1. Maven 约定的目录结构（约定即大家承认的规则）

HelloMaven/

---/src

└---/mian #你的主程序java代码和配置文件

└---/java #程序包以及包中的配置文件

└---/rescources #java程序中要使用的配置文件

└---/test #存放测试程序的代码和文件

└---/java #测试程序包以及包中的配置文件

└---/rescources #测试程序中要使用的配置文件

---/target #执行mvn compile后生成的结果目录

#Maven编译的java程序，最后的class文件都放在target目录中

---/pom.xml #(Maven的核心配置文件)

1. Maven仓库的地址修改：

备份conf/setting.xml文件

修改<localRepository>x/x/X</localRepository>

1. Maven仓库源修改

修改：

<mirrors>

<!-- 阿里云仓库 -->

<mirror>

<id>alimaven</id>

<mirrorOf>\*</mirrorOf>

<name>aliyun maven</name>

<url>http://maven.aliyun.com/nexus/content/repositories/central/</url>

</mirror>

</mirrors>

1. Maven仓库的概念

仓库是什么？

仓库是存放东西的，存放maven使用的jar以及项目使用的jar（maven使用的插件、项目使用的第三方工具jar）

仓库的分类

a 本地仓库 计算机上的文件夹，存放各种jar

b 远程仓库 在互联网上的，使用网络才能使用的仓库

1 中央仓库 最权威，所有开发人员都共享使用的一个集中的仓库

2 镜像仓库 中央仓库的镜像

3 私服仓库 用在公司内部局域网中对内使用

仓库的使用：

Maven仓库的使用不需要人为的干预

eg：开发人员需要使用mysql 驱动：maven首先检查本地仓库---私服---镜像---中央仓库---得到需要jar包

1. POM pom.xml

**A 模型版本**

**<modelVersion>**声明项目描述符遵循哪一个POM模型版本。模型本身的版本很少改变，虽然如此，但它仍然是必不可少的，这是为了当Maven引入了新的特性或者其他模型变更的时候，确保稳定性。对于Maven2/3来说，只能是4.0.0**</modelVersion>**

**B 坐标**

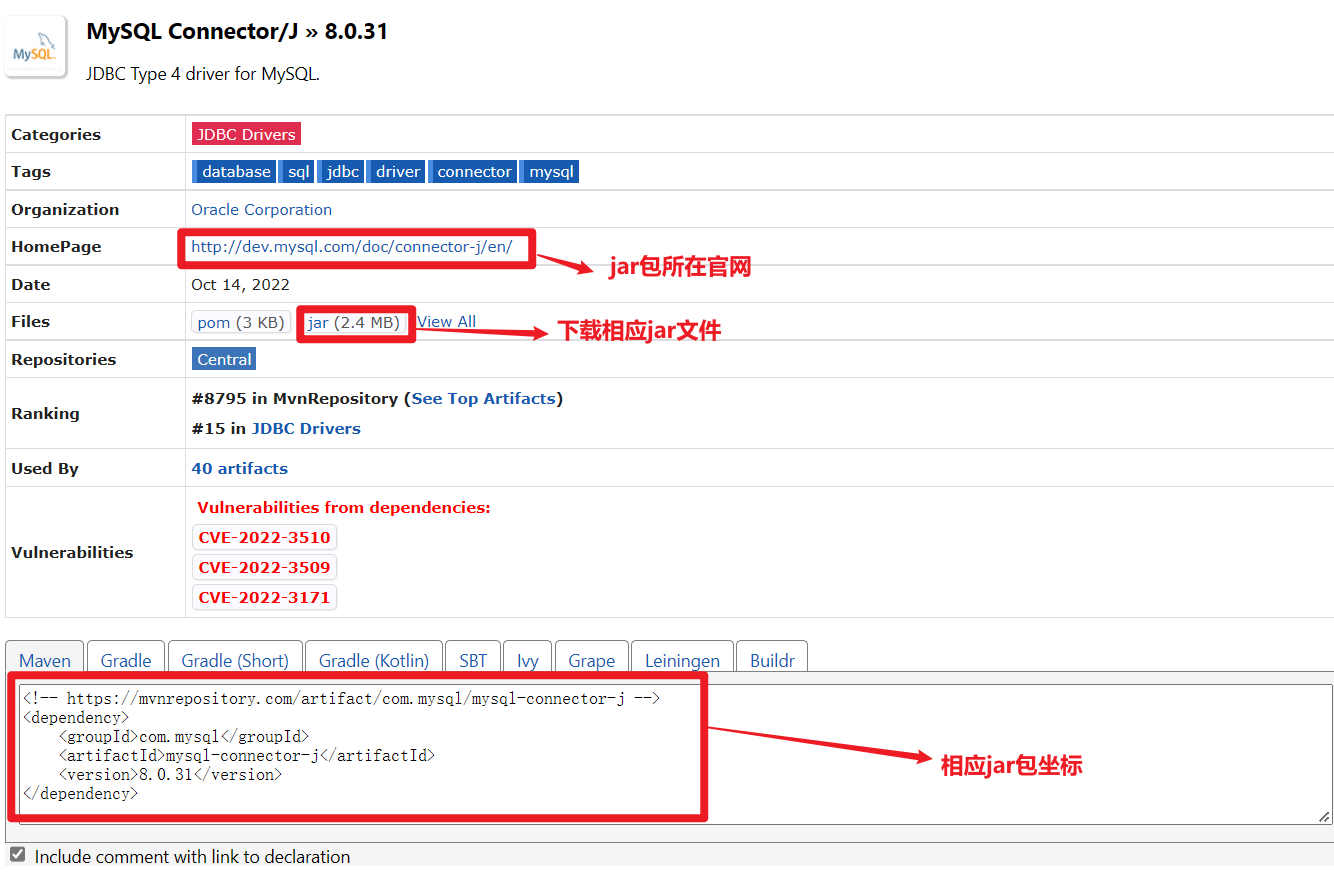
**<groupId>**公司或者组织名称的倒写，配置时生成的路径也是由此生成**</groupId>**

