

# Maven开发笔记

---

## 学习内容

---

### 1.Maven概述

---

### 2.Maven基本使用

---

### 3.Maven基本命令

---

### 4.Maven与开发工具的集成

---

### 5.聚合工程和父子工程

---

### 6.Maven私服

---

## 一. Maven概述

---

### 1. 什么是Maven

---

- Maven是一个项目管理工具，主要用来做项目构建
- 项目构建的概念
  - 项目有生命周期：可行性分析-->需求分析-->概要设计-->详细设计-->编码-->测试-->部署-->试运行
  - 一组标准工作集合
- 基于**项目对象模型**（ POM Project Object Model ）
  - Maven可以通过一小段描述信息（ pom.xml ）来管理项目的构建
- 基于项目依赖管理系统（ Dependency Management System ）
  - 自动导包
- Maven 除了以程序构建能力为特色之外，还提供高级项目管理工具。由于 Maven 的缺省构建规则有较高的可重用性，所以常常用两三行 Maven 构建脚本就可以构建简单的项目。由于 Maven 的面向项目的方法，许多 Apache Jakarta 项目发文时使用 Maven，而且公司项目采用 Maven 的比例在持续增长。
- Maven这个单词来自于意第绪语（犹太语），意为知识的积累，最初在Jakarta Turbine项目中用来简化构建过程。当时有一些项目（有各自Ant build文件），仅有细微的差别，而JAR文件都由[CVS](#)来维护。于是希望有一种标准化的方式构建项目，一个清晰的方式定义项目的组成，一个容易的方式发布项目的信息，以及一种简单的方式在多个项目中共享JARs

### 2. Maven的概述

---

#### 2.1 官网

- <https://maven.apache.org/>

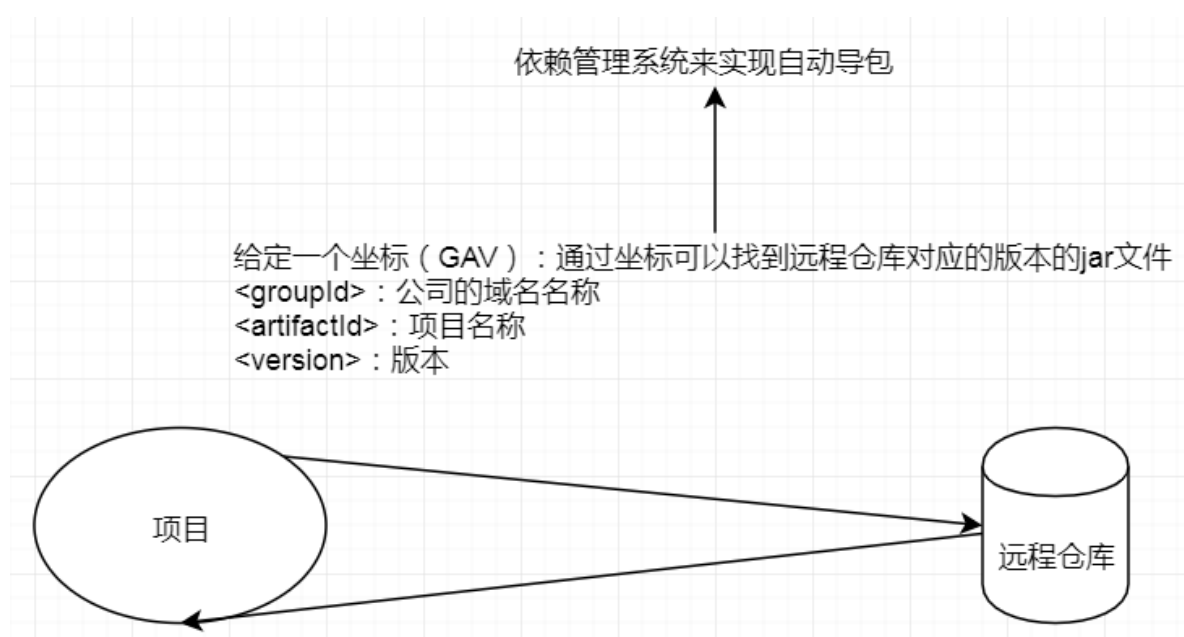
- 1 Maven, a [\[Yiddish word\]\(https://en.wikipedia.org/wiki/Maven\)](https://en.wikipedia.org/wiki/Maven) meaning *\*accumulator of knowledge\**, began as an attempt to simplify the build processes in the Jakarta Turbine project. There were several projects, each with their own Ant build files, that were all slightly different. JARs were checked into CVS. We wanted a standard way to build the projects, a clear definition of what the project consisted of, an easy way to publish project information and a way to share JARs across several projects.
- 2 The result is a tool that can now be used for building and managing any Java-based project. We hope that we have created something that will make the day-to-day work of Java developers easier and generally help with the comprehension of any Java-based project.

