

第二章 Maven项目管理

-Maven(二)2018/11/17 [泽林.王峰]

授课目标

- 1、上章回顾
- 2、Maven中的传递依赖
- 3、Maven分模块开发（非常重要）
- 4、私服（nexus）搭建（了解）

授课内容

1、 上章回顾

2、 Maven中的传递依赖

2.1) 关于引入的jar中出现版本冲突的解决方案：

方案一：第一优先原则，即谁放到前面就以谁为准。

```
1      <!--解决jar包中内置包冲突方案一：第一声明优先原则，即谁放到前面，就以谁的版本为主，
2          这种情况的版本实际上就是3.0.1版本了
3      -->
4      <dependency>
5          <groupId>org.springframework</groupId>
6          <artifactId>spring-struts</artifactId>
7          <version>3.0.1.RELEASE</version>
8      </dependency>
9      <dependency>
10         <groupId>org.springframework</groupId>
11         <artifactId>spring-context</artifactId>
12         <version>4.2.4.RELEASE</version>
13     </dependency>
```

方案二：路径近者优先原则，即将想依赖的包的对应的版本直接引入即可。

```
1      <!--解决版本冲突的第二种方案，路径近者优先原则，即将想要用的哪个包的对应版本直接引入即可，下面
2          这种情况
3          spring-beans的版本就是4.2.4了
4      -->
5      <dependency>
```

```

5      <groupId>org.springframework</groupId>
6      <artifactId>spring-struts</artifactId>
7      <version>3.0.1.RELEASE</version>
8  </dependency>
9  <dependency>
10     <groupId>org.springframework</groupId>
11     <artifactId>spring-context</artifactId>
12     <version>4.2.4.RELEASE</version>
13 </dependency>
14 <dependency>
15     <groupId>org.springframework</groupId>
16     <artifactId>spring-beans</artifactId>
17     <version>4.2.4.RELEASE</version>
18 </dependency>

```

方案三：排除不想要的内置版本的包，余下都是我们想要的

```

1  <!--解决jar包中内置包冲突方案三：直接在不想要的引赖包中排除相关的内层的
2      jar包版本即可，本例中直接排除的就是3.0.1RELEASE这个版本中自带的spring-beans
3      经过排除后，最后的spring-beans的版本就是4.2.4
4  -->
5  <dependency>
6      <groupId>org.springframework</groupId>
7      <artifactId>spring-struts</artifactId>
8      <version>3.0.1.RELEASE</version>
9      <exclusions>
10         <exclusion>
11             <groupId>org.springframework</groupId>
12             <artifactId>spring-beans</artifactId>
13         </exclusion>
14     </exclusions>
15 </dependency>
16 <dependency>
17     <groupId>org.springframework</groupId>
18     <artifactId>spring-context</artifactId>
19     <version>4.2.4.RELEASE</version>
20 </dependency>

