

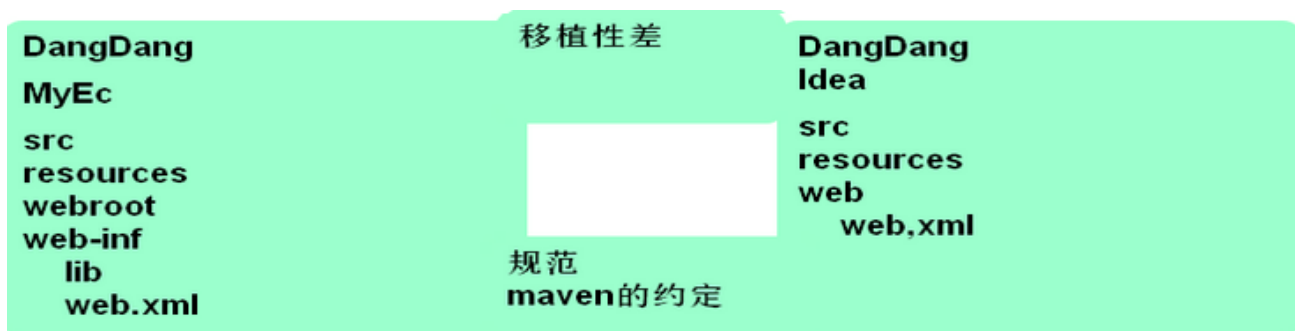
# Maven

## 1.maven的概念

•Maven是Apache开源组织提供的一个基于POM(Project Object Model)的项目构建（Build）工具，所谓的构建指的是代码清除（clean）、编译（compile）、测试（test）、打包(package)、安装（install）、部署（deploy）等一系列流程。

