

第五讲 Maven项目管理

初版时间: 2017年8月15日

初版制作者: 林焕彬

教材版本号: szlanyou-V1.1

修订时间:

修订者:

教材版本号: szlanyou-V1.1

目录



Maven基础

Maven关键知识点

pom.xml文件

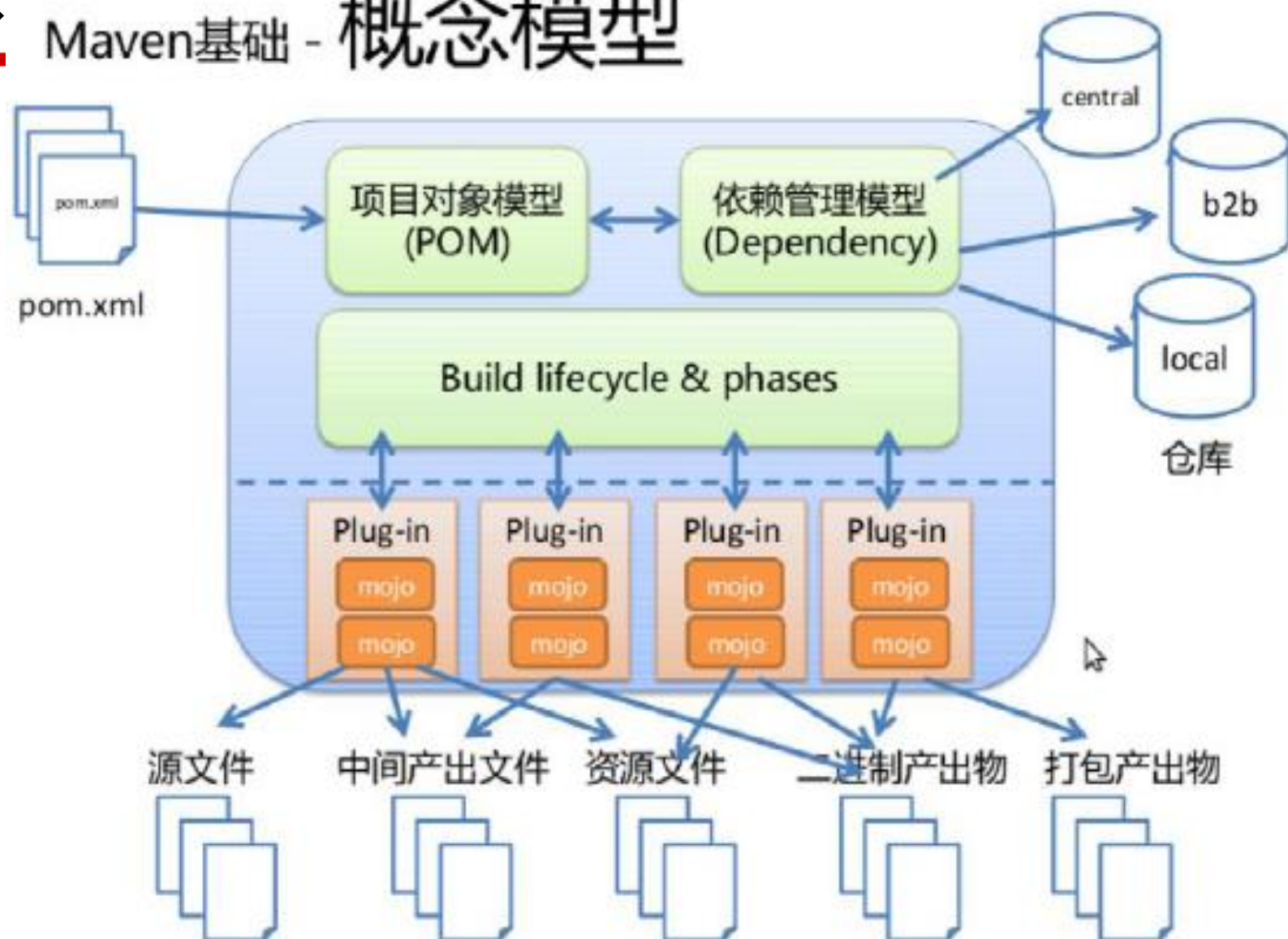
Maven常用命令

Maven实践

一、Maven介绍 1-1

- Maven是一个项目管理工具，它包含了一项目对象模型(Project Object Model)，一组标准集合，一个项目生命周期(Project Lifecycle)，一个依赖管理系统(Dependency Manangement System)，和用来运行定义在生命周期阶段中插件目标的逻辑。
- Maven是面向技术层面，针对Java开发项目管理工具，它提供了构建工具所提供功能的超集，除了构建功能之外，Maven还可以管理项目结构、管理依赖关系、生成报告、生成Web站点、有助于团队成员之间的交流与协作。

