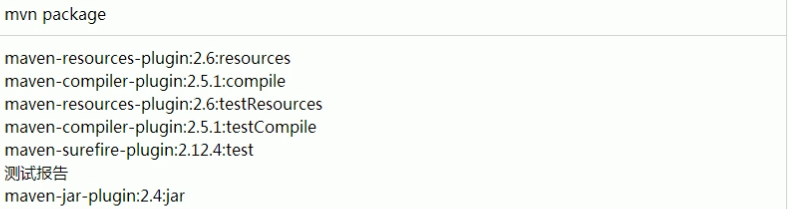




生命周期：不论要执行生命周期的哪一个阶段都要从生命周期最初的位置开始执行。

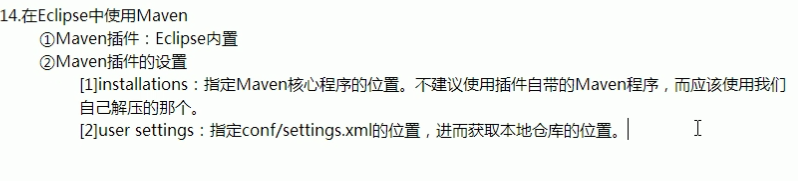


例如：



插件和目标：

在Eclipse中使用Maven:IDEA里面类似，就是在setting里面找到maven设置。



把maven默认jdk修改成1.8的方法：

<https://blog.csdn.net/qq_38225558/article/details/84999872>



1. 创建Maven版的Java工程：

<https://www.cnblogs.com/adaonling/p/11669008.html>

1. 创建Maven版的Web工程：

<https://blog.csdn.net/czc9309/article/details/80304074>

如果有问题可以看这里：

<https://www.cnblogs.com/zhangchengzi/p/9865546.html>

Maven创建的web工程没有java文件夹：

<https://www.jianshu.com/p/099eeca28c0f>

Idea怎么自动下载包：

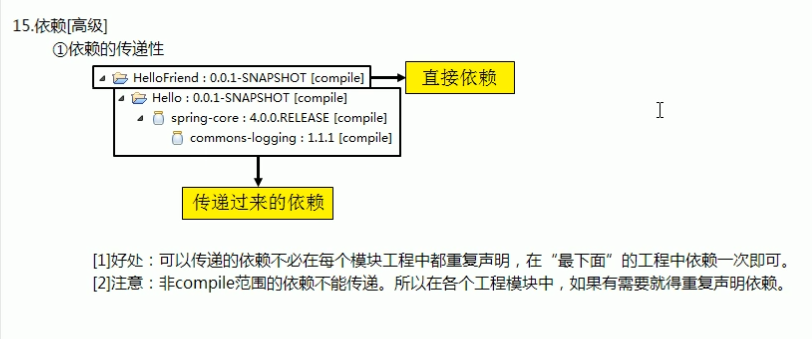
<https://blog.csdn.net/qq_42388853/article/details/108365053>

1. 执行Maven命令：

  
注意导入依赖的时候范围要写好，不然很容易出现冲突，而且报错很奇葩，例如空指针异常！！

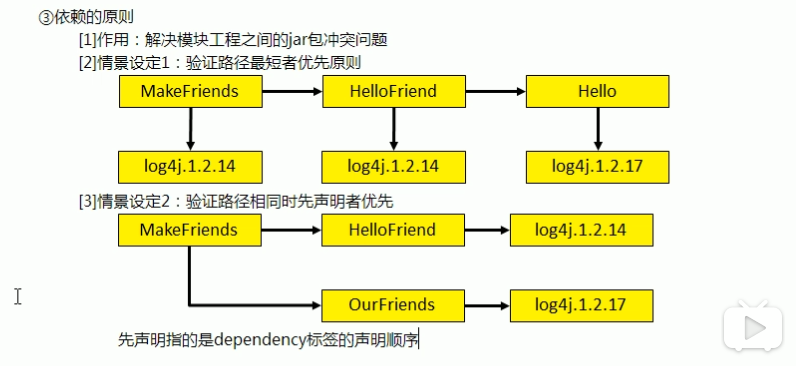
1. 依赖[高级]

依赖的传递性：一个项目导入后其他项目也可以获取到该依赖。



Idea移除依赖：<https://www.cnblogs.com/geekdc/p/9399619.html>

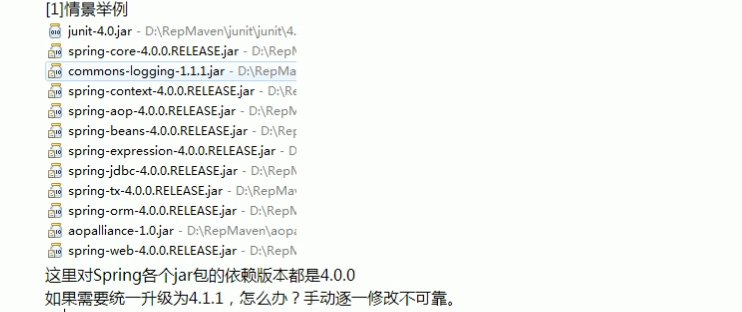
依赖的原则：



先声明的为准：意思是在pom文件里先写出依赖的，例如MakeFriend依赖hellofriend和ourfriend但是hellofriend的依赖在pom文件前面，就使用hellofriend的依赖。

