# Maven

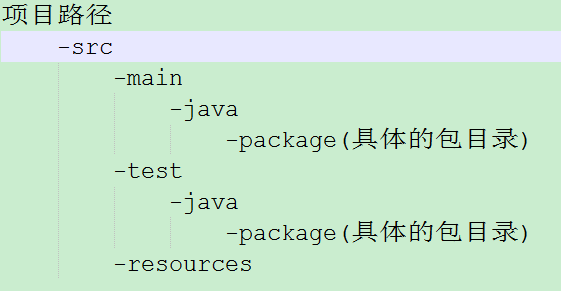
## Maven的安装

1. 新建环境变量 M2\_HOME 指向maven 的安装目录

D:\apache-maven-3.3.3

1. Path 添加 ;%M2\_HOME%\bin
2. 验证是否安装成功 mvn-v 安装成功将显示maven、java等其他信息

## Maven 的目录结构



## Maven的常用命令

mvn-v 查看maven版本

mvn compile 编译项目

mvn test 测试项目

mvn package 打包项目

mvn clean 删除target

mvn install 安装jar包到本地仓库

## Maven 自动构建目录骨架

### 根据提示步骤建立骨架

mvn archetype:generate 此命令会在maven中央仓库中寻找可创建maven的所有模板项目，大多数是无用的

解决办法

{1.使用mvn archetype:crawl命令,它会在C:\Users\buha\.m2\repository目录下生成一个archetype-catalog.xml文件

2.将archetype-catalog.xml移到上一层目录,也就是C:\Users\buha\.m2

3.这时再运行mvn archetype:generate -DarchetypeCatalog=local就可以达到你想要的目的了。}

{mvn archetype:generate-DarchetypeCatalog=internal}

### 将所有参数直接设置

mvn archetype:generate-DgroupId=com.rzp.maven02 -DartifactId=maven02 -Dversion=1.0.1 -Dpackage=com.rzp.maven02 -DarchetypeCatalog=loacl

## Maven的构件

Maven中所有的依赖、插件、项目构建都称之为 构件

## Maven的坐标

构件的唯一标识就是坐标

以下的groupId、artifactId、version组成一组坐标

<groupId>com.rzp.maven02</groupId>

<artifactId>maven02</artifactId>

<version>1.0.1</version>

## Maven的仓库

本地仓库、远程仓库

镜像仓库

中央仓库的地址

maven-model-builder-3.3.3.jar –pom.xml

### 镜像仓库的设置

conf\ settings.xml 146行

<mirror>

<id>maven.net.cn</id>

<mirrorOf>central</mirrorOf>

<name>central mirror in china</name>

<url>http://maven.net.cn/content/groups/public</url>

</mirror>

### 更改本地仓库的位置

本地仓库默认位置 ${user.home}/.m2/repository

更改位置

conf\ settings.xml 53行

复制<localRepository>E:/repo</localRepository>更改为新的仓库地址 把setting复制一份至新仓库

## Eclipse与maven集成

1. 设置本地maven

Maven installations 设置为本地maven路径

1. 设置 userSettings

设置为 本地仓库下复制的setting.xml

1. 设置java运行maven

在jdk安装位置编辑VM运行参数

-Dmaven.multiModuleProjectDirectory=$M2\_HOME

## POM依赖

### 依赖范围<scope></scope>

1. Compile 编译 测试 运行
2. Provided 编译 测试 J2EE对于servlet API
3. Runtime 测试 运行 JDBC
4. Test 测试 junit
5. System 与本机系统关联
6. Import 导入的范围，只使用在dependencyManagement中，表示从其他的pom中导入dependcy的配置

### 依赖传递

在<dependencies>内添加 <dependency> 内通过坐标 来进行依赖

C依赖 B B依赖A C就依赖A 为依赖传递

若要打断对A的依赖传递 添加<exclusions> <exclusion>坐标来排除依赖

### 依赖冲突

1. 短路优先

A B C X

A D X

将为A依赖 短路的X版本

1. 路径相同，先声明 先优先

## Maven项目的聚合

1. Packingg改为pom
2. 添加<modules> <module>
3. 运行选择install

## Maven项目的继承

1. 使用<dependencyManagement>来管理依赖
2. 可以在<properties>中自定义属性，通过EL表达式来获取值
3. Package改为pom
4. 在子项目POM中添加parent在parent中添加父项目的坐标

## Maven创建WEB项目

1. 创建maven时选择webapp
2. 在properties属性中—Project Facets中将Web Model改为目标模式
3. 在Deployment Assembly 注意将Maven中的依赖输出至WEB-INF/lib文件夹

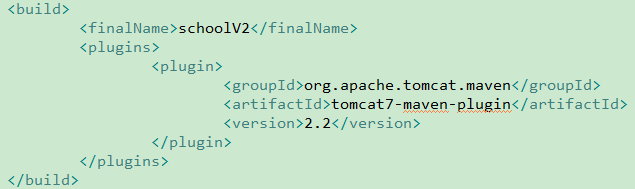
注意将test目录移除

注意将classpath下的文件输出至WEB-INF/classes

1. 添加servlet-api编译依赖



1. POM.xml中在build添加 tomcat集成测试插件，运行时以命令tomcat:run即可启动项目



## Maven修改默认JDK

<profile>

<id>jdk-1.7</id>

<activation>

<activeByDefault>true</activeByDefault>

<jdk>1.7</jdk>

</activation>

<properties>

<maven.compiler.source>1.7</maven.compiler.source>

<maven.compiler.target>1.7</maven.compiler.target>

<maven.compiler.compilerVersion>1.7</maven.compiler.compilerVersion>

</properties>

</profile>