## 2.maven的优势

•可以摆脱IDE的规范 maven约定



•通过命令快速构建项目

mysql sql语句 linux 操作tomcat

•自动管理依赖 告诉maven 自动下载jar包、并且下载依赖jar包 xwork 用struts

•节省磁盘空间（集中式管理依赖）

•支持远程部署等

## 3.小结

### 1.仓库

#### 1.本地仓库

本地仓库

本地磁盘 存放相关jar

远程仓库

不需要咱们维护 它是由maven官方维护的 全球知名的开源的jar

#### 2.坐标

groupId 组织名 公司域名的倒叙

artifactId 项目名

version 版本号

### 3.约定

src/main/java 源代码

src/main/resources 配置文件

src/test/java 测试代码

src/test/resources 测试配置文件

target .class文件 字节码文件

pom.xml maven的配置文件

## 4.安装

### 1.下载安装 解压

bin

boot

conf

lib

LICENSE

NOTICE

README.txt

2018/8/8 10:20 maven 启动 maven 关闭

2018/8/8 10:20 指引远程仓库的jar下载导本地

2018/8/8 10:20

2018/8/8 10:20 配置文件目录 settings.xml

2015/11/10 1

2015/11/10 1

2015/11/10 1

本地仓库的位置 阿里云的镜像

### 2.配置maven的环境变量

Maven\_Home maven安装的目录

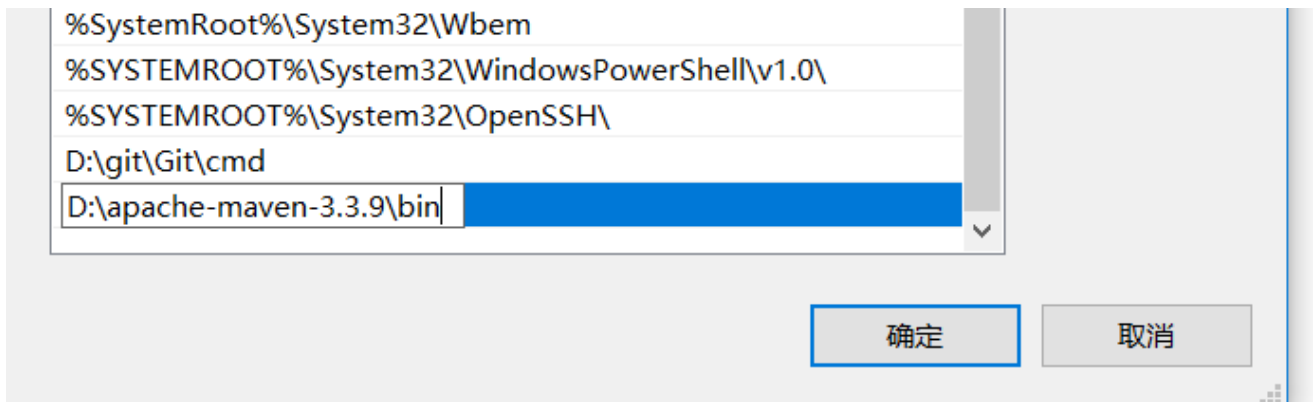
新建系统变量

变量名(N): Maven\_Home

变量值(V): D:\apache-maven-3.6.0

浏览目录(D)... 浏览文件(F)... 确定 取消

Path maven安装的binmulu



### 3.测试maven环境是否正确

```
C:\Users\11453>mvn -v
Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-11T00:41:47+08:00)
Maven home: D:\apache-maven-3.3.9\bin\..
Java version: 1.8.0_131, vendor: Oracle Corporation
Java home: C:\Program Files\Java\jdk1.8.0_131\jre
Default locale: zh_CN, platform encoding: GBK
OS name: "windows 10", version: "10.0", arch: "amd64", family: "dos"
```

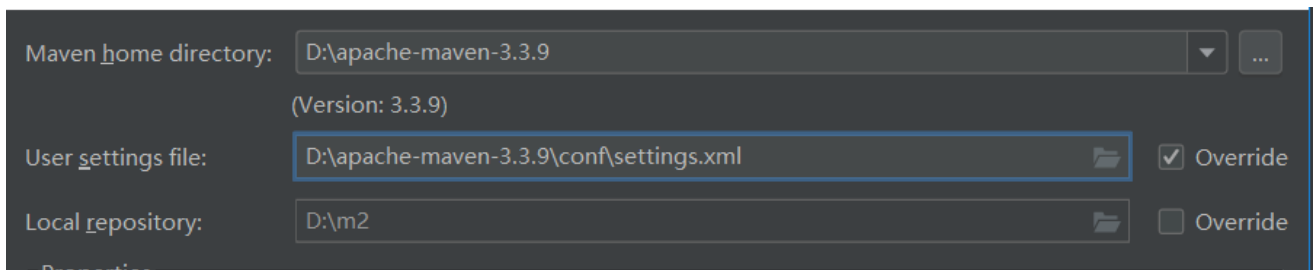
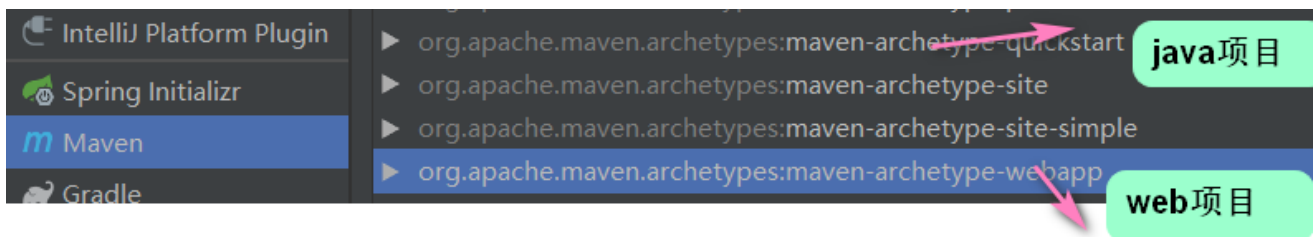
### 4.maven通过指令快速的构建项目