```

方案四：版本锁定原则(开发中经常使用)

```

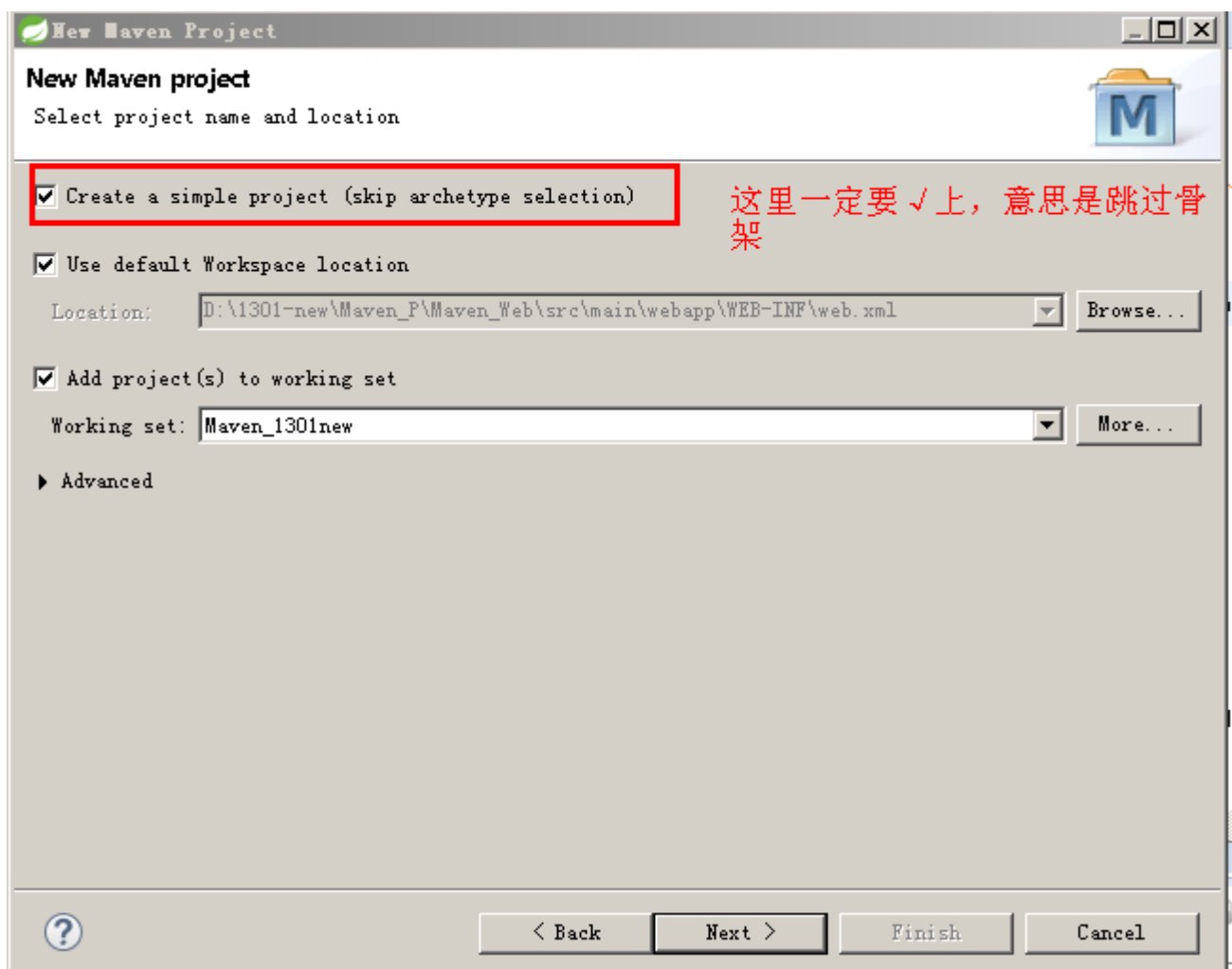
1  <!--解决jar包中内置包冲突方案四：对有冲突的jar包直接锁定版本号-->
2  <dependencyManagement>
3      <dependencies>
4          <dependency>
5              <groupId>org.springframework</groupId>
6              <artifactId>spring-beans</artifactId>
7              <version>4.2.4.RELEASE</version>
8          </dependency>
9      </dependencies>
10 </dependencyManagement>
11
12 <dependencies>

```

```
13     <dependency>
14         <groupId>org.springframework</groupId>
15         <artifactId>spring-struts</artifactId>
16         <version>3.0.1.RELEASE</version>
17     </dependency>
18     <dependency>
19         <groupId>org.springframework</groupId>
20         <artifactId>spring-context</artifactId>
21         <version>4.2.4.RELEASE</version>
22     </dependency>
23
```

3、 Maven分模块开发（非常重要）

3.1)新建父模块（聚合工程，在新建maven工程时选择pom类型）



New Maven Project

Enter a group id for the artifact.

Artifact

Group Id:

Artifact Id:

Version: 0.0.1-SNAPSHOT

Packaging: pom

Name:

Description:

Parent Project

Group Id:

Artifact Id:

Version: Browse... Clear

Advanced

< Back Next > Finish Cancel

这里选择pom, 代表创建的是聚合父工程

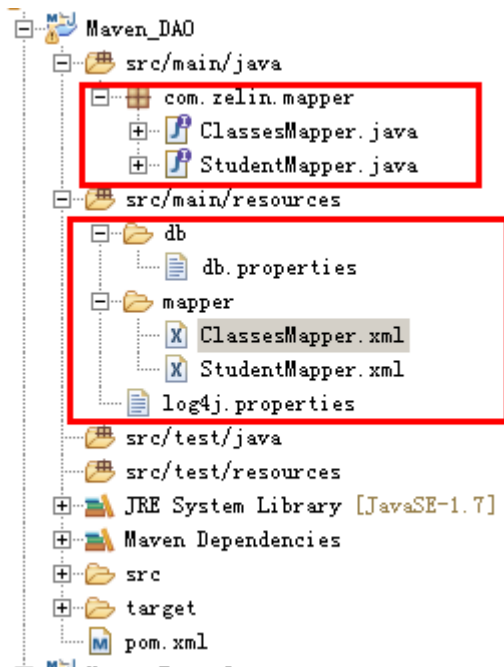
3.2)用同样的方法在Maven-P工程下新建Maven-pojo模块(专门用于放实体类), 是一个jar工程

3.3)用同样的方法在Maven-P工程下新建Maven-dao模块(专门用于放dao, 即Mybatis相关的内容), 其中的pom.xml文件:

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
3     <modelVersion>4.0.0</modelVersion>
4     <parent>
5         <groupId>com.zelin</groupId>
6         <artifactId>Maven_P</artifactId>
7         <version>0.0.1-SNAPSHOT</version>
8     </parent>
9     <artifactId>Maven_DAO</artifactId>
10    <dependencies>
11
12        <!-- 日志处理 -->
13        <dependency>
14            <groupId>org.slf4j</groupId>
15            <artifactId>slf4j-log4j12</artifactId>
```

```
16     </dependency>
17     <!-- Mybatis -->
18     <dependency>
19         <groupId>org.mybatis</groupId>
20         <artifactId>mybatis</artifactId>
21     </dependency>
22     <dependency>
23         <groupId>org.mybatis</groupId>
24         <artifactId>mybatis-spring</artifactId>
25     </dependency>
26     <dependency>
27         <groupId>com.github.miemiedev</groupId>
28         <artifactId>mybatis-paginator</artifactId>
29     </dependency>
30     <dependency>
31         <groupId>com.github.pagehelper</groupId>
32         <artifactId>pagehelper</artifactId>
33     </dependency>
34     <!-- MySql -->
35     <dependency>
36         <groupId>mysql</groupId>
37         <artifactId>mysql-connector-java</artifactId>
38     </dependency>
39     <!-- 连接池 -->
40     <dependency>
41         <groupId>com.alibaba</groupId>
42         <artifactId>druid</artifactId>
43     </dependency>
44
45     <!-- 添加对pojo的依赖 -->
46     <dependency>
47         <groupId>com.zelin</groupId>
48         <artifactId>Maven_POJO</artifactId>
49         <version>0.0.1-SNAPSHOT</version>
50     </dependency>
51 </dependencies>
52 </project>
```

3.4)将访问数据库相关的内容添加到Maven-dao工程中



在dao模块的pom.xml文件内容如下：

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4       http://maven.apache.org/xsd/maven-4.0.0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6   <parent>
7     <groupId>com.zelin</groupId>
8     <artifactId>Maven_P</artifactId>
9     <version>0.0.1-SNAPSHOT</version>
10  </parent>
11  <artifactId>Maven_DAO</artifactId>
12  <dependencies>
13
14    <!-- 日志处理 -->
15    <dependency>
16      <groupId>org.slf4j</groupId>
17      <artifactId>slf4j-log4j12</artifactId>
18    </dependency>
19    <!-- Mybatis -->
20    <dependency>
21      <groupId>org.mybatis</groupId>
22      <artifactId>mybatis</artifactId>
23    </dependency>
24    <dependency>
25      <groupId>org.mybatis</groupId>
26      <artifactId>mybatis-spring</artifactId>
27    </dependency>
28    <dependency>
29      <groupId>com.github.miemiedev</groupId>
30      <artifactId>mybatis-paginator</artifactId>
31    </dependency>
32  </dependencies>
```

