# 第11章：SSM框架整合

## ssm整合

### 步骤

1.添加ssm整合jar文件

2.添加mybatis配置文件

3.添加spring配置文件

主配置文件applicationContext.xml

声明式事务applicationContext-tx.xml

4.在web.xml文件加载spring配置文件

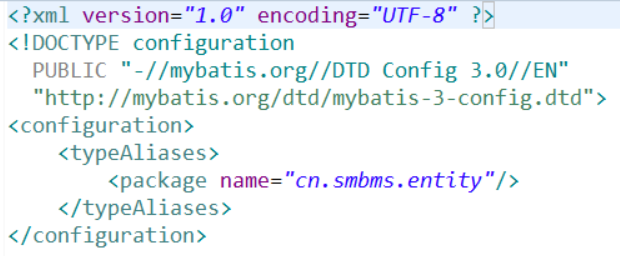
5.添加springmvc配置文件

6.在web.xml文件加载springmvc配置文件

7.创建包结构，检查各种配置文件的包结构路径

### 添加ssm整合jar文件

### 添加mybatis配置文件



### 添加spring配置文件

主配置文件applicationContext.xml

声明式事务applicationContext-tx.xml

### 在web.xml文件加载spring配置文件



### 添加springmvc配置文件



### 在web.xml文件加载springmvc配置文件



### 创建包结构，检查各种配置文件的包结构路径

### 在web.xml配置文件过滤器配置



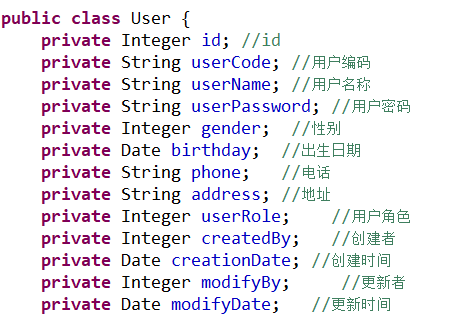
## 加载静态资源

Springmvc静态资源

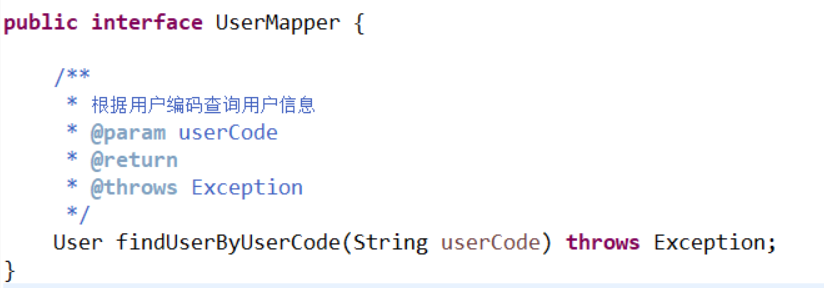


