

第一章 Maven项目管理

-Maven入门、整合SSM2018/11/17 [泽林.王峰]

授课目标

- 1、Maven简介
- 2、Maven入门
- 3、在Idea中配置Maven
- 4、在Idea中开发第一个Maven项目
- 5、Maven依赖管理

授课内容

1、Maven简介

1.1) Maven是什么？

Maven是Apache下的一个纯Java开发的开源项目，它是一个项目管理工具，使用Maven对Java项目进行构建、依赖管理。当前使用Maven的项目正在持续增长，大公司基本都是使用Maven来作项目管理的。

1.2) Maven项目构建过程：

Maven将项目的构建过程进行标准化，每个阶段使用一个命令完成，下图展示了构建过程：



上图中部分阶段对应命令如下：

清理阶段对应 **maven** 的命令是 **clean**，清理输出的 **class** 文件

编译阶段对应 **maven** 的命令是 **compile**，将 **java** 代码编译成 **class** 文件。

打包阶段对应 **maven** 的命令是 **package**，**java** 工程可以打成 **jar** 包，**web** 包可以打成 **war** 包

运行一个 **maven** 工程（**web** 工程）需要一个命令：**tomcat:run**

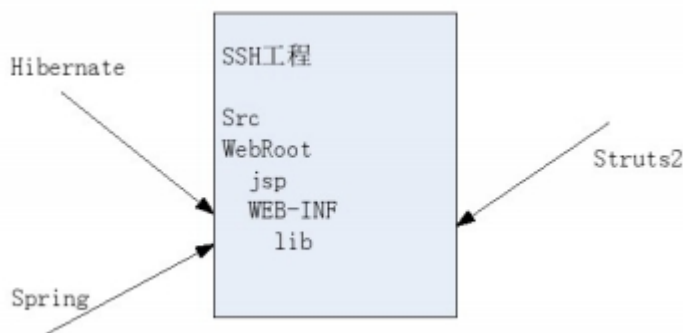
maven 工程构建的优点：

- 1、一个命令完成构建、运行，方便快捷。
- 2、**maven** 对每个构建阶段进行规范，非常有利于大型团队协作开发。

1.3) 传统项目与Maven项目的依赖管理：

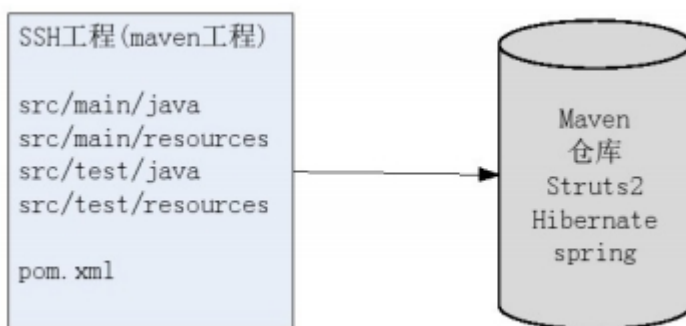
1.3.1) 传统项目的依赖管理：

传统项目要管理项目中使用的jar包，完全要靠手工进行，开发人员要从网上下载相应的jar包到当前的项目中，如图所示：



1.3.2) Maven项目的依赖管理：

Maven项目管理所依赖的jar包不需要手动向工程添加jar包，只需要在pom.xml中添加需要的jar包的坐标，自动从maven仓库中下载jar包，如下图：



Maven依赖管理添加jar的好处：

- ①通过pom.xml文件对jar包的版本进行统一管理，可避免版本冲突。
- ②maven团队维护了一个非常全的maven仓库，里边包括了当前正在使用的所有jar包，maven工程可自动从maven仓库下载jar包，非常方便。

