

Maven中文

- [博客](#)
 - [微博](#)
 - [相册](#)
 - [收藏](#)
 - [留言](#)
 - [关于我](#)

Maven最佳实践：Maven仓库

- 博客分类：

 - [Maven](#)

[maven配置管理](#)
[项目管理XML](#)
[junit](#)

什么是Maven仓库

在不用Maven的时候，比如说以前我们用Ant构建项目，在项目目录下，往往会看到一个名为/lib的子目录，那里存放着各类第三方依赖jar文件，如log4j.jar，junit.jar等等。每建立一个项目，你都需要建立这样的一个/lib目录，然后复制一对jar文件，这是很明显的重复。重复永远是噩梦的起点，多个项目不共用相同的jar文件，不仅会造成磁盘资源的浪费，也使得版本的一致性管理变得困难。此外，如果你使用版本管理工具，如SVN（你没有使用版本管理工具？马上试试SVN吧，它能帮你解决很多头疼的问题），你需要将大量的jar文件提交到代码库里，可是版本管理工具在处理二进制文件方面并不出色。

Maven仓库就是放置所有JAR文件（WAR，ZIP，POM等等）的地方，所有Maven项目可以从同一个Maven仓库中获取自己所需要的依赖JAR，这节省了磁盘资源。此外，由于Maven仓库中所有的JAR都有其自己的坐标，该坐标告诉Maven它的组ID，构件ID，版本，打包方式等等，因此Maven项目可以方便的进行依赖版本管理。你也不在需要提交JAR文件到SCM仓库中，你可以建立一个组织层次的Maven仓库，供所有成员使用。

简言之，Maven仓库能帮助我们管理构件（主要是JAR）。


本地仓库 vs. 远程仓库

运行Maven的时候，Maven所需要的任何构件都是直接从本地仓库获取的。如果本地仓库没有，它会首先尝试从远程仓库下载构件至本地仓库，然后再使用本地仓库的构件。

比如说，你的项目配置了junit-3.8的依赖，在你运行**mvn test**的时候，Maven需要使用junit-3.8的jar文件，它首先根据坐标查找本地仓库，如果找到，就直接使用。如果没有，Maven会检查可用的远程仓库配置，然后逐个尝试这些远程仓库去下载junit-3.8的jar文件，如果远程仓库存在该文件，Maven会将其下载到本地仓库中，继而使用。如果尝试过所有远程仓库之后，Maven还是没能够下载到该文件，它就会报错。

Maven缺省的本地仓库地址为 *`${user.home}/.m2/repository`*。也就是说，一个用户会对应的拥有一个本地仓库。

你也可以自定义本地仓库的位置，修改 `${user.home}/.m2/settings.xml`：

Xml代码 

```
1. <settings>
2. ...
3. <localRepository>D:\java\repository</localRepository>
4. ...
5. </settings>
```

你还可以在运行时指定本地仓库位置：


```
mvn clean install -Dmaven.repo.local=/home/juven/myrepo/
```

还有一点需要理解的是，当我们运行install的时候，Maven实际上是将项目生成的构件安装到了本地仓库，也就是说，只有install了之后，其它项目才能使用此项目生成的构件。

了解了本地仓库，接着了解一下Maven缺省的远程仓库，即Maven中央仓库。

安装好Maven之后，我们可以建立一个简单的项目，配置一些简单的依赖，然后运行mvn clean install，项目就构建好了。我们没有手工的去下载任何jar文件，这一切都是因为Maven中央仓库的存在，当Maven在本地仓库找不到需要的jar文件时，它会查找远程仓库，而一个原始的Maven安装就自带了一个远程仓库——Maven中央仓库。

这个Maven中央仓库是在哪里定义的呢？在我的机器上，我安装了maven-2.0.10，我可以找到这个文件：`${M2_HOME}/lib/maven-2.0.10-uber.jar`，打开该文件，能找到超级POM：`\org\apache\maven\project\pom-4.0.0.xml`，它是所有Maven POM的父POM，所有Maven项目继承该配置，你可以在这个POM中发现如下配置：

Xml代码 


```
1. <repositories>
2. <repository>
3.   <id>central</id>
4.   <name>Maven Repository Switchboard</name>
5.   <layout>default</layout>
6.   <url>http://repo1.maven.org/maven2</url>
7.   <snapshots>
8.     <enabled>>false</enabled>
9.   </snapshots>
10. </repository>
11. </repositories>
```

关于远程仓库的配置，下面的小节我会详细解释，这里我们只要知道，中央仓库的id为central，远程url地址为http://repo1.maven.org/maven2，它关闭了snapshot版本构件下载的支持。

