1. 需要什么样的sql语句(想好参数和返回值)
2. 写dao层方法
3. 在Controller里写handler
4. 写jsp界面

# jQuery Ajax



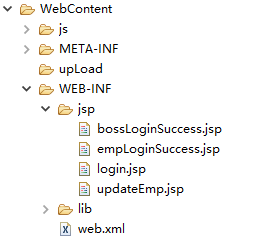
直接跳转登录

@RequestMapping(value={"/login","/"})

**public** String toLogin(){

**return** "login";

}

因为把登录的jsp放到了

WebContent下的WEB-INF下的jsp文件夹下，只能通过springMVC跳转

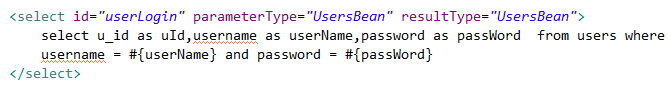
# 1.登录(springMVC+MyBatis)

Sql语句：

<!--登录-->

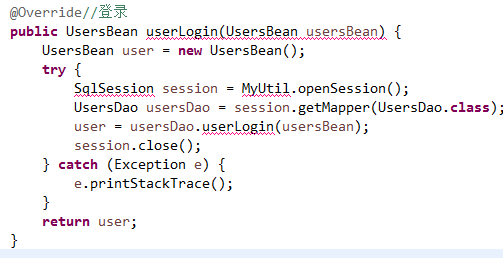
参数：通过表单传入一个bean,通过这个bean查数据库，然后返回一个bean

一般select都有返回值，因为是查询，而增删改返回值只有int，只能返回数据库有几行被修改了



id属性对应dao层的方法名

dao层方法:



Controller里写handler



Jsp页面

<h1>请登录</h1>

<h4>${requestScope.loginFiled}</h4>

<form action=*"doLogin"* method=*"post"*>

用户名:<input type=*"text"* name=*"userName"* />

密码:<input type=*"password"* name=*"passWord"* />

<input type=*"submit"* value=*"登录"* />

</form>

用户名和密码的input的name属性要和实体类的属性名对应

流程是这样的

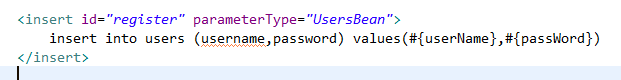
1. 用户在jsp界面输入值,form表单提交
2. 走到controller的handler方法
3. Handler调用dao层方法，查询数据库
4. Handler返回给用户一个界面

# 2.注册

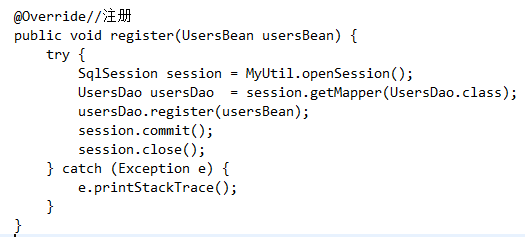
Sql语句:

