

第二章 企业项目开发--maven父子模块

2.1、maven父子模块

在实际开发中，我们基本都会用maven父子分模块的方式进行项目的开发。

2.2、实际操作

2.2.1、手工建立一个ssmm0的文件夹，并在该文件夹中加入一个pom.xml文件，该pom.xml文件内容如下：

```
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/maven-v4_0_0.xsd">
4
5     <modelVersion>4.0.0</modelVersion>
6
7     <groupId>com.xxx</groupId>
8     <artifactId>ssmm0</artifactId>
9     <version>1.0-SNAPSHOT</version>
10
11     <name>ssmm0</name>
12     <packaging>pom</packaging><!-- 父模块 -->
13
14     <!-- 管理子模块 -->
15     <modules>
16         <module>ssmm</module>
17     </modules>
18
19     <properties>
20         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
21         <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
22     </properties>
23 </project>
```

注意：

- 父模块的pom.xml文件的<packaging>标签内容为pom；而需要部署的子项目为war；只是作为其他项目的工具的子项目为jar
- 使用<modules> 标签管理所有的子模块，以后再有新的子模块，只需要在<modules>添加新的<module>子标签即可

2.2.2、将上一章建好的那个项目ssmm放入ssmm0文件夹下，修改pom.xml如下：

```
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/maven-v4_0_0.xsd">
4
5     <modelVersion>4.0.0</modelVersion>
6
7     <!-- 指定父模块 -->
8     <parent>
9         <groupId>com.xxx</groupId>
10        <artifactId>ssmm0</artifactId>
11        <version>1.0-SNAPSHOT</version>
12    </parent>
13
14    <groupId>com.xxx.ssmm0</groupId>
15    <artifactId>ssmm0-ssmm</artifactId>
16    <!--<version>1.0-SNAPSHOT</version>--><!-- 父模块已经指定了版本号，这里就不用了-->
17
18    <name>ssmm0-ssmm</name>
19    <packaging>war</packaging>
20
```

```
21 <properties>
22     <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
23     <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
24 </properties>
25
26 <!-- 引入实际依赖 -->
27 <dependencies>
28     <!-- json -->
29     <dependency>
30         <groupId>com.alibaba</groupId>
31         <artifactId>fastjson</artifactId>
32         <version>1.1.39</version>
33     </dependency>
34     <!-- servlet -->
35     <dependency>
36         <groupId>javax.servlet</groupId>
37         <artifactId>javax.servlet-api</artifactId>
38         <version>3.0.1</version>
39         <scope>provided</scope>
40     </dependency>
41     <!-- spring -->
42     <dependency>
43         <groupId>org.springframework</groupId>
44         <artifactId>spring-core</artifactId>
45         <version>3.2.6.RELEASE</version>
46     </dependency>
47     <dependency>
48         <groupId>org.springframework</groupId>
49         <artifactId>spring-beans</artifactId>
50         <version>3.2.6.RELEASE</version>
51     </dependency>
52     <dependency>
53         <groupId>org.springframework</groupId>
54         <artifactId>spring-context</artifactId>
55         <version>3.2.6.RELEASE</version>
56     </dependency>
57     <dependency>
58         <groupId>org.springframework</groupId>
59         <artifactId>spring-web</artifactId>
60         <version>3.2.6.RELEASE</version>
61     </dependency>
62     <dependency>
63         <groupId>org.springframework</groupId>
64         <artifactId>spring-webmvc</artifactId>
65         <version>3.2.6.RELEASE</version>
66     </dependency>
67     <!-- 这个是使用velocity的必备包 -->
68     <dependency>
69         <groupId>org.springframework</groupId>
70         <artifactId>spring-context-support</artifactId>
71         <version>3.2.6.RELEASE</version>
72         <scope>compile</scope>
73     </dependency>
74     <!-- mysql -->
75     <dependency>
76         <groupId>mysql</groupId>
77         <artifactId>mysql-connector-java</artifactId>
78         <version>5.1.27</version>
79         <scope>runtime</scope>
80     </dependency>
81     <!-- 数据源 -->
82     <dependency>
83         <groupId>org.apache.tomcat</groupId>
84         <artifactId>tomcat-jdbc</artifactId>
85         <version>7.0.47</version>
```

```
86         </dependency>
87         <!-- mybatis -->
88         <dependency>
89             <groupId>org.mybatis</groupId>
90             <artifactId>mybatis</artifactId>
91             <version>3.1.1</version>
92         </dependency>
93         <dependency>
94             <groupId>org.mybatis</groupId>
95             <artifactId>mybatis-spring</artifactId>
96             <version>1.1.1</version>
97         </dependency>
98         <!-- velocity -->
99         <dependency>
100             <groupId>org.apache.velocity</groupId>
101             <artifactId>velocity</artifactId>
102             <version>1.5</version>
103         </dependency>
104         <dependency>
105             <groupId>velocity-tools</groupId>
106             <artifactId>velocity-tools-generic</artifactId>
107             <version>1.2</version>
108         </dependency>
109         <!-- 用于加解密 -->
110         <dependency>
111             <groupId>commons-codec</groupId>
112             <artifactId>commons-codec</artifactId>
113             <version>1.7</version>
114         </dependency>
115         <dependency>
116             <groupId>org.bouncycastle</groupId>
117             <artifactId>bcprov-jdk15on</artifactId>
118             <version>1.47</version>
119         </dependency>
120         <!-- 集合工具类 -->
121         <dependency>
122             <groupId>org.apache.commons</groupId>
123             <artifactId>commons-collections4</artifactId>
124             <version>4.0</version>
125         </dependency>
126         <!-- http -->
127         <dependency>
128             <groupId>org.apache.httpcomponents</groupId>
129             <artifactId>httpclient</artifactId>
130             <version>4.2.6</version>
131         </dependency>
132     </dependencies>
133 </project>
```