## 2.2 Maven目标

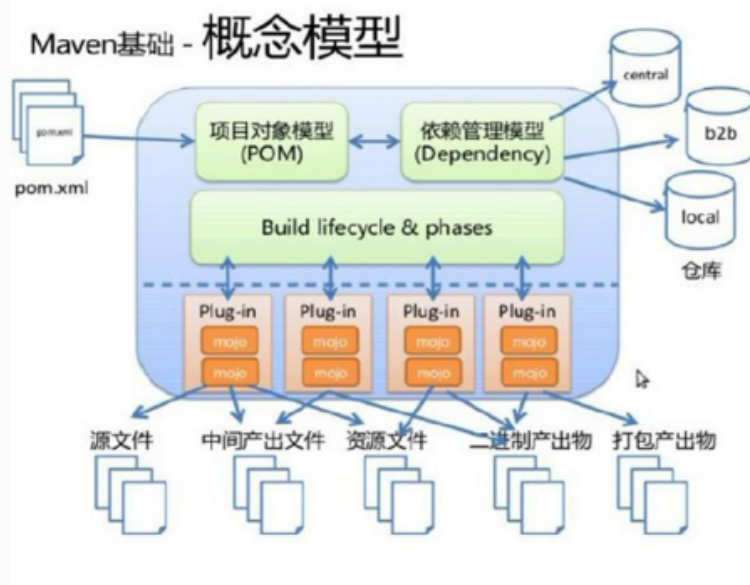
- 1 Maven's primary goal is to allow a developer to comprehend the complete state of a development effort in the shortest period of time. In order to attain this goal, there are several areas of concern that Maven attempts to deal with:
- 2
- 3 Making the build process easy
- 4 Providing a uniform build system
- 5 Providing quality project information
- 6 Providing guidelines for best practices development
- 7 Allowing transparent migration to new features

- 自动导入第三方jar包，能够解决包的版本的冲突
- 自动编译、测试、打包、安装
- 提供了大量的插件
- 父子工程和聚合工程符合当今分布式和微服务开发的架构思想

## 3. Maven基本工作原理



## 4. Maven体系结构



- POM (项目对象模型) : 描述了坐标以及插件...
- Dependency Management System :
- Build Lifecycle
- Plug-in
- 生成的文件以及报表等

## 二. Maven的基本使用

### 2.1 Maven的安装

#### 2.1.1 下载

- <https://maven.apache.org/download.cgi>

#### 2.1.2 安装

- 解压下载的文件



- 配置环境变量 (重点)
  - 配置MAVEN\_HOME : D:\Program Files\apache-maven-3.5.4
  - 配置PATH : D:\Program Files\apache-maven-3.5.4\bin
- 测试环境变量是否配置成功

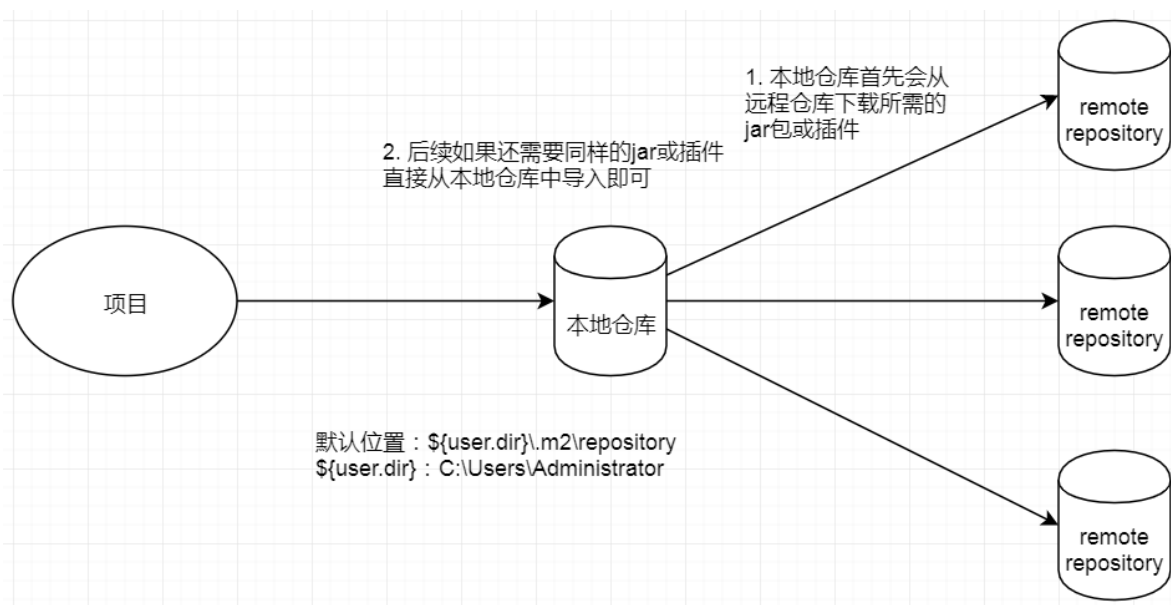
```
选定 管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>mvn -v
Apache Maven 3.5.4 (1edded0938998edf8bf061f1ceb3cfdeccf443fe; 2018-06-18T02:33:14+08:00)
Maven home: D:\Program Files\apache-maven-3.5.4\bin\..
Java version: 1.8.0_45, vendor: Oracle Corporation, runtime: D:\Program Files\Java\jdk1.8.0_45\jre
Default locale: zh_CN, platform encoding: GBK
OS name: "windows 7", version: "6.1", arch: "amd64", family: "windows"

C:\Users\Administrator>
```