<!--注册-->

注册肯定需要参数，一般是个bean,不需要返回值

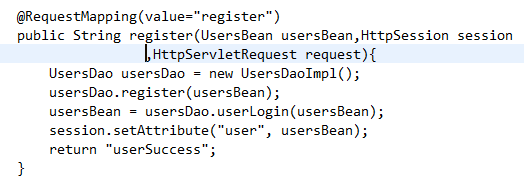


Dao层方法



Controller里的handler

//注册



注册成功后也返回一个登陆后的界面

Jsp界面

<h1>请注册</h1>

<form action=*"register"* method=*"post"*>

用户名:<input type=*"text"* name=*"userName"* />

密码:<input type=*"password"* name=*"passWord"* />

<input type=*"submit"* value=*"注册"* />

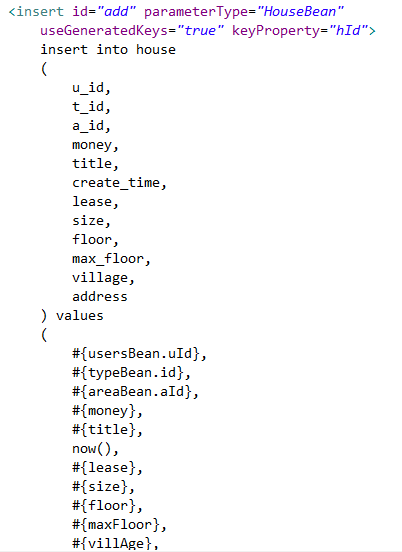
</form>

# 3.插入房屋信息

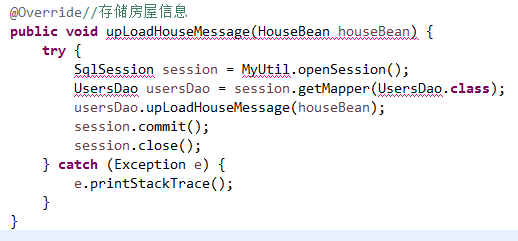
Sql语句

<!--插入房屋信息-->

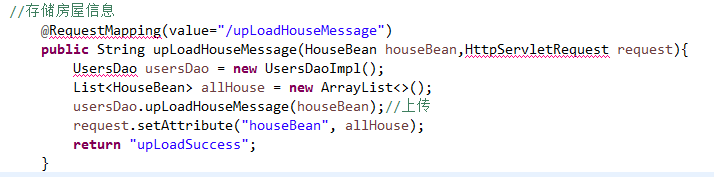
参数肯定是一个bean,不需要返回值



Dao层方法



Controller里的handler



传完了直接把信息放到request中,便于展示

Jsp界面





这里要注意几点：

1. 用户id存在于session中，用隐藏域把用户id通过表单传给后台
2. Select下拉框标签。

1.Name属性要和实体类的属性名对应,springMVC支持级联属性

2.里面的option标签是从数据库里得来的,所以在到这个页面之前,要查询好这个属性，并通过foreach标签遍历

3.option的value属性是单个条条的值，选中后，是把这个value传到后台，

Option中间的字只用于显示

没选中的时候，给一个默认的值<option value=*"0"*>请选择</option>

# 4.查所有房源(以及条件查询)

Sql语句:

<!--查所有-->

参数是一个map,返回值是MyBatis的resultMap,下面是map的映射

<select id=*"selectByMap"* parameterType=*"Map"* resultMap=*"selectAllHouse"*>

select

h.h\_id,

h.t\_id,t.type\_value,

h.a\_id,a.area\_value,

h.u\_id,u.username,

h.title,h.money,

h.size,h.floor,

h.max\_floor,h.village,h.address,

h.create\_time

from house h

left join type t on t.t\_id = h.t\_id

left join area a on a.a\_id = h.a\_id

left join users u on u.u\_id = h.u\_id

where 1=1

<if test=*"usersBean != null"*>

and h.u\_id = #{usersBean.uId}

</if>

<if test=*"houseBean.title != null and houseBean.title != ''"*>

and h.title like concat('%',#{houseBean.title},'%')

</if>

<if test=*"houseBean.money != null and houseBean.money != ''"*>

and h.money >= CONVERT(#{houseBean.money},SIGNED)

</if>

<if test=*"houseBean.maxMoney !=null and houseBean.maxMoney !=''"*>

and h.money<![CDATA[<=]]>CONVERT(#{houseBean.maxMoney},SIGNED)

</if>

<if test=*"houseBean.typeBean !=null and houseBean.typeBean.id !=0 "*>

and h.t\_id = #{houseBean.typeBean.id}

</if>

<if test=*"houseBean.areaBean !=null and houseBean.areaBean.aId !=0 "*>and h.a\_id = #{houseBean.areaBean.aId}</if>