统—管理依赖的版本：





配置方式:



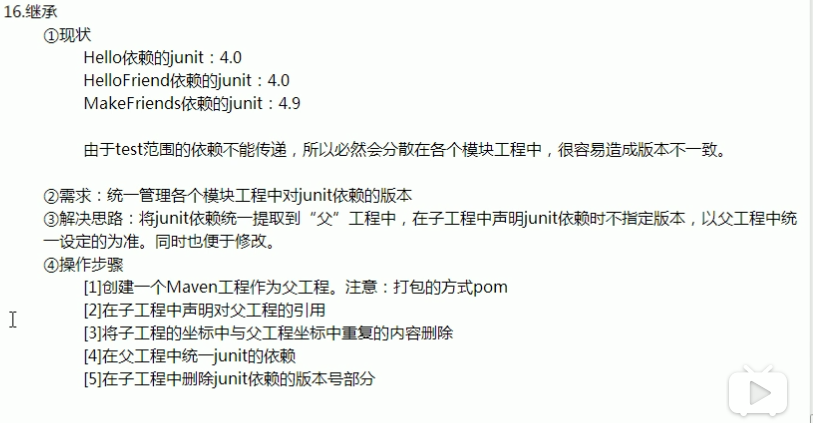
举例说明：



]其实properties标签配合自定义标签声明数据的配置并不是只能用于声明依赖的版本号：



继承：





在子工程中声明对父工程的引用：



将子工程的坐标中与父工程坐标中重复的内容删除：

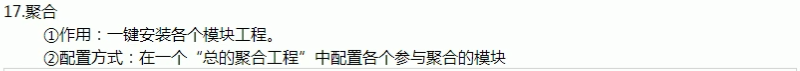





继承以后一定要先安装（install）父工程然后在安装子工程！！！

聚合：就是为了一次性安装多个，并且先安装父工程然后再安装子工程，保证安装成功。





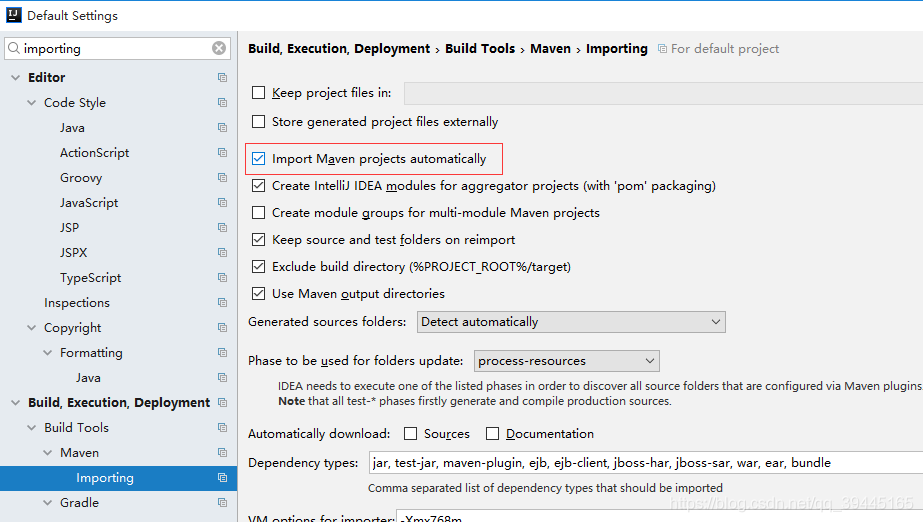
顺序无所谓，他会根据需求决定先后顺序。

自动部署：（一般不用，还是在idea上跑）

查战依赖信息的网站：

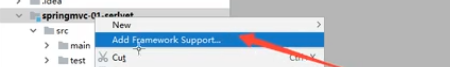
<https://mvnrepository.com/>

