目录

# Maven基础 Maven功能 构建、文档生成、报告、依赖、SCMs、发布、分发、邮件列表

## Maven目录结构

|  |  |
| --- | --- |
| ${basedir} | 存放pom.xml和所有的子目录 |
| ${basedir}/src/main/java | 项目的java源代码 |
| ${basedir}/src/main/resources | 项目的资源，比如说property文件，springmvc.xml |
| ${basedir}/src/test/java | 项目的测试类，比如说Junit代码 |
| ${basedir}/src/test/resources | 测试用的资源 |
| ${basedir}/src/main/webapp/WEB-INF | web应用文件目录，web项目的信息，比如存放web.xml、本地图片、jsp视图页面 |
| ${basedir}/target | 打包输出目录 |
| ${basedir}/target/classes | 编译输出目录 |
| ${basedir}/target/test-classes | 测试编译输出目录 |
| Test.java | Maven只会自动运行符合该命名规则的测试类 |
| ~/.m2/repository | Maven默认的本地仓库目录位置 |

# Maven POM

POM( Project Object Model，项目对象模型 ) 是 Maven 工程的基本工作单元，是一个XML文件，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，等等。

执行任务或目标时，Maven 会在当前目录中查找 POM。它读取 POM，获取所需的配置信息，然后执行目标。

POM 中可以指定以下配置：

* 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

在命令控制台，进入 MVN/project 目录，执行以下命令：

C:\MVN\project>mvn help:effective-pom

Maven 将会开始处理并显示 effective-pom。

Effective POM 的结果就像在控制台中显示的一样，经过继承、插值之后，使配置生效。

# 总结 1.在IDEA安装目录Maven目录下，修改Settings.xml中的仓库地址指向项目包文件地址 2.将项目需要的包，在仓库中找到包的配置代码，复制后放到项目的pom.xml文件配置中 注： IDEA可安装Maven Helper插件