1.图片批量上传(框架已搞定,利用ckFinder可以配置放到固定位置，然后通过servlet做显示 com.fj.common.servlet.UserfilesDownloadServlet)

2.地图坐标获取(已解决，参考开发文档)

http://developer.baidu.com/map/jsdemo.htm#h0\_5

<script type="text/javascript" src="http://api.map.baidu.com/api?v=2.0&ak=B2e75d2a46ef88bab0e6af581d57ce26"></script>

3.sso单点登录

4.mybatis会根据相应的类属性做映射 如下面代码

<!-- 插入用户 -->

<insert id="insert">

INSERT INTO sys\_user(

id,

company\_id,

office\_id,

login\_name,

password,

no,

name,

email,

phone,

mobile,

user\_type,

create\_by,

create\_date,

update\_by,

update\_date,

remarks,

login\_flag,

photo,

del\_flag

) VALUES (

#{id},

#{company.id},

#{office.id},

#{loginName},

#{password},

#{no},

#{name},

#{email},

#{phone},

#{mobile},

#{userType},

#{createBy.id},

#{createDate},

#{updateBy.id},

#{updateDate},

#{remarks},

#{loginFlag},

#{photo},

#{delFlag}

)

</insert>

5.ckFinder文件上传 

http://fj.com/static/ckfinder/ckfinder.html?

type=files

start=files:/house/smallArea/2015/04/(固定的目录，有bug无法上传到指定的文件夹下)

action=js

func=remarksSelectAction

thumbFunc=remarksThumbSelectAction

cb=remarksCallback

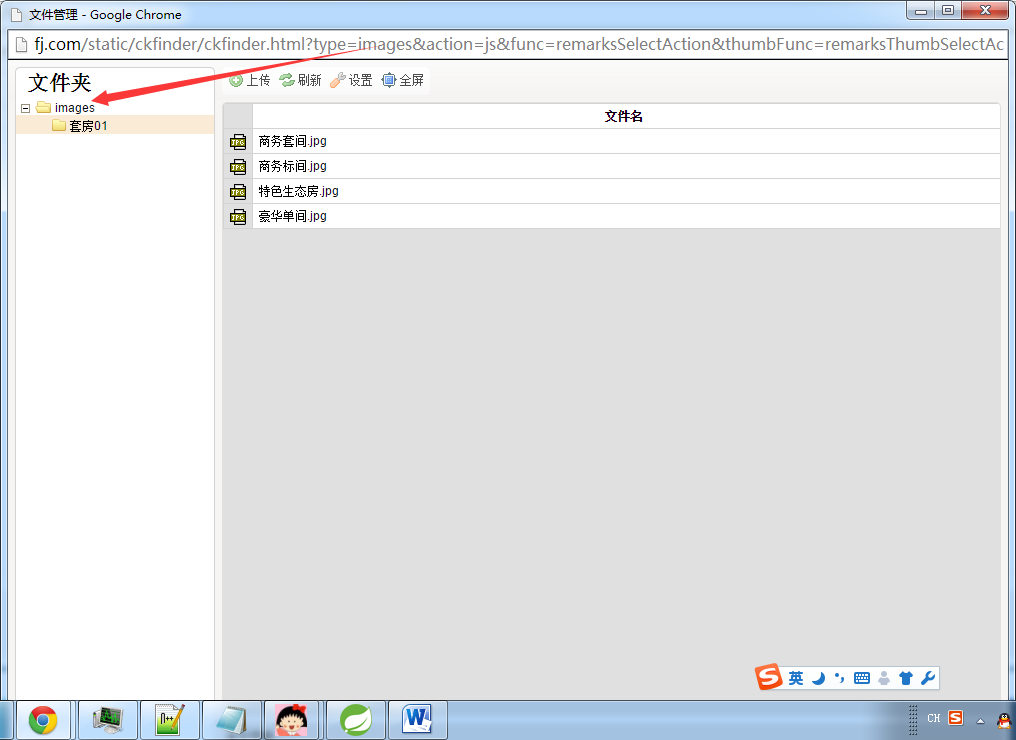
dts=0

sm=1

CKFinderConfig.java 配置文件是否可编辑上传及用户的初始化目录

http://fj.com/static/ckfinder/ckfinder.html?type=images&action=js&func=remarksSelectAction&thumbFunc=remarksThumbSelectAction&cb=remarksCallback&dts=0&sm=1