Maven基础 - 概念模型



一、使用Maven的好处 1-3

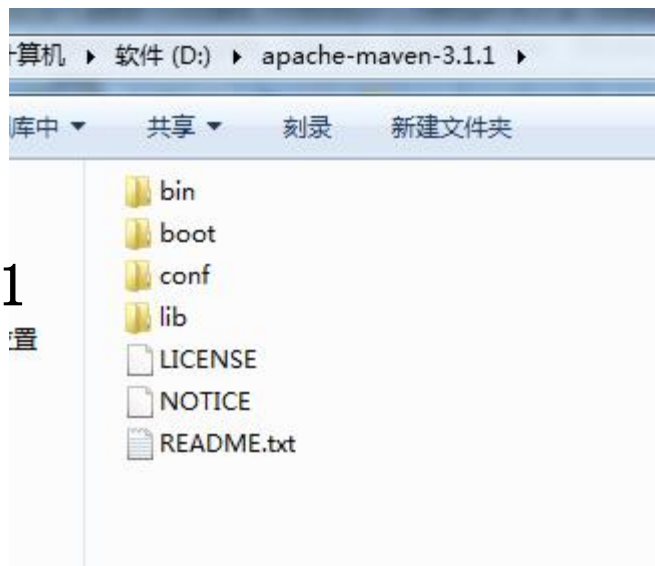
- **指导开发：**提供了**Java**项目的最佳开发实践，自由开发项目骨架而可自动生成项目结构。
- **自动编译：**不仅仅只像**Ant**自动编译，还包括测试，打包，发布，文档生成，项目站点生成.....
- **依赖管理：****Maven**可以方便地管理应用程序依赖，例如第三方依赖、模型依赖
- **无限扩展：**插件模式可以无限增强**Maven**功能，例如通过**Tomcat**、**Jetty**插件可以自由控制其服务器。

一、使用Maven的好处 1-4

- 持续集成：鼓励开发者积极提交代码，更早地发现程序错误，在并行开发中稳妥推进。
- 开发协作：更简单和谐的团队协作

一、Maven的安装配置 1-5

- 机器安装JDK5以上，并配置好环境变量。
- 下载Maven，解压，配置环境变量：
 - 增加M2_HOME指向D:\apache-maven-3.1.1
 - 把%M2_HOME%\bin变量加入Path变量
 - 查看Maven是否安装成功（cmd下）：

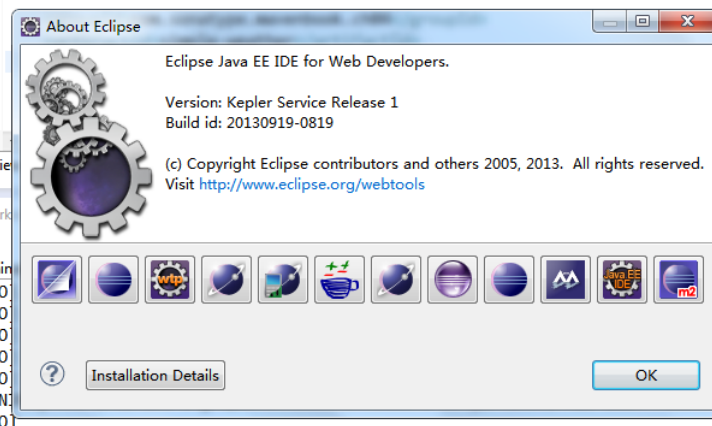
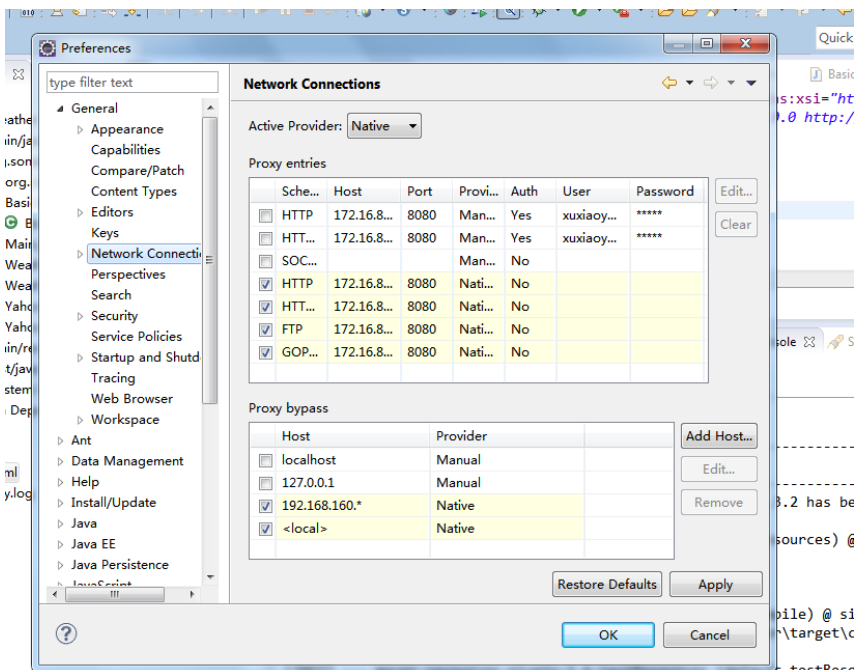


```
1051 E:\Projects\simple-weather>mvn -v
1052 Apache Maven 3.1.1 (0728685237757ffbf44136acec0402957f723d9a; 2013-09-17 23:22:22+0800)
1053 Maven home: D:\apache-maven-3.1.1
1054 Java version: 1.6.0_16, vendor: Sun Microsystems Inc.
1055 Java home: C:\Java\jdk1.6.0_16\jre
1056 Default locale: zh_CN, platform encoding: GBK
1057 OS name: "windows 7", version: "6.1", arch: "x86", family: "windows"
```

