

# Maven多模块管理

Maven管理多模块应用的实现是互联网项目中多使用分布式开发，那么每个独立的服务都会使用独立的项目进行维护，那么这样就需要使用多模块应用管理，来实现项目的高度统一。

## maven父工程必须遵循一下两点要求

1. packaging标签的文本内容必须设置为pom
2. 把src目录删除掉

## 父模块pom

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <!--
        pom是项目对象模型（Project Object Module），该文件是可以被继承的
        maven多模块管理，其实就是让它的子模块的pom文件来继承父工程的pom文件
    -->
    <groupId>com.qiangliu8</groupId>
    <artifactId>mavenModulesProjects</artifactId>
    <packaging>pom</packaging>
    <version>1.0-SNAPSHOT</version>
    <modules>
        <module>002-son</module>
        <module>003-web</module>
    </modules>

    <!--packing标签的是指定的打包方法
        默认是jar，如果pom文件中没有packaging标签那么默认就是打开jar
    -->
    <properties>
        <maven.compiler.source>15</maven.compiler.source>
        <maven.compiler.target>15</maven.compiler.target>
    </properties>
</project>
```

## 子模块pom

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <!--指向父工程的GAV坐标-->
    <parent>
        <artifactId>mavenModulesProjects</artifactId>
        <groupId>com.qiangliu8</groupId>
        <version>1.0-SNAPSHOT</version>
    </parent>
```

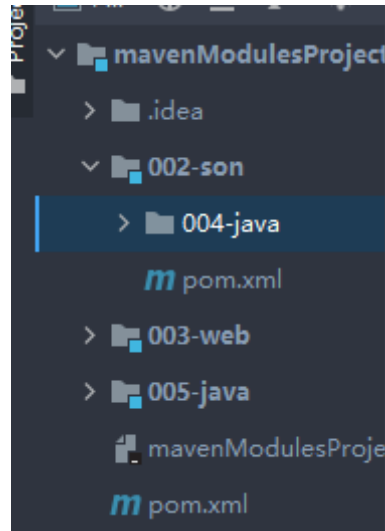
```
<modelVersion>4.0.0</modelVersion>

<artifactId>002-son</artifactId>

<packaging>pom</packaging>

</project>
```

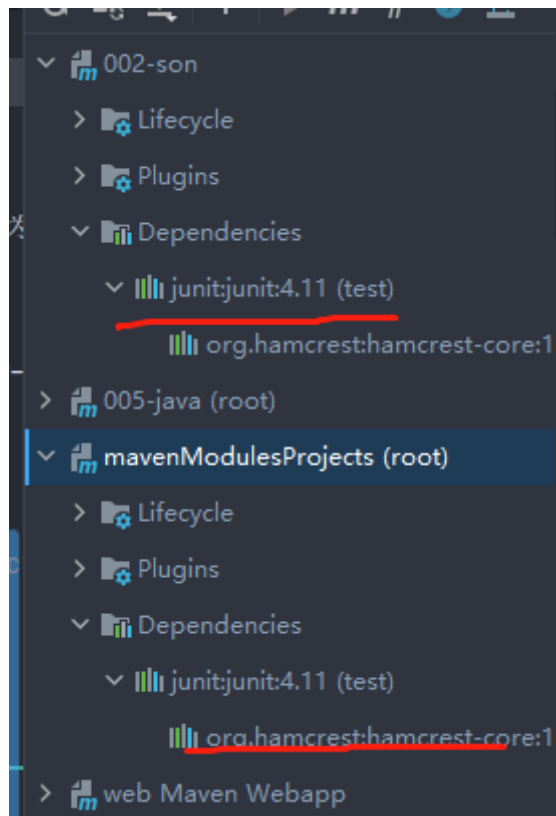
子模块也可以当作父模块



## 父工程添加的依赖，子模块会无条件继承

父模块添加一个依赖后，所有子模块默认都添加该依赖

```
//父POM
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>
</dependencies>
```



这个时候需要加强管理，使用**dependencyManagement**

```
<!--父工程要加强管理子模块的所有依赖-->
<dependencyManagement>

    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.11</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
</dependencyManagement>
```

**所有子模块依赖版本号继承父工程**

父

```
<dependencies>
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.13</version>
    </dependency>
</dependencies>
```

子

```
<dependencies>
  <!--声明式依赖-->
  <!--所有子模块依赖版本号继承父工程
  如果子模块指定依赖的版本号那就不会继承父工程依赖的版本号
  -->
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
  </dependency>
</dependencies>
```

## 自定义版本号

```
<properties>

  <spring-webmvc-version>5.2.5.RELEASE</spring-webmvc-version>
  <mysql-connector-java-version>8.0.13</mysql-connector-java-version>
</properties>

<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>${spring-webmvc-version}</version>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>${mysql-connector-java-version}</version>
  </dependency>
</dependencies>
```