## 登录

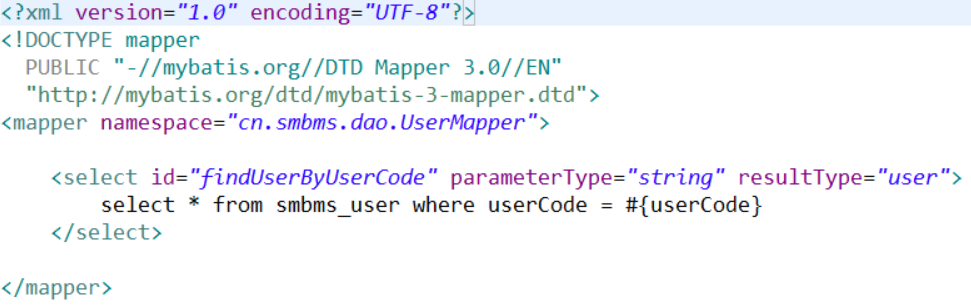
### 实体类



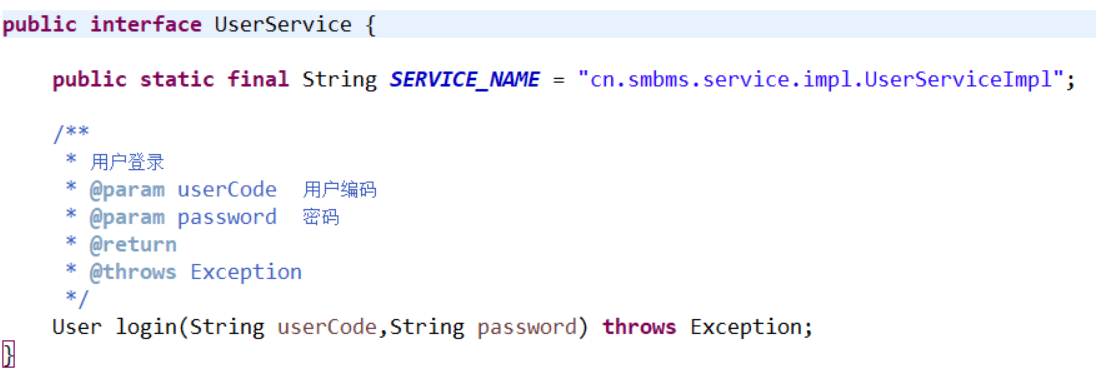
### UserMapper接口



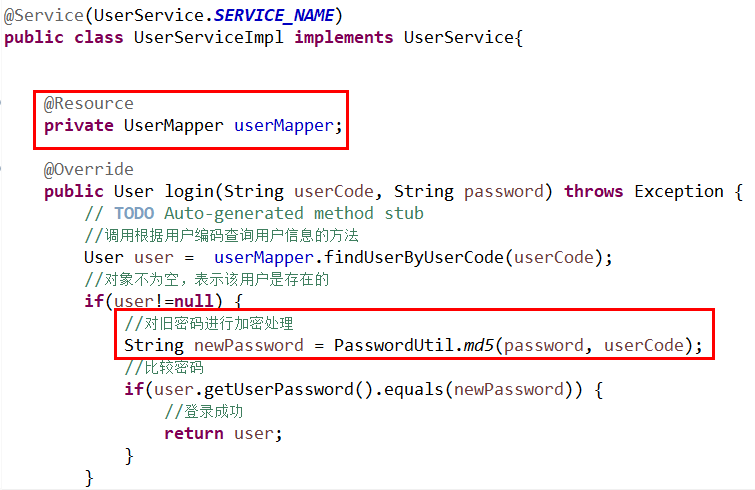
### 映射文件



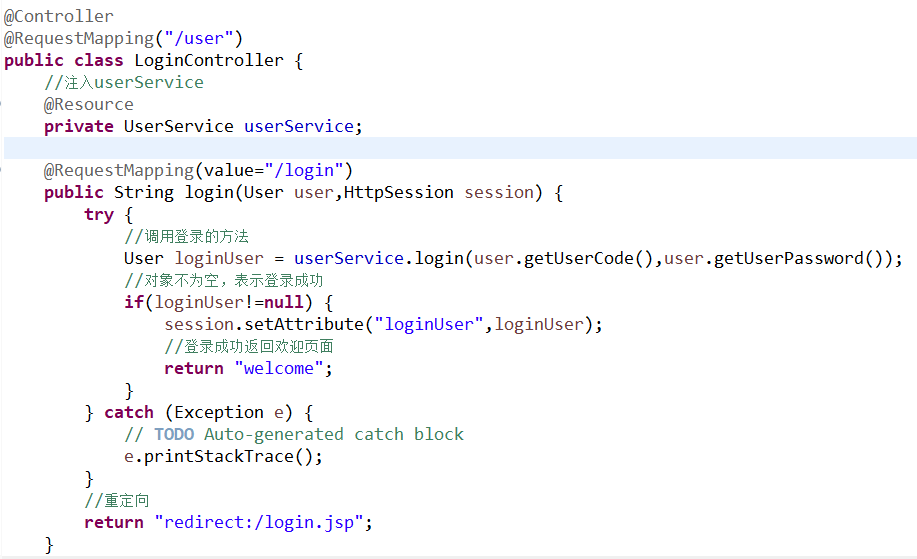
### UserService业务层



### UserService业务层实现类



### 控制器



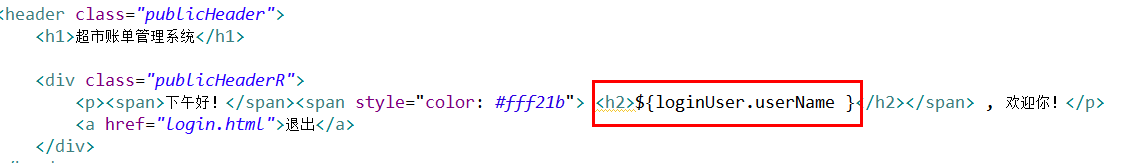
### 登录页面



### 欢迎页面

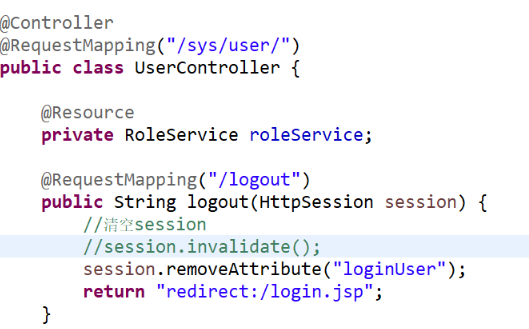


### 头部head页面



## 注销

### Controller

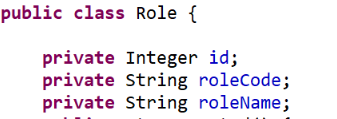


### Head页面

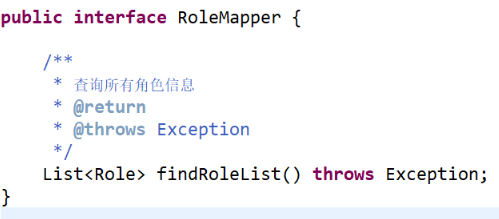


## 用户管理页面角色列表查询

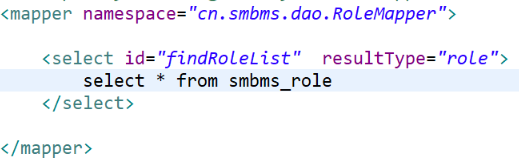
### 实体类



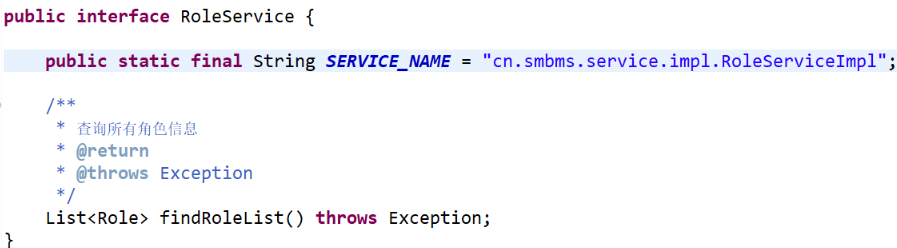