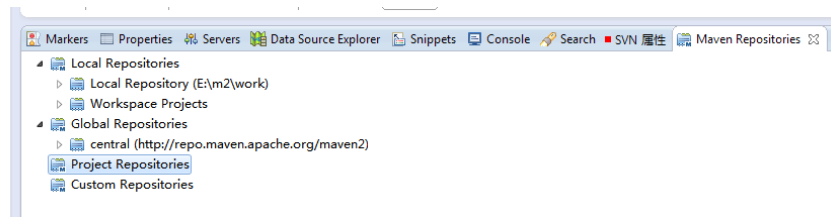
一、Maven的安装配置 1-6

Eclipse官方Kepler版本自带M2Eclipse插件：

配置Eclipse的http代理：

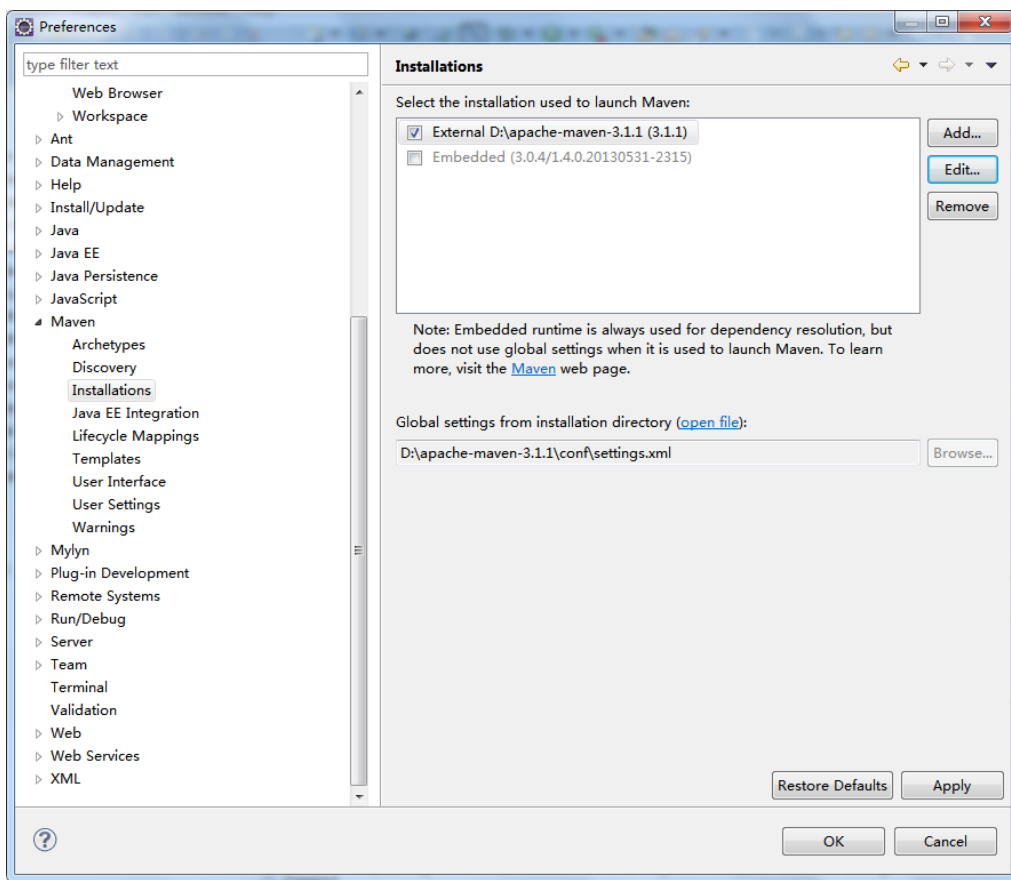


Eclipse中打开Maven视图：

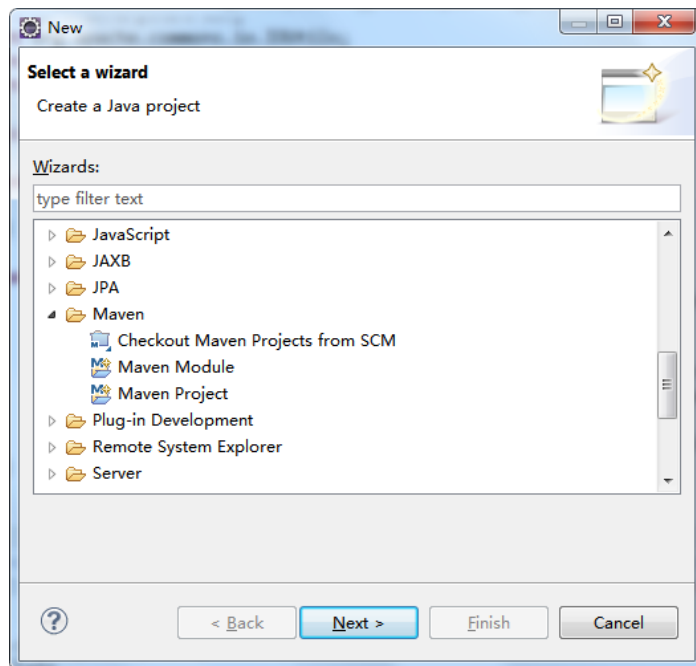


一、Maven的安装配置 1-7

Eclipse里设置Maven安装的路径和settings.xml文件：



Eclipse对Maven工程的支持：



目录



Maven基础

Maven关键知识点

pom.xml文件

Maven常用命令

Maven实践

二、Maven关键知识点 2-1

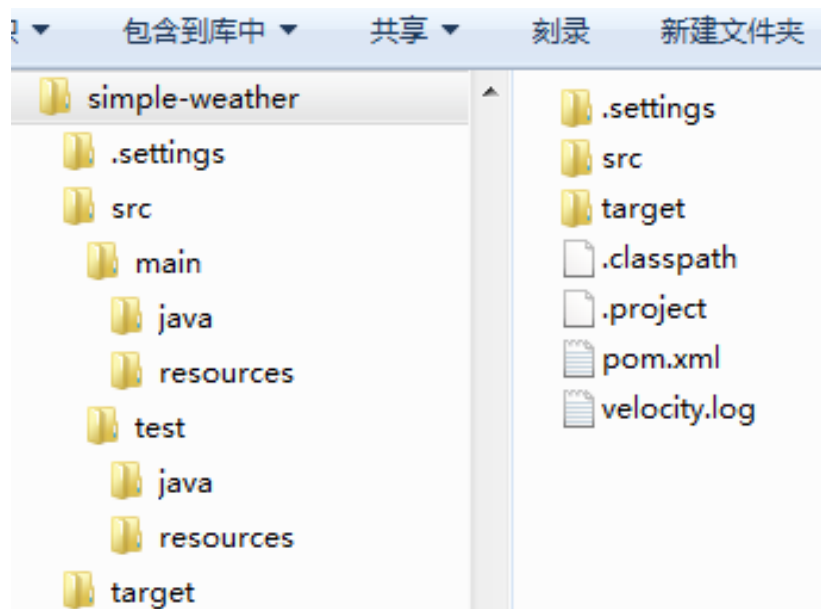
- **Project:** 任何您想build的事物，Maven都可以认为它们是工程。这些工程被定义为工程对象模型(POM, Project Object Model)。一个工程可以依赖其它的工程；一个工程也可以由多个子工程构成。
- **POM:** POM(pom.xml)是Maven的核心文件，它是指示Maven如何工作的元数据文件，类似于Ant中的build.xml文件。POM文件位于每个工程的根目录中。
- **GroupId:** groupId是一个工程的在全局中唯一的标识符，一般地，它就是工程名。groupId有利于使用一个完全的包名，将一个工程从其它有类似名称的工程里区别出来。

二、Maven关键知识点 2-2

- **Artifact:** artifact 是工程将要产生或需要使用的文件，它可以是jar文件，源文件，二进制文件，war文件，甚至是pom文件。每个artifact都由groupId和 artifactId组合的标识符唯一识别。需要被使用(依赖)的artifact都要放在仓库(见Repository)中，否则Maven无法找到(识别)它们。
- **Dependency:** 为了能够build或运行，一个典型的Java工程会依赖其它的包。在Maven中，这些被依赖的包就被称为dependency。dependency一般是其它工程的artifact。
- **Plug-in:** Maven是由插件组织的，它的每一个功能都是由插件提供的。插件提供goal(类似于Ant中的target)，并根据在POM中找到的元数据去完成工作。主要的Maven插件要是由Java写成的，但它也支持用Beanshell或Ant脚本写成的插件。

二、Maven关键知识点 2-3

运算符	说 明
<code>\${basedir}</code>	存放pom.xml和所有子目录
<code>\${basedir}/src/main/java</code>	项目的java源代码
<code>\${basedir}/src/main/resources</code>	项目的资源文件
<code>\${basedir}/src/test/java</code>	项目测试类，比如JUnit代码
<code>\${basedir}/src/test/resources</code>	测试使用的资源文件