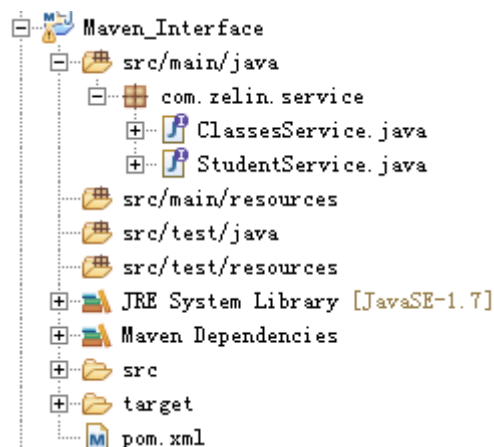
```

31         <groupId>com.github.pagehelper</groupId>
32         <artifactId>pagehelper</artifactId>
33     </dependency>
34     <!-- MySql -->
35     <dependency>
36         <groupId>mysql</groupId>
37         <artifactId>mysql-connector-java</artifactId>
38     </dependency>
39     <!-- 连接池 -->
40     <dependency>
41         <groupId>com.alibaba</groupId>
42         <artifactId>druid</artifactId>
43     </dependency>
44
45     <!-- 添加对pojo的依赖 -->
46     <dependency>
47         <groupId>com.zelin</groupId>
48         <artifactId>Maven_POJO</artifactId>
49         <version>0.0.1-SNAPSHOT</version>
50     </dependency>
51 </dependencies>
52 </project>

```

注意：因为dao模块要访问数据库，所以，需要依赖pojo模块。

3.5)在Maven-P工程下新建Maven-interface模块，用于定义服务层的接口



3.6)在Maven-P工程下新建Maven-service模块，用于定义服务层的实现：

1. 其中的pom文件的内容如下：

```

1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <parent>
5       <groupId>com.zelin</groupId>
6       <artifactId>Maven_P</artifactId>
7       <version>0.0.1-SNAPSHOT</version>

```

```
8      </parent>
9      <artifactId>Maven_Service</artifactId>
10     <packaging>jar</packaging>
11     <dependencies>
12         <!-- Jackson Json处理工具包 -->
13         <dependency>
14             <groupId>com.fasterxml.jackson.core</groupId>
15             <artifactId>jackson-databind</artifactId>
16         </dependency>
17         <!-- Spring -->
18         <dependency>
19             <groupId>org.springframework</groupId>
20             <artifactId>spring-context</artifactId>
21         </dependency>
22         <dependency>
23             <groupId>org.springframework</groupId>
24             <artifactId>spring-beans</artifactId>
25         </dependency>
26         <dependency>
27             <groupId>org.springframework</groupId>
28             <artifactId>spring-webmvc</artifactId>
29         </dependency>
30         <dependency>
31             <groupId>org.springframework</groupId>
32             <artifactId>spring-jdbc</artifactId>
33         </dependency>
34         <dependency>
35             <groupId>org.springframework</groupId>
36             <artifactId>spring-aspects</artifactId>
37         </dependency>
38         <dependency>
39             <groupId>org.springframework</groupId>
40             <artifactId>spring-jms</artifactId>
41         </dependency>
42         <dependency>
43             <groupId>org.springframework</groupId>
44             <artifactId>spring-context-support</artifactId>
45         </dependency>
46         <!-- 引入服务层接口 -->
47         <dependency>
48             <groupId>com.zelin</groupId>
49             <artifactId>Maven_Interface</artifactId>
50             <version>0.0.1-SNAPSHOT</version>
51         </dependency>
52         <dependency>
53             <groupId>com.zelin</groupId>
54             <artifactId>Maven_DAO</artifactId>
55             <version>0.0.1-SNAPSHOT</version>
56         </dependency>
57     </dependencies>
58     <!-- 3、添加一些工具插件 -->
59     <build>
60         <finalName>${project.artifactId}</finalName>
```



```

61     <plugins>
62         <!-- 资源文件拷贝插件 -->
63         <plugin>
64             <groupId>org.apache.maven.plugins</groupId>
65             <artifactId>maven-resources-plugin</artifactId>
66             <version>2.7</version>
67             <configuration>
68                 <encoding>UTF-8</encoding>
69             </configuration>
70         </plugin>
71         <!-- java编译插件 -->
72         <plugin>
73             <groupId>org.apache.maven.plugins</groupId>
74             <artifactId>maven-compiler-plugin</artifactId>
75             <version>3.2</version>
76             <configuration>
77                 <source>1.7</source>
78                 <target>1.7</target>
79                 <encoding>UTF-8</encoding>
80             </configuration>
81         </plugin>
82     </plugins>
83 </build>
84
85 </project>