在POM中配置远程仓库

前面我们看到超级POM配置了ID为central的远程仓库，我们可以在POM中配置其它的远程仓库。这样做的原因有很多，比如你有一个局域网的远程仓库，使用该仓库能大大提高下载速度，继而提高构建速度，也有可能你依赖的一个jar在central中找不到，它只存在于某个特定的公共仓库，这样你也不得不添加那个远程仓库的配置。

这里我配置一个远程仓库指向中央仓库的中国镜像：

Xml代码 

```
1. <project>
2. ...
```

```
3. <repositories>
4.   <repository>
5.     <id>maven-net-cn</id>
6.     <name>Maven China Mirror</name>
7.     <url>http://maven.net.cn/content/groups/public/</url>
8.     <releases>
9.       <enabled>>true</enabled>
10.    </releases>
11.    <snapshots>
12.      <enabled>>false</enabled>
13.    </snapshots>
14.  </repository>
15.</repositories>
16.<pluginRepositories>
17.  <pluginRepository>
18.    <id>maven-net-cn</id>
19.    <name>Maven China Mirror</name>
20.    <url>http://maven.net.cn/content/groups/public/</url>
21.    <releases>
22.      <enabled>>true</enabled>
23.    </releases>
24.    <snapshots>
25.      <enabled>>false</enabled>
26.    </snapshots>
27.  </pluginRepository>
28.</pluginRepositories>
29. ...
30.</project>
```

我们先看一下<repositories>的配置，你可以在它下面添加多个<repository>，每个<repository>都有它唯一的ID，一个描述性的name，以及最重要的，远程仓库的url。此外，<releases><enabled>true</enabled></releases>告诉Maven可以从这个仓库下载releases版本的构件，而<snapshots><enabled>>false</enabled></snapshots>告诉Maven不要从这个仓库下载snapshot版本的构件。禁止从公共仓库下载snapshot构件是推荐的做法，因为这些构件不稳定，且不受你控制，你应该避免使用。当然，如果你想使用局域网内组织内部的仓库，你可以激活snapshot的支持。

关于<repositories>的更详细的配置及相关解释，请参考：http://www.sonatype.com/books/maven-book/reference_zh/apas02s08.html。


至于<pluginRepositories>，这是配置Maven从什么地方下载插件构件（Maven的所有实际行为都由其插件完成）。该元素的内部配置和<repository>完全一样，不再解释。

在settings.xml中配置远程仓库

我们知道了如何在POM中配置远程仓库，但考虑这样的情况：在一个公司内部，同时进行这3个项目，而且以后随着这几个项目的结束，越来越多的项目会开始；同时，公司内部建立一个Maven仓库。我们统一为所有这些项目配置该仓库，于是不得不为每个项目提供同样的配置。问题出现了，这是**重复**！

其实我们可以做到只配置一次，在哪里配置呢？就是settings.xml。

不过事情没有那么简单，不是简单的将POM中的<repositories>及<pluginRepositories>元素复制到settings.xml中就可以，setting.xml *不直接支持* 这两个元素。但我们还是有一个并不复杂的解决方案，就是利用profile，如下：

Xml代码 

```
1. <settings>
2. ...
3. <profiles>
4.   <profile>
5.     <id>dev</id>
6.     <!-- repositories and pluginRepositories here-->
7.   </profile>
8. </profiles>
9. <activeProfiles>
10.   <activeProfile>dev</activeProfile>
11. </activeProfiles>
12. ...
13. </settings>
```


这里我们定义一个id为dev的profile，将所有repositories以及pluginRepositories元素放到这个profile中，然后，使用<activeProfiles>元素自动激活该profile。这样，你就不用再为每个POM重复配置仓库。

使用profile为settings.xml添加仓库提供了一种用户全局范围的仓库配置。

镜像


如果你的地理位置附近有一个速度更快的central镜像，或者你想覆盖central仓库配置，或者你想为所有POM使用唯一的一个远程仓库（这个远程仓库代理的所有必要的其它仓库），你可以使用settings.xml中的mirror配置。

以下的mirror配置用maven.net.cn覆盖了Maven自带的central：

Xml代码 

```
1. <settings>
2. ...
3. <mirrors>
4.   <mirror>
5.     <id>maven-net-cn</id>
6.     <name>Maven China Mirror</name>
7.     <url>http://maven.net.cn/content/groups/public/</url>
8.     <mirrorOf>central</mirrorOf>
9.   </mirror>
10. </mirrors>
11. ...
12. </settings>
```