注意：在上一章的基础上做了以下几点修改

- 添加了<parent>标签，用来指定父模块，该标签内有父模块的坐标（三要素：groupId/artifactId/version）
- **子模块不需要再有版本号了，由父模块来指定就好**
- 将子模块中下边这一块儿代码也删掉（因为在父模块中已经指定了）

```
<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
</properties>
```

建议：

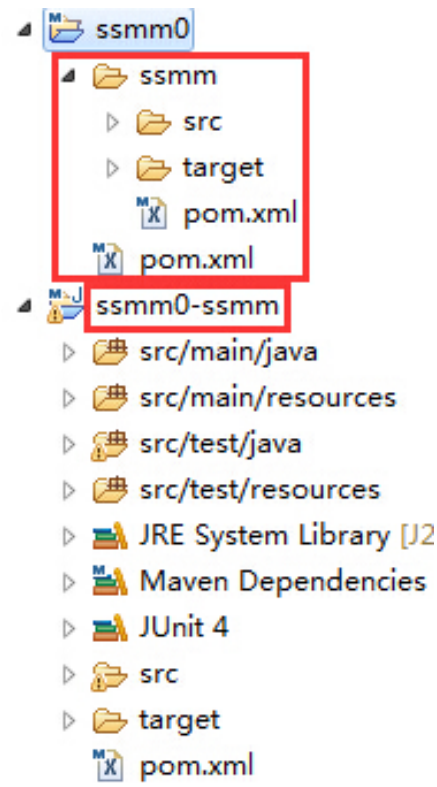
- "子模块的groupId"设为"**父模块的groupId.父模块的artifactId**"
- "子模块的artifactId"设为"**父模块的artifactId-子模块的名称**"，"父模块的artifactId-子模块的名称"也就是子模块的项目名
- 无论父模块还是子模块，建议同一个pom.xml文件中的artifactId与name标签内容相同