<select />

resultMap的一对一关系

<resultMap type=*"HouseBean"* id=*"selectAllHouse"* autoMapping=*"true"*>

<id column=*"h\_id"* property=*"hId"* />

<association property=*"usersBean"* javaType=*"UsersBean"*>

<id column=*"u\_id"* property=*"uId"* />

<result column=*"username"* property=*"userName"* />

<result column=*"password"* property=*"passWord"* />

</association>

<association property=*"typeBean"* javaType=*"TypeBean"*>

<id column=*"t\_id"* property=*"id"* />

<result column=*"type\_value"* property=*"typeValue"* />

</association>

<association property=*"areaBean"* javaType=*"AreaBean"*>

<id column=*"a\_id"* property=*"aId"* />

<result column=*"area\_value"* property=*"areaValue"* />

</association>

</resultMap>

这里用到了MyBatis的if判断标签

如果userbean不为空,就在全部查询后面跟where条件,查出来的就是该用户的房源,不拼接的话就是全部房源

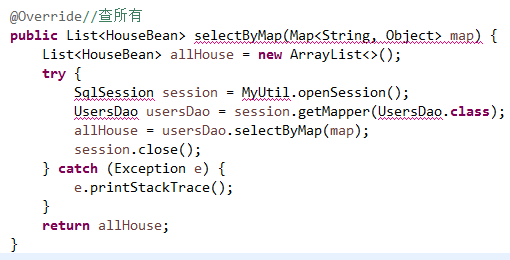
Where 1=1标识无论如何,后边都会拼and …

这些if标签是条件查询的关键

如果我页面上传来对应的值，就会走if条件，里面的and就会拼接到select语句后面，形成条件查询

也可以不写where 1=1,在所有if条件外边写上一个<where><where/>标签即可

Dao层方法



Controller里的handler

//查所有房源(以及条件查询)

@RequestMapping(value={"/getAllHouse","/getMyHouse"})

**public** String getAllHouse(HouseBean houseBean,HttpServletRequest request,Map<String, Object> requestMap){

UsersDao usersDao = **new** UsersDaoImpl();

String uri = request.getRequestURI();//返回请求行中的资源名部分

Map<String, Object> map = **new** HashMap<>();

map.put("houseBean", houseBean);

List<HouseBean> allHouse = **new** ArrayList<>();

**if** (uri.equals("/getMyHouse")) {//如果拼接路径为getMyHouse

//从session中获取user的属性，放到map里

UsersBean u = (UsersBean) request.getSession().getAttribute("user");

map.put("usersBean", u);

allHouse = usersDao.selectByMap(map);//调dao层方法，把含有userBean的map当做参数

}**else** **if**(uri.equals("/getAllHouse")){//如果点的是getAllHouse

map.put("usersBean", **null**);//map里放一个空的userbean

allHouse = usersDao.selectByMap(map);

}

//查付款方式

List<TypeBean> typeList = usersDao.payWay();

//查区域

List<AreaBean> areaList = usersDao.selectArea();

requestMap.put("allHouse", allHouse);

requestMap.put("typeList", typeList);

requestMap.put("areaList", areaList);

//将请求的uri保存到request中

requestMap.put("uri", uri);

**return** "showHouse";

}

这里有很多需要注意的地方

1. 这个handler映射了两个jsp页面的路径
2. 根据不同的映射，查询不同的数据

Jsp界面

<a href=*"/getMyHouse"*>查看自己发布的</a>

<a href=*"/getAllHouse"*>查看所有房源</a>

查询成功后，跳转的jsp

显示条件查询的表单，然后是显示的页面



查询的信息：

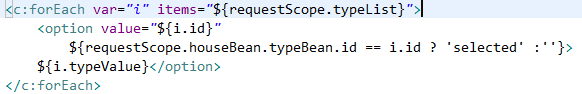


这里也有很多要注意的地方

1. form表单的action属性是动态的，放的是上一个页面拼接的请求,why?
2. 给input标签的value属性写上上一次请求的信息，意思是说

如果我搜索的话，搜索后跳到的界面，input标签里显示内容会是之前Input标签里输入的信息

1. option标签的属性如何绑定呢?