maven设置自动导入依赖：

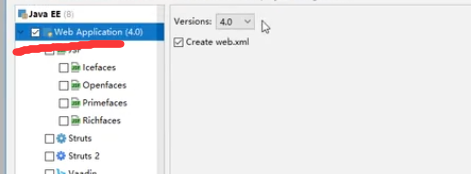


如何把一个普通的maven项目转变为web项目：

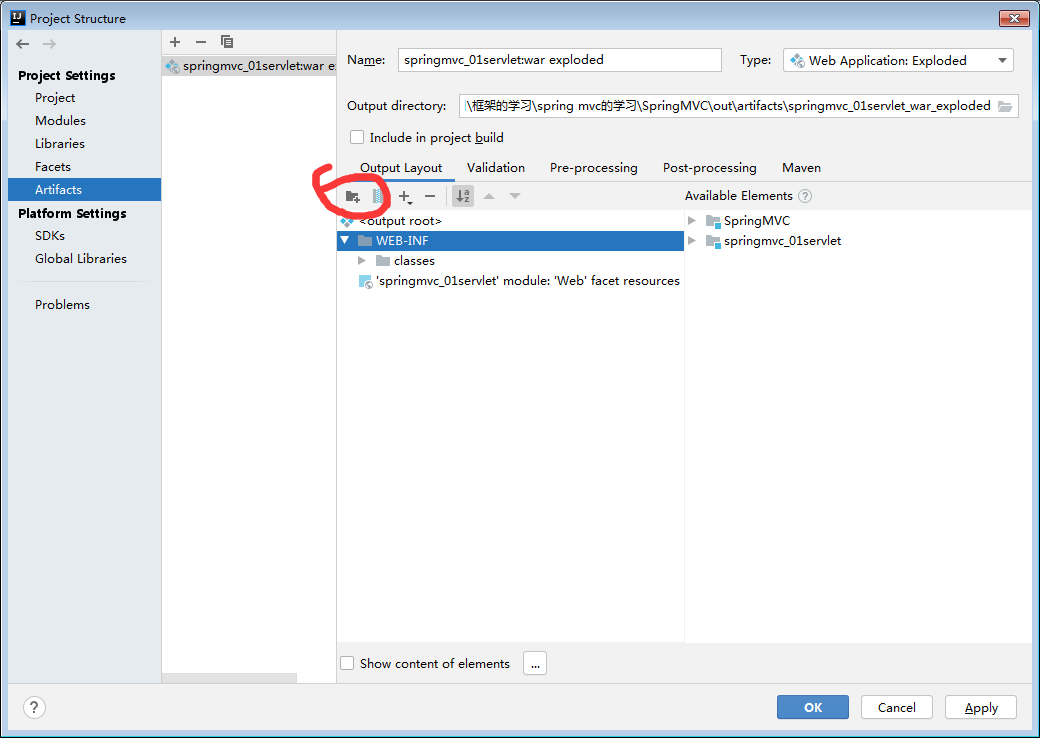
第一步添加框架支持

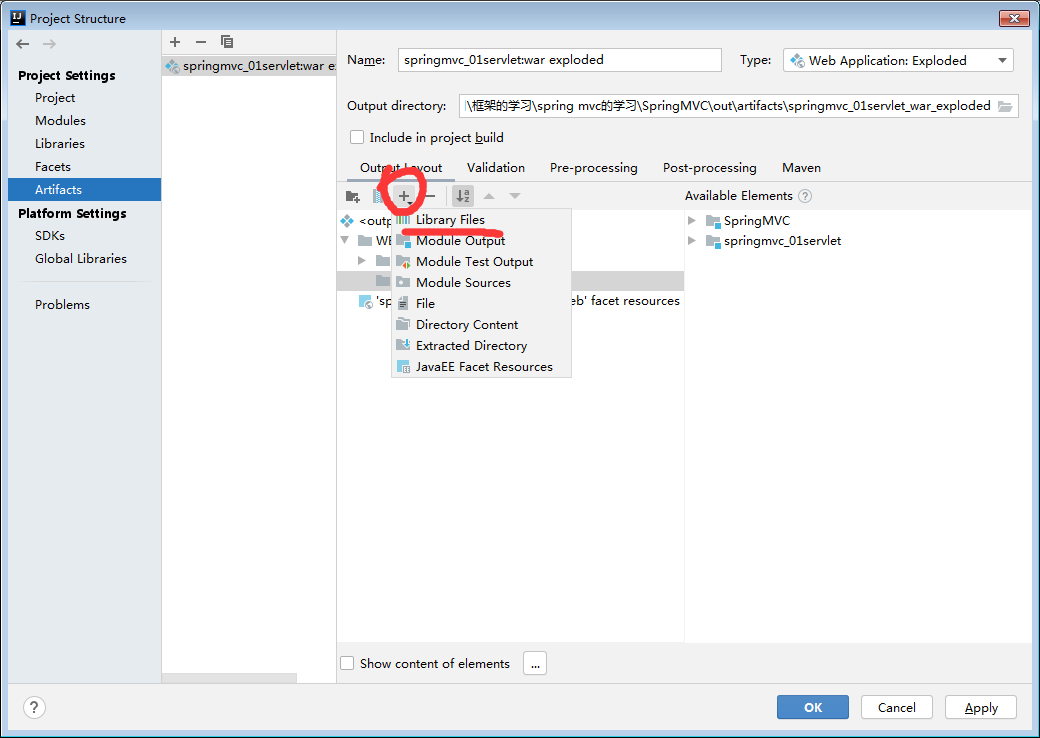


第二步勾选web application



手动添加lib：





阿里云镜像配置：

<mirror>

<id>nexus-aliyun</id>

<mirrorOf>central</mirrorOf>

<name>Nexus aliyun</name>

<url>http://maven.aliyun.com/nexus/content/groups/public</url>

</mirror>

仓库位置配置，repo：

<localRepository>D:\maven\repo</localRepository>

