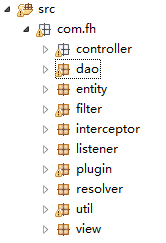
# 简介FH Q313596790

## Springmvc + mybatis 组合框架

# 1各包说明

## Src



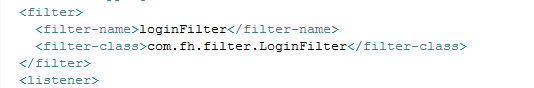
**1．controller:业务处理包 （日常代码维护主要包）**

2．dao:增删改查的接口 （无需操作，不用管它）

3．entity:实体类包 (存放实体类)

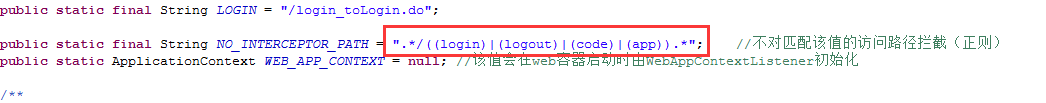
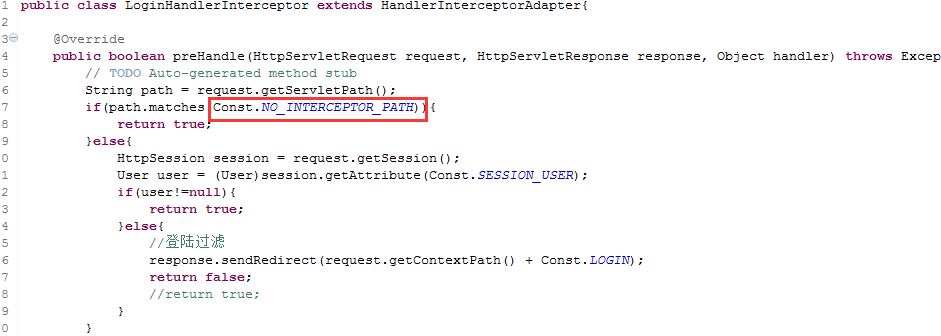
4. filter:登录顾虑验证器(可以在此添加一段代码，让tomcat启动后立即自动执行

需要配置web.xml



5．interceptor：session有效期验证



请求的连接中\*xx.do 不包含login，logout，code，app 等字符的，都会被判断session存在与否，否: 跳转到登录，是: 跳转到相应地址

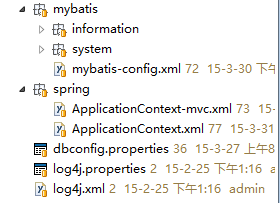
6．Listener ：在web容器启动时由WebAppContextListener初始化

7．Plugin :分页插件(已经处理好，无需更改)

8．Listene ：MyExceptionResolver 异常处理

9．Util 所有工具类(发邮件，发短信，日期格式化等)

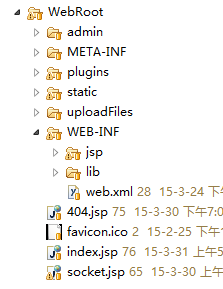
## 1.2 resources



1. mybatis :对应的配置文件
2. spring ：spring的配置文件ApplicationContext.xml
3. log4j 日志处理配置，可设置生成日志文件到硬盘的某个目录下
4. dbconfigerties : 数据库链接池配置
5. shior 配置，在spring/ApplicationContext.xml



## WebRoot



admin :存放配置文件，代码生成器生成的代码(相对tomcat的目录)

plugins : 插件存放目录

static : js css img 等存放目录

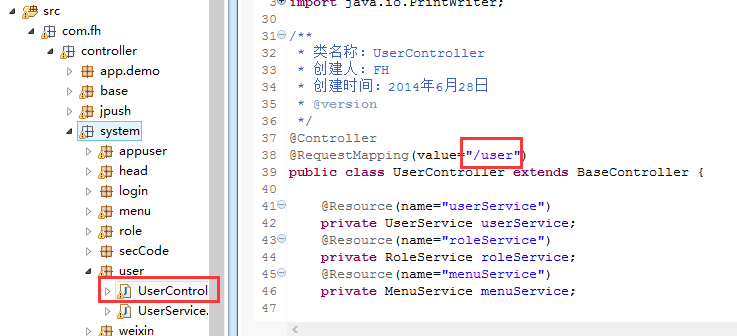
jsp : 在WEB-INF 目录下

# 增删改查流程

## 增加：(form表单提交数到后台在存入数据库)

form表单action=”user/saveU.do”