### Mapper接口



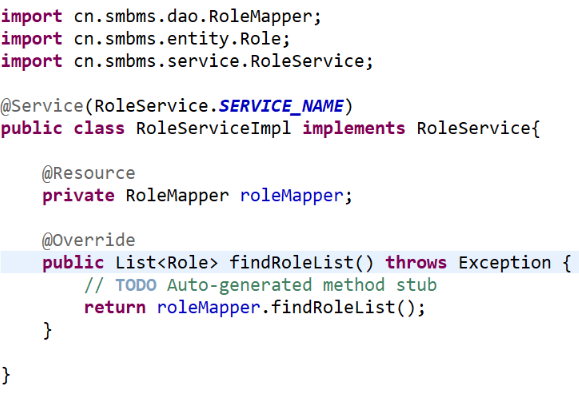
### Mapper映射文件



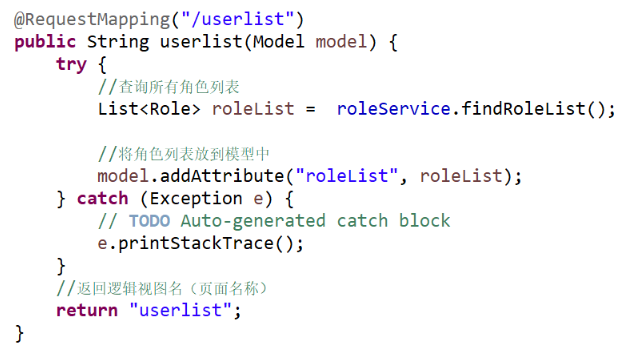
### 业务层接口



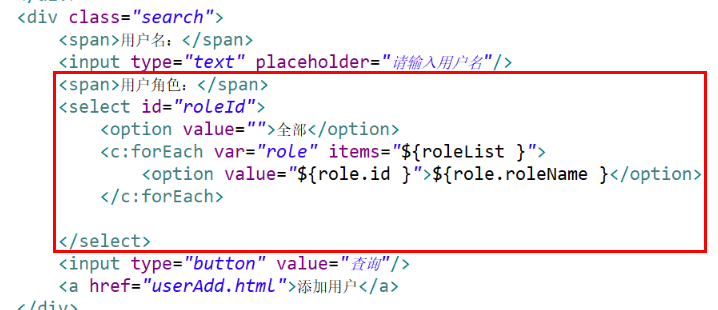
### 业务层实现类



### UserController

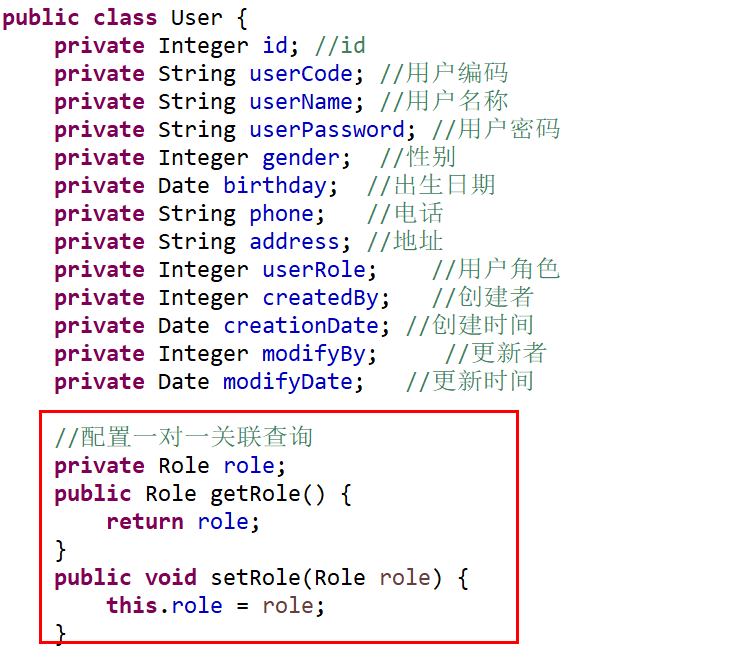


### Userlist页面

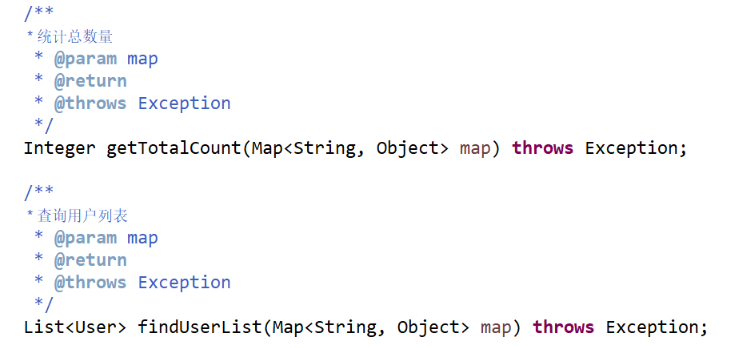


## 用户列表查询

### 实体类

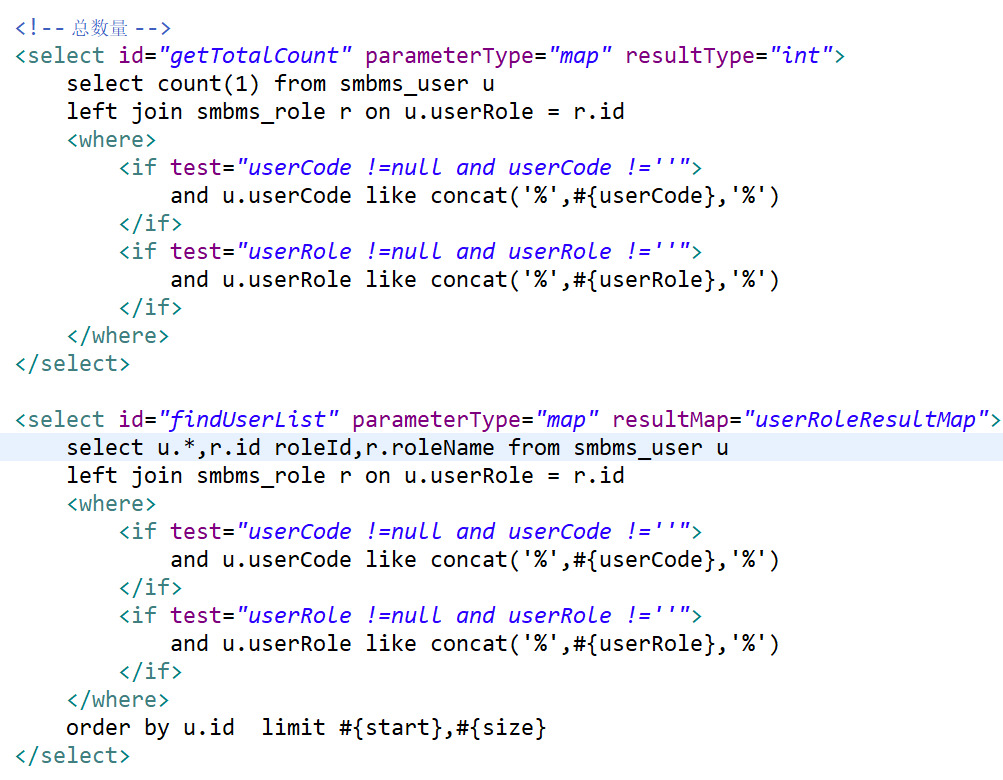


### Mapper接口

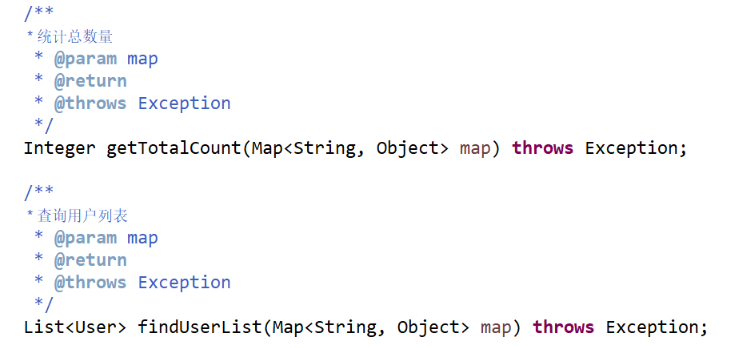


### Mapper映射文件

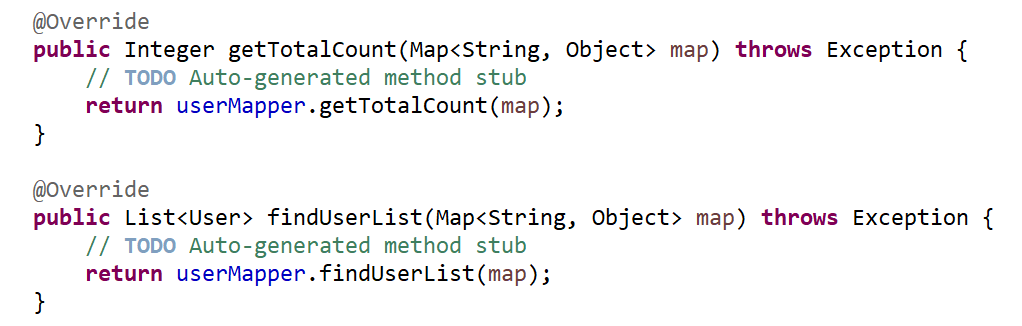




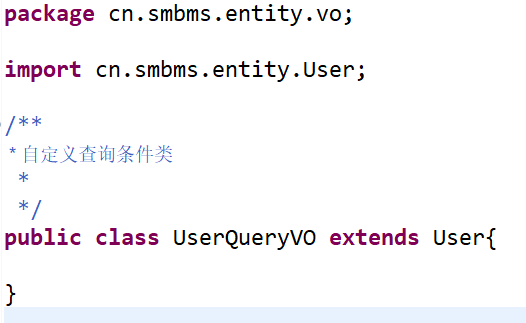
### 业务层接口



### 业务层实现类



### 创建自定义条件类



### Controller

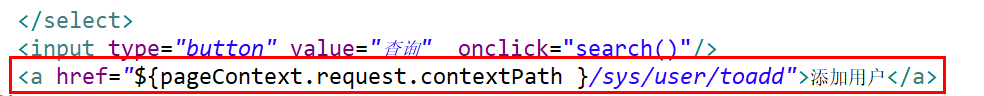