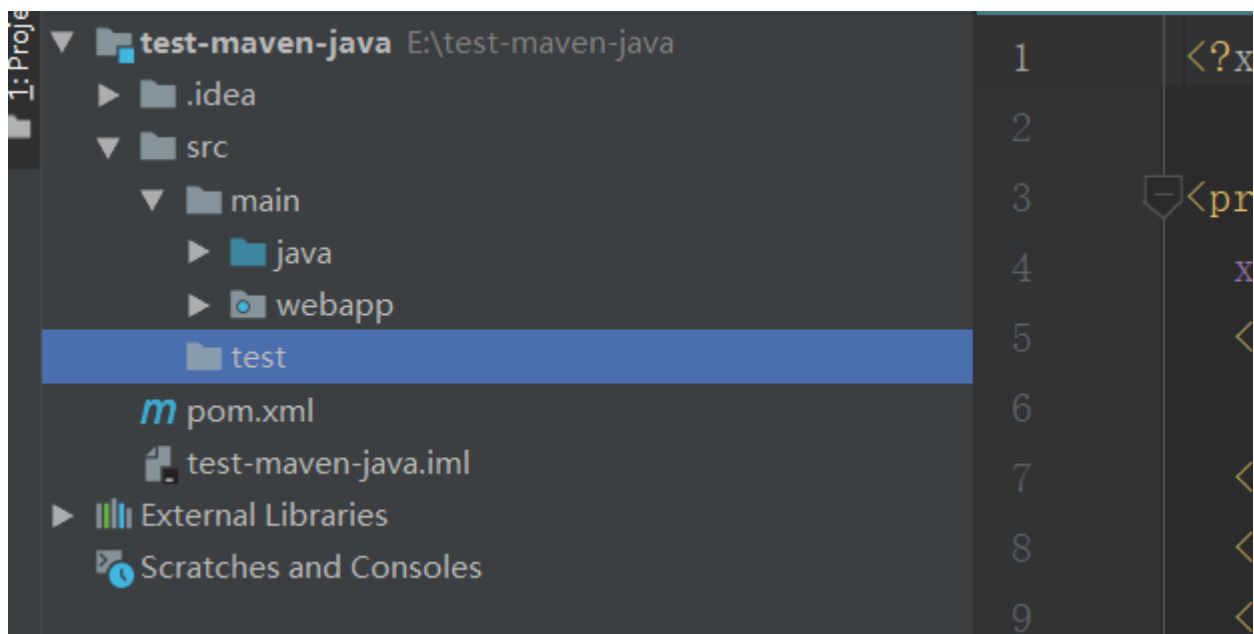
archetype:骨架

```
mvn archetype:generate -DgroupId=com.baizhi -DartifactId=test-maven -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

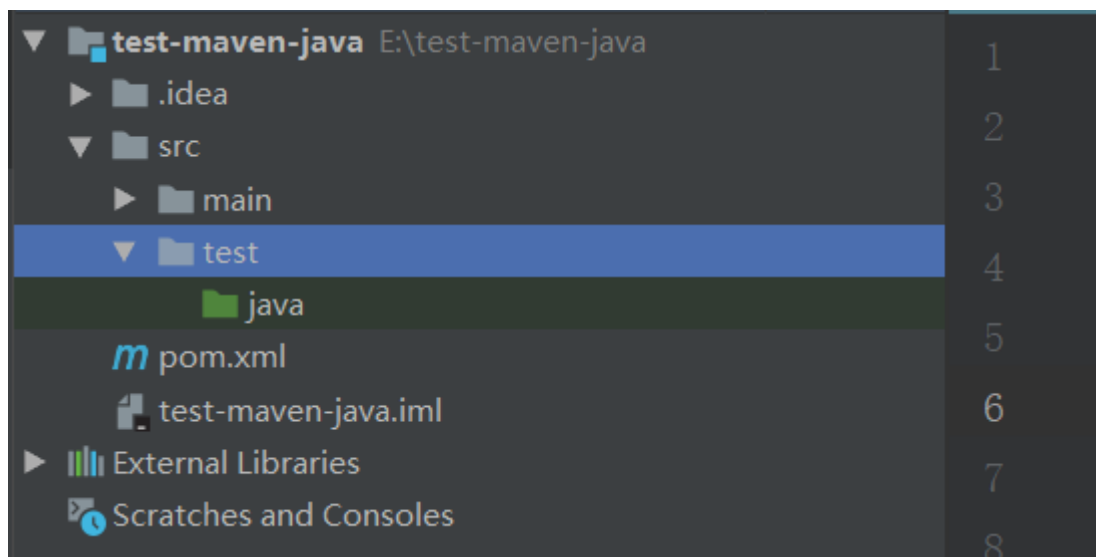
## 4.idea集成maven

### 1.创建项目



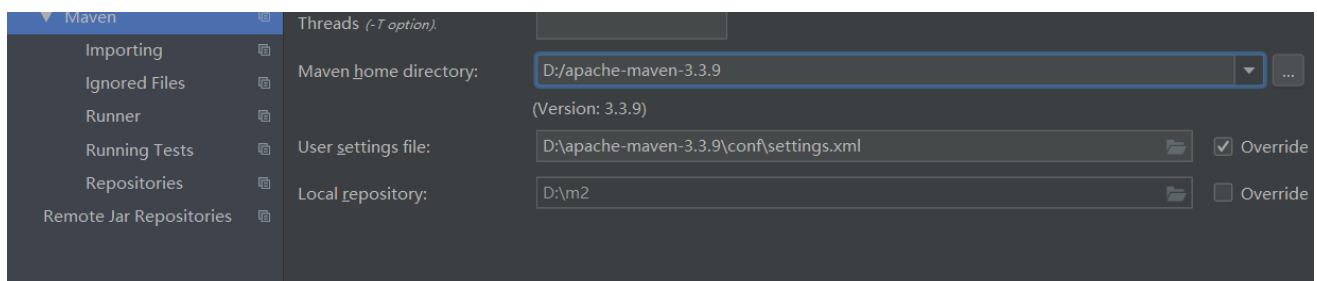


创建 src/main/java 源代码



创建 src/test/java 测试

## 2.maven相关配置



全局配置

## 3.全局配置版本号