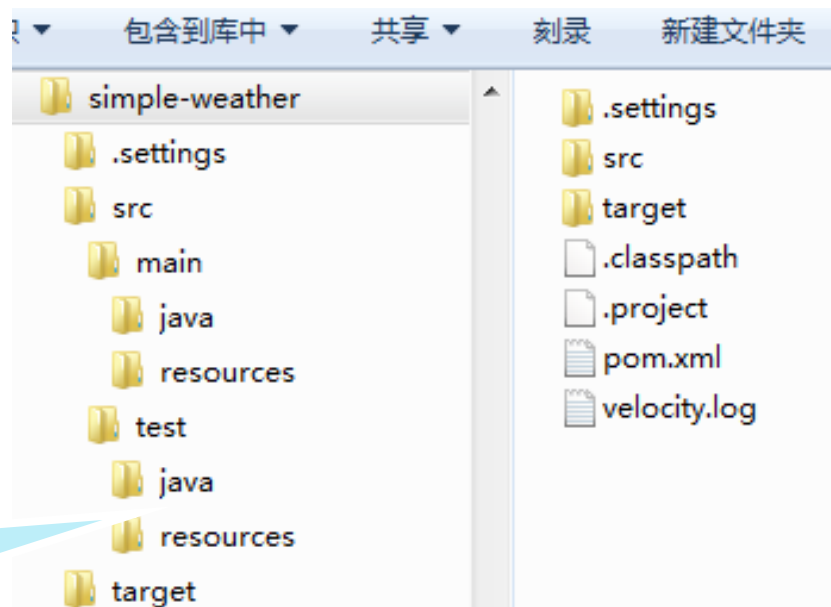


二、Maven关键知识点 2-3

概念：是一系列构建工具的集合

pom.xml：项目对象模型（Project Object Model, POM）

约定大于配置：Maven项目约定目录如右图



Maven坐标：groupId:artifactId:version，唯一确定一个项目

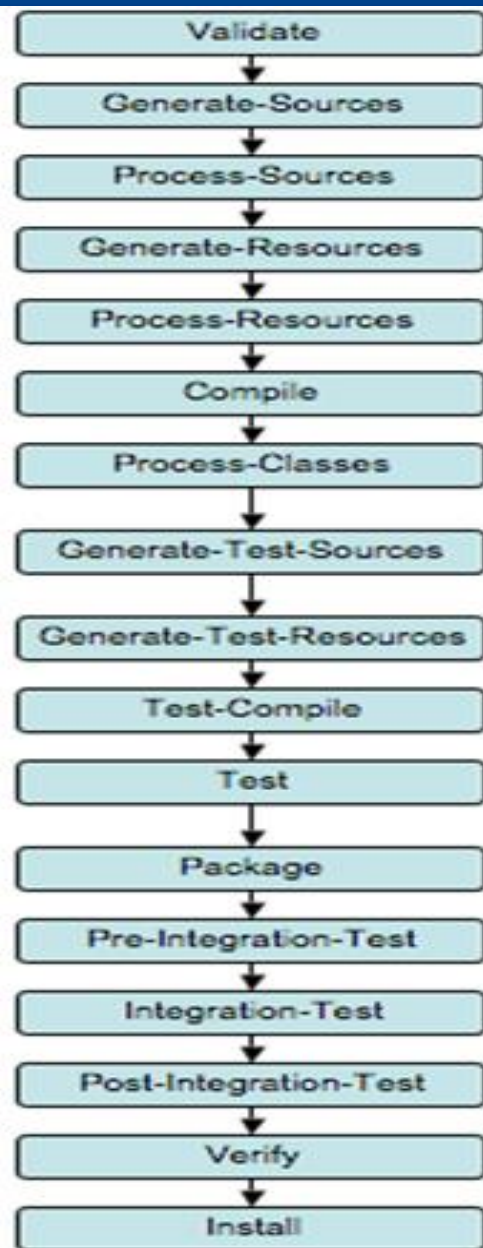
```
<groupId>org.sonatype.mavenbook.multispring</groupId>  
<artifactId>parent</artifactId>  
<version>0.8-SNAPSHOT</version>
```

二、Maven关键知识点 2-4

项目生命周期

几个重要的过程:

compile: 编译
test: 测试
package: 打包
install: 安装
deploy: 发布



二、Maven关键知识点 2-5

● 远程公用仓库：

- Maven内置了远程公用仓库：<http://repo1.maven.org/maven2>
- 这个公共仓库是由Maven自己维护，里面有大量的常用类库，并包含了世界上大部分流行的开源项目构件。目前是以java为主。

● 内部中心仓库：

- 也称**私有共享仓库(私服)**。一般是由公司自己设立的，只为本公司内部共享使用。它既可以作为公司内部构件协作和存档，也可作为公用类库镜像缓存，减少在外部访问和下载的频率。

二、Maven关键知识点 2-6

● 本地仓库：

- Maven会将工程中依赖的构件(Jar包)从远程下载到本机一个目录下管理，通常默认在\$user.home/.m2/repository下。
- 修改本地库位置：在\$M2_HOME/conf/setting.xml文件的<localRepository>元素中指定路径，例如：
<localRepository>D:/my_repository</localRepository>