## 2.2 Maven的仓库

- 本地仓库
  - 用来存储从远程仓库或中央仓库下载的jar文件
  - 项目优先查找本地仓库
- 远程仓库 ( remote repository )
  - 如果本地仓库没有jar或插件，那么就从远程仓库下载
  - 远程仓库可以是在同一个局域网内也可以是外网



- 中央仓库 ( central repository )
  - Maven库
  - 服务于整个互联网，由Maven团队成员来管理
  - <https://repo1.maven.org/maven2>
  - ...
- 配置本地库 ( 重点 )
  - 将原有的repository.rar文件解压
  - 找到D:\Program Files\apache-maven-3.5.4\conf下的文件settings.xml文件
  - 修改settings.xml文件

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache.org/xsd/maven-1.0.0.xsd">
  <!-- localRepository
    | The path to the local repository maven will use to store artifacts
    |
    | Default: ${user.home}/.m2/repository
  -->
  <localRepository>D:\Maven\repository</localRepository>
```

将此处修改成本地仓库的路径

- 添加一个阿里云镜像（加速）

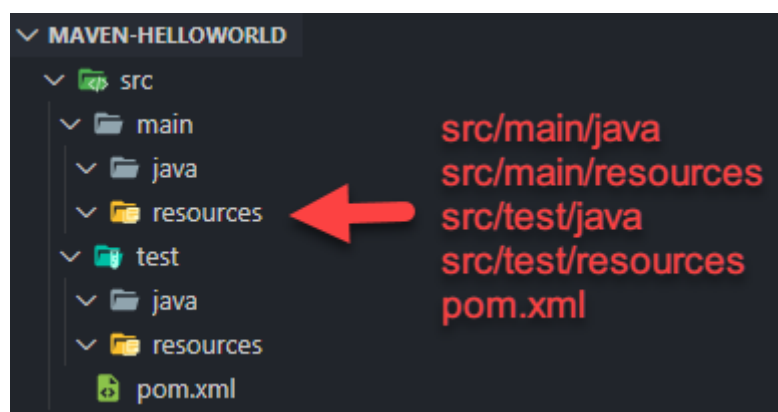
```
1 <repository>
2   <id>alimaven</id>
3   <name>aliyun maven</name>
4   <url>http://maven.aliyun.com/nexus/content/groups/public/</url>
5 </repository>
```

- 全局配置和用户配置

- 全局配置：所有的项目都使用settings.xml文件中指定的仓库
- 用户配置：仅仅这个项目使用这个本地库，其他项目不一定使用
  - D:\Program Files\apache-maven-3.5.4\conf\settings.xml文件拷贝一份到 \${user.dir}/.m2/下
  - 再进行上述的修改
- Maven首先找用户配置，如果找到则使用，如果没有找到则使用全局配置

## 三. Maven基本命令

### 3.1 Maven项目的基本结构



### 3.2 编写pom.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
5     http://maven.apache.org/xsd/maven-4.0.0.xsd">
6   <modelVersion>4.0.0</modelVersion>
```

```
7     <groupId>com.bjlemon</groupId>
8     <artifactId>maven-helloworld</artifactId>
9     <version>1.0-SNAPSHOT</version>
10
11     <dependencies>
12         <dependency>
13             <groupId>junit</groupId>
14             <artifactId>junit</artifactId>
15             <version>4.12</version>
16             <scope>test</scope>
17         </dependency>
18     </dependencies>
19 </project>
```

## 3.3 编写程序

```
1 package com.bjlemon.maven;
2
3 public class Helloworld {
4
5     public String sayHello(String content) {
6         return "Helloworld " + content;
7     }
8 }
```

## 3.4 编写测试

```
1 package com.bjlemon.maven.test;
2
3 import com.bjlemon.maven.Helloworld;
4 import org.junit.*;
5
6 public class HelloworldTest {
7
8     @Test
9     public void testSayHello() {
10         Helloworld helloworld = new Helloworld();
11         String result = helloworld.sayHello("Maven");
12         System.out.println(result);
13     }
14 }
```

## 3.5 Maven命令构建项目

### 3.5.1 编译命令

```
1 mvn compile
```

```
选定 管理员: C:\Windows\System32\cmd.exe
3 个目录 47,426,093,056 可用字节

F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld>mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.bjlemon:maven-helloworld >-----
[INFO] Building maven-helloworld 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ maven-hello
oworld ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e
. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ maven-helloworl
d ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding GBK, i.e. buil
d is platform dependent!
[INFO] Compiling 1 source file to F:\Course Workspace\bjlemon\Workspace15\day1\c
ode\maven-helloworld\target\classes
[INFO]
[INFO] BUILD SUCCESS
半:
```

