**大数据实训工具**

**Maven**

|  |  |  |
| --- | --- | --- |
|  | 文件标识： |  |
| 当前版本： | 1.0 |
| 作 者： | 马卫花 |
| 完成日期： | 2022-10 |

**目 录**

[1项目依赖的概念 2](#_Toc13084)

[2依赖与传递依赖 2](#_Toc13457)

[3版本冲突 2](#_Toc5370)

[4 Maven 3](#_Toc31318)

[5 Maven核心配置文件 3](#_Toc22445)

[6 项目结构 6](#_Toc29650)

## 1项目依赖的概念

我们任何一个人都是站在巨人的肩膀上

我们开发的项目也是基于前人的经验。“不要重复造轮子”意思是不要重复的去做别人已经做过的事情。所以我们的项目会使用别人已做好的工具包或框架。

## 2依赖与传递依赖

假设项目webproject依赖了Druid，Druid是否会依赖其他的工具或框架呢？答案是肯定的。那么webproject和Druid会有多少个这类的依赖呢？答案是很多个。所以一个项目的依赖关系就如同一颗树一样，具备层级关系。例如

Note

druid

commons

castjson

......

说明：webproject依赖了druid,druid依赖了commons,那么显然在运行时需要commons的代码才可以的，所以note间接也依赖了commons。

## 3版本冲突

版本是某个时间点上的代码。一旦版本发布，代码就固定了。版本升级，代码就会变化。一般版本升级会对代码做哪些操作呢?

* 增加功能或者修改bug,代码增加
* 删除功能，比如废弃老旧的功能，代码减少。
* 更改功能，有可能更改包名。

所以版本升级可能是兼容老版本，也可能不兼容老版本。假设一个项目utools是我们团队的开发神器工具包。有可能utools2.0和utools1.0是不兼容的。如果不兼容，会出现什么问题呢？轻则编译报错，重则运行时代码逻辑出错。如果在编码过程中出现版本冲突的问题，可以通过**Maven工具**进行管理。

## 4 Maven

Apache Maven时一个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（POM）概念，Maven利用一个中央信息片断能管理一个项目的**构建**、**报告和文档**等步骤。

Maven时一个项目管理及自动构建工具。

Maven配置步骤:

1. 下载Maven压缩包然后解压。如解压的路径为：C:\Program Files\apache-maven-3.6.3\
2. 打开Maven目录conf文件夹中的settings.xml文件，设置本地仓库的路径何和下载的镜像源，如下所示：

<localRepository>D:\develop\repository</localRepository>

<mirror>

<id>alimaven</id>

<name>aliyun maven</name>

<url>http://maven.aliyun.com/nexus/content/groups/public/</url>

<mirrorOf>central</mirrorOf>

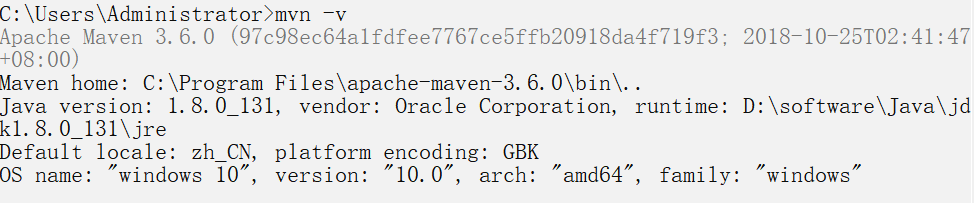
</mirror>

1. 设置系统的环境变量，如下图所示：



4）验证安装是否成功

环境变量设置完后，打开命令提示符，然后输入mvn -v 如果显示如下提示信息，说明安装成功。



## 5 Maven核心配置文件

Maven核心配置文件settings.xml，如下配置:

<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/settings-1.0.0.xsd">

<!-- localRepository

| The path to the local repository maven will use to store artifacts.

|

| Default: ${user.home}/.m2/repository

-->

<localRepository>D:\develop\repository</localRepository>

<!-- interactiveMode

| This will determine whether maven prompts you when it needs input. If set to false,

| maven will use a sensible default value, perhaps based on some other setting, for

| the parameter in question.

|

| Default: true

<interactiveMode>true</interactiveMode>

-->

<!-- offline

| Determines whether maven should attempt to connect to the network when executing a build.

| This will have an effect on artifact downloads, artifact deployment, and others.

|

| Default: false

<offline>false</offline>

-->

<!-- pluginGroups

| This is a list of additional group identifiers that will be searched when resolving plugins by their prefix, i.e.

| when invoking a command line like "mvn prefix:goal". Maven will automatically add the group identifiers

| "org.apache.maven.plugins" and "org.codehaus.mojo" if these are not already contained in the list.

|-->

<pluginGroups>

<!-- pluginGroup

| Specifies a further group identifier to use for plugin lookup.

<pluginGroup>com.your.plugins</pluginGroup>

-->

</pluginGroups>

<!-- proxies

| This is a list of proxies which can be used on this machine to connect to the network.

| Unless otherwise specified (by system property or command-line switch), the first proxy

| specification in this list marked as active will be used.

|-->

<proxies>

<!-- proxy

| Specification for one proxy, to be used in connecting to the network.

|

<proxy>

<id>optional</id>

<active>true</active>

<protocol>http</protocol>

<username>proxyuser</username>

<password>proxypass</password>

<host>proxy.host.net</host>

<port>80</port>

<nonProxyHosts>local.net|some.host.com</nonProxyHosts>

</proxy>

-->

</proxies>

<!-- Another sample, using keys to authenticate.

<server>

<id>siteServer</id>

<privateKey>/path/to/private/key</privateKey>

<passphrase>optional; leave empty if not used.</passphrase>

</server>

-->

</servers>

<mirror>

<id>alimaven</id>

<name>aliyun maven</name>

<url>http://maven.aliyun.com/nexus/content/groups/public/</url>

<mirrorOf>central</mirrorOf>

</mirror>

</mirrors>

</settings>

## 6 项目结构

src 源码

|--main 主要内容

| |--java Java代码

| |--com 以下是Java包的路径

| |--example

| |-demo

| |--resources 资源包括配置文件等

|--test 测试内容

|----

pom.xml 最核心的配置文件