1.4) Maven好处：

1) 一步构建：

2) 依赖管理：

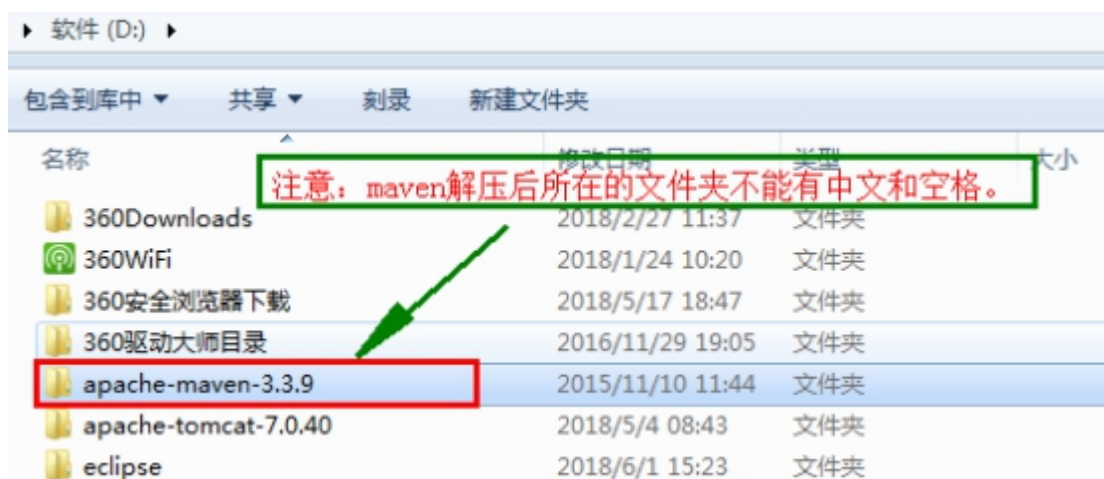
3) 跨平台，可以任意平台，如windows、linux上使用：

4) Maven遵循规范开发有利于提高大型团队的开发效率，降低项目的维护成本，大公司都会使用maven来构建和管理项目。

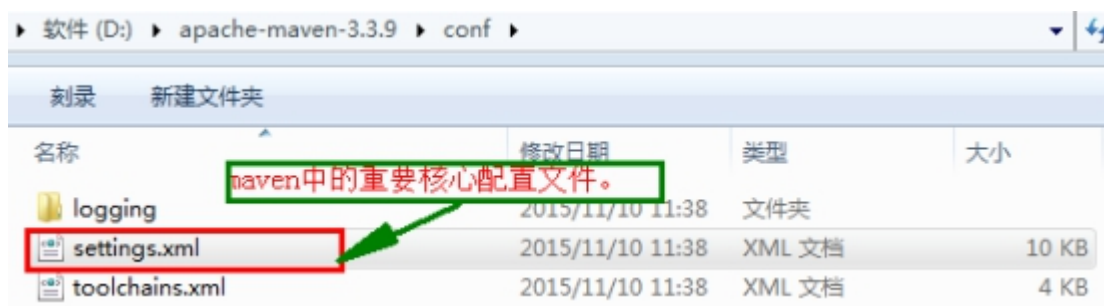
2、Maven入门

2.1) Maven的安装配置过程：

第一步：在<http://maven.apache.org/>下载maven的zip文件并解压缩：



第二步：maven文件夹中的重要文件：



第三步：在系统的环境变量中配置%JAVA_HOME%,再配置%MAVEN_HOME%(代表maven的安装目录)，然后在path路径下添加%MAVEN_HOME%/bin:



第四步：配置完成后，打开“cmd”控制台，输入:mvn -v,出现如下画面，代表配置成功：

```
C:\Users\Administrator.USER-20161129NL>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: D:\Java\jdk1.8.0_131\jre
Default locale: zh_CN, platform encoding: GBK
OS name: "windows 7", version: "6.1", arch: "amd64", family: "dos"
```

第五步：新建普通java项目

```
1 mvn archetype:generate -DgroupId=com.zelin -DartifactId=HelloWorld -
  DarchetypeArtifactId=maven-archetype-quickstart
```

2.5) Maven常用命令：

2.5.1) mvn clean:清理编译后的class文件：

```
E:\maven-helloworld>mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building 第一个maven工程 0.0.1-SNAPSHOT
[INFO] -----
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/22/maven-plugins-22.pom
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/22/maven-plugins-22.pom (13 KB at
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/21/maven-parent-21.pom
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/21/maven-parent-21.pom (26 KB at 33.3 KB/
Downloading: https://repo.maven.apache.org/maven2/org/apache/apache/10/apache-10.pom
Downloaded: https://repo.maven.apache.org/maven2/org/apache/apache/10/apache-10.pom (15 KB at 23.7 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ maven-helloworld ---
```

clean: 代表清理字节码文件，即target目录下的class文件。

2.5.2) mvn compile：代表编译maven项目：

```
信息: Retrying request to {s}->https://repo.maven.apache.org:443
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/maven-model/3.0/maven-model-3.0.jar (1
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to E:\maven-helloworld\target\classes
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 05:08 min
[INFO] Finished at: 2018-06-14T10:08:03+08:00
[INFO] Final Memory: 17M/145M
[INFO] -----
```

mvn compile命令：代表编译maven项目。

2.5.3) mvn test:代表测试maven项目：

TESTS

Running cn.itcast.maven.test.HelloTest

mvn test: 代表运行测试，会首先进行编译。

ello test...

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.176 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]

[INFO] BUILD SUCCESS

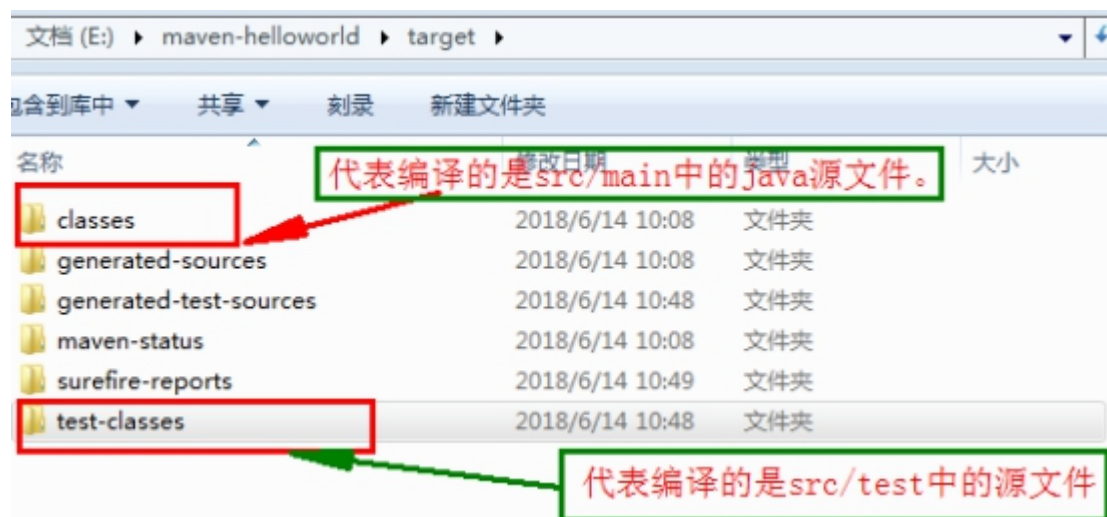
[INFO]