- 成功后会在工程下生成一个target文件夹

### 3.5.2 清除命令

```
1 | mvn clean
```

```
PS F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld> mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.bjlemon:maven-helloworld >-----
[INFO] Building maven-helloworld 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ maven-helloworld ---
[INFO] Deleting F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld\target
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 0.845 s
[INFO] Finished at: 2020-02-27T22:13:19+08:00
[INFO]
PS F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld>
```

### 3.5.3 单元测试命令

```
1 | mvn test
```

```

[INFO] Building maven-helloworld 1.0-SNAPSHOT
PS F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld> mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.bjlemon:maven-helloworld >-----
[INFO] Building maven-helloworld 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ maven-helloworld ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ maven-helloworld ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding GBK, i.e. build is platform dependent!
[INFO] Compiling 1 source file to F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld\target\classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ maven-helloworld ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ maven-helloworld ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding GBK, i.e. build is platform dependent!
[INFO] Compiling 1 source file to F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld\target\test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ maven-helloworld ---
[INFO] Surefire report directory: F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld\target\surefire-reports

-----
T E S T S
-----
Running com.bjlemon.maven.test.HelloworldTest
Helloworld Maven

```

## 3.5.4 打包命令

1 | mvn package

```

PS F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld> mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.bjlemon:maven-helloworld >-----
[INFO] Building maven-helloworld 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ maven-helloworld ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ maven-helloworld ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ maven-helloworld ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ maven-helloworld ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ maven-helloworld ---
[INFO] Surefire report directory: F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld\target\surefire-reports

-----
T E S T S
-----
Running com.bjlemon.maven.test.HelloworldTest
Helloworld Maven
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.146 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ maven-helloworld ---
[INFO] Building jar: F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld\target\maven-helloworld-1.0-SNAPSHOT.jar
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 5.344 s
[INFO] Finished at: 2020-02-27T22:16:58+08:00
[INFO]
PS F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld>

```

## 3.5.5 安装命令

1 | mvn install



```

PS F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld> mvn install
[INFO] Scanning for projects...
[INFO]
[INFO] < com.bjlemon:maven-helloworld >-----
[INFO] Building maven-helloworld 1.0-SNAPSHOT
[INFO] [ jar ]-----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ maven-helloworld ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ maven-helloworld ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ maven-helloworld ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ maven-helloworld ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ maven-helloworld ---
[INFO] Surefire report directory: F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld\target\surefire-reports

-----
T E S T S
-----
Running com.bjlemon.maven.test.HelloworldTest
Helloworld Maven
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.147 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ maven-helloworld ---
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ maven-helloworld ---
[INFO] Installing F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld\target\maven-helloworld-1.0-SNAPSHOT.jar to D:\Maven\repository\com\bjlemon\maven-helloworld\1.0-SNAPSHOT\maven-helloworld-1.0-SNAPSHOT.jar
[INFO] Installing F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld\pom.xml to D:\Maven\repository\com\bjlemon\maven-helloworld\1.0-SNAPSHOT\maven-helloworld-1.0-SNAPSHOT.pom
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 6.289 s
[INFO] Finished at: 2020-02-27T22:18:29+08:00
[INFO]
PS F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld>