```

注意：在此pom文件中要除了引入spring相关的jar包外，还需要引入pojo工程模块及dao工程模块。

2. 在spring/applicationContext.xml文件中定义

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xmlns:context="http://www.springframework.org/schema/context"
5      xsi:schemaLocation="http://www.springframework.org/schema/beans
6          http://www.springframework.org/schema/beans/spring-beans.xsd
7          http://www.springframework.org/schema/context
8          http://www.springframework.org/schema/context/spring-context-4.2.xsd">
9
10     <!-- 读取属性文件 -->
11     <context:property-placeholder
12         location="classpath*:db/db.properties"/>
13
14     <!-- 配置包扫描器 -->
15     <context:component-scan base-package="com.zelin.service"/>
16
17     <!-- 配置数据源 -->
18     <bean id="dataSource"
19         class="com.alibaba.druid.pool.DruidDataSource">
20         <property name="driverClassName" value="${db.driver}"/>
21         <property name="url" value="${db.url}"/>
22         <property name="username" value="${db.user}"/>
23         <property name="password" value="${db.password}"/>
24     </bean>
25
26     <!-- 配置SqlSessionFactoryBean -->

```

```

20     <bean id="sqlSessionFactoryBean"
class="org.mybatis.spring.SqlSessionFactoryBean">
21         <property name="dataSource" ref="dataSource"/>
22         <property name="typeAliasesPackage" value="com.zelin.pojo"/>
23         <property name="plugins">
24             <list>
25                 <bean class="com.github.pagehelper.PageHelper">
26                     <property name="properties">
27                         <value>
28                             dialect=mysql
29                         </value>
30                     </property>
31                 </bean>
32             </list>
33         </property>
34         <property name="mapperLocations"
value="classpath*:mapper/*.xml"/>
35     </bean>
36     <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
37         <property name="basePackage" value="com.zelin.mapper"/>
38     </bean>
39 </beans>
40

```

3.7)在Maven-P工程下新建Maven-web模块，用于定义web的实现：

1.maven-web模块的pom.xml文件

```

1  <project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
3      <modelVersion>4.0.0</modelVersion>
4      <parent>
5          <groupId>com.zelin</groupId>
6          <artifactId>Maven_P</artifactId>
7          <version>0.0.1-SNAPSHOT</version>
8      </parent>
9      <artifactId>Maven_Web</artifactId>
10     <packaging>war</packaging>
11     <dependencies>
12
13         <!-- JSP相关 -->
14         <dependency>
15             <groupId>jstl</groupId>
16             <artifactId>jstl</artifactId>
17         </dependency>
18         <dependency>
19             <groupId>javax.servlet</groupId>
20             <artifactId>servlet-api</artifactId>

```

```

21         <scope>provided</scope>
22     </dependency>
23     <dependency>
24         <groupId>javax.servlet</groupId>
25         <artifactId>jsp-api</artifactId>
26         <scope>provided</scope>
27     </dependency>
28
29     <dependency>
30         <groupId>com.zelin</groupId>
31         <artifactId>Maven_Service</artifactId>
32         <version>0.0.1-SNAPSHOT</version>
33     </dependency>
34 </dependencies>
35
36 <!-- 3、添加一些工具插件 -->
37 <build>
38     <finalName>${project.artifactId}</finalName>
39     <plugins>
40         <!-- 资源文件拷贝插件 -->
41         <plugin>
42             <groupId>org.apache.maven.plugins</groupId>
43             <artifactId>maven-resources-plugin</artifactId>
44             <version>2.7</version>
45             <configuration>
46                 <encoding>UTF-8</encoding>
47             </configuration>
48         </plugin>
49         <!-- java编译插件 -->
50         <plugin>
51             <groupId>org.apache.maven.plugins</groupId>
52             <artifactId>maven-compiler-plugin</artifactId>
53             <version>3.2</version>
54             <configuration>
55                 <source>1.7</source>
56                 <target>1.7</target>
57                 <encoding>UTF-8</encoding>
58             </configuration>
59         </plugin>
60
61     </plugins>
62     <pluginManagement>
63         <plugins>
64             <!-- 配置Tomcat插件 -->
65             <plugin>
66                 <groupId>org.apache.tomcat.maven</groupId>
67                 <artifactId>tomcat7-maven-plugin</artifactId>
68                 <version>2.2</version>
69                 <configuration>
70                     <port>9100</port>
71                     <path>/</path>
72                     <contextReloadable>true</contextReloadable>
73
74                 </configuration>

```