[INFO] Total time: 01:33 min

[INFO] Finished at: 2018-06-14T10:49:28+08:00

[INFO] Final Memory: 17M/142M

[INFO]



2.5.4) mvn package:代表对maven项目进行打包：

[INFO] Packaging webapp

[INFO] Assembling webapp [maven-helloworld] in [E:\maven-helloworld\target\maven-helloworld-0.0.1-SNAP

[INFO] Processing war project

[INFO] Copying webapp resources [E:\maven-helloworld\src\main\webapp]

[INFO] Webapp assembled in [83 msecs]

[INFO] Building war: E:\maven-helloworld\target\maven-helloworld-0.0.1-SNAPSHOT.war

[INFO] WEB-INF\web.xml already added, skipping

[INFO]

[INFO] BUILD SUCCESS

[INFO]

[INFO] Total time: 21.230 s

[INFO] Finished at: 2018-06-14T10:54:22+08:00

[INFO] Final Memory: 15M/142M

[INFO]

对maven项目进行打包，如果是普通的java项目，则打包成jar，如果是web项目，则打包为war包

2.5.6) mvn install:把项目发布到本地仓库：


```
[INFO] Installing E:\maven-helloworld\target\maven-helloworld-0.0.1-SNAPSHOT.war to E:\myrepository\cn\itcast\maven-helloworld-0.0.1-SNAPSHOT.war
[INFO] Installing E:\maven-helloworld\pom.xml to E:\myrepository\cn\itcast\maven\maven-helloworld\0.0.1-SNAPSHOT\pom.xml
[INFO] BUILD SUCCESS
[INFO] mvn install:代表将本地生成的jar包或war包发布到maven的本地仓库中。
[INFO] Total time: 36.832 s
[INFO] Finished at: 2018-06-14T10:58:21+08:00
[INFO] Final Memory: 15M/136M
[INFO]
```

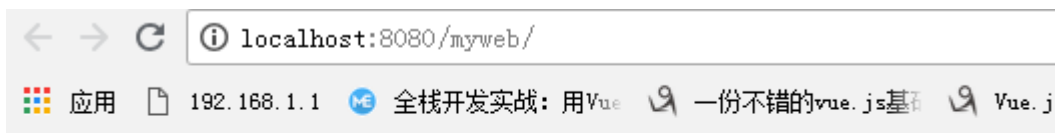
第六步：新建java web项目

```
1 mvn archetype:generate -DgroupId=com.zelin -DartifactId=myweb -DarchetypeArtifactId=maven-archetype-webapp
```

2.5.7) mvn tomcat:run运行web项目：

```
1 mvn tomcat:run
```

运行效果如下：



Hello World!

2.6) Maven生命周期：

Compile test package install deploy (发布到私服)

三种生命周期

Clean生命周期 (一般了解)

Clean

Default生命周期 (了解)

Compile test package install deploy

Site生命周期 (非常一般了解)