### 用户列表页面



## 去到添加用户页面

### userlist页面



### Controller控制器



### useradd页面

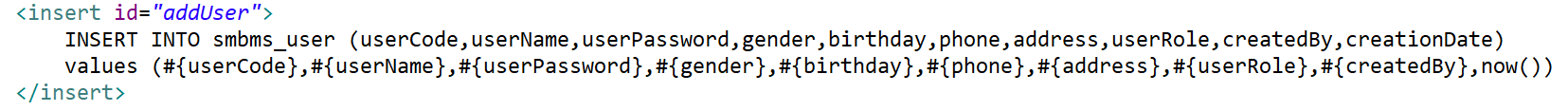


## 添加用户

### mapper接口



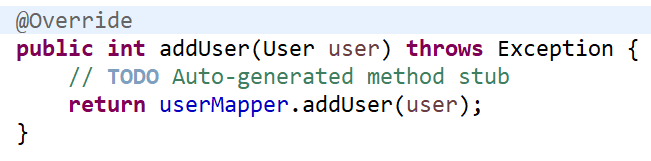
### mapper映射文件



### 业务层接口



### 业务层实现类



### Controller控制器



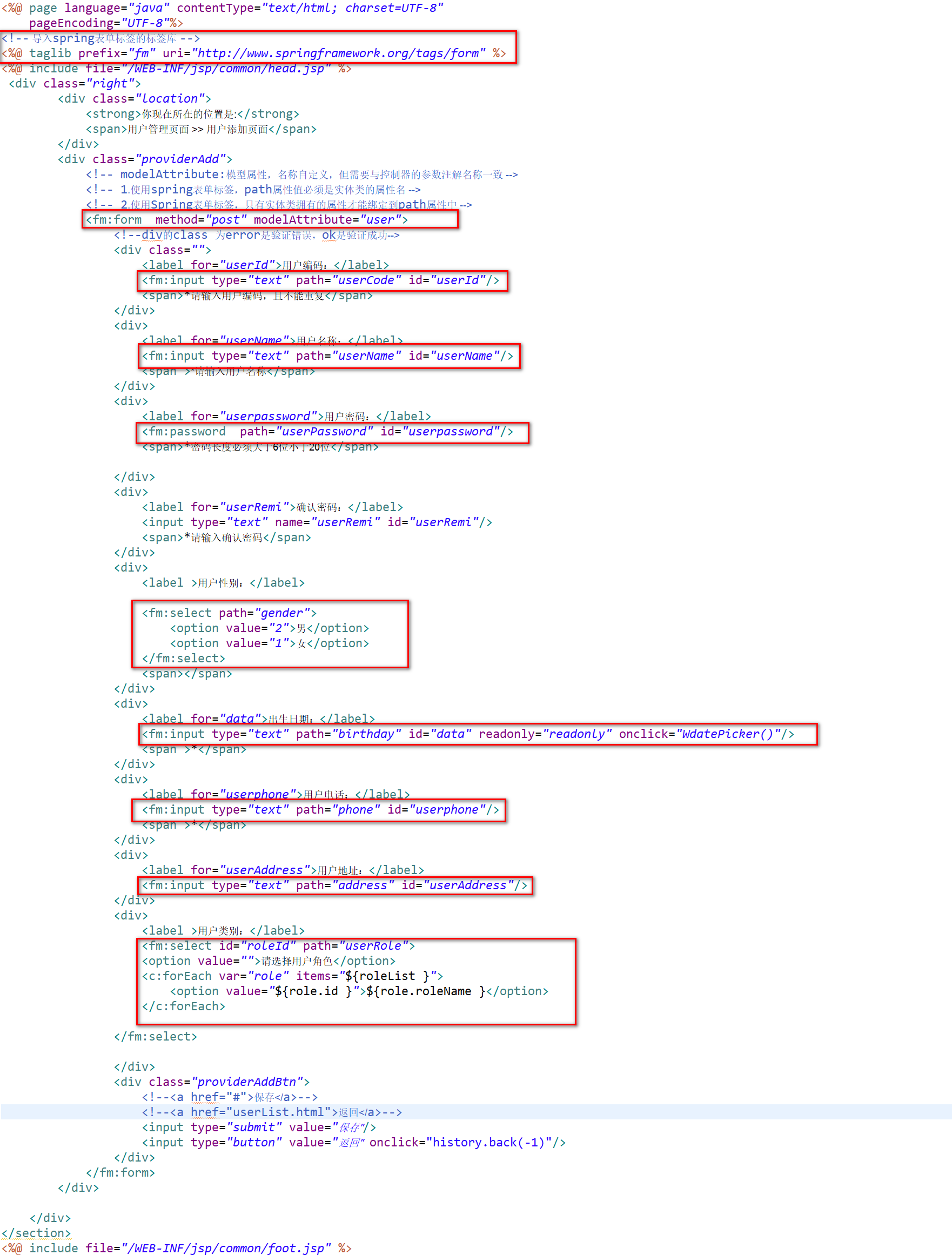
### useradd页面

使用日历插件，首先需要导入js，在foot.jsp页面导入js，然后在日期输入框中添加点击事件



## 使用spring表单标签实现添加

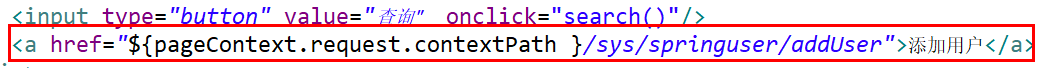
### 页面



### SpringUserController



### userlist页面



注意

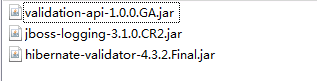
1. 使用spring表单标签必须引入标签库
2. 使用spring表单标签，path属性值必须是实体类的属性名
3. 使用Spring表单标签，只有实体类拥有的属性才能绑定到path属性中
4. 控制器提供2个方法，请求路径一致且分别是get、post提交

