

02 仓库

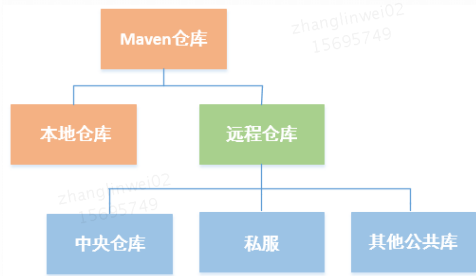
C-2 创建: 张林伟, 最后修改: 张林伟 2018-08-01 20:11

仓库布局

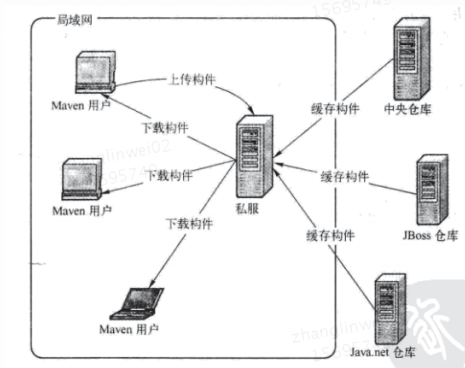
以构件: groupId = org.testng、artifactId = testng、version = 5.8、classifier = jdk5、packaging = jar 为例

- 1. 基于 groupId 准备路径: 句点分隔符转换成路径分隔符, 之后再加一个路径分隔符, 即 org.testng 就成为 org/testng/。
- 2. 基于 artifactId 准备路径: 在前面基础上加上 artifactId 以及一个路径分隔符, 结果为 org/testng/testng/。
- 3. 使用版本信息: 在前面基础上加上 version 以及一个路径分隔符, 结果为 org/testng/testng/5.8/。
- 4. 依次加上 artifactId、构件分隔符和 version。结果为 org/testng/testng/5.8/testng-5.8。
- 5. 如果有 classifier, 就加上构件分隔符和 classifier。结果为 org/testng/testng/5.8/testng-5.8-jdk5。
- 6. 检查构件的 extension, 若 extension 存在, 则加上句点分隔符和 extension。packaging 决定 extension。结果为 org/testng/testng/5.8/testng-5.8-jdk5.jar。

仓库分类



私服用途



远程仓库配置

可以在POM中配置其他远程仓库

^ 代码块XML

```
1 <project>
2   ...
3   <repositories>
4     ...
5     <repository>
6       <id>jboss</id>
7       <name>JBoss Repository</name>
8       <url>http://repository.jboss.com/maven2/</url>
9       <releases>
10        <enabled>true</enabled>
11      </releases>
12      <snapshots>
13        <enabled>false</enabled>
14      </snapshots>
15      <layout>default</layout>
16    </repository>
17    ...
```

```
18     </repositories>
19     ...
20 </project>
```

id必须唯一。

releases的enabled值为true，snapshots的enabled值为false，表示开启此仓库发布版本下载支持和关闭快照版本下载支持。

layout值为default表示仓库布局是maven2及maven3的默认布局，而不是maven1的布局。

releases和snapshots

^ 代码块XML

```
1     <snapshots>
2       <enabled>false</enabled>
3       <updatePolicy>daily</updatePolicy>
4       <checksumPolicy>ignore</checksumPolicy>
5     </snapshots>
```

updatePolicy

- daily：每天检查一次。默认值。
- never：从不检查更新。
- always：每次构建都检查更新。
- interval：每隔多少分钟检查一次。

checksumPolicy（验证校验和文件策略）

- warn：输出警告信息。默认值。
- fail：校验错误就构建失败。
- ignore：忽略校验和错误。

远程仓库认证

部分远程仓库（如私服）需要认证信息才能访问

在settings.xml配置认证信息

^ 代码块XML

```
1     <settings>
2     ...
3     <servers>
4     ...
5     <server>
6       <id>repo-id</id>
7       <username>repo-user</username>
8       <password>repo-pwd</password>
9     </server>
10    ...
11  </servers>
12  ...
13 </settings>
```

部署至远程仓库

在 POM 中配置

^ 代码块XML

```
1 <project>
2   ...
3   <distributionManagement>
4     <repository>
5       <id>meituan-nexus-releases</id>
6       <name>Meituan Nexus Releases Repository</name>
7       <url>http://maven.sankuai.com/nexus/content/repositories/releases/</url>
8     </repository>
9   </distributionManagement>
```

```
10     <snapshotRepository>
11         <id>meituan-nexus-snapshots</id>
12         <name>Meituan Nexus Snapshots Repository</name>
13         <url>http://maven.sankuai.com/nexus/content/repositories/snapshots</url>
14     </snapshotRepository>
15 </distributionManagement>
16 ...
17 </project>
```

repository 表示发布版本构件的仓库，snapshotRepository 表示快照版本构件的仓库。

认证信息配置方式同上。

运行 **mvn clean deploy** 进行部署。

快照版本

maven 快照版本机制会自动为快照版本（2.1-SNAPSHOT）打上时间戳（如2.1-20171214.221414-13，表示2017年12月14日22点14分14秒的第13次快照）。有个该时间戳，maven 就能随时找到仓库中2.1-SNAPSHOT版本的最新文件。

其他依赖该版本的项目，maven 会自动从仓库中检查该依赖快照版本的最新构件。

镜像

如果仓库 X 可以提供仓库 Y 存储的所有内容，那么就可以认为 X 是 Y 的一个镜像。

编辑 settings.xml 来配置镜像

^ 代码块

XML

```
1 <settings>
2 ...
3 <mirrors>
4 ...
5 <mirror>
6     <id>maven.net.cn</id>
7     <name>one of the central mirrors in China</name>
8     <url>http://maven.net.cn/content/groups/public</url>
9     <mirrorOf>central</mirrorOf>
10 </mirror>
11 ...
12 </mirrors>
13 ...
14 </settings>
```

任何对于中央仓库（central）的请求都会转至该镜像。

认证信息配置方式同上。