Site

2.7) Maven生命周期与各个命令之间的关系：

1) mvn clean package ：在这个命令过程中，会执行如下命令：

Clean:清理maven项目中的字节码文件

Compile 编译项目

Test 测试项目

Package 打包项目

2) mvn clean install:在这个命令过程中，会执行如下命令：

Clean:清理

Compile:编译

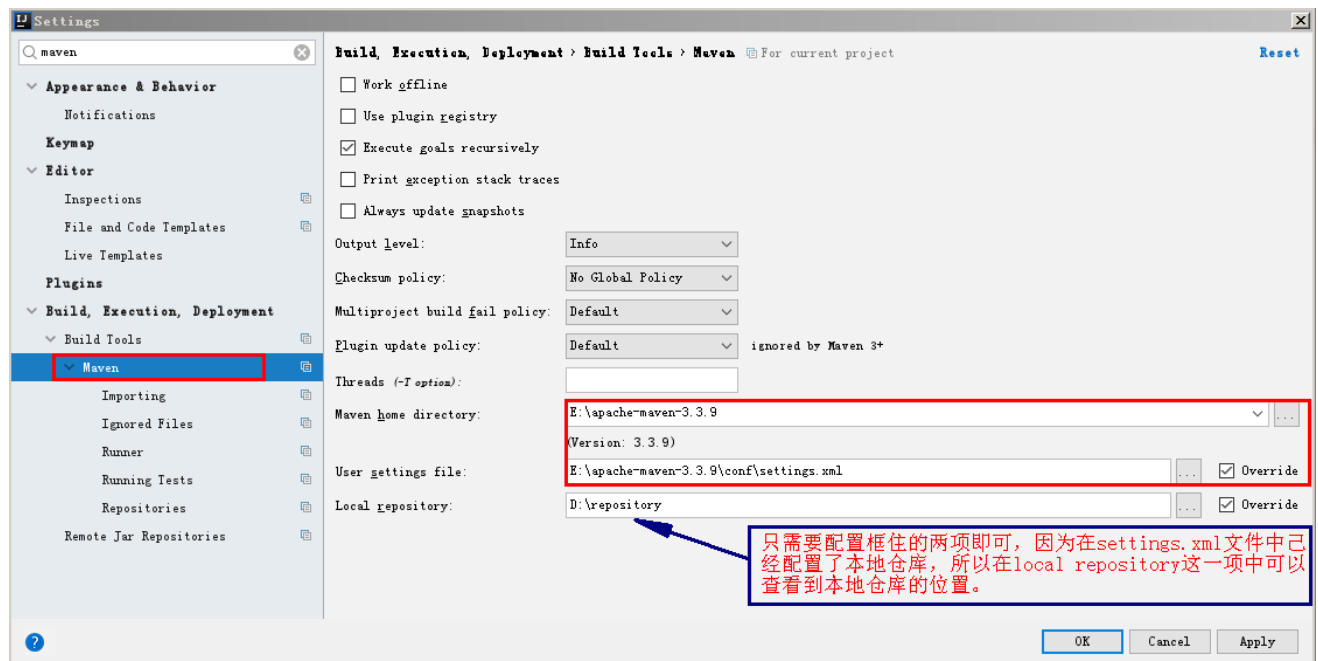
Test:测试

Package:打包

Install : 安装

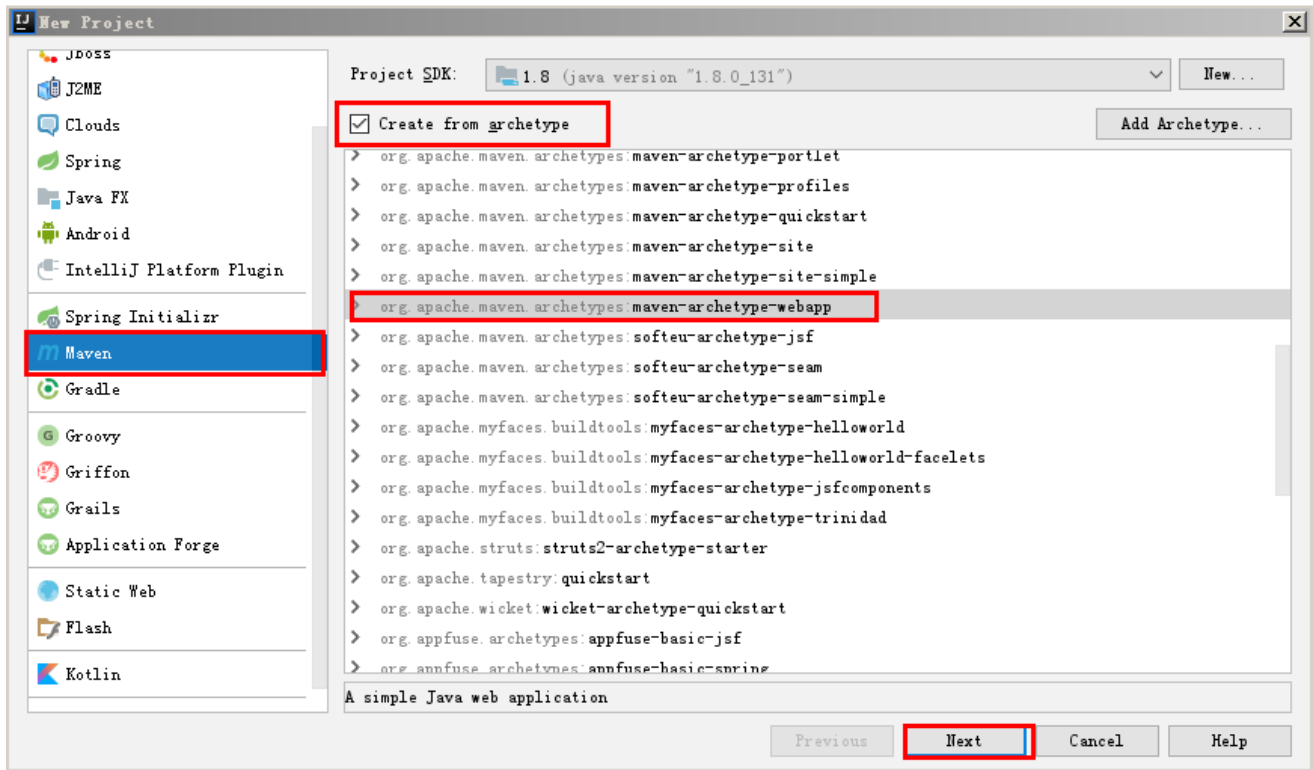
3、 在Idea中配置Maven

3.1) 在Idea中的settings中配置Maven



4、 在Idea中整合SSM + Maven + AngularJS

4.1)选择新建项目：



4.2)新建项目完成后，还要将项目的目录结构改成如下形式：

MavenSSMAngular C:\Users\Administrator.USER-2

> .idea

> src

> main

resource

> webapp

> WEB-INF

index.jsp

MavenSSMAngular.iml

m pom.xml

External Libraries

1 <?xml version="1.0" encodi

2

3 <project xmlns="http://mav

New

Cut Ctrl+X

Copy Ctrl+C

Copy Path Ctrl+Shift+C

Copy Relative Path Ctrl+Alt+Shift+C

Paste Ctrl+V

Find Usages Alt+F7

Find in Path... Ctrl+Shift+F

Replace in Path... Ctrl+Shift+R

Analyze

Refactor

Show Image Thumbnails Ctrl+Shift+T

Reformat Code Ctrl+Alt+L

Optimize Imports Ctrl+Alt+O

Delete... Delete

Build Module 'MavenSSMAngular'

Local History

Synchronize 'resource'

Show in Explorer

Directory Path Ctrl+Alt+F12

Compare With... Ctrl+D

Mark Directory as

SchemaLocation="http

Version>4.0.0</mode

Id>com.zelin</group