1.比如新增用户，”user” 对应的是



saveU.do对应的是如下图



学过springmvc的朋友都知道这样映射方式，和struts2区别很大(放到xml配置中)

2.提交的表达所有的参数都通过pd = **this**.getPageData();

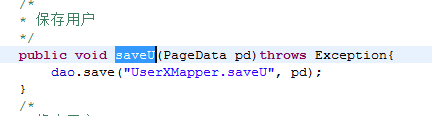
获取到，pd就是封装的map。

比如表单中新增的用户名 <input name=”username” value=”张三”>

后台pd = this.getPageData();后，pd里面就有username的值了

打印(pd.getString(“username”))输入的值为“张三”

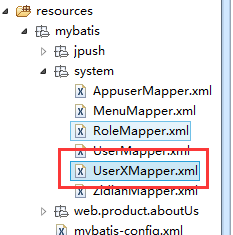
1. 3.最后把充满数据的pd 传递给service层userService.saveU(pd);

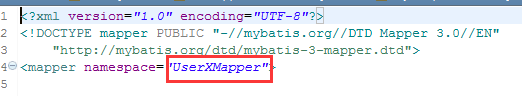


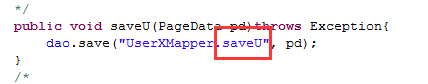
dao.save（”字符串1 . 字符串2”，pd ） 。save就是保存，update就是更新