## 服务器端表单校验

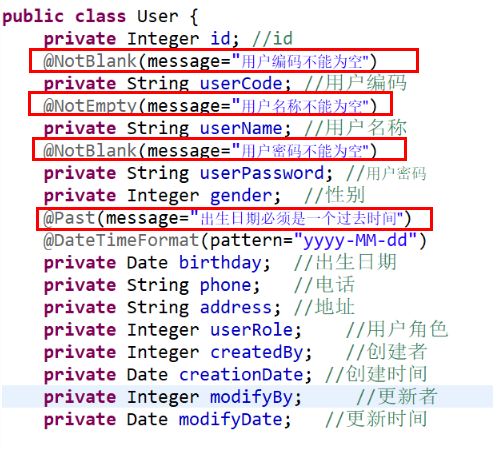
### 校验注解



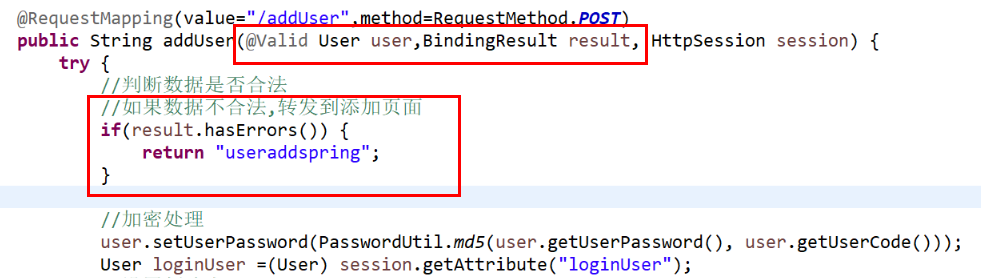
### 加入数据验证jar文件



### 实体类



### 控制器

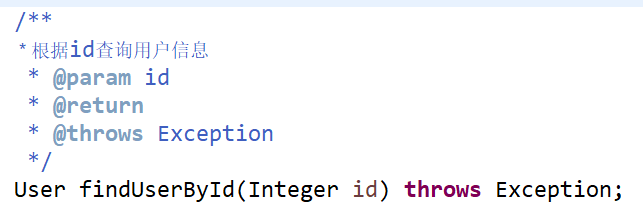


### 页面

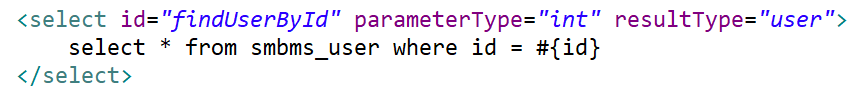


## Restful风格----去到修改用户页面

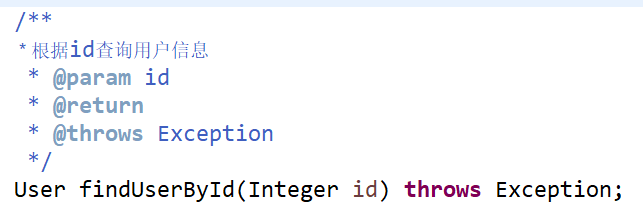
### mapper接口



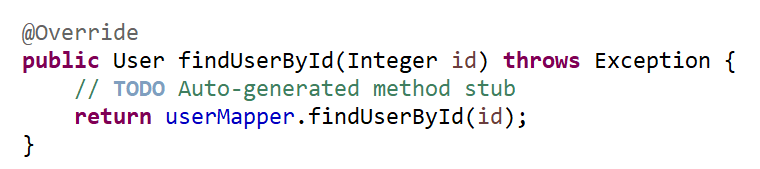
### mapper映射文件



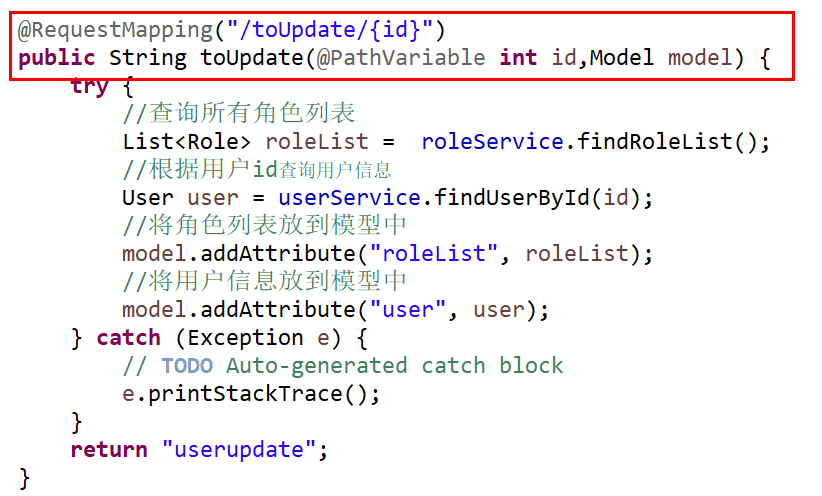
### 业务层接口



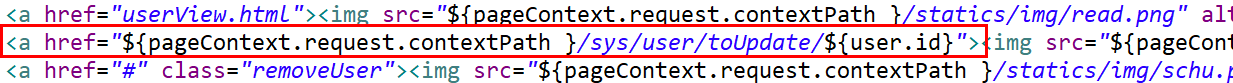
### 业务层实现类



### Controller控制器



### userlist页面

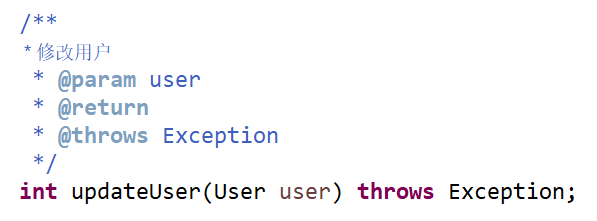


### userupdate页面



## 修改用户

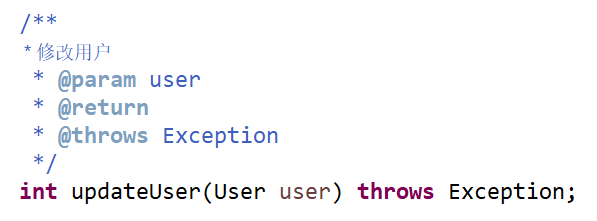
### mapper接口



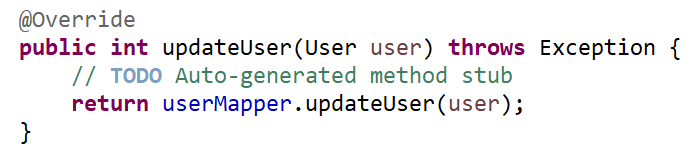
### mapper映射文件



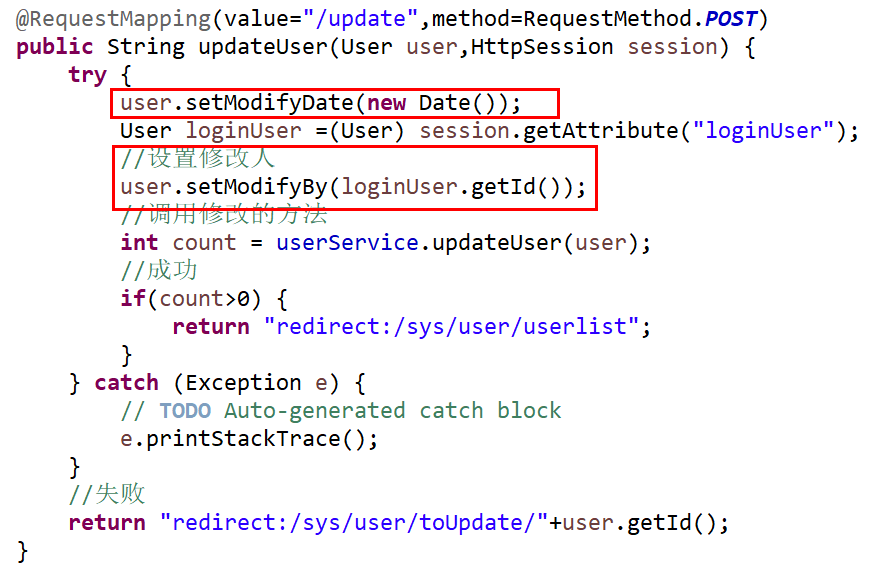