actId>Maven-SSM-Angu

on>1.0-SNAPSHOT</ver

ging>war</packaging

Maven-SSM-Angular Ma

XME change it to

Sources Root

Test Sources Root

Resources Root

Test Resources Root

Excluded

Generated Sources Root

essages Maven Goal

[INFO] -----

[INFO] Total time: 01:

[INFO] Finished at: 20

[INFO] Final Memory: 1

[INFO] -----

[INFO] Maven execution

MavenSSMAngular C:\Users\Administrator.USER-20161129NL\IdeaProjects\MavenSSMAngular

1 <?xml version="1.0" encoding="UTF-8" ?>

2

3 <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/maven-v4_0_0.xsd">

java

resource

webapp

WEB-INF

index.jsp

MavenSSMAngular.iml

pom.xml

External Libraries

Messages Maven Goal

[INFO] -----

[INFO] Total time: 0.001 s

[INFO] Finished at: 2016-11-29 14:00:00

[INFO] Final Memory: 0.001 MB

[INFO] -----

[INFO] MavenSSMAngular:war:compile: OK

Build Module 'MavenSSMAngular'

Local History

Synchronize 'java'

Show in Explorer

Directory Path Ctrl+Alt+F12

Compare With... Ctrl+D

Mark Directory as

Sources Root

Test Sources Root

Resources Root

Test Resources Root

Excluded

Generated Sources Root

FIXME change it

MavenSSMAngular C:\Users\Administrator.USER-20161129NL\IdeaProjects\MavenSSMAngular