这几点建议，在我们编写和部署项目的时候都一目了然；以上几点建议，请对照着以上两个pom.xml文件对号入座。

2.2.3、在ssmm0的文件夹中，即父模块pom.xml所在的文件夹中运行命令窗口，执行"mvn clean compile"命令

2.2.4、编译成功后，将项目ssmm0以maven项目引入eclipse（具体方法：见第一章）

引入的项目结构如下：



注意：

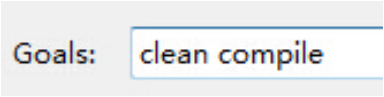
- 以上第一个框处的内容是ssmm0-ssmm编译出来的
- 第二个红框正好印证了"父模块的artifactId-子模块的名称"也就是子模块的项目名这句

2.2.5、运行ssmm项目，进行测试

这样，就建立起了一个maven的父子模块项目了。

注：若以后再增加一个新的模块该怎么做？

- 1) 我们在父模块的pom.xml文件中添加<module>
- 2) 在ssmm0文件夹下手工创建一个子maven项目（创建方式见第一章），假设为ssmm2，ssmm2的pom.xml文件类似于ssmm子项目的pom.xml即可
- 3) 在ssmm2文件夹中，打开命令窗口，执行"mvn compile"
- 4) 将ssmm2引入eclipse
- 5) 用eclipse的maven插件编译在ssmm0上执行："Run as"-->"Maven build"-->"clean compile"



这样，就新建了一个子项目了。

针对maven这一块有几点建议：（很重要）

- 安装maven后，最好在M2_HOME/conf/setting.xml文件添加如下代码，将之后项目中用到的jar包全部下载到下边的文件夹中（当然，在修改setting.xml文件时，最好先复制一份最备份），防止C盘撑爆。

```

<localRepository>E:/Java/mavenLocalRepository</localRepository>

```

- 在实际使用中，我们会架设私服（一般采用nexus），这样做主要是为了加快jar的下载速度，之后需要在父pom.xml文件中指定私服的位置

```

<!-- nexus -->
<repositories>
  <repository>
    <id>xxx-Nexus</id>
    <name>xxx Maven Repository</name>
    <url>http://xxx.com/nexus/</url>
  </repository>
</repositories>