### 业务层接口



### 业务层实现类



### Controller控制器



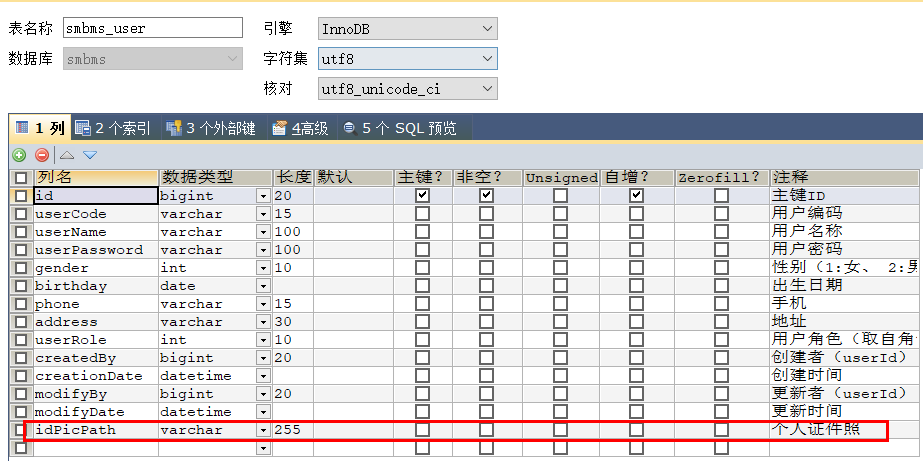
### userupdate页面



## 单文件上传

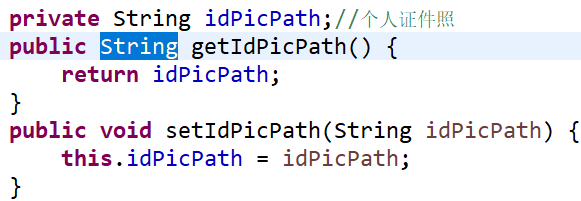
### 修改user表

添加字段，个人证件照



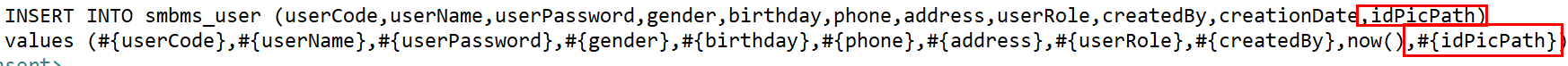
### 修改实体类

添加个人证件照的属性，提供get、set方法



### 修改usermapper映射文件

修改sql与占位符

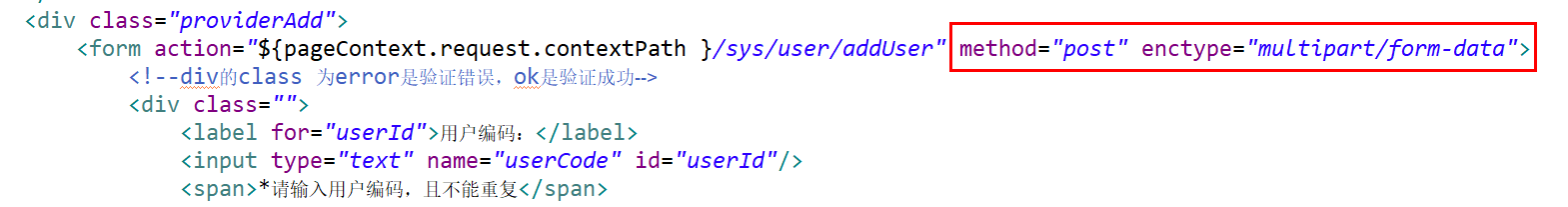


### 修改useradd添加用户页面

必须是post提交

添加enctype=”multipart/fom-data”

添加个人证件照的input标签





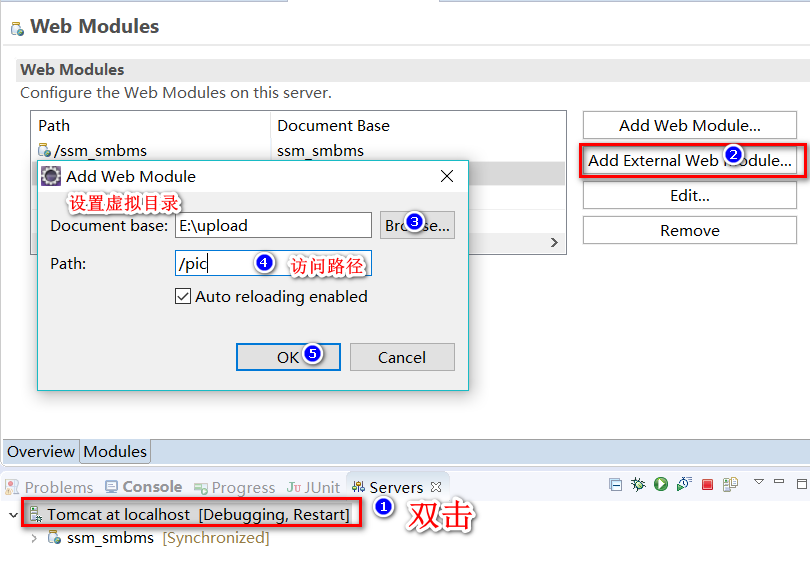
### 在springmvc配置文件中添加文件上传配置

|  |
| --- |
| <!-- 文件上传 -->  <bean id=*"multipartResolver"*  class=*"org.springframework.web.multipart.commons.CommonsMultipartResolver"*>  <!-- 设置上传文件的最大尺寸为5MB -->  <property name=*"maxUploadSize"*>  <value>5242880</value>  </property>  <property name=*"defaultEncoding"* value=*"UTF-8"*></property>  </bean> |