.idea

src

main

java

resource

db

mapper

spring

applicationContext.xml

applicationContext-mvc.xml

log4j.properties

webapp

plugins

WEB-INF

web.xml

index.html

index.jsp

MavenSSMAngular.iml

pom.xml

注意：

1. 如上图所示，需要在src/main下新建java及resource目录，分别代表存放java源程序及资源文件的位置，并且，java要设置为 sourceRoot，而resource要设置为resourceRoot。
2. 将原来项目中的src目录下的所有的包复制到java目录下，而将原来的resource目录下的内容复制到resource这个目录下。
3. 如果想要进行单元测试的话，还需要添加src/test目录，也要在其中添加java与resource目录，代表存放单元测试的java代码及资源。
4. web.xml文件的内容也与上一章的整合案例一致，其它配置文件也一样。
5. 下面是关于pom.xml文件中的内容。

```
1 <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">
2   <modelVersion>4.0.0</modelVersion>
3   <groupId>com.zelin</groupId>
4   <artifactId>Maven-SSM-Angular</artifactId>
5   <version>0.0.1-SNAPSHOT</version>
6   <packaging>war</packaging>
7
8   <!-- 集中定义依赖版本号 -->
9   <properties>
10    <junit.version>4.12</junit.version>
11    <spring.version>4.2.4.RELEASE</spring.version>
12    <mybatis.version>3.2.8</mybatis.version>
13    <mybatis.spring.version>1.2.2</mybatis.spring.version>
14    <mybatis.paginator.version>1.2.15</mybatis.paginator.version>
15    <mysql.version>5.1.32</mysql.version>
16    <slf4j.version>1.6.4</slf4j.version>
17    <jackson.version>2.4.2</jackson.version>
18    <druid.version>1.0.9</druid.version>
19    <jstl.version>1.2</jstl.version>
20    <servlet-api.version>2.5</servlet-api.version>
21    <jsp-api.version>2.0</jsp-api.version>
22    <joda-time.version>2.5</joda-time.version>
23    <commons-lang3.version>3.3.2</commons-lang3.version>
24    <commons-io.version>1.3.2</commons-io.version>
25    <commons-net.version>3.3</commons-net.version>
26    <pagehelper.version>3.4.2-fix</pagehelper.version>
27    <jsqlparser.version>0.9.1</jsqlparser.version>
28    <commons-fileupload.version>1.3.1</commons-fileupload.version>
29    <quartz.version>2.2.2</quartz.version>
30  </properties>
31
32  <dependencies>
33    <!-- 时间操作组件 -->
34    <dependency>
35      <groupId>joda-time</groupId>
36      <artifactId>joda-time</artifactId>
```

```
37     <version>${joda-time.version}</version>
38 </dependency>
39 <!-- Apache工具组件 -->
40 <dependency>
41     <groupId>org.apache.commons</groupId>
42     <artifactId>commons-lang3</artifactId>
43     <version>${commons-lang3.version}</version>
44 </dependency>
45 <dependency>
46     <groupId>org.apache.commons</groupId>
47     <artifactId>commons-io</artifactId>
48     <version>${commons-io.version}</version>
49 </dependency>
50 <dependency>
51     <groupId>commons-net</groupId>
52     <artifactId>commons-net</artifactId>
53     <version>${commons-net.version}</version>
54 </dependency>
55 <!-- Jackson Json处理工具包 -->
56 <dependency>
57     <groupId>com.fasterxml.jackson.core</groupId>
58     <artifactId>jackson-databind</artifactId>
59     <version>${jackson.version}</version>
60 </dependency>
61
62 <!-- quartz任务调度框架 -->
63 <dependency>
64     <groupId>org.quartz-scheduler</groupId>
65     <artifactId>quartz</artifactId>
66     <version>${quartz.version}</version>
67 </dependency>
68 <!-- 单元测试 -->
69 <dependency>
70     <groupId>junit</groupId>
71     <artifactId>junit</artifactId>
72     <version>${junit.version}</version>
73     <scope>test</scope>
74 </dependency>
75 <!-- 日志处理 -->
76 <dependency>
77     <groupId>org.slf4j</groupId>
78     <artifactId>slf4j-log4j12</artifactId>
79     <version>${slf4j.version}</version>
80 </dependency>
81 <!-- Mybatis -->
82 <dependency>
83     <groupId>org.mybatis</groupId>
84     <artifactId>mybatis</artifactId>
85     <version>${mybatis.version}</version>
86 </dependency>
87 <dependency>
88     <groupId>org.mybatis</groupId>
89     <artifactId>mybatis-spring</artifactId>
```

```
90     <version>${mybatis.spring.version}</version>
91 </dependency>
92 <dependency>
93     <groupId>com.github.miemiedev</groupId>
94     <artifactId>mybatis-paginator</artifactId>
95     <version>${mybatis.paginator.version}</version>
96 </dependency>
97 <dependency>
98     <groupId>com.github.pagehelper</groupId>
99     <artifactId>pagehelper</artifactId>
100     <version>${pagehelper.version}</version>
101 </dependency>
102 <!-- MySql -->
103 <dependency>
104     <groupId>mysql</groupId>
105     <artifactId>mysql-connector-java</artifactId>
106     <version>${mysql.version}</version>
107 </dependency>
108 <!-- 连接池 -->
109 <dependency>
110     <groupId>com.alibaba</groupId>
111     <artifactId>druid</artifactId>
112     <version>${druid.version}</version>
113 </dependency>
114 <!-- Spring -->
115 <dependency>
116     <groupId>org.springframework</groupId>
117     <artifactId>spring-context</artifactId>
118     <version>${spring.version}</version>
119 </dependency>
120 <dependency>
121     <groupId>org.springframework</groupId>
122     <artifactId>spring-beans</artifactId>
123     <version>${spring.version}</version>
124 </dependency>
125 <dependency>
126     <groupId>org.springframework</groupId>
127     <artifactId>spring-webmvc</artifactId>
128     <version>${spring.version}</version>
129 </dependency>
130 <dependency>
131     <groupId>org.springframework</groupId>
132     <artifactId>spring-jdbc</artifactId>
133     <version>${spring.version}</version>
134 </dependency>
135 <dependency>
136     <groupId>org.springframework</groupId>
137     <artifactId>spring-aspects</artifactId>
138     <version>${spring.version}</version>
139 </dependency>
140 <dependency>
141     <groupId>org.springframework</groupId>
142     <artifactId>spring-jms</artifactId>
```

```
143     <version>${spring.version}</version>
144 </dependency>
145 <dependency>
146     <groupId>org.springframework</groupId>
147     <artifactId>spring-context-support</artifactId>
148     <version>${spring.version}</version>
149 </dependency>
150 <!-- JSP相关 -->
151 <dependency>
152     <groupId>jstl</groupId>
153     <artifactId>jstl</artifactId>
154     <version>${jstl.version}</version>
155 </dependency>
156 <dependency>
157     <groupId>javax.servlet</groupId>
158     <artifactId>servlet-api</artifactId>
159     <version>${servlet-api.version}</version>
160     <scope>provided</scope>
161 </dependency>
162 <dependency>
163     <groupId>javax.servlet</groupId>
164     <artifactId>jsp-api</artifactId>
165     <version>${jsp-api.version}</version>
166     <scope>provided</scope>
167 </dependency>
168 <!-- 文件上传组件 -->
169 <dependency>
170     <groupId>commons-fileupload</groupId>
171     <artifactId>commons-fileupload</artifactId>
172     <version>${commons-fileupload.version}</version>
173 </dependency>
174
175
176 </dependencies>
177 <build>
178     <finalName>${project.artifactId}</finalName>
179     <plugins>
180         <!-- 资源文件拷贝插件 -->
181         <plugin>
182             <groupId>org.apache.maven.plugins</groupId>
183             <artifactId>maven-resources-plugin</artifactId>
184             <version>2.7</version>
185             <configuration>
186                 <encoding>UTF-8</encoding>
187             </configuration>
188         </plugin>
189         <!-- java编译插件（指定编译版本） -->
190         <plugin>
191             <groupId>org.apache.maven.plugins</groupId>
192             <artifactId>maven-compiler-plugin</artifactId>
193             <version>3.2</version>
194             <configuration>
195                 <source>1.7</source>
```

