

# Maven简介

2021年9月18日 20:08

Maven是Apache旗下一个纯Java的开源项目，主要用于管理一个项目的构建、依赖、生成报告和文档等步骤  
Maven与IDE无关

Maven最主要的作用是用来**编译打包应用程序**，其次是**管理我们在程序中需要用到Jar包**  
Maven管理Jar包的方式是提供中央仓库和本地仓库

对于在我们项目中的所有文件，Maven提供了一个标准的目录结构，规定了生产代码、测试代码放在什么样的目录下，以及这些目录的命名方式，以便所有程序员在共同协作时所有东西都会按照一定的规定来存放

## Maven默认的目录结构

- src目录
  - 所有的源代码以及资源文件都放在src目录下
    - main目录
      - 生产代码以及生产资源文件放在main目录下
        - java目录
          - 所有的java代码都放在java目录下
        - resources目录
          - 配置文件、静态文件放在resources目录下
        - webapp/WEB-INF
          - web应用文件目录，存储web项目的信息，比如存放web.xml、本地图片、jsp视图页面
    - test目录
      - 测试用的生产代码以及测试用的资源文件放在test目录下，结构同main
  - target目录
    - 所有编译的代码，以及所有要用的资源文件都会拷贝到target目录下，最后打包出来的Jar包以及war包也会在target目录下
  - POM文件
    - pom文件位于整个目录的根目录下

## POM文件

POM文件定义了项目的一些基本信息，描述项目如何构建，声明项目依赖  
使用pom文件可以简化开发和构建的过程，使整个项目的管理与具体的IDE无关

### pom文件简单举例（以autowiredemo为例）

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4
5      <!-- 模型版本需要设置为 4.0. -->
6      <modelVersion>4.0.0</modelVersion>
7
8      <!-- 定义pom继承关系-->
9      <parent>
10         <groupId>org.springframework.boot</groupId>
11         <artifactId>spring-boot-starter-parent</artifactId>
12         <version>2.5.3</version>
13         <relativePath/> <!-- lookup parent from repository -->
14     </parent>
15
16     <!-- 工程组的标识。它在一个组织或者项目中通常是唯一的-->
17     <groupId>cn.edu.xmu</groupId>
18
19     <!-- 工程的标识。它通常是工程的名称。-->
20     <artifactId>autowiredemo</artifactId>
21
22     <!-- 工程的版本号。在 artifact 的仓库中，它用来区分不同的版本。-->
23     <version>0.0.1-SNAPSHOT</version>
24
25     <name>autowiredemo</name>
26     <description>Demo project for Spring Boot</description>
27
28     <!-- 打包的格式-->
29     <packaging>jar</packaging>
30
31     <!-- 定义属性，用在后面的定义中-->
32     <properties>
33         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
34     </properties>
35
```

- xml标识，标识本文件是一个xml文件
- project标识，这个头不要动  
pom文件的所有内容都在project标签内
- modelVersion标识版本，不要动
- parent标签定义pom文件的继承关系  
pom文件是可以项目继承的，可以继承父pom文件中所有的定义，在当前pom文件中可以修改以及增加新的定义
- groupId是工程组标识，artifactId是工程名称  
两个合起来构成了这个工程唯一的标识
- version是工程的版本号，每次发布一个新的版本都需要给它一个版本号  
name和description是描述信息
- packaging定义了打包的方式  
SpringBoot工程主要打包为两种方式：jar和war  
jar带了Servlet容器，war不带  
打包为war包需要部署到Servlet应用服务器中
- properties是xml文件的一种常用标签，用来定义一些需要重复用到的值，也可以用来设定一些默认属性

```

32 <properties>
33   <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
34 </properties>
35
36 <!-- 定义依赖关系-->
37 <dependencies>
38   <dependency>
39     <groupId>org.springframework.boot</groupId>
40     <artifactId>spring-boot-starter</artifactId>
41   </dependency>
42
43   <dependency>
44     <groupId>org.springframework.boot</groupId>
45     <artifactId>spring-boot-starter-web</artifactId>
46   </dependency>
47
48   <dependency>
49     <groupId>org.springframework.boot</groupId>
50     <artifactId>spring-boot-starter-test</artifactId>
51     <scope>test</scope>
52     <exclusions>
53       <exclusion>
54         <groupId>org.junit.vintage</groupId>
55         <artifactId>junit-vintage-engine</artifactId>
56       </exclusion>
57     </exclusions>
58   </dependency>
59
60   <dependency>
61     <groupId>org.springframework.boot</groupId>
62     <artifactId>spring-boot-starter-actuator</artifactId>
63   </dependency>
64
65   <dependency>
66     <groupId>de.codecentric</groupId>
67     <artifactId>spring-boot-admin-starter-client</artifactId>
68     <version>2.4.3</version>
69   </dependency>
70
71   <dependency>
72     <groupId>org.jolokia</groupId>
73     <artifactId>jolokia-core</artifactId>
74   </dependency>
75
76   <dependency>
77     <groupId>com.googlecode.json-simple</groupId>
78     <artifactId>json-simple</artifactId>
79     <version>1.1</version>
80   </dependency>
81 </dependencies>
82
83 <!-- 定义构建的方法-->
84 <build>
85   <plugins>
86     <plugin>
87       <groupId>org.springframework.boot</groupId>
88       <artifactId>spring-boot-maven-plugin</artifactId>
89     </plugin>
90   </plugins>
91 </build>
92
93 <!-- maven的仓库-->
94 <repositories>
95   <repository>
96     <id>alimaven</id>
97     <name>aliyun maven</name>
98     <url>http://maven.aliyun.com/nexus/content/groups/public/</url>
99   </repository>
100 </repositories>
101 </project>
102

```

properties是xml文件的一种常用标签，用来定义一些需要重复用到的值，也可以用来设定一些默认属性

dependencies是定义依赖关系，工程所需要的各种jar包都写在dependencies里，每个dependency都是一个独立jar包

依赖关系后是Maven的各种插件  
Maven是依赖于各种插件来完成各种功能的

repositories是仓库的定义，如果不加这一部分默认用的是国外的仓库  
一般我们定义一个国内的Maven仓库镜像，这里是阿里云仓库镜像