### 修改添加方法Controller



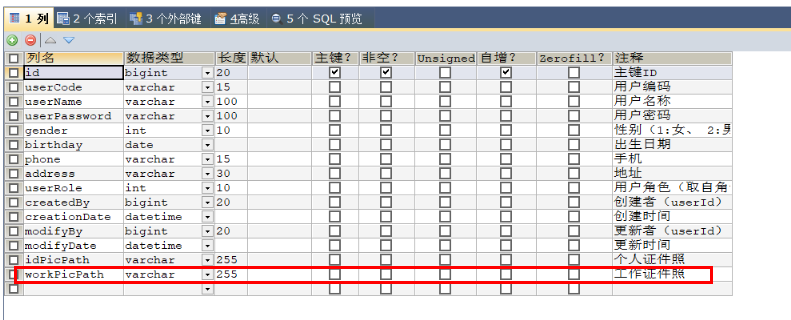
### 添加虚拟目录



## 多文件上传

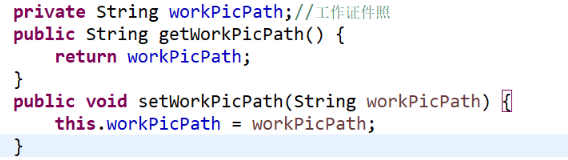
### 修改smbms\_user表

添加工作照字段



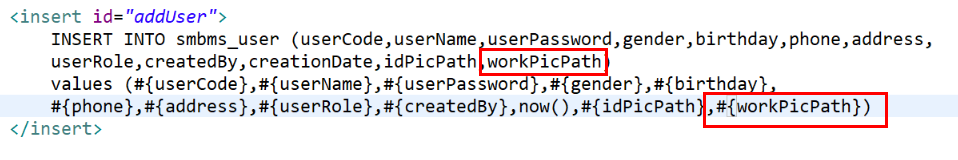
### 修改user类

添加工作照属性，提供get、set



### 修改映射文件

sql语句与占位符



### 修改添加用户的表单

添加工作照的文件表单

注意：多文件上传，表单name属性必须一致

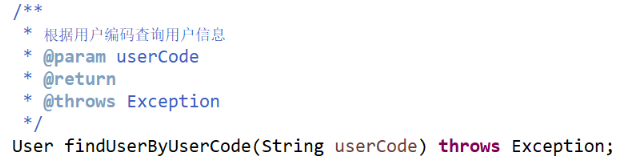


### 控制器

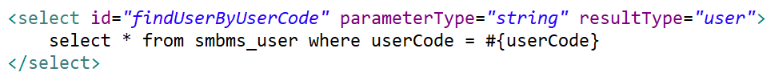


## ajax验证用户名是否存在

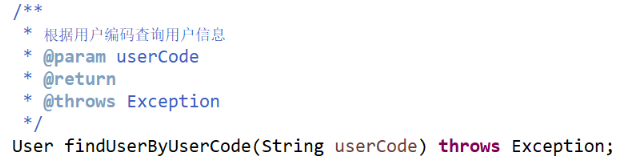
### dao接口



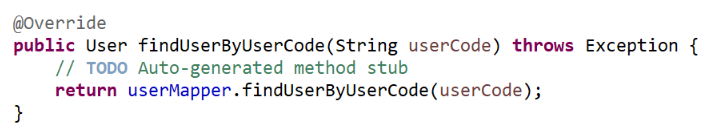
### 映射文件



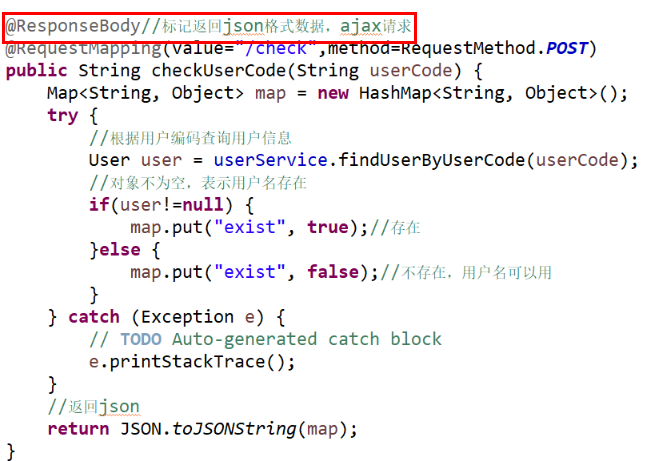
### service接口



### service实现类



### 控制器



### 页面



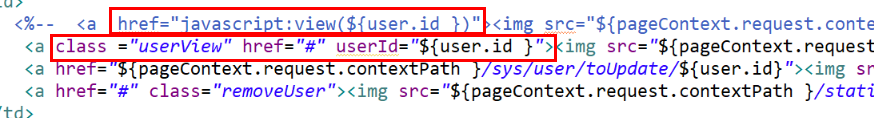
## ajax查询用户详情

### 修改userlist页面

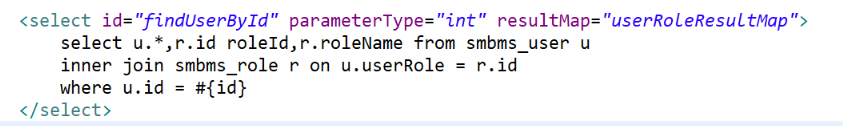
#### 添加表单，在分页代码下添加

|  |
| --- |
| <div class=*"providerAdd"* style="margin-top:*35px*;display: *none*">  <div>  <label>用户编码：</label>  <input type=*"text"* id=*"v\_userCode"* value=*""* readonly=*"readonly"*>  </div>  <div>  <label>用户名称：</label>  <input type=*"text"* id=*"v\_userName"* value=*""* readonly=*"readonly"*>  </div>  <div>  <label>用户性别：</label>  <input type=*"text"* id=*"v\_gender"* value=*""* readonly=*"readonly"*>  </div>  <div>  <label>出生日期：</label>  <input type=*"text"* Class=*"Wdate"* id=*"v\_birthday"* value=*""*  readonly=*"readonly"* onclick="WdatePicker();">  </div>  <div>  <label>用户电话：</label>  <input type=*"text"* id=*"v\_phone"* value=*""* readonly=*"readonly"*>  </div>  <div>  <label>用户角色：</label>  <input type=*"text"* id=*"v\_userRoleName"* value=*""* readonly=*"readonly"*>  </div>  <div>  <label>用户地址：</label>  <input type=*"text"* id=*"v\_address"* value=*""* readonly=*"readonly"*>  </div>  <div>  <label>创建日期：</label>  <input type=*"text"* Class=*"Wdate"* id=*"v\_create"* value=*""*  readonly=*"readonly"* onclick="WdatePicker();">  </div>  </div> |

