从这一讲开始，正式开始讲解maven一个一个的知识点853769620

提一点点，大家也会看过一些其他的maven培训视频，我为什么还要自己重新讲解一遍呢？这个原因之前解释过了，这里我再提一点，就是哪怕是有些东西别的视频讲过了，但是我不太认可。

做这个架构班的课程，市面上有的东西，我都会去看一下。他们讲解的细的程度，还有讲解的思路，我不认可。不够细，不够讲的透彻，不够讲的明白；还有讲解的思路，都是按照一种书本上的知识体系组织的结构去讲解的。我也不太认可。

我个人后面架构班的思路，其实都是希望按照公司里，在做项目的过程中，如果需要学习一块技术，按照先入门，然后跟着项目开发进度走，一步一步顺着项目的进度，项目的需求，项目遇到的挑战，把这个技术一个一个的知识点给你讲解出来，结合项目来讲解，告诉你这个技术的这个知识点能够解决这个项目的什么问题。

我个人觉得这样子讲解，效果是最好的，最理想的。

而不是按照书本上的组织方式来讲解，比如maven生命周期，依赖管理，仓库，插件。书本是可以给你学习和参考，但是效果不是最理想的。

大家有没有一个感触，就是自己掌握的最好的技术，和印象最深刻的，永远是那些自己公司里遇到过的项目中的挑战，促使你去学习和研究的东西，然后搞定了之后，解决了项目里的问题，这一块你会特别有底气，特别扎实，印象特别深刻。

反之，如果你学习了很多的东西，各种看书，各种看视频，结果没有跟最实际的项目结合起来，导致你后面发现自己印象很淡薄，甚至都忘了。

1、maven坐标的介绍

每个maven项目都有一个坐标

groupId + artifactId + version + packaging + classifier，五个维度的坐标，唯一定位一个依赖包

任何一个项目，都是用这五个维度唯一定位一个发布包

实际上后面两个维度较为少用，99%的场景下，唯一定位一个依赖的就是三个维度，groupId + artifactId + version

几乎所有你需要使用的依赖包的各个版本，都在maven的中央仓库里

简单来说，就是在pom.xml里配置你需要的依赖的坐标，然后maven自动从中央仓库下载后缓存到你本地仓库里

打包的时候，可以通过插件直接将本地仓库中的依赖打入你的jar包中，形成一个完整可用的发布包

而我们自己的maven项目，也可能要发布到maven仓库中，给别的项目使用，因此maven是强制性要求每个项目都要定义一个坐标的

2、企业级的坐标设置

groupId：不是你的公司或者组织，但是以你的公司或者组织的官网的域名倒序来开头，然后加上项目名称

比如说，你现在在百度公司，www.baidu.com，公司里任何一个项目的开头，就可以用com.baidu来打头

或者说，你在阿里巴巴，www.alibaba.com，公司里任何一个项目的开头，就是可以用com.alibaba来打头

然后你可能会说，我的公司有点low，连域名都没有，网站都没有，比如你的公司叫中华石杉，没有网站，一般是这样，国内会用自己公司名字的拼音的缩写，或者是公司名称的英文的简写，加上com来组成groupId

com.zhss

咱们这个架构班里所有的课程，所有的工程，groupId都是com.zhss打头

maven-helloworld是属于一个maven这个课程里的一个demo项目，com.zhss.maven

接下来，我们不是要开始做一个oa案例，com.zhss.oa

com.zhss + 项目名称，或者是系统名称

com.zhss.hr

com.zhss.finance

com.zhss.crm

比如我们这个课程，是中华石杉做的，一般会用中文的拼音简写，就是zhss，然后我们虽然没有自己的网站，但是一般可以用常规的com来打头，同时我们当前讲解的是这个课程中的maven部分，因此可以认为是一个maven单独的项目

所以最终我们的groupId，就是com.zhss.maven

artifactId：项目中的某个模块，或者某个服务

一般在一个大项目最终会打成一个war包，可能里面有几十个工程组成，上百万行代码，发布到一个tomcat里面去。一个一个的模块，每个模块基本就是一个工程，最终各个工程会被打到一个war包里去而已。artifactId就是这个项目里的某个模块

com.zhss.oa，oa-organ，organ是缩写，organization

com.zhss.oa，oa-auth，auth是authorization

com.zhss.oa，oa-flow，flow就是流程的意思

咱们这个架构师课程，整个思路，就是完全真实还原一个完整的大型电商网站的开发过程，演进过程

第一个阶段：单块应用

第二个阶段：微服务化架构

第三个阶段：高并发架构+高可用架构+安全性架构+稳定性架构+伸缩性架构+扩展性架构

第四个阶段：亿级流量超大型架构

第五个阶段：企业级框架开发

第六个阶段：企业级中间件开发

第七个阶段：大数据架构

你去任何一个公司，都可以说你作为架构师，搞起来公司java这一块项目的任何一块的架构

我们会发现，一个系统不是由多个模块组成了，不是最终打成一个war包发布。一个系统是由多个服务组成，artifactId就变成了服务的名称。

com.zhss.oa，oa-organ服务

com.zhss.oa，oa-auth服务

com.zhss.oa，oa-flow服务

比如我们刚才写的第一个maven工程，就是个hello world工程，所以artifactId就是hello-world

version：这个工程的版本号，在最后几讲的时候，我们会讲解maven里的企业级的版本管理，是怎么玩儿的

packaging：这个工程的发布包打包方式，一般常用的就jar和war两种，java -cp执行一个jar包，war可以放到一个tomcat容器里去跑的web工程

classifier：很少用，定义某个工程的附属项目，比如hello-world工程的，hello-world-source工程，就是源码，可能是类似于hello-world-1.0-SNAPSHOT-source.jar这样的东西。

3、设置了坐标的作用是啥？

很简单，你每一次一个工程写好了，都有版本，groupId+artifactId+version就成了这一次这个工程目前这个状态的唯一的标识和定位。

一个groupId+artifactId+version，就定位了这个项目某个时间点的一个特定版本的代码，也就是一个特定的版本的代码的jar包。

然后呢在maven世界里，特别是开源的那些项目，junit，spring，mybatis，都是在每个版本的代码开发好之后，设置好对应版本的坐标，主要是version。然后将这个版本的代码的jar包，上传到maven的中央仓库。供其他任何人来使用这个版本的代码的jar包。

然后，我们在自己的maven项目里，就可以通过<dependency>加上那个依赖的坐标，groupId+artifactId+version，去声明我要用那个依赖的哪个版本的代码的jar包。maven通过这个坐标唯一定位到那个项目的那个版本的代码的jar包。从中央仓库下载下来，给你放到本地仓库里去。

然后，你自己的maven项目为什么也要一个坐标，因为可能你写好了一个模块之后，你就会需要将这个版本的代码打成一个jar包，放到仓库里去，给公司里其他人去依赖和使用。只有你也有一个坐标，才能让别人唯一定位你的项目某个版本的代码的jar包，让maven下载了之后给别人用。

坐标的思路，三步走

（1）坐标有哪些元素

（2）坐标的每个元素都是什么意思，怎么设置

（3）设置了坐标之后的意义在哪儿