```

## 3.5.6 组合命令

1 | mvn clean install

```

PS F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld> mvn clean install
[INFO] Scanning for projects...
[INFO]
[INFO] < com.bjlemon:maven-helloworld >-----
[INFO] Building maven-helloworld 1.0-SNAPSHOT
[INFO] [ jar ]-----
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ maven-helloworld ---
[INFO] Deleting F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld\target
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ maven-helloworld ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ maven-helloworld ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding GBK, i.e. build is platform dependent!
[INFO] Compiling 1 source file to F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld\target\classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ maven-helloworld ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ maven-helloworld ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding GBK, i.e. build is platform dependent!
[INFO] Compiling 1 source file to F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld\target\test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ maven-helloworld ---
[INFO] Surefire report directory: F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld\target\surefire-reports

-----
T E S T S
-----
Running com.bjlemon.maven.test.HelloworldTest
Helloworld Maven
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.136 sec

Results :

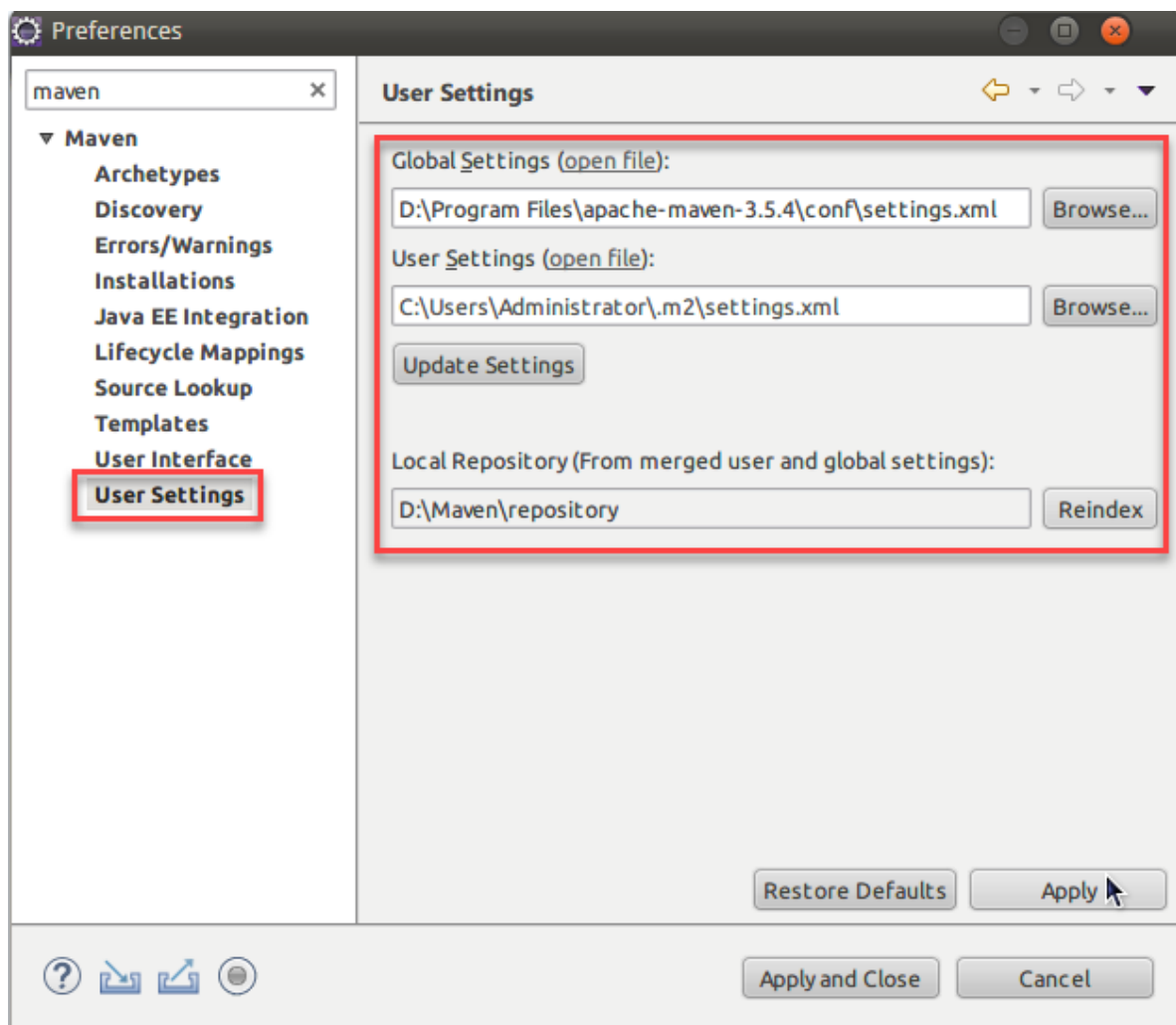
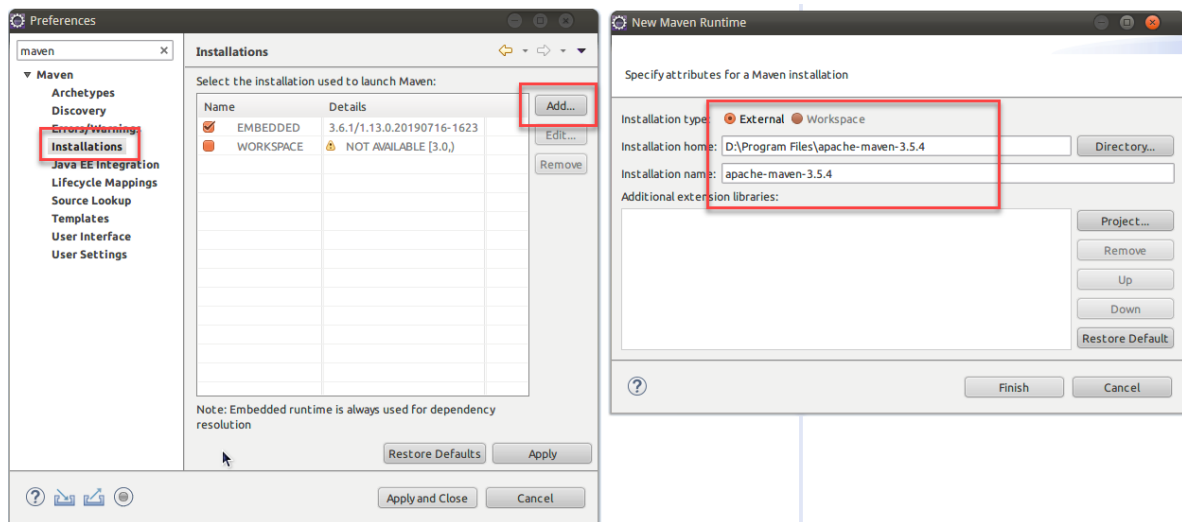
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ maven-helloworld ---
[INFO] Building jar: F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld\target\maven-helloworld-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ maven-helloworld ---
[INFO] Installing F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld\target\maven-helloworld-1.0-SNAPSHOT.jar to D:\Maven\repository\com\bjlemon\maven-helloworld\1.0-SNAPSHOT\maven-helloworld-1.0-SNAPSHOT.jar
[INFO] Installing F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld\pom.xml to D:\Maven\repository\com\bjlemon\maven-helloworld\1.0-SNAPSHOT\maven-helloworld-1.0-SNAPSHOT.pom
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 6.289 s
[INFO] Finished at: 2020-02-27T22:18:29+08:00
[INFO]
PS F:\Course Workspace\bjlemon\Workspace15\day1\code\maven-helloworld>