```

74         </plugin>
75     </plugins>
76 </pluginManagement>
77 </build>
78 </project>

```

说明：maven-web模块除了引用jsp的相关依赖外，还需要引用Maven-service的相关依赖。

2. web.xml文件的配置如下:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     xmlns="http://java.sun.com/xml/ns/javaee"
4     xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
5     version="2.5">
6     <display-name>Maven_Web</display-name>
7     <!-- 1.加载spring所有的配置文件（不包含springmvc的配置文件） -->
8     <context-param>
9         <param-name>contextConfigLocation</param-name>
10        <!-- 注意：classpath与classpath*区别：（1）classpath:代表访问指定类路径
下的资源文件（2）classpath*:代表访问指定类路径及引入的第三方包中的资源文件。 -->
11        <param-value>classpath*:spring/applicationContext*.xml</param-value>
12    </context-param>
13    <!-- 2.配置spring的监听器 -->
14    <listener>
15        <listener-
16class>org.springframework.web.context.ContextLoaderListener</listener-class>
17    </listener>
18    <!-- 3.配置DispatcherServlet -->
19    <servlet>
20        <servlet-name>springmvc</servlet-name>
21        <servlet-
22class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
23        <init-param>
24            <param-name>contextConfigLocation</param-name>
25            <param-value>classpath:spring/applicationContext-mvc.xml</param-
26value>
27        </init-param>
28    </servlet>
29    <servlet-mapping>
30        <servlet-name>springmvc</servlet-name>
31        <url-pattern>*.action</url-pattern>
32    </servlet-mapping>
33    <!-- 4.处理中文乱码的过滤器 -->
34    <filter>
35        <filter-name>characterEncoding</filter-name>
36        <filter-
37class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
38        <init-param>
39            <param-name>encoding</param-name>

```

```

38         <param-value>UTF-8</param-value>
39     </init-param>
40 </filter>
41 <filter-mapping>
42     <filter-name>characterEncoding</filter-name>
43     <url-pattern>/*</url-pattern>
44 </filter-mapping>
45 <welcome-file-list>
46     <welcome-file>index.html</welcome-file>
47     <welcome-file>index.htm</welcome-file>
48     <welcome-file>index.jsp</welcome-file>
49     <welcome-file>default.html</welcome-file>
50     <welcome-file>default.htm</welcome-file>
51     <welcome-file>default.jsp</welcome-file>
52 </welcome-file-list>
53 </web-app>

```

applicationContext-mvc.xml文件的配置如下:

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xmlns:context="http://www.springframework.org/schema/context"
5      xmlns:mvc="http://www.springframework.org/schema/mvc"
6      xsi:schemaLocation="http://www.springframework.org/schema/beans
7          http://www.springframework.org/schema/beans/spring-beans.xsd
8          http://www.springframework.org/schema/context
9          http://www.springframework.org/schema/context/spring-context-4.2.xsd
10         http://www.springframework.org/schema/mvc
11         http://www.springframework.org/schema/mvc/spring-mvc-4.2.xsd">
12
13      <context:component-scan base-package="com.zelin.controller"/>
14      <mvc:annotation-driven/>
15
16 </beans>

```

3.8) 程序最终运行效果如下：

学生管理系统							
学号	姓名	性别	年龄	住址	生日	所在班级	操作
2	王小五	男	28	广州	2010-10-27	1301班	修改 删除
37	余小明	男	20	上海	2000-1-1	1301班	修改 删除

« 1 » 第 1 页 每页 5 条 /共2条

添加学生 小 输入住址关键字 请选择班级: 1301班 查询

泽林信息版权所有 2000-2018.

4、私服 (nexus) 搭建 (了解)