这里唯一需要解释的是<mirrorOf>，这里我们配置central的镜像，我们也可以配置一个所有仓库的镜像，以保证该镜像Maven唯一使用的仓库：

Xml代码 

```
1. <settings>
2. ...
3. <mirrors>
```


```
4.   <mirror>
5.     <id>my-org-repo</id>
6.     <name>Repository in My Orgnization</name>
7.     <url>http://192.168.1.100/maven2</url>
8.     <mirrorOf>*</mirrorOf>
9.   </mirror>
10. </mirrors>
11. ...
12. </settings>
```

关于更加高级的镜像配置，可以参考：<http://maven.apache.org/guides/mini/guide-mirror-settings.html>。

分发构件至远程仓库

mvn install 会将项目生成的构件安装到本地Maven仓库，**mvn deploy** 用来将项目生成的构件分发到远程Maven仓库。本地Maven仓库的构件只能供当前用户使用，在分发到远程Maven仓库之后，所有能访问该仓库的用户都能使用你的构件。


我们需要配置POM的distributionManagement来指定Maven分发构件的位置，如下：

Xml代码 

```
1. <project>
2.   ...
3.   <distributionManagement>
4.     <repository>
5.       <id>nexus-releases</id>
6.       <name>Nexus Release Repository</name>
7.       <url>http://127.0.0.1:8080/nexus/content/repositories/releases/</url>
8.     </repository>
9.     <snapshotRepository>
10.      <id>nexus-snapshots</id>
11.      <name>Nexus Snapshot Repository</name>
12.      <url>http://127.0.0.1:8080/nexus/content/repositories/snapshots/</url>
13.    </snapshotRepository>
14.  </distributionManagement>
15.  ...
16. </project>
```

Maven区别对待release版本的构件和snapshot版本的构件，snapshot为开发过程中的版本，实时，但不稳定，release版本则比较稳定。Maven会根据你项目的版本来判断将构件分发到哪个仓库。

一般来说，分发构件到远程仓库需要认证，如果你没有配置任何认证信息，你往往会得到401错误。这个时候，如下在settings.xml中配置认证信息：

Xml代码 

```
1. <settings>
2.   ...
3.   <servers>
4.     <server>
5.       <id>nexus-releases</id>
```



```
6.   <username>admin</username>
7.   <password>admin123</password>
8. </server>
9. <server>
10.  <id>nexus-snapshots</id>
11.  <username>admin</username>
12.  <password>admin123</password>
13. </server>
14. </servers>
15. ...
16. </settings>
```

需要注意的是，settings.xml中server元素下id的值必须与POM中repository或snapshotRepository下id的值完全一致。将认证信息放到settings下而非POM中，是因为POM往往是它人可见的，而settings.xml是本地的。

小结

本文介绍了Maven仓库，它是什么？本地仓库，远程仓库，中央仓库具体是指什么？并介绍了如何在POM中配置项目层次的仓库，在settings中配置用户层次的仓库，以及mirror。本文还介绍了如何安装构件到本地仓库，如何分发构件至仓库。



65

顶

3

踩

分享到：

[Maven最佳实践：版本管理](#) | [Maven中央仓库的中国镜像](#)

- 2009-04-01 17:42
- 浏览 93583
- [评论\(19\)](#)
- [查看更多](#)

相关资源推荐

- [AtCommands相关学习资料 \(中文+英文\)](#)
- [Delphi7高级应用开发随书源码](#)
- [LTE AT指令](#)
- [封装了DAO对象用于直接操纵access数据库](#)
- [MAVEN使用最佳实践](#)
- [数据仓库与商务智能最佳实践](#)
- [maven3.2.5压缩包 \(本地仓库路径修改过的\)](#)
- [Maven本地仓库搭建工具nexus-2.11.4-01-bundle](#)

- [At Commands：从白痴到大师的修炼历程 \(一\)](#)
- [聊天机器人2002](#)
- [迅捷全站功能模块](#)
- [Maven版的JPA最佳实践](#)
- [大型数据仓库最佳实践](#)
- [linux下使用Nexus做Maven仓库管理](#)
- [maven项目仓库包](#)
- [Maven：仓库搜索服务](#)

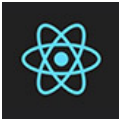
[eclipse json编辑器依赖的maven仓库](#)
[maven 连接阿里云仓库setting.xml配置](#)

[Maven本地仓库](#)
[本地maven仓库3](#)

参考知识库



[Android知识库](#) 38192 关注 / 3162 收录



[React知识库](#) 3708 关注 / 393 收录



[人工智能基础知识库](#) 17148 关注 / 212 收录



[Java 知识库](#) 36117 关注 / 3748 收录

评论

19 楼 [charis0721](#) 2016-02-03

zhuyanjavaeye 写道

请教一个问题，如果在settings.xml中配置了repository，在自己的pom.xml中也配置了repository，那么下载jar的时候，顺序是什么样的？先从settings.xml中指定的下载？没有了再到自己的pom.xml中指定的去下载？