```

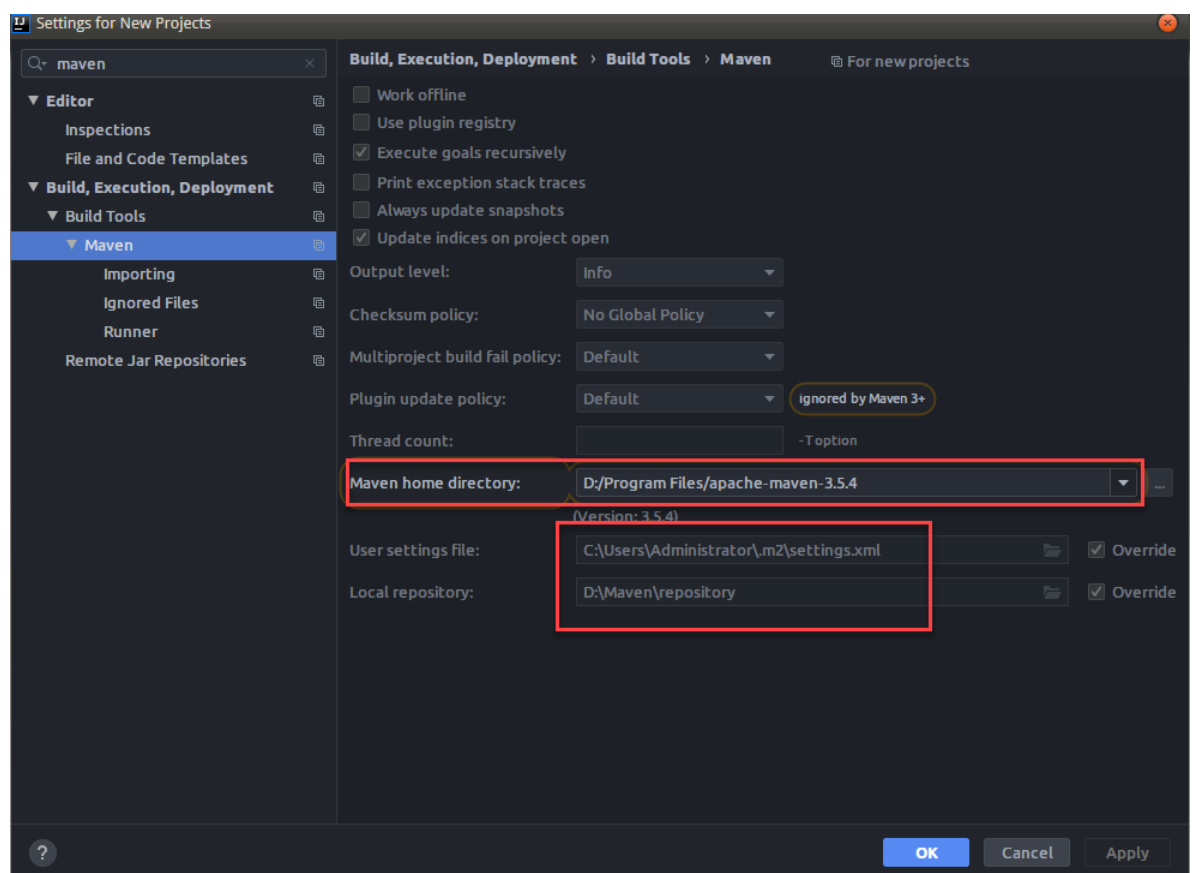
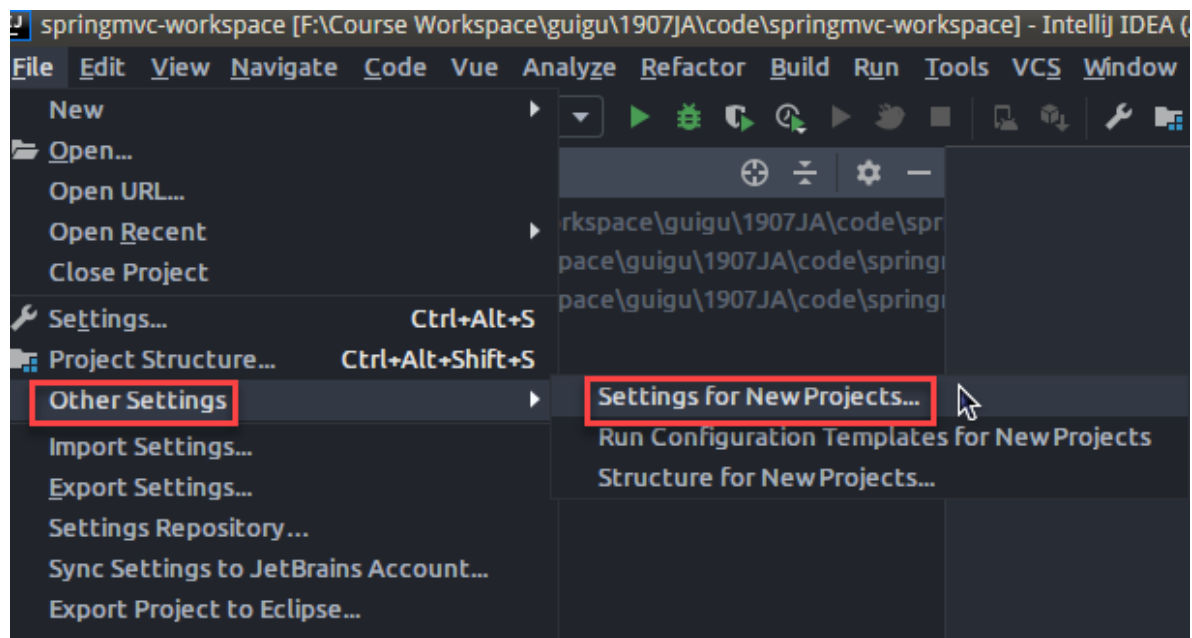
# 四.Maven与开发工具的集成