delete就是删除

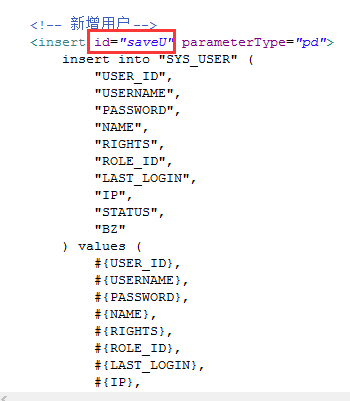
“UserXMapper”对应的mybatis的配置文件为



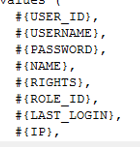
映射关联是根据此xml文件顶部的相关联

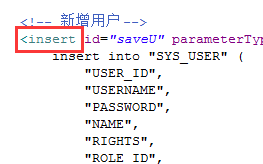


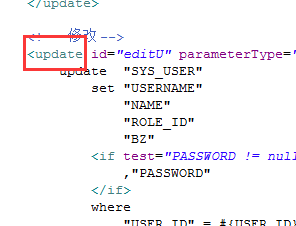
saveU 对应的是 id

  
传过来的 parameterType 意思是传入值的类型



读取，根据pd中的键名，读取其值

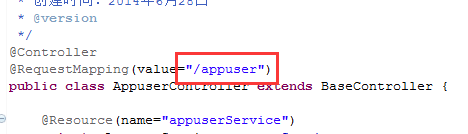
新增标签

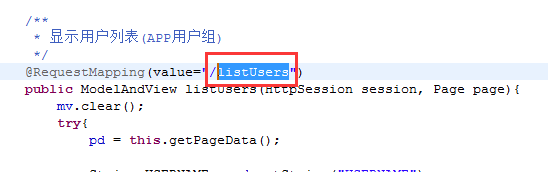
 修改标签

删除标签，等以此类推

## 重点介绍下列表

用户列表：请求连接/appuser/listUsers.do

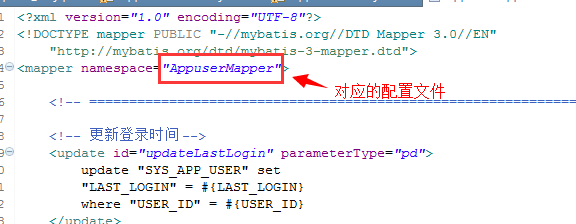


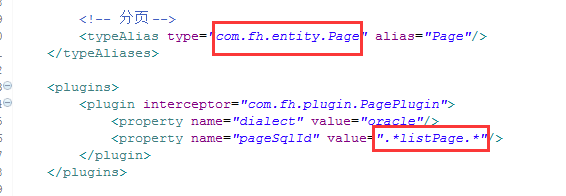






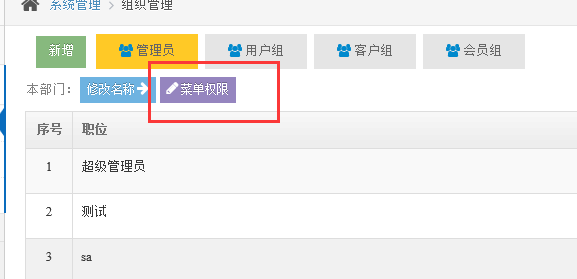




分页处理



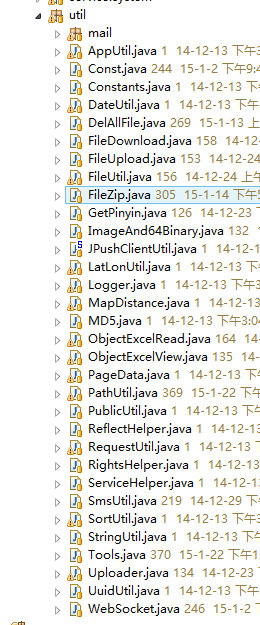
### 菜单权限分配



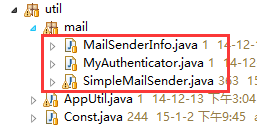
**Admin 用户只能这里分配菜单权限起作用，admin不是超级用户，是最高用户**

所有权限，菜单，都放入到session中了，所有多菜单、权限的操作，需要退出重新登录才能看到效果

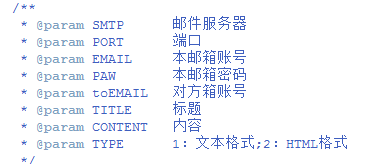
# 工具类说明



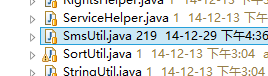
### 邮箱服务器：



SimpleMailSender.sendEmail(1, 2, 3, 4, 5, 6, 7, 8);//调用发送邮件函数



### 调用短信接口:



SmsUtil.sendSms1(phone, CONTENT); //调用发短信函数1

SmsUtil.sendSms2(phone CONTENT); //调用发短信函数2



### 代码生成器

进入系统后，右上角，系统设置下面

生成的压缩包里面有 操作说明，安装说明步骤把代码放到项目里面

### 信息管理模块

只有小项目代码里面有信息管理模块

其它只有系统模块

注意切换菜单按钮

在左上角最边上的那个小方格

数据库连接池(用的阿里的 druid )  
查看监控地址  
<http://127.0.0.1:8080/xxx/druid/index.html>

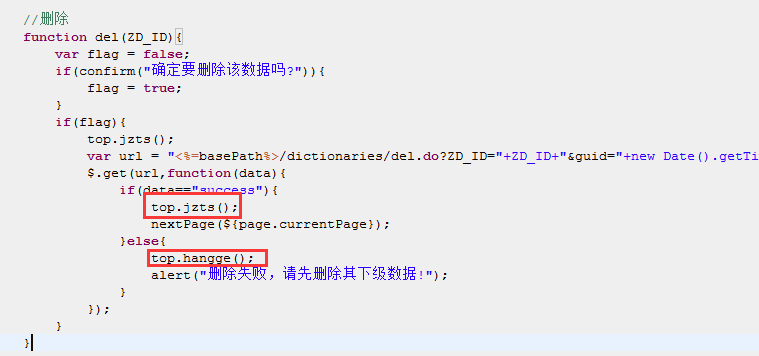
# 页面相关

### 页面加载状态开关

开启加载状态 js代码：top.jzts();

关闭加载状态 js代码：top.hangge();

例：



# Ehcache 缓存

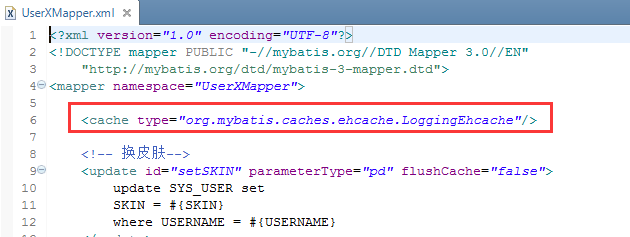
## 合理使用缓存

在\*\*mapper.xml 文件如同加入

<!-- 以下两个<cache>标签二选一,第一个可以输出日志,第二个不输出日志 -->

**<cache** type="org.mybatis.caches.ehcache.LoggingEhcache"**/>**

<!-- <cache type="org.mybatis.caches.ehcache.EhcacheCache"/> -->



这样本页面所有都默认加入缓存，请注意不能乱加，具体如何合理加入缓存，请百度搜索详细资料，我就不在此啰嗦

单个开关

Insert update delete flushCache=*"false"*

Select useCache=*"false"*