```



一个<repository>标签就是一个私服，可以加多个私服。

- 在实际使用中，我们会在父pom.xml文件中加入一个依赖管理池<dependencyManagement>，在这个池中指定了jar的版本号<version>和范围<scope>，之后在子项目中实际引入jar的坐标的时候，就不需要写<version>标签和<scope>了；当然，这样做，也可以保证在整个项目中，我们使用的同一个jar都是同一个版本；同时，需要指出的是，为了子模块重复代码的减少，在父模块处，会先引入一些所有子模块都会使用的jar（例如：log4j等），这样在子项目中就不需要再引入这些jar了。这里给出父pom.xml与子项目ssmm的pom.xml的完整版。对着代码，理解一下这一条。



```
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/maven-v4_0_0.xsd">
4
5      <modelVersion>4.0.0</modelVersion>
6
7      <groupId>com.xxx</groupId>
8      <artifactId>ssmm0</artifactId>
9      <version>1.0-SNAPSHOT</version>
10
11     <name>ssmm0</name>
12     <packaging>pom</packaging><!-- 父模块 -->
13
14     <!-- 管理子模块 -->
15     <modules>
16         <module>ssmm</module>
17     </modules>
18
19     <properties>
20         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
21         <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
22     </properties>
23
24     <!-- dependencyManagement不会引入实际的依赖，只是作为一个依赖池，供其和其子类使用 -->
25     <dependencyManagement>
26         <dependencies>
27             <!-- json -->
28             <dependency>
29                 <groupId>com.alibaba</groupId>
30                 <artifactId>fastjson</artifactId>
31                 <version>1.1.39</version>
32             </dependency>
33             <!-- servlet -->
34             <dependency>
35                 <groupId>javax.servlet</groupId>
36                 <artifactId>javax.servlet-api</artifactId>
37                 <version>3.0.1</version>
38                 <scope>provided</scope>
39             </dependency>
40             <!-- spring -->
41             <dependency>
42                 <groupId>org.springframework</groupId>
43                 <artifactId>spring-core</artifactId>
44                 <version>3.2.6.RELEASE</version>
45             </dependency>
46             <dependency>
47                 <groupId>org.springframework</groupId>
48                 <artifactId>spring-beans</artifactId>
49                 <version>3.2.6.RELEASE</version>
50             </dependency>
51             <dependency>
52                 <groupId>org.springframework</groupId>
```

```
53         <artifactId>spring-context</artifactId>
54         <version>3.2.6.RELEASE</version>
55     </dependency>
56     <dependency>
57         <groupId>org.springframework</groupId>
58         <artifactId>spring-web</artifactId>
59         <version>3.2.6.RELEASE</version>
60     </dependency>
61     <dependency>
62         <groupId>org.springframework</groupId>
63         <artifactId>spring-webmvc</artifactId>
64         <version>3.2.6.RELEASE</version>
65     </dependency>
66     <!-- 这个是使用velocity的必备包 -->
67     <dependency>
68         <groupId>org.springframework</groupId>
69         <artifactId>spring-context-support</artifactId>
70         <version>3.2.6.RELEASE</version>
71     </dependency>
72     <!-- mysql -->
73     <dependency>
74         <groupId>mysql</groupId>
75         <artifactId>mysql-connector-java</artifactId>
76         <version>5.1.27</version>
77         <scope>runtime</scope>
78     </dependency>
79     <!-- 数据源 -->
80     <dependency>
81         <groupId>org.apache.tomcat</groupId>
82         <artifactId>tomcat-jdbc</artifactId>
83         <version>7.0.47</version>
84     </dependency>
85     <!-- mybatis -->
86     <dependency>
87         <groupId>org.mybatis</groupId>
88         <artifactId>mybatis</artifactId>
89         <version>3.1.1</version>
90     </dependency>
91     <dependency>
92         <groupId>org.mybatis</groupId>
93         <artifactId>mybatis-spring</artifactId>
94         <version>1.1.1</version>
95     </dependency>
96     <!-- velocity -->
97     <dependency>
98         <groupId>org.apache.velocity</groupId>
99         <artifactId>velocity</artifactId>
100         <version>1.5</version>
101     </dependency>
102     <dependency>
103         <groupId>velocity-tools</groupId>
104         <artifactId>velocity-tools-generic</artifactId>
105         <version>1.2</version>
106     </dependency>
107     <!-- 用于加解密 -->
108     <dependency>
109         <groupId>commons-codec</groupId>
110         <artifactId>commons-codec</artifactId>
111         <version>1.7</version>
112     </dependency>
113     <dependency>
114         <groupId>org.bouncycastle</groupId>
115         <artifactId>bcprov-jdk15on</artifactId>
116         <version>1.47</version>
117     </dependency>
```

```
118         <!-- 集合工具类 -->
119         <dependency>
120             <groupId>org.apache.commons</groupId>