## 4.1 Maven与Eclipse集成（了解）

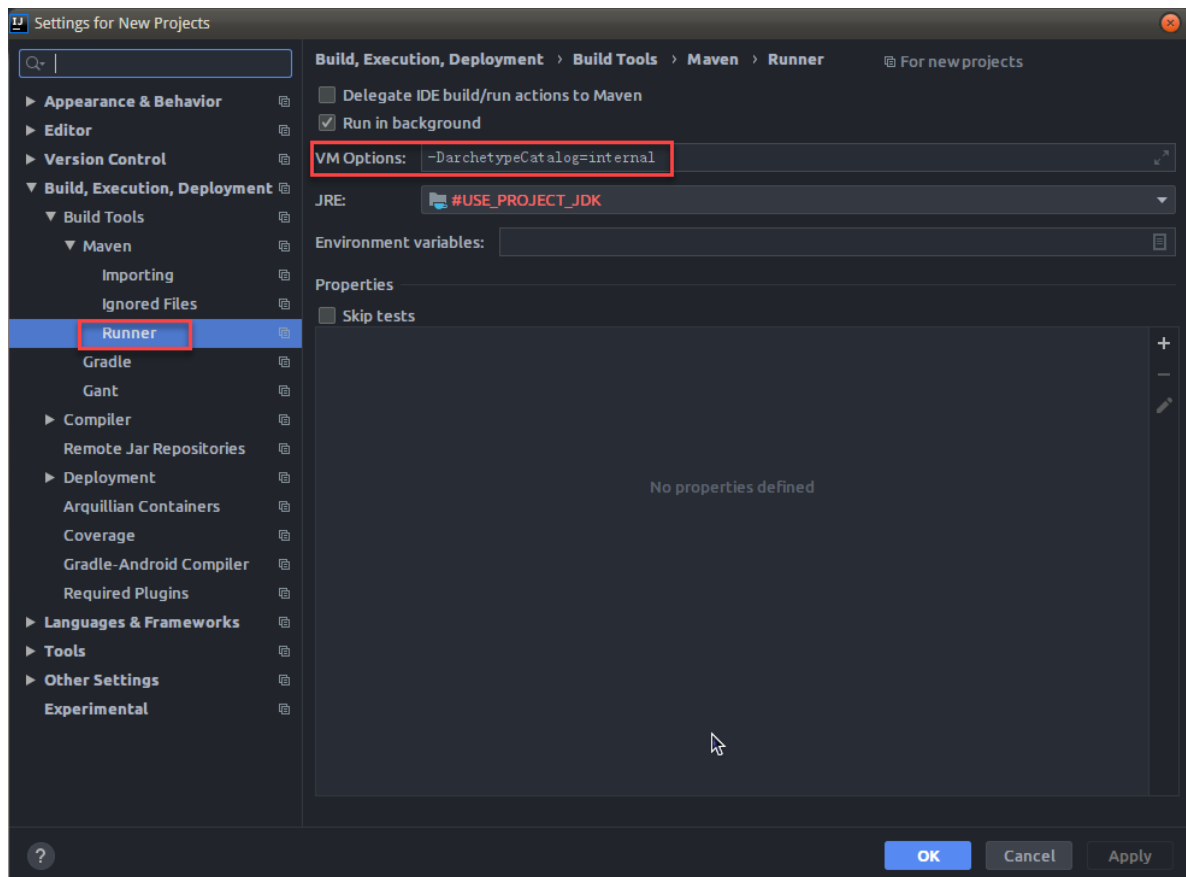


## 4.2 Maven与IDEA集成

### 4.2.1 设置默认配置



- 设置VM options ( -DarchetypeCatalog=internal )