目录



Maven基础

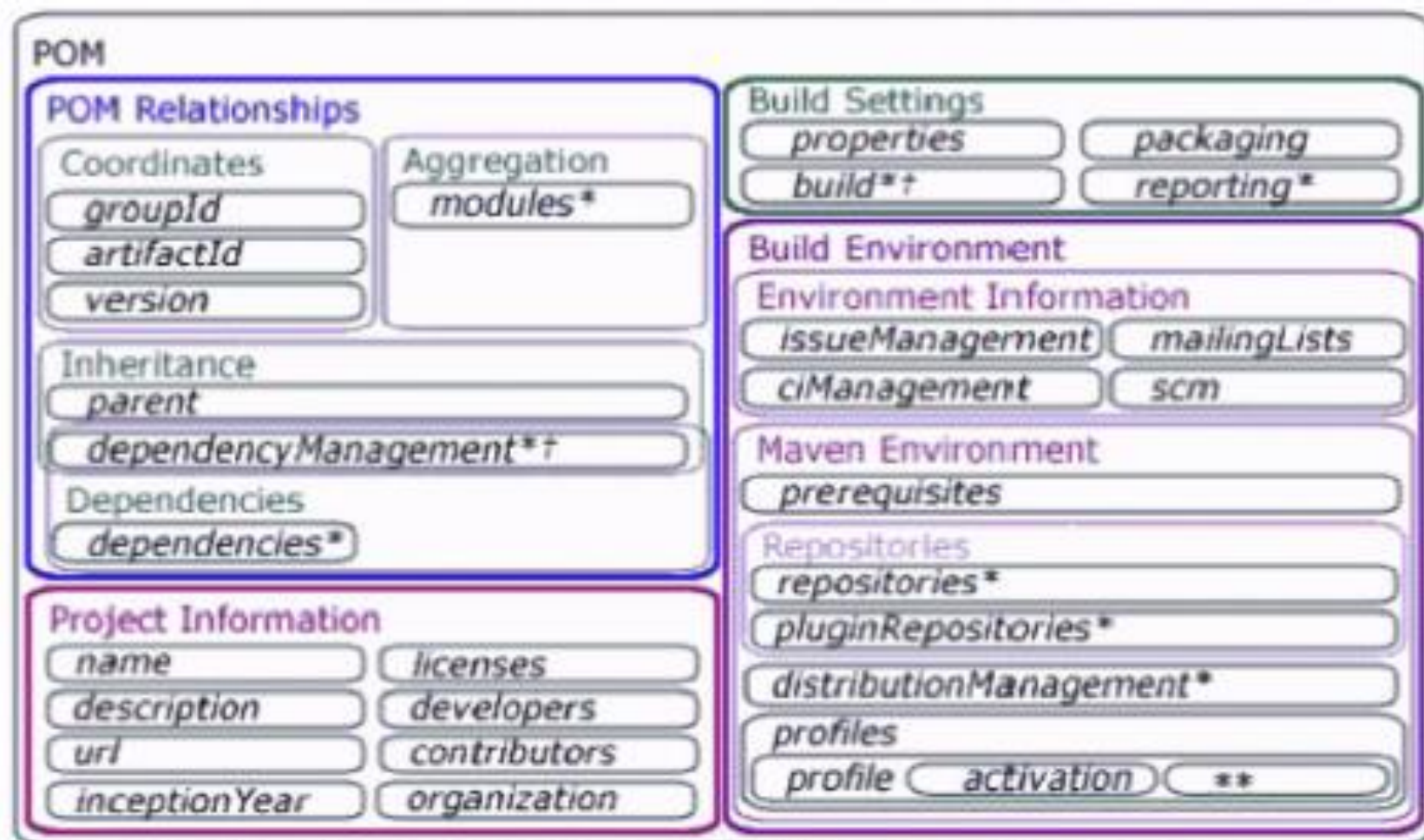
Maven关键知识点

pom.xml文件

Maven常用命令

Maven实践

三、pom.xml文件 3-1



* Element may be overridden (at least mostly) by *profile* element settings

** Profile elements are the *-suffixed elements

† Contains elements for meant for inheritance

三、pom.xml文件 3-2

- **pom.xml**是**Maven**项目的核心配置文件，位于每个工程的根目录，指示**Maven**工作的元数据文件。：
 - `<project>`：文件的根节点。
 - `<modelversion>`：`pom.xml`使用的对象模型版本。
 - `<groupId>`：创建项目的组织或团体的唯一 Id。
 - `<artifactId>`：项目的唯一 Id, 可视为项目名。
 - `<packaging>`：打包类型，一般有JAR,WAR,EAR 等
 - `<version>`：产品的版本号。
 - `<name>`：项目的显示名，常用于 Maven 生成的文档。
 - `<url>`：组织的站点，常用于 Maven 生成的文档。
 - `<description>`：项目描述，常用于 Maven 生成的文档。

三、pom.xml文件 3-3

- **pom.xml**是**Maven**项目的核心配置文件，位于每个工程的根目录，指示**Maven**工作的元数据文件。
 - **<dependencies>**: 构件依赖**<parent>**: 模型继承
 - **<dependencyManagement>**: 依赖管理
 - **<reporting>**: 创建报告
 - **<build>**: 构建
 - **<repositories>**: 引用第三方仓库
 - **<licenses>**: 许可

目录



Maven基础

Maven关键知识点

pom.xml文件

Maven常用命令

Maven实践

四、Maven常用命令 4-1

检测Maven、JDK版本：`mvn -v` 或者 `mvn -version`

获取帮助选项：`mvn -h` 或者 `mvn -help`

显示详细错误信息：`mvn -e`

创建Java项目：`mvn archetype:create`
 `-DgroupId=${groupId}`
 `-DartifactId=${artifactId}`
 `-DarchetypeArtifactId=maven-archetype-webapp`