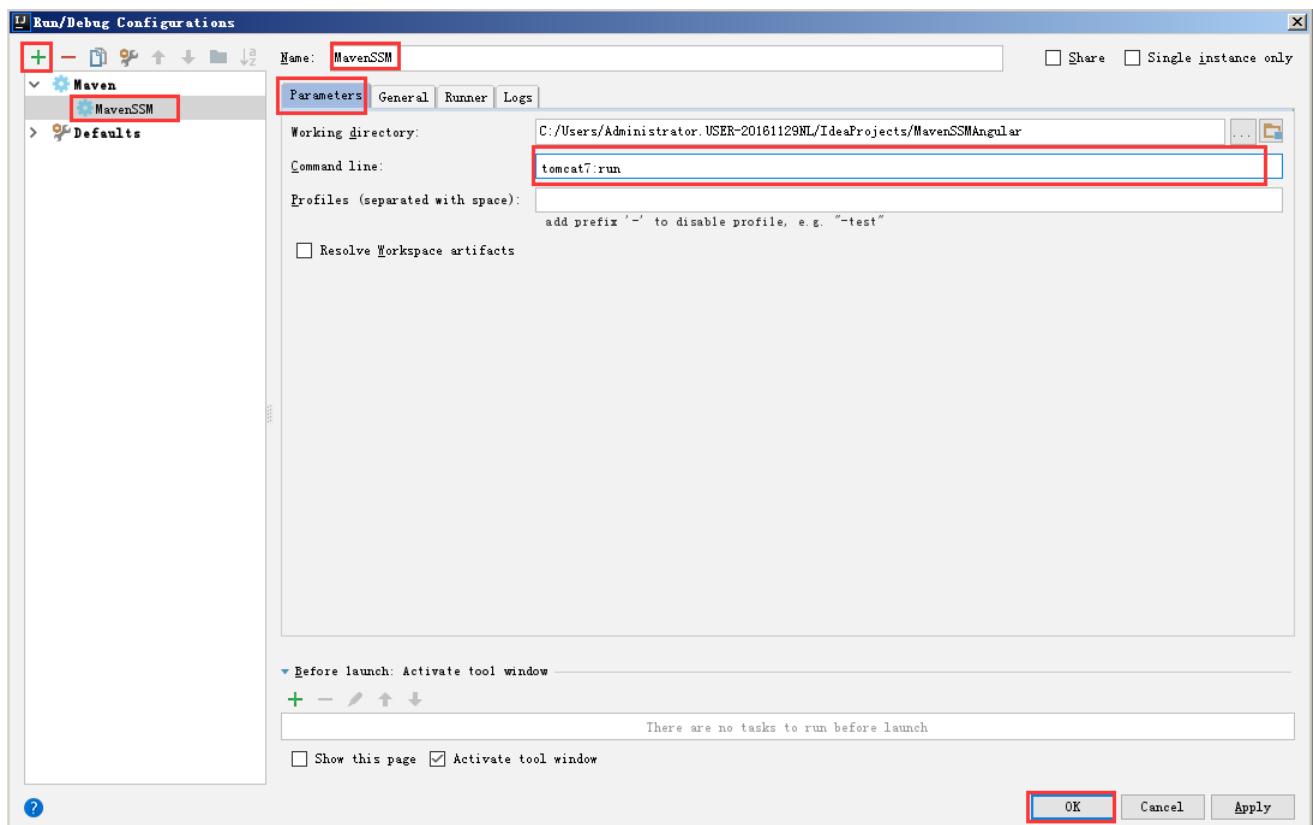


```

196         <target>1.7</target>
197         <encoding>UTF-8</encoding>
198     </configuration>
199 </plugin>
200 </plugins>
201 <pluginManagement>
202     <plugins>
203         <!-- 配置Tomcat插件 -->
204         <plugin>
205             <groupId>org.apache.tomcat.maven</groupId>
206             <artifactId>tomcat7-maven-plugin</artifactId>
207             <version>2.2</version>
208             <configuration>
209                 <path>/</path>
210                 <contextReloadable>true</contextReloadable>
211             </configuration>
212         </plugin>
213     </plugins>
214 </pluginManagement>
215 </build>
216 </project>