如果settings.xml中有mirror，三者的顺序又是怎样的？

我的理解：如果没有mirror，优先项目pom中加载，没有再去settings.xml中加载；
如果有mirror，优先项目pom中加载，没有再去settings.xml中加载，再没有去mirror中加载。

但是我现在遇到个问题，感觉有mirror的时候优先mirror了，而且没有再用pom中指定的库，求解？？

18 楼 [zhuyanjavaeye](#) 2016-01-11

请教一个问题，如果在settings.xml中配置了repository，在自己的pom.xml中也配置了repository，那么下载jar的时候，顺序是什么样的？先从settings.xml中指定的下载？没有了再到自己的pom.xml中指定的去下载？

17 楼 [spring_springmvc](#) 2015-07-30

可以参考最新的文档：

[如何在eclipse jee中检出项目并转换为Maven project,最后转换为Dynamic web project](#)，地址：<http://www.zuidaima.com/blog/1618180875144192.htm>

[如何在eclipse jee中创建Maven project并且转换为Dynamic web project](#)

地址：<http://www.zuidaima.com/blog/1618162161323008.htm>

16 楼 [map612](#) 2014-07-23

已阅，有收获，多谢

15 楼 [永志_爱戴](#) 2014-02-28

好文章，果断收藏，期待楼主的下一篇

14 楼 [Allen_J_Will](#) 2014-01-09

Martin_Bonnie 写道

按照大牛说明 已经成功弄好 , 但是有一个问题, 我在项目中使用了

Java代码 ☆

```
1. <plugin>
2.     <groupId>org.apache.maven.plugins</groupId>
3.     <artifactId>maven-assembly-plugin</artifactId>
4.     <version>2.4</version>
5.     <executions>
6.         <execution>
7.             <id>make-client-jar</id>
8.             <phase>package</phase>
9.             <goals>
10.                <goal>single</goal>
11.            </goals>
12.            <configuration>
13.                <skipAssembly>>false</skipAssembly>
14.                <finalName>express-basicdata-service</finalName>
15.                <appendAssemblyId>>false</appendAssemblyId>
16.                <descriptors>
17.                    <descriptor>src/main/assembly/xingng-client-jar.xml</descriptor>
18.                </descriptors>
19.            </configuration>
20.        </execution>
21.    </executions>
22. </plugin>
```

maven-assembly-plugin的插件, 我想将这个插件打出来的jar文件自动发布到远程的snapshot库中, 配置的deployManage好像是当前工程的pom文件的。 不知道怎么弄了, 求指教 谢谢。

这个配置和构件发布到远程无关, 不要混淆了。

构件发布远程, 需要配置<distributionManagement>标签, 一般还要求授权

13 楼 [Martin_Bonnie](#) 2013-12-09

按照大牛说明 已经成功弄好 , 但是有一个问题, 我在项目中使用了

Java代码 ☆

```
1. <plugin>
2.     <groupId>org.apache.maven.plugins</groupId>
3.     <artifactId>maven-assembly-plugin</artifactId>
4.     <version>2.4</version>
5.     <executions>
6.         <execution>
7.             <id>make-client-jar</id>
8.             <phase>package</phase>
9.             <goals>
10.                <goal>single</goal>
11.            </goals>
12.            <configuration>
13.                <skipAssembly>>false</skipAssembly>
```



```
14.         <finalName>express-basicdata-service</finalName>
15.         <appendAssemblyId> false</appendAssemblyId>
16.         <descriptors>
17.             <descriptor>src/main/assembly/xingng-client-jar.xml</descriptor>
18.         </descriptors>
19.     </configuration>
20. </execution>
21. </executions>
22. </plugin>
```

maven-assembly-plugin的插件，我想将这个插件打出来的jar文件自动发布到远程的snapshot库中，配置的deployManage好像是当前工程的pom文件的。 不知道怎么弄了，求指教 谢谢。

12 楼 [semmy](#) 2013-10-22

按我的经验，这两个都要配置的，如果没有配置mirror，只是单单配置了一个profile，那么还是会从默认的：http://repo.maven.apache.org/maven2 中央仓库下载依赖包。

galen888 写道

请问想配置访问自己的私服仓库 mirror 和 <profiles> 都需要配置吗

例如：

```
<mirrors>
  <mirror>
    <id>my-org-repo</id>
    <name>Repository in My Orgnization</name>
    <url>http://192.168.1.100/maven2</url>
    <mirrorOf>*</mirrorOf>
  </mirror>
</mirrors>
```

```
<profiles>
  <profile>
    <id>dev</id>
    <!-- repositories and pluginRepositories here-->
```

