

Maven学习

Maven学习

- Maven简介
- 项目结构
- POM
 - 必需字段
 - 生命周期
- Maven仓库
 - 本地仓库
 - 中央仓库
 - 远程仓库
- 引入外部依赖

Maven简介

知识的积累，利用一个中央信息片段能管理一个项目的构建、报告和文档等步骤，在任意工程中可共享。

- 一个清晰的方式定义项目的组成
- 一个容易的方式发布项目的信息
- 一种简单的方式在多个项目中共享JARs

项目结构

```
.
├─ MavenStudy.iml
├─ pom.xml           //Maven的标识文件
└─ src
    ├─ main
    │   ├─ java       //项目java源代码
    │   └─ resources  //项目的资源
    └─ test
        └─ java        //测试类
```

POM

项目对象模型（Project Object Model），包含了项目的基本信息，用于描述项目如何构建，声明项目依赖等

执行任务时，Maven会读取POM，获取所需的配置信息，然后执行目标

可指定的配置
项目依赖
插件
执行目标
项目构建profile
项目版本
项目开发者列表
相关邮件列表信息

必需字段

```
<!-- 公司或者组织的唯一标志，并且配置时生成的路径也是由此生成。  
如com.companyname.project-group，maven会将该项目打成的jar包放本地路  
径： /com/companyname/project-group -->  
<groupId>com.tongji.doubleZ</groupId>  
  
<!-- 项目的唯一ID，一个groupId下面可能多个项目，就是靠artifactId来区分的 -->  
<artifactId>MavenStudy</artifactId>  
  
<!-- 版本号 -->  
<version>1.0-SNAPSHOT</version>
```

生命周期

阶段	描述
clean	清理，项目清理
validate	验证，验证项目是否正确且所有必须信息可用
compile	编译，源代码编译
test	测试，使用单元测试框架进行测试
package	打包，创建JAR/WAR包
verify	检查，对集成测试的结果进行检查，保证质量达标
install	安装，安装打包的项目到本地仓库
site	项目站点文档创建的处理
deploy	部署，拷贝最终的工程包到远程仓库

Maven仓库

本地仓库

- 存放在：/用户目录/.m2/repository
- 运行Maven时，所需的任何构建都是直接从本地仓库获取的，如果本地仓库没有，会尝试从远程仓库下载构建到本地仓库，然后再使用本地仓库的构建

中央仓库

- [官方maven仓库](#)
- [maven仓库](#)
- 包含了绝大多数流行的开源java构建

远程仓库

- 开发人员自己定制的仓库，包含了所需的代码库或者其他工程中用到的jar文件

引入外部依赖

```
<dependencies>

<!-- mysql -->
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
```

```
        <version>8.0.17</version>
    </dependency>

    <!-- gav框架 -->
    <dependency>
        <groupId></groupId>
        <artifactId></artifactId>
        <version></version>
    </dependency>

</dependencies>
```