**<artifactId>**自定义的项目名称**</artifactId>**

**<version>**自定义的版本号**</version>**

其中信息：groupId、artifactId、version三个元素构成了一个Maven项目的基本**坐标**，在众多的Maven项目中可以唯一定位到某一个项目。坐标也决定着将来项目在仓库中的路径以及名称

[www.mvnrepository.com](http://www.mvnrepository.com) 搜索使用的中央仓库，



**C 打包类型**

**<packageing>**项目打包的类型，含jar/wer/rar/ear/pom，可不写，默认为jar**</packageing>**

**D 添加依赖**

**<dependencies>**

**<dependency>**

<groupId>org.mybatis.spring.boot</groupId>

<artifactId>mybatis-spring-boot-starter</artifactId>

<version>1.2.0</version>

**<dependency>**

**</dependencies>**

**F 属性信息**

定义一些配置属性

**<properties>**

<java.version>1.8</java.version>

< !--项目构建源码方式-->

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

<spring-boot.version>2.3.7.RELEASE</spring-boot.version>

……

**</properties>**

**G 配置信息**

maven在进行项目的构建时，配置信息，例如指定编译java代码使用的jdk的版本等等

<build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
</build>

**H 其它**

parent 继承

modules 聚合

1. Maven的生命周期

每一个阶段都对应着一个命令：

1 clean 清理

2 compile 编译 主程序

3 test-compile 测试 编译测试程序

4 test 报告 会生成目录surefire-reports保存测试结果

5 package 打包

6 install 安装

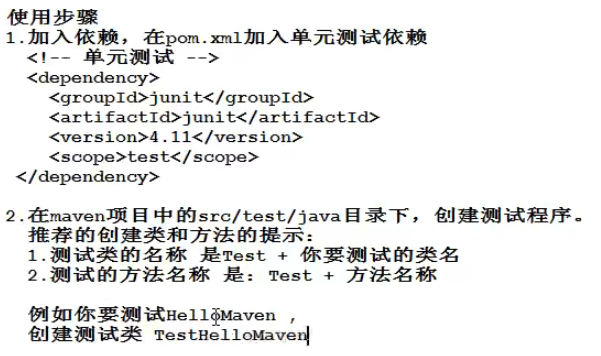
7 deploy 部署

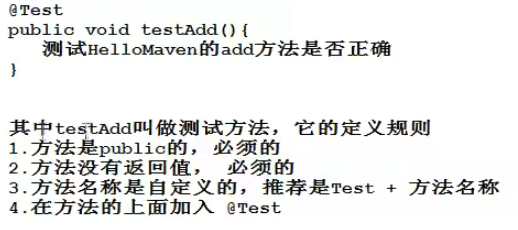
maven真正完成其功能的是一些插件，即jar文件





1. 单元测试junit





1. 练习maven命令