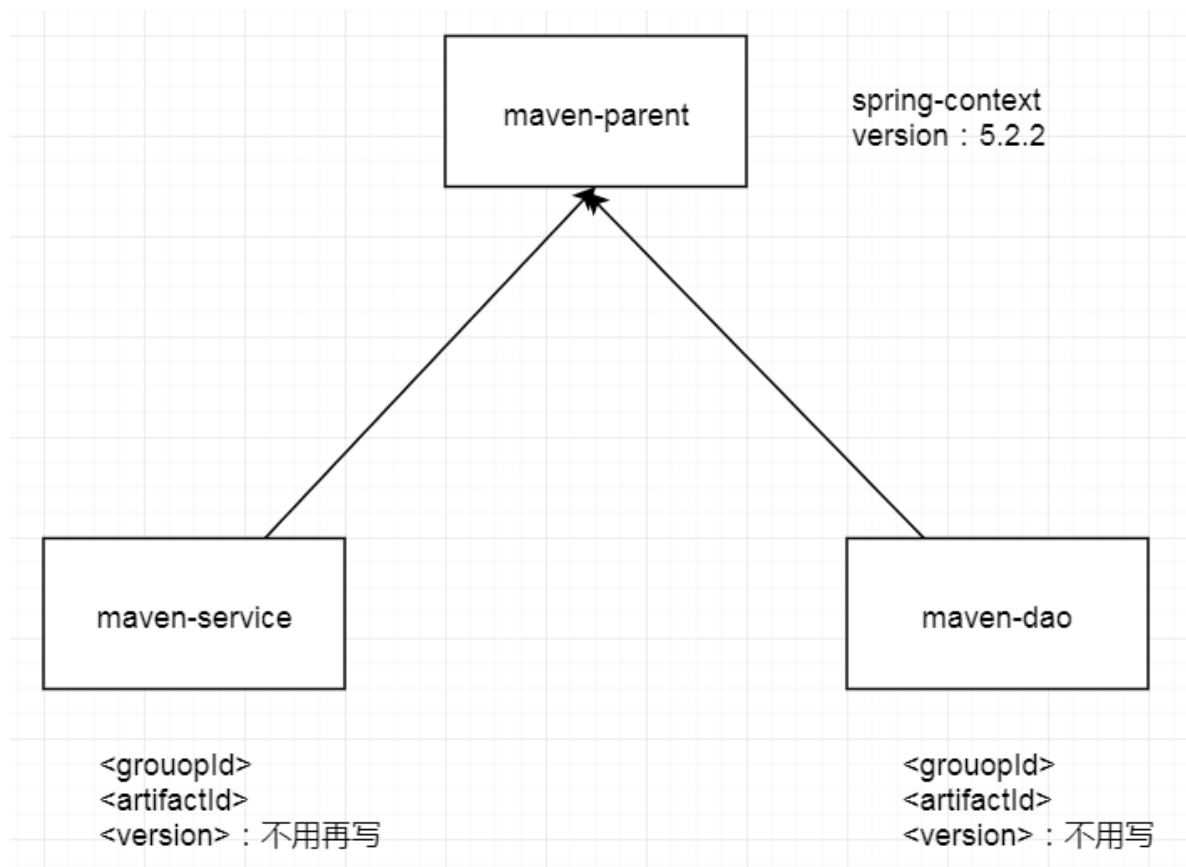


- 装上插件Maven Helper

## 五. 父子工程和聚合工程（重点 SSM）

### 5.1 父子工程

- 项目中包很多，版本很乱，那么开发时很容易出现包的版本冲突问题
- Maven有种机制，能够将包的版本进行统一管理。父子工程就是用来进行包的版本控制。
- **父子工程必须满足打包方式为pom**
- 父子工程将项目中所用到的所有的包进行了统一的版本管理，那么子工程继承父工程，当子工程用到哪个包还是需要将坐标显示写出，但是就不用再写版本。
- 父工程不需要编写任何java代码，因此可以将src文件夹删掉



## 5.2 聚合工程

- 项目的拆分，各自完成工程的功能，但是最终需要将各个工程整合在一起才能运行
- mall-dao(jar), mall-service(jar), mall-web(war)...
- 项目的整合需要将上述三个工程整合在一起（Maven有这个聚合的能力）
- 每一个工程的配置文件最终进行整合，能够共享
- 聚合工程必须是父子工程，父子工程不一定是聚合工程。

## 六. Maven私服

## 七. 将第三方的jar文件导入到本地库或私服