121             <artifactId>commons-collections4</artifactId>
122             <version>4.0</version>
123         </dependency>
124         <!-- http -->
125         <dependency>
126             <groupId>org.apache.httpcomponents</groupId>
127             <artifactId>httpclient</artifactId>
128             <version>4.2.6</version>
129         </dependency>
130     </dependencies>
131 </dependencyManagement>
132
133 <!-- 引入实际依赖 -->
134 <dependencies>
135     <!-- json -->
136     <dependency>
137         <groupId>com.alibaba</groupId>
138         <artifactId>fastjson</artifactId>
139     </dependency>
140     <!-- servlet -->
141     <dependency>
142         <groupId>javax.servlet</groupId>
143         <artifactId>javax.servlet-api</artifactId>
144     </dependency>
145     <!-- spring -->
146     <dependency>
147         <groupId>org.springframework</groupId>
148         <artifactId>spring-core</artifactId>
149     </dependency>
150     <dependency>
151         <groupId>org.springframework</groupId>
152         <artifactId>spring-beans</artifactId>
153     </dependency>
154     <dependency>
155         <groupId>org.springframework</groupId>
156         <artifactId>spring-context</artifactId>
157     </dependency>
158     <dependency>
159         <groupId>org.springframework</groupId>
160         <artifactId>spring-web</artifactId>
161     </dependency>
162     <dependency>
163         <groupId>org.springframework</groupId>
164         <artifactId>spring-webmvc</artifactId>
165     </dependency>
166     <!-- 这个是使用velocity的必备包 -->
167     <dependency>
168         <groupId>org.springframework</groupId>
169         <artifactId>spring-context-support</artifactId>
170     </dependency>
171     <!-- mysql -->
172     <dependency>
173         <groupId>mysql</groupId>
174         <artifactId>mysql-connector-java</artifactId>
175     </dependency>
176     <!-- 数据源 -->
177     <dependency>
178         <groupId>org.apache.tomcat</groupId>
179         <artifactId>tomcat-jdbc</artifactId>
180     </dependency>
181     <!-- mybatis -->
182     <dependency>
```



```
183         <groupId>org.mybatis</groupId>
184         <artifactId>mybatis</artifactId>
185     </dependency>
186     <dependency>
187         <groupId>org.mybatis</groupId>
188         <artifactId>mybatis-spring</artifactId>
189     </dependency>
190     <!-- velocity -->
191     <dependency>
192         <groupId>org.apache.velocity</groupId>
193         <artifactId>velocity</artifactId>
194     </dependency>
195     <dependency>
196         <groupId>velocity-tools</groupId>
197         <artifactId>velocity-tools-generic</artifactId>
198     </dependency>
199     <!-- 集合工具类 -->
200     <dependency>
201         <groupId>org.apache.commons</groupId>
202         <artifactId>commons-collections4</artifactId>
203     </dependency>
204 </dependencies>
205 </project>
```



```
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/maven-v4_0_0.xsd">
4
5     <modelVersion>4.0.0</modelVersion>
6
7     <!-- 指定父模块 -->
8     <parent>
9         <groupId>com.xxx</groupId>
10        <artifactId>ssmm0</artifactId>
11        <version>1.0-SNAPSHOT</version>
12    </parent>
13
14    <groupId>com.xxx.ssmm0</groupId>
15    <artifactId>ssmm0-ssmm</artifactId>
16    <!--<version>1.0-SNAPSHOT</version>--><!-- 父模块已经指定了版本号，这里就不用了-->
17
18    <name>ssmm0-ssmm</name>
19    <packaging>war</packaging>
20
21    <properties>
22        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
23        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
24    </properties>
25
26    <!-- 引入实际依赖 -->
27    <dependencies>
28        <!-- 用于加解密 -->
29        <dependency>
30            <groupId>commons-codec</groupId>
31            <artifactId>commons-codec</artifactId>
32        </dependency>
33        <dependency>
34            <groupId>org.bouncycastle</groupId>
35            <artifactId>bcprov-jdk15on</artifactId>
36        </dependency>
```



```
37         <!-- http -->
38         <dependency>
39             <groupId>org.apache.httpcomponents</groupId>
40             <artifactId>httpclient</artifactId>
41         </dependency>
42     </dependencies>
43 </project>
```



对于maven3来讲，常用的scope有以下4种：

- compile：默认，作用于编译、测试和运行
- provided：常见的就是servlet-api，作用于编译和测试（运行的时候由容器提供了）
- runtime：常见的就是mysql-connector，作用在运行和测试时
- test：常见的就是junit，spring-test，作用在测试时

在配置pom.xml的时候，需要将这些写上scope写上。

- 在实际使用中，我们会在父pom.xml配置<profiles>标签，在其中配置多套运行环境（一般而言，配置三套：开发，预上线，上线），每套环境配置不同的数据库和服务器。