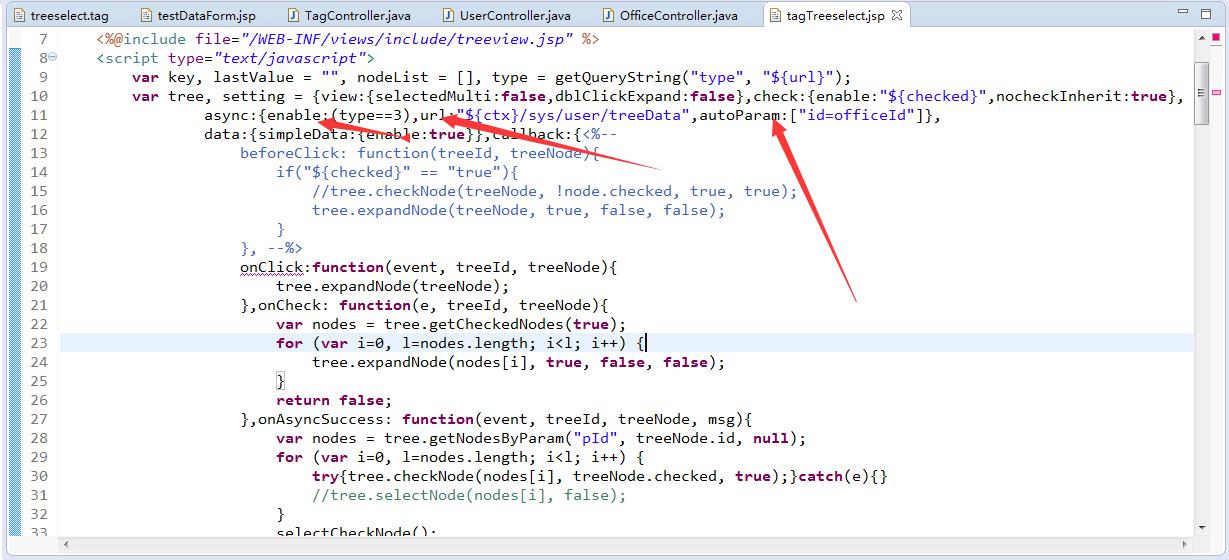
type:设置初始化目录



6.zTree分析<http://www.ztree.me/>

如果有需要二次异步加载的，设置enable:true，并设置其他参 例如：async:{enable:(type==3),url:"${ctx}/sys/user/treeData",autoParam:["id=officeId"]},

Enable/autoParam参数说明详见http://www.ztree.me/v3/api.php



6.mybatis传递多个参数

**Mybatis传多个参数（三种解决方案）**

# 第一种方案

DAO层的函数方法

[?](http://www.2cto.com/database/201409/338155.html)

|  |  |
| --- | --- |
| 1 | Public User selectUser(String name,String area); |

对应的Mapper.xml

[?](http://www.2cto.com/database/201409/338155.html)

|  |  |
| --- | --- |
| 1  2  3 | <select id="selectUser" resultMap="BaseResultMap">      select  \*  from user\_user\_t   where user\_name = #{0} and user\_area=#{1}  </select> |

其中，#{0}代表接收的是dao层中的第一个参数，#{1}代表dao层中第二参数，更多参数一致往后加即可。

# 第二种方案

此方法采用Map传多参数.

Dao层的函数方法

[?](http://www.2cto.com/database/201409/338155.html)

|  |  |
| --- | --- |
| 1 | Public User selectUser(Map paramMap); |

对应的Mapper.xml

[?](http://www.2cto.com/database/201409/338155.html)

|  |  |
| --- | --- |
| 1  2  3 | <select id=" selectUser" resultMap="BaseResultMap">     select  \*  from user\_user\_t   where user\_name = #{userName，jdbcType=VARCHAR} and user\_area=#{userArea,jdbcType=VARCHAR}  </select> |

Service层调用

[?](http://www.2cto.com/database/201409/338155.html)

|  |  |
| --- | --- |
| 1  2  3  4  5 | Private User xxxSelectUser(){  Map paramMap=new hashMap();  paramMap.put(“userName”,”对应具体的参数值”);  paramMap.put(“userArea”,”对应具体的参数值”);  User user=xxx. selectUser(paramMap);} |

个人认为此方法不够直观，见到接口方法不能直接的知道要传的参数是什么。

# 第三种方案

Dao层的函数方法

[?](http://www.2cto.com/database/201409/338155.html)

|  |  |
| --- | --- |
| 1 | Public User selectUser(@param(“userName”)Stringname,@param(“userArea”)String area); |

对应的Mapper.xml

[?](http://www.2cto.com/database/201409/338155.html)

|  |  |
| --- | --- |
| 1  2  3 | <select id=" selectUser" resultMap="BaseResultMap">     select  \*  from user\_user\_t   where user\_name = #{userName，jdbcType=VARCHAR} and user\_area=#{userArea,jdbcType=VARCHAR}  </select> |

个人觉得这种方法比较好，能让开发者看到dao层方法就知道该传什么样的参数，比较直观，个人推荐用此种方案。