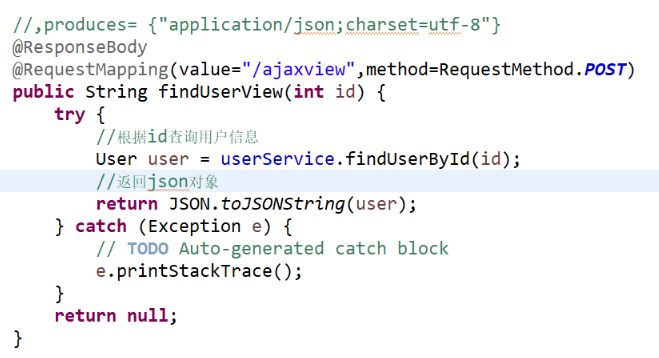
#### 修改查看按钮



### userMapper映射文件



### 控制器



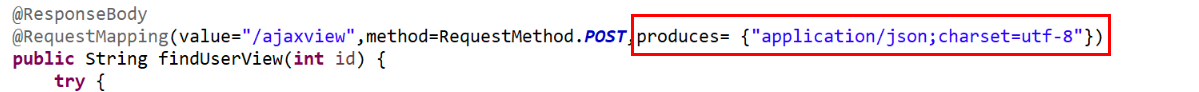
### userlist页面添加js





### 处理中文乱码

#### 方法一：局部配置

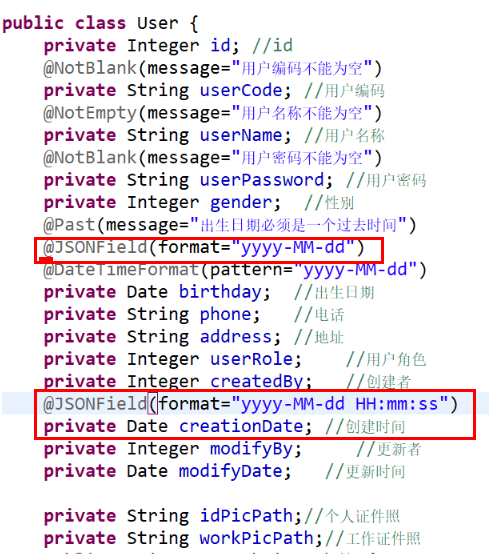


#### 方法二：全局配置

|  |
| --- |
| <!-- 注入注解驱动 -->  <mvc:annotation-driven>  <mvc:message-converters>  <!-- json中文乱码处理 -->  <bean class=*"org.springframework.http.converter.StringHttpMessageConverter"*>  <property name=*"supportedMediaTypes"*>  <list>  <value>application/json;charset=UTF-8</value>  </list>  </property>  </bean>  </mvc:annotation-driven> |

### 处理日期格式

#### 方法一：局部配置（使用注解）



#### 方法二：全局配置（会失效）

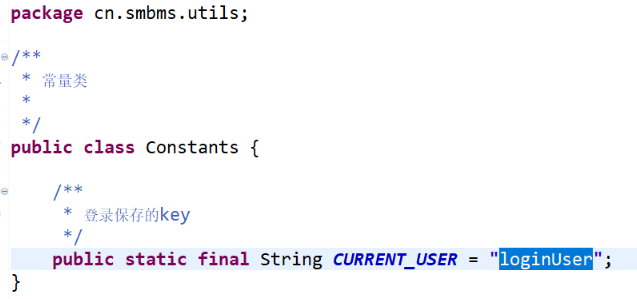
|  |
| --- |
| <!-- 注入注解驱动 -->  <mvc:annotation-driven>  <mvc:message-converters>  <!-- json中文乱码处理 -->  <bean class=*"org.springframework.http.converter.StringHttpMessageConverter"*>  <property name=*"supportedMediaTypes"*>  <list>  <value>application/json;charset=UTF-8</value>  </list>  </property>  </bean>  <!-- json默认将日期解析成毫秒数，处理毫秒数问题 -->  <bean class=*"com.alibaba.fastjson.support.spring.FastJsonHttpMessageConverter"*>  <property name=*"supportedMediaTypes"*>  <list>  <value>text/html;charset=UTF-8</value>  <value>application/json</value>  </list>  </property>  <property name=*"features"*>  <list>  <!-- Date的日期转换器 -->  <value>WriteDateUseDateFormat</value>  </list>  </property>  </bean>  </mvc:message-converters>    </mvc:annotation-driven> |

## 拦截器

### 概念

拦截器类似于过滤器，在进入控制器之前或之后做相应的拦截处理，例如：登录拦截，用户没有登录则不能去后台

### 定义常量类或枚举



### 定义拦截器



### 在springmvc配置文件中加载拦截器

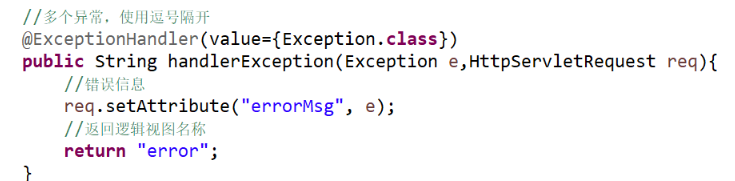


## 异常处理

### 局部异常

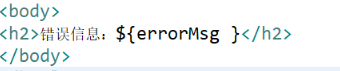
#### 控制器

在控制器中添加以下方法即可



注意:局部异常，在每一个控制器都需要加入，可以采用继承公共的控制或者全局异常处理，解决冗余问题

#### 添加错误页面



### 全局异常处理

在springmvc配置文件中加入以下配置即可

|  |
| --- |
| <bean class=*"org.springframework.web.servlet.handler.SimpleMappingExceptionResolver"*>  <property name=*"exceptionMappings"*>  <!-- 配置运行时异常 -->  <props>  <!-- 其中error是页面名称 -->  <prop key=*"java.lang.Exception"*>error</prop>  </props>  </property>  </bean> |