```

4.3) 配置Maven运行环境：



4.4) 复制原来项目中的resource目录下的所有的内容到src/main/resource目录下

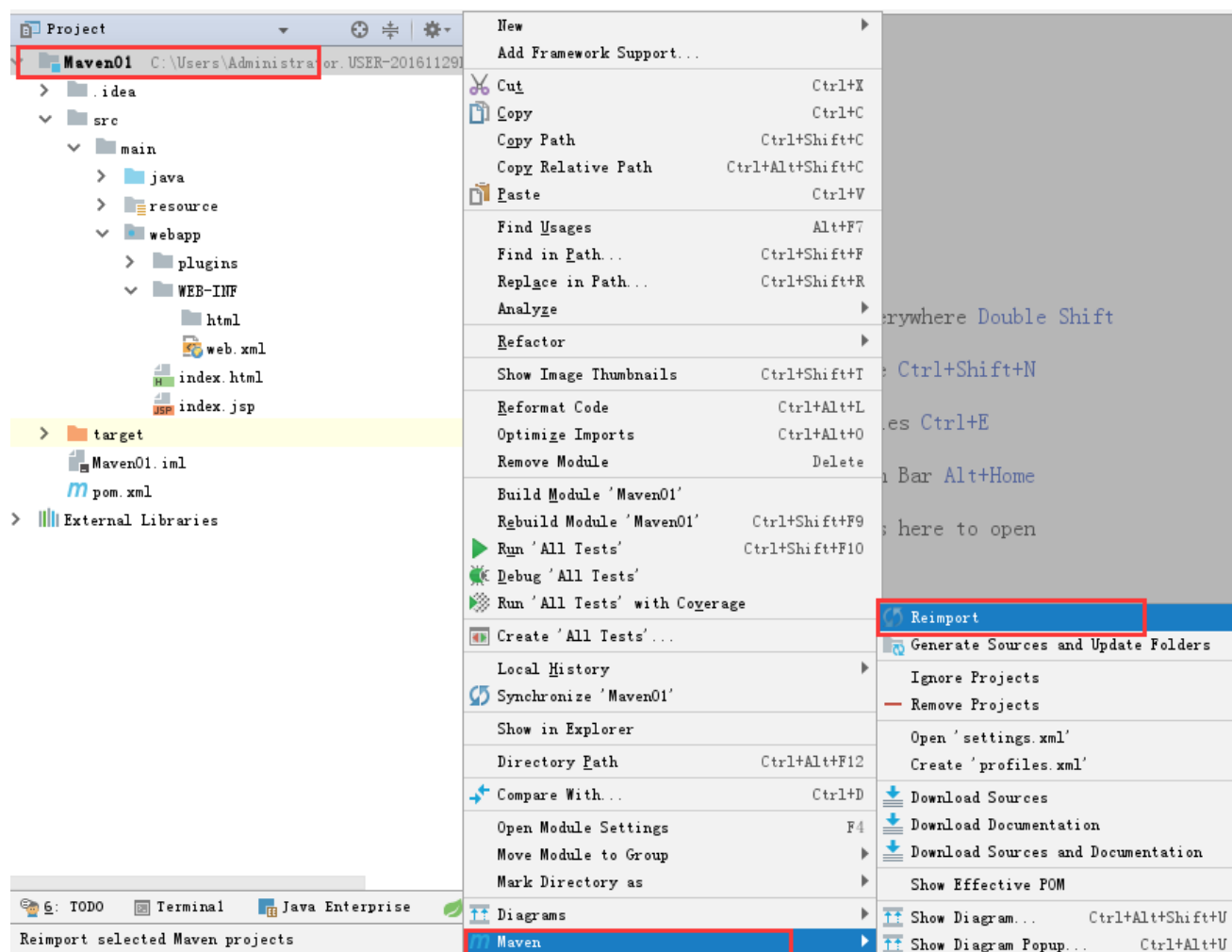
4.5) 复制原来项目中的src下的所有的包到src/main/java目录下

4.6) 复制原来项目中的web/WEB-INF/web.xml复制到src/webapp/WEB-INF/目录下，覆盖原文件即可。

4.7) 复制/web/plugins目录下的所有内容到src/webapp/目录下。

4.8) 将index.html文件也复制到src/webapp/目录下

4.9) 重新导入项目：



4.10) 运行效果如下：

学生管理系统

学号	姓名	性别	年龄	住址	生日	所在班级	操作
1	张小三	男	25	广州123	2018-9-6	1302班	修改 删除
2	王小五	男	28	广州	2010-10-27	1301班	修改 删除
3	李小红	女	190	杭州	2008-8-21	1302班	修改 删除
4	王二小	男	12	岳阳	2010-2-16	1302班	修改 删除
5	张小二	男	28	上海123	2018-10-30	1303班	修改 删除

« 1 2 3 4 5 » 第 1 页 每页 5 条 /共21条

[添加学生](#) 请选择班级： [查询](#)

泽林信息版权所有 2000-2018.