四、Maven常用命令 4-2

转换成Eclipse工程: `mvn eclipse:eclipse`

清除Eclipse设置信息: `mvn eclipse:clean`

编译: `mvn compile`

编译测试代码: `mvn test-compile`

测试: `mvn test`

清除: `mvn clean`

打包: `mvn package`

发布: `mvn install`

目录



Maven基础

Maven关键知识点

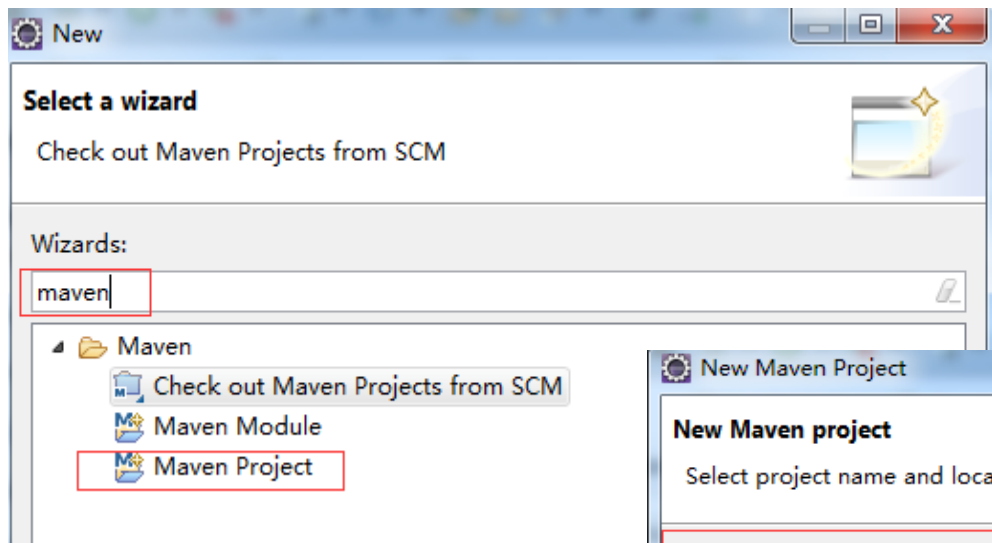
pom.xml文件

Maven常用命令

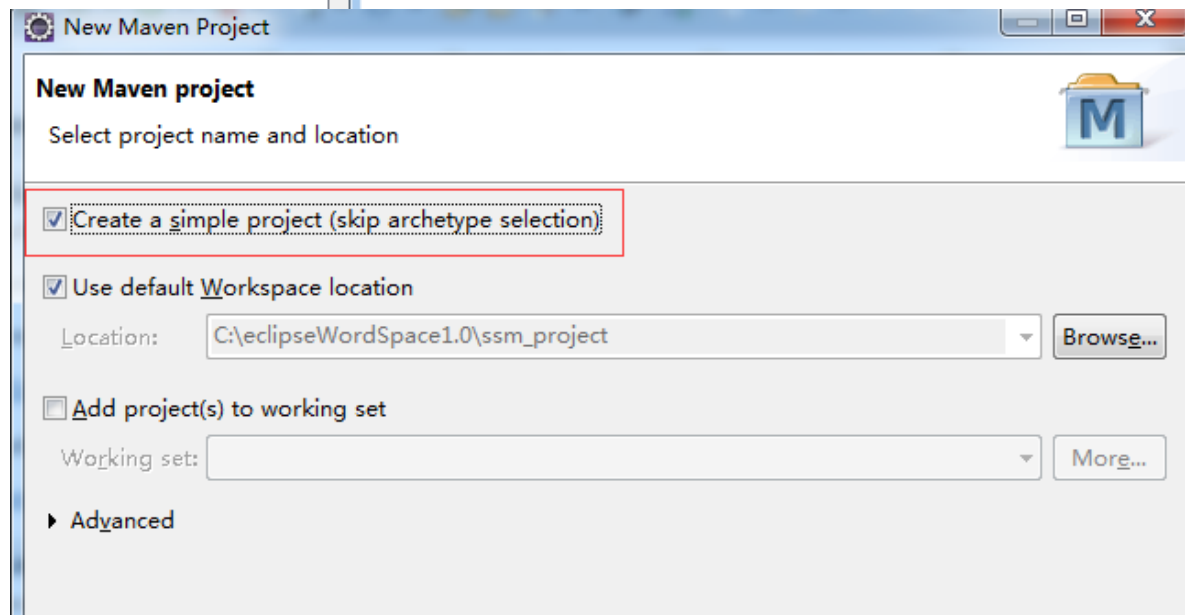
Maven实践

五、Maven的安装配置 5-1

1、Maven工程创建：

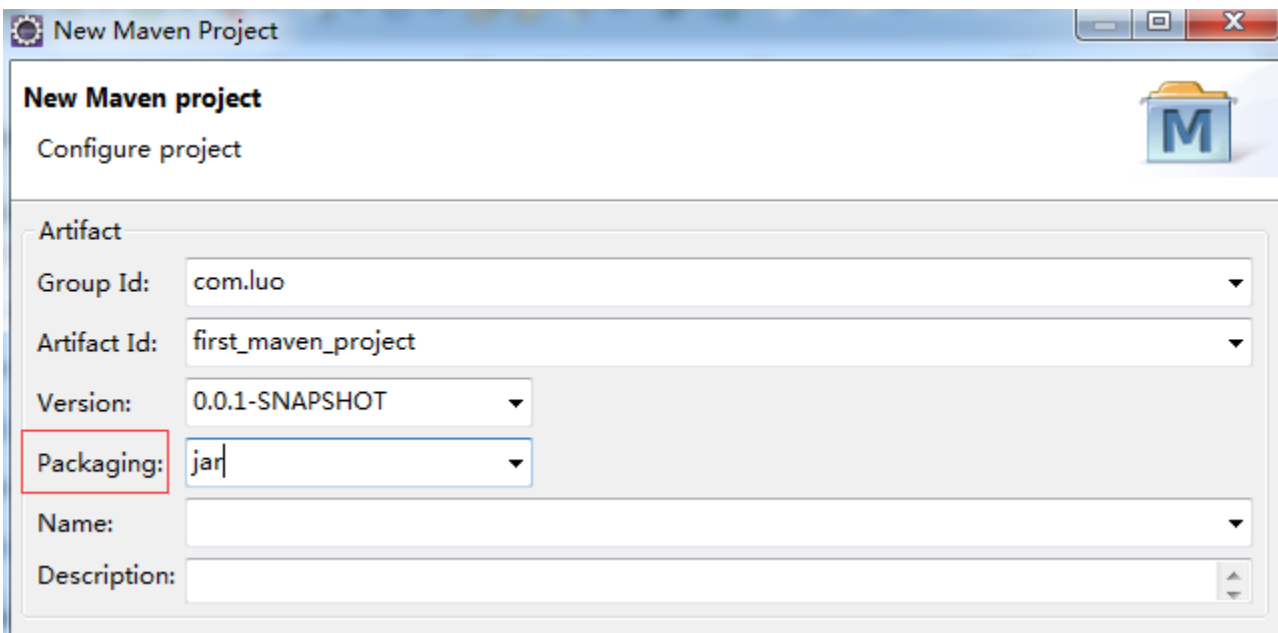


2、选择快速框架：

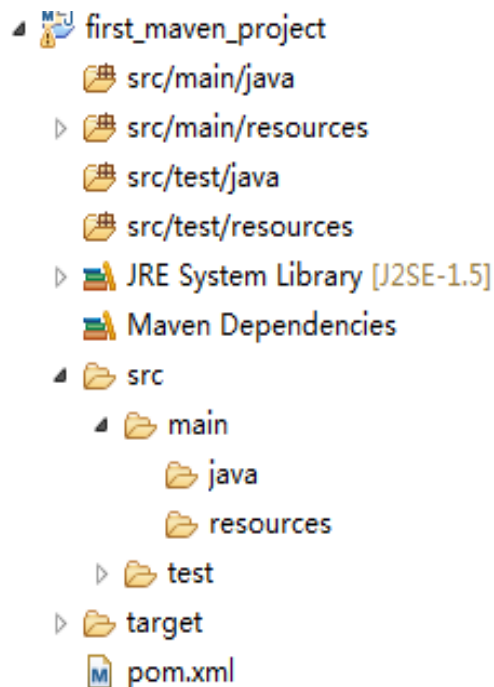


五、Maven的安装配置 5-2

3、输出项目名，包（Packaging，如果只是普通的项目，选jar就好了，如果是web项目就选war，这里我们选择jar）：



4、创建好的项目目录：



五、Maven的安装配置 5-3

5、Maven命令：mvn compile

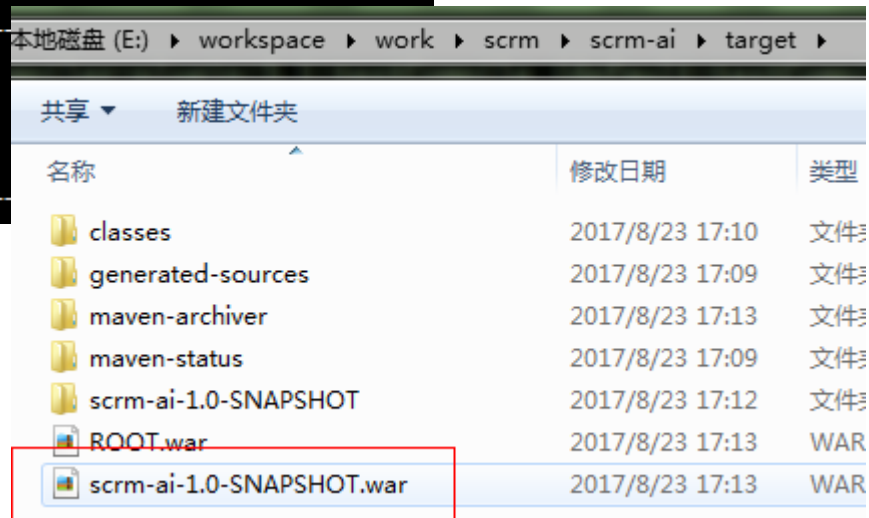
```
E:\workspace\work\scrm\scrm-authorize>mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ scrm-ai ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 3 resources
[INFO] Copying 2 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ scrm-ai ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to E:\workspace\work\scrm\scrm-ai\target\classes
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 13.014s
[INFO] Finished at: Wed Aug 23 17:09:58 CST 2017
[INFO] Final Memory: 42M/345M
[INFO] -----
```

五、Maven的安装配置 5-4

6、Maven命令：mvn package -Dmaven.test.skip=true