```

<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <maven.compiler.source>1.7</maven.compiler.source>
  <maven.compiler.target>1.7</maven.compiler.target>
  <junit.version>4.11</junit.version>
</properties>

<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>${junit.version}</version>
    <scope>test</scope>
  </dependency>

```

注意:

使用 插件test 类名必须以Test开头或者结尾



打包 是以artifactId+version.jar 方式命名

#### 4.maven本地仓库中怎么去找一个jar

groupId+artifactId+version+

5.maven 下载jar不完整

不完整的jar .lastUpdated结尾

名称	修改日期	类型	大小
 spring-context-1.jar.lastUpdated	2019/2/18 15:14	LASTUPDATED ...	1 KB
 spring-context-1.pom.lastUpdated	2019/2/18 15:14	LASTUPDATED ...	1 KB

解决本地仓库中存在的不完整jar(.lastUpdated结尾的文件)

运行脚本

1.编辑脚本中本地仓库的路径

2.运行脚本

#### 5.手动导入非开源的jar(oracle)

mvn install:install-file -DgroupId=com.baizhi -DartifactId=ojdbc -Dversion=1.0.0 -Dpackaging=jar -Dfile=F:\144中期项目\lib\ojdbc6.jar

#### 6.scope 属性

1.test 1.只能在src/test/Java中使用

2.不随项目的发布而发布

2.compile 默认的 跟随项目发布而发布

3.runtime 运行时可以使用,但是不随项目发布而发布

4.provided 由容器或jdk提供