拿之前请求的request中存的typeBean中的id和当前遍历的集合分别对比

相等就给option标签来一个selected属性，标识当前已选中

# 5.单文件上传

Sql语句:根据用户id插入到user表的photo字段

update users set photo = ? where u\_id = ?

dao层方法

…省略了

Controller里的handler



需要注意的:

1.参数有个MultipartFile file，是springMVC用于上传的类

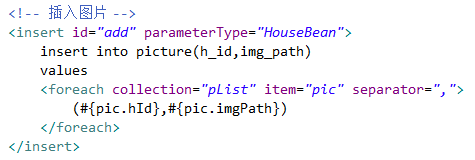
2.需要导两个包 

Jsp页面:



# 6.批量上传

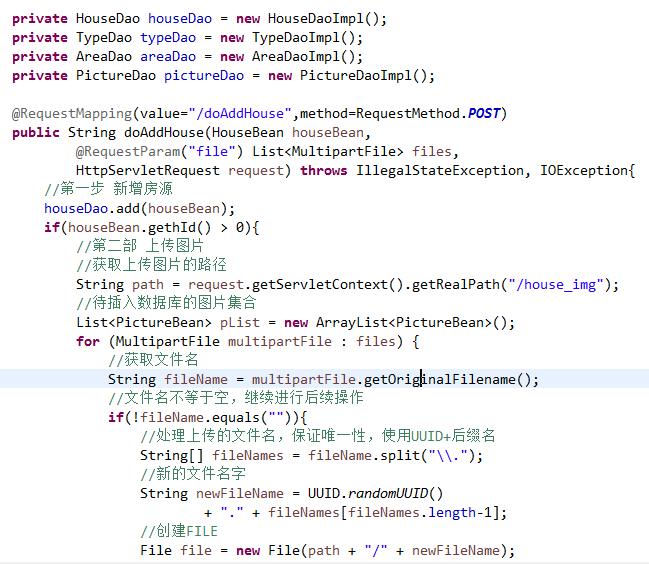
Sql语句:

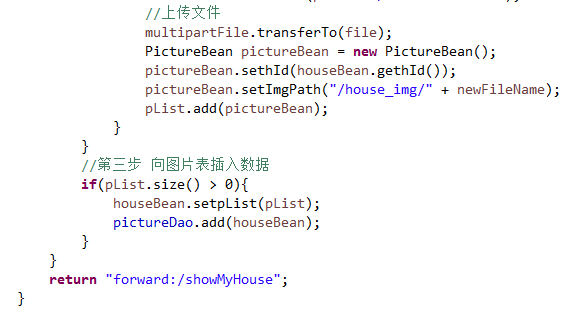


Dao层方法

…

Controller里的handler



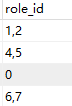


# 7.根据登录信息不同跳转不同页面



# 8.根据id查询用户权限

数据库是这样的：



权限id有多种，以逗号分隔

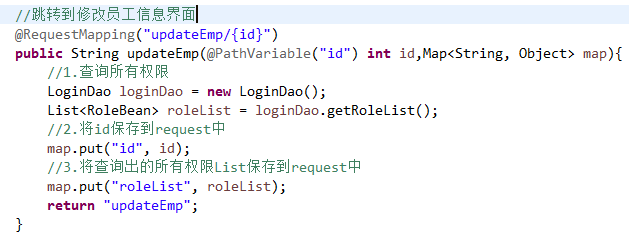
Dao层处理方式：



# 9.修改用户信息，对应的位置显示数据库的信息



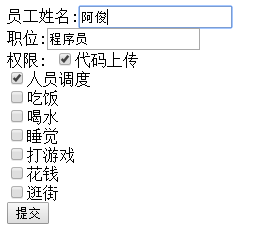
修改的时候，把员工id传过去,然后跳转到修改信息界面



修改员工信息界面：



如何跳过来就让标签绑定当前员工的信息？Like this



通过ajax可以



通过ajax提交:

