

maven的介绍与使用

作者：徐庶

maven的介绍与使用
1、maven的简单介绍
2、maven的安装
3、maven的基本常识
设置本地仓库：
设置镜像：
设置默认jdk版本：
4、maven常用命令

1、maven的简单介绍

Maven是Apache下的项目管理工具，它由纯Java语言开发，可以帮助我们更方便的管理和构建Java项目。

maven的优点

- 1、 jar包管理：
  - a) 从Maven中央仓库获取标准的规范的jar包以及相关依赖的jar包，避免自己下载到错误的jar包；
  - b) 本地仓库统一管理jar包，使jar包与项目分离，减轻项目体积。
- 2、 maven是跨平台的可以在window、linux上使用。
- 3、 清晰的项目结构；

---src	源码
--- ---main	存放主程序
--- --- ---java	存放Java源文件
--- --- ---resources	存放框架或其他工具的配置文件
--- ---test	存放测试程序
--- --- ---java	存放Java 测试的源文件
--- --- ---resources	存放测试的配置文件
---pom.xml	Maven 工程的核心配置文件

4、 多工程开发，将模块拆分成若干工程，利于团队协作开发。

5、 一键构建项目：使用命令可以对项目进行一键构建。

## 2、maven的安装

maven官网： <https://maven.apache.org/>

maven仓库： <https://mvnrepository.com/>

安装步骤：

- 1、安装jdk
- 2、从官网中下载对应的版本
- 3、解压安装，然后配置环境变量，需要配置MAVEN\_HOME,并且将bin目录添加到path路径下
- 4、在命令行中输入mvn -v,看到版本信息表示安装成功

## 3、maven的基本常识

### maven如何获取jar包

maven通过坐标的方式来获取 jar包，坐标组成为： 公司/组织（groupId）+项目名（artifactId）+版本（version）组成，可以从互联网，本地等多种仓库源获取jar包

### maven仓库的分类

本地仓库：本地仓库就是开发者本地已经下载下来的或者自己打包所有jar包的依赖仓库，本地仓库路径配置在maven对应的conf/settings.xml配置文件。

设置本地仓库：

```
1 <localRepository>C:\repmvn</localRepository>
```

私有仓库：私有仓库可以理解为自己公司的仓库，也叫Nexus私服

中央仓库：中央仓库即maven默认下载的仓库地址，是maven维护的

### maven的常用仓库

由于网络访问的原因，在国内如果需要下载国外jar包的时候会受限，因此一般在使用过程中需要修改maven的配置文件，将下载jar包的仓库地址修改为国内的源，常用的是阿里云的mvn仓库，修改配置如下：

maven目录/conf/setting.xml

### 设置镜像：

```
1      <mirror>
2          <id>nexus-aliyun</id>
3          <name>Nexus aliyun</name>
4          <url>http://maven.aliyun.com/nexus/content/groups/public</url>
5          <mirrorOf>central</mirrorOf>
6      </mirror>
7      <mirror>
8          <id>central</id>
9          <name>Maven Repository Switchboard</name>
10         <url>http://repo.maven.org/maven2/</url>
11         <mirrorOf>central</mirrorOf>
12     </mirror>
13     <mirror>
14         <id>central</id>
15         <name>Maven Repository Switchboard</name>
16         <url>http://repo2.maven.org/maven2/</url>
17         <mirrorOf>central</mirrorOf>
18     </mirror>
19
```

### 设置默认jdk版本：

```
1  <!--修改默认jdk编译版本，默认jdk1.5-->
2  <profile>
```

```
3      <id>jdk-1.8</id>
4      <activation>
5          <activeByDefault>true</activeByDefault>
6          <jdk>1.8</jdk>
7      </activation>
8  <properties>
9  <maven.compiler.source>1.8</maven.compiler.source>
10 <maven.compiler.target>1.8</maven.compiler.target>
11 <maven.compiler.compilerVersion>1.8</maven.compiler.compilerVersion>
12 </properties>
13 </profile>
```

## 4、maven常用命令

- clean：清理编译后的目录
- compile：编译，只编译main目录，不编译test中的代码
- test-compile：编译test目录下的代码
- test：运行test中的代码
- package：打包，将项目打包成jar包或者war包
- install：发布项目到本地仓库，用在打jar包上，打成的jar包可以被其他项目使用
- deploy：打包后将其安装到pom文件中配置的远程仓库
- site：生成站点目录

### 1.添加依赖

```
1 <dependencies>•
2     <dependency>
3         <groupId>org.springframework</groupId>
4         <artifactId>spring-context</artifactId>
```

```
5      <version>5.2.6.RELEASE</version>
6    </dependency>
7    <dependency>
8      <groupId>junit</groupId>
9      <artifactId>junit</artifactId>
10     <version>4.12</version>
11     <scope>test</scope>
12   </dependency>
13
14 </dependencies>
```

## 2. 配置xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.x
5
6   <bean class="cn.tulingxueyuan.dao.UserOrcDao" id="userDao" ></bean>
7
8   <bean class="cn.tulingxueyuan.service.UserService" id="userService">
9     <property name="dao" ref="userDao"></property>
10   </bean>
11
12 </beans>
```

## 3.新建测试类

```
1 public class TestIoC {
2
3   @Test
```

```
4     public void test01(){
5         ApplicationContext ioc=new ClassPathXmlApplicationContext("spring.xml");
6
7         User bean = ioc.getBean(User.class);
8         System.out.println(bean);
9
10    }
11 }
```