链滴

maven 总结 (1)

作者：cxmnb
原文链接：https://ld246.com/article/1589359023960
来源网站：链滴

1. Maven入门指南

1.1 Maven是什么

Maven的正确发音是[mev n]，而不是“马瘟”以及其他什么瘟。Maven在美国是一个口语化的词，代表专家、内行的意思，约等于北京话中的老炮儿。

Apache Maven，是一个软件（特别是Java软件）**项目管理及自动构建工具**，由Apache软件基金会提供。基于**项目对象模型（Project Object Model缩写：POM）**概念，Maven利用一个中央信片断能管理一个项目的构建、报告和文档等步骤。Maven也可被用于构建和管理各种项目，例如C#，uby，Scala和其他语言编写的。

1.2 为什么要用Maven

Maven 是一个项目管理和整合工具。**Maven 为开发者提供了一套完整的构建生命周期框架。**开发团几乎不用花多少时间就能够自动完成工程的基础构建配置，因为 **Maven 使用了一个标准的目录结构一个默认的构建生命周期。**

最主要优势可以总结一下三点：

- 生命周期管理，便捷的构建过程；
- 依赖管理，方便引入所需依赖 Jar 包；
- 仓库管理，提供统一管理所有 Jar 包的工具；
- 目录结构管理，提供了一套标准的目录结构（基本上所有的web项目，目录结构几乎都是相同的）

当然还有其他的优点：

- 插件式架构，大量的可重用插件；
- 很方便集成IDE；
- 开源项目都使用Maven

1.3 Maven安装

1.首先要安装好Java环境，然后需要下载Maven的二进制包

2.然后直接解压指令：

```
tar zxvf apache-maven-3.5.0-bin.tar.gz
```

3.设置Maven全局变量：在.bash_prodile或者.zshrc中添加对应的内容

```
export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.7.0_76.jdk/Contents/Home/

export M2_HOME=/Users/zhangguanghui/Public/apache-maven-3.3.9

export M2=$M2_HOME/bin:$PATH

export MAVEN_OPTS="-Xms256m -Xmx512m"
```

JAVA_HOME是电脑中的Java路径，M2_HOME是Maven的解压路径，MAVEN_OPTS是maven的jv配置。这个地方根据自己的位置而定，这个是MAC的安装配置。

1. 高级配置

● 每个公司或者和小组织一般都有自己的私有仓库，所以加入团队要首先配置自己的settings.xml文， 当然最直接的方式是直接从同事那里进行拷贝。

● 接下来是配置本地仓库位置，默认位置是**${user.home}/.m2/repository/**, 如果想将仓库位置成自己想要的位置，在settings.xml中修改localRepository的属性就可以

```
<settings>
      ...
      <localRepository>/path/to/local/repo/</localRepository>
      ...
  </settings>
```

● **配置私有仓库和仓库的注册用户名、密码。**公司的远程私有仓库部署定义在一个项目中的pom.ml文件中，通过\来定义发布仓库位置，有几个仓库就在里面定义几个\标签，每个仓库都有唯一的\签和\标签。公司的私有仓库一般需要用户名，密码去认证才能进行下载，但是这些不能保存在项目，pom文件是要上传到git服务器上，所有人都能看到，不安全，基于这个考虑可以在setting.xml进添加，通过定义\标签来定义多个私有仓库认证信息，每个仓库都有一个\相对应，然后通过定义\标签pom文件中的\中的\标签相同来对应，然后在里面定义\标签和\标签。如下分别是pom.xml中定义仓位置，setting.xml定义对应的私有仓库用户名和密码。

pom.xml中定义仓库

```
  <project>
    ....
    <distributionManagement>
      <repository>
        <id>libs-releases</id>
        <url>http://mvn.hz.netease.com/artifactory/libs-releases</url>
      </repository>
      <snapshotRepository>
        <id>libs-snapshots</id>
        <url>http://mvn.hz.netease.com/artifactory/libs-snapshots</url>
      </snapshotRepository>
    </distributionManagement>
  </project>
```

setting.xml定义仓库的用户名和密码

```
  <servers>
    <server>
      <id>libs-snapshots</id>
      <username>******</username>
      <password>******</password>
    </server>
    <server>
      <id>libs-releases</id>
      <username>******</username>
      <password>******</password>
    </server>
  </servers>
```

可以发现pom中的仓库的id与setting的id是相对应的。

更多配置请参见 4.Maven之setting.xml文件详解。

1.4 创建第一个Maven工程

- **使用archetype模板创建工程**

```
mvn -B archetype:generate \

 -DarchetypeGroupId=org.apache.maven.archetypes \

 -DgroupId=com.mycompany.app \

 -DartifactId=my-app
```

命令执行成功后，产生一个目录文件 my-app，该目录下有一个文件 pom.xml。Maven 就是通过 pom.xml 来构建工程。

- **pom.xml文件内容**

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/
MLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-
4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>my-app</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

该pom.xml文件内包含了最基础的信息，主要有：

标识符 含义

groupId 一般用该项目的组织或团体的域名来标识，例如:org.apache.maven.plugins

artifactId 代表唯一的工程名

version 版本号

packaging 标识打包的类型，例如有:jar, war, tar

dependencies 该工程内依赖的其他 jar 包

想了解更多信息，请参考 5. Maven之Pom.xml配置文件详解

- **坐标**

每一个 Jar 包都需要定义一个唯一标识，方便管理维护，因此 **Maven 使用 groupId, artifactId, versionId 三元素组成一个 Jar 的坐标。**当我们依赖该 Jar 包时，同样需要指定该 Jar 包的坐标

- **工程的目录结构**

```
.
|____pom.xml
|____src
| |____main
| | |____java
| | | |____com
| | | | |____mycompany
| | | | | |____app
| | | | | | |____App.java
| |____test
| | |____java
| | | |____com
| | | | |____mycompany
| | | | | |____app
| | | | | | |____AppTest.java
```

如果你想对 Maven 工程的目录结构更多的了解，请阅读: 请参考 6. Maven之约定大于配置

1.5 创建一个documentation

```
mvn archetype:generate \

  -DarchetypeGroupId=org.apache.maven.archetypes \

  -DarchetypeArtifactId=maven-archetype-site \

  -DgroupId=com.mycompany.app \

  -DartifactId=my-app-site`
```

1.6 创建一个web应用

```
mvn archetype:generate \

    -DarchetypeGroupId=org.apache.maven.archetypes \

    -DarchetypeArtifactId=maven-archetype-webapp \

    -DgroupId=com.mycompany.app \

    -DartifactId=my-webapp
```