```
E:\workspace\work\scrm\scrm-ai>mvn package -Dmaven.test.skip=true
[INFO] Scanning for projects...
[INFO] Webapp assembled in [23858 msecs]
[INFO] Building war: E:\workspace\work\scrm\scrm-ai\target\ROOT.war
[INFO]
[INFO] --- spring-boot-maven-plugin:1.4.4.RELEASE:repackage (default) @ scrm-ai ---
[INFO] Replacing main artifact E:\workspace\work\scrm\scrm-ai\target\ROOT.war to
E:\workspace\work\scrm\scrm-ai\target\scrm-ai-1.0-SNAPSHOT.war
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 45.031s
[INFO] Finished at: Wed Aug 23 17:13:18 CST 2017
[INFO] Final Memory: 43M/357M
[INFO] -----
```

7、target目录下创建的war/jar包：



本地磁盘 (E:) > workspace > work > scrm > scrm-ai > target >		
共享 ▾ 新建文件夹		
名称	修改日期	类型
classes	2017/8/23 17:10	文件夹
generated-sources	2017/8/23 17:09	文件夹
maven-archiver	2017/8/23 17:13	文件夹
maven-status	2017/8/23 17:09	文件夹
scrm-ai-1.0-SNAPSHOT	2017/8/23 17:12	文件夹
ROOT.war	2017/8/23 17:13	WAR
scrm-ai-1.0-SNAPSHOT.war	2017/8/23 17:13	WAR

总结



提问

Maven是一个项目管理工具，是一系列构建工具的集合

Maven关键的几个知识点

Maven几个重要过程的命令

pom.xml是Maven项目的核心配置文件

Maven安装配置及M2Eclipse插件的安装

上机练习



需求说明

- 安装配置Maven环境，并使用Maven创建Java Web项目，实践compile编译，test测试，package打包，install安装，deploy发布这几个重要过程的命令。

完成时间：10分钟

共性问题集中讲解

Thank you

Q&A