省略自己的IP

```
</profile>
</profiles>
<activeProfiles>
  <activeProfile>dev</activeProfile>
</activeProfiles>
```

11 楼 [galen888](#) 2013-10-10

请问想配置访问自己的私服仓库 mirror 和 <profiles> 都需要配置吗

例如：

```
<mirrors>
```

```
<mirror>
  <id>my-org-repo</id>
  <name>Repository in My Orgnization</name>
  <url>http://192.168.1.100/maven2</url>
  <mirrorOf>*</mirrorOf>
</mirror>
</mirrors>
```

```
<profiles>
  <profile>
    <id>dev</id>
    <!-- repositories and pluginRepositories here-->

    省略自己的IP
  </profile>
</profiles>
<activeProfiles>
  <activeProfile>dev</activeProfile>
</activeProfiles>
```

10 楼 [b_l_east](#) 2013-07-04

GOOD👍

9 楼 [succeedloveaaaa](#) 2013-03-05

😊 在网上找了很多，只有你的我看懂了，谢谢博主

8 楼 [eksliang](#) 2012-09-10

真的讲的很好。

7 楼 [zhubinhua](#) 2012-05-12

请问一下：

Snapshots 这个Repositories如何上传一个maven中央仓库没有的jar包呀？？

6 楼 [yangpeihai](#) 2011-12-28

刚刚学习maven，楼主博客给了我非常大的帮帮助，非常谢谢👍

5 楼 [LXJ_hmxc](#) 2010-01-25

讲的挺详细和挺实用的

👍👍👍

4 楼 [juvenshun](#) 2009-08-06

q3691223 写道

问下如何在pom.xml里面开启snapshot的jar包下载？给个例子指导下

```
# <repositories>
#   <repository>
#     <id>maven-net-cn</id>
#     <name>Maven China Mirror</name>
#     <url>http://maven.net.cn/content/groups/public/</url>
```

```
# <releases>
#   <enabled>true</enabled>
# </releases>
# <snapshots>
#   <enabled>true</enabled>
# </snapshots>
# </repository>
# </repositories>
```

3 楼 [q3691223](#) 2009-08-05

问下如何在pom.xml里面开启snapshot的jar包下载？给个例子指导下

2 楼 [juvenshun](#) 2009-04-06

hxirui 写道

引用其实我们可以做到只配置一次，在哪里配置呢？就是settings.xml。
这句没看明白，所谓只配置一次指的是什么？引用settings.xml是哪个地方的？
请指教。

全局范围:M2_HOME/conf/settings.xml(安装自带这个文件)
用户范围:USER_HOME/.m2/settings.xml（该文件默认不存在，可复制全局范围settings.xml至此）
“只配置一次”是指“只需要在settings.xml中进行一次仓库配置，那么只要是在该settings.xml的影响范围内，所有的项目（POM）就都能自动使用该仓库配置”

1 楼 [hxirui](#) 2009-04-05

引用

其实我们可以做到只配置一次，在哪里配置呢？就是settings.xml。

这句没看明白，所谓只配置一次指的是什么？

引用

settings.xml

是哪个地方的？

请指教。

发表评论



[您还没有登录,请您登录后再发表评论](#)



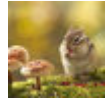
juvenshun

- 浏览: 1330056 次
- 性别:
- 来自: 苏州
- 我现在离线

最近访客 [更多访客>>](#)



[张洪财](#)



[itace](#)



[爱吃小面](#)



[regsaveme](#)

文章分类

- [全部博客 \(56\)](#)
- [Maven \(61\)](#)

社区版块

- [我的资讯](#) (5)
- [我的论坛](#) (15)
- [我的问答](#) (2)

存档分类

- [2010-11](#) (1)
- [2010-04](#) (1)
- [2010-03](#) (1)
- [更多存档...](#)

最新评论

- [远方_张涛](#)：赞赞赞,生动有趣还说明问题
[Maven最佳实践：版本管理](#)
- [smartdog](#)：新的网站访问不了啊~~博主快更新
[再见JavaEye](#)
- [smartdog](#)：言简意赅，厉害
[Maven最佳实践：遵循约定](#)
- [liuyuan_jl](#)：这篇文章，真心的不错，受益匪浅
[Maven最佳实践：划分模块](#)
- [u010835702](#)：谢谢您，非常赞
[按需构建多模块，玩转Maven反应堆](#)