# 班级成绩管理系统报告

目录

[SSM开发步骤 1](#_Toc30225)

[一、 系统概述 3](#_Toc12496)

[二、 系统功能介绍 3](#_Toc13997)

[三、 组建系统架构 3](#_Toc24692)

[1. 建立数据库： 3](#_Toc22521)

[2. 建立项目 3](#_Toc24077)

[3. 建立文件夹 3](#_Toc25684)

[4. 建立包 4](#_Toc11435)

[5. 建立以上的框架目录如下图： 4](#_Toc5139)

[四、 引入jar包 4](#_Toc26307)

[1. Mybatis jar包 4](#_Toc13671)

[2. Spring jar包 5](#_Toc4370)

[3. Web开发： 5](#_Toc5180)

[4. 其他包 5](#_Toc16746)

[1、 表现层 6](#_Toc27218)

[2、 服务层（业务逻辑层） 6](#_Toc8439)

[3、 持久层 6](#_Toc24799)

[六、配置文件 6](#_Toc19919)

[1、 log4j.properties 6](#_Toc14455)

[2、 Mybatis.xml 6](#_Toc32192)

[3、 applicationContext.xml 7](#_Toc4375)

[4、 Springmvc.xml配置 8](#_Toc5296)

[5、 配置web.xml 8](#_Toc23415)

[6、 配置UserMapper.xml 9](#_Toc13088)

[七、导入所需外部资源（图片，js） 10](#_Toc1560)

[1. js前台框架 essyui 框架（开源） 10](#_Toc9606)

[2. 导入背景图片 10](#_Toc19373)

[3. 设置跳转路径 10](#_Toc9105)

[八、实现功能 10](#_Toc30155)

[1. 通过controller实现页面跳转 10](#_Toc1698)

[2. 新增功能 11](#_Toc15868)

[3. 查询功能 12](#_Toc6398)

[4. 修改功能 12](#_Toc2232)

[5. 删除功能 13](#_Toc27515)

[6. 登录功能 13](#_Toc20683)

[7. 上述功能实现代码 13](#_Toc19903)

[九、功能扩展 14](#_Toc3881)

[1. 拦截器（拦截未登录不合法跳转） 14](#_Toc9543)

[2. 退出功能 15](#_Toc2637)

[3. 限制增改用户名重复问题 15](#_Toc459)

[4. 文件上传下载 16](#_Toc4492)

[5. 分页功能 16](#_Toc456)

[十、学生信息管理 17](#_Toc3078)

[1. 新增功能 17](#_Toc28713)

[2. 查询功能 18](#_Toc16410)

[3. 修改功能 19](#_Toc3177)

[4. 删除功能 19](#_Toc8075)

[5. 文件上传下载 20](#_Toc14504)

[6.分页功能 20](#_Toc1462)

[7.上该管理结合了五、六的全部功能实现 20](#_Toc25993)

[十一、总结 23](#_Toc137)

[1. 有关配置文件 23](#_Toc4560)

[2. 有关注解 23](#_Toc16920)

[3. 有关引包 23](#_Toc9239)

[4. 有关web服务的理解 23](#_Toc2056)

[5. 有关前端easyui 24](#_Toc22296)

1. 系统概述

本系统是基于SSM+easyui的设计模式进行设计的，面对信息化时代的发展，班级信息的管理同样也需要信息化服务平台的支持，由此设计了本次的班级信息管理系统，利用ssm的后台架构降低了层与层之间的耦合性，并且增强了代码的复用性。

1. 系统功能介绍

本系统包括了用户信息的管理以及班级成绩的管理，实现了模糊搜索的查询功能单条信息修改，多条信息的删除。

在程序的健壮性方面，设置了拦截器功能，阻止了不合法的页面跳转功能，在存储信息方面，设置了不允许添加重复的用户功能。

1. 组建系统架构
2. 建立数据库：

数据库名：ssm

建立的表为：user



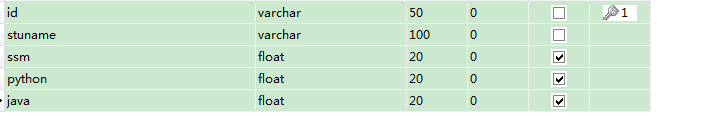
Id：用户的标识

Username：用户名

Password：密码

Status：状态（0：可用 1：停用）

建立表：stu 学生成绩表



1. 建立项目

项目名为ssm

需勾选web.xml

1. 建立文件夹
2. 建立config用来存放配置文件，在构建是需要选择： 

Config文件夹中存放的配置文件分别为：

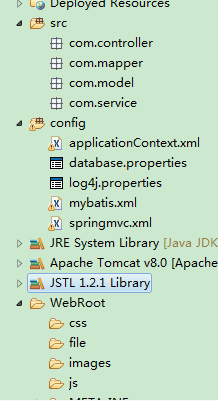
applicationContext.xml 用来配置spring

mybatis.xml 用来配置mybatis

springmvc.xml 配置springmvc

log4j.properties用来打印日志即日志管理

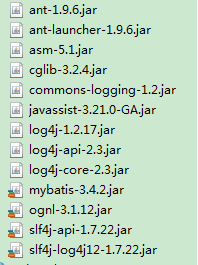
1. 在webRoot下建立js文件夹用来存放js文件
2. 在webRoot下建立css文件夹用来存放外部引用的css文件
3. 在webRoot下建立images文件夹用来用来存放引用的背景图片和logo
4. 在webRoot下建立file文件夹用来存放下载的文件
5. 在webRoot/webINF下建立jsp文件夹用来存放jsp页面
6. 建立包
7. 在src下建立com.controller包用来放表现层
8. 在src下建立com.service 包用来存放服务层（业务逻辑层）
9. 在src下建立com.mapper 包用来存放持久层
10. 在src下建立com.model 包用来存放数据库中对应的表数据
11. 建立以上的框架目录如下图：



1. 引入jar包
2. Mybatis jar包

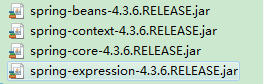
这13个包为一个mybatis基础包 mybatis-3.4.2.jar mybatis 基础包

和12个依赖包



1. Spring jar包

四个基本jar包



面向切面编程：



事务管理：spring-tx-4.3.6.RELEASE.jar

Jdbc数据库：spring-jdbc-4.3.6.RELEASE.jar

1. Web开发：



1. 其他包

处理jsion数据



专门处理面向切面编程的



第三方数据库连接池： commons-dbcp2-2.1.1.jar

commons-pool2-2.4.2.jar

数据库连接包： mysql-connector-java-5.1.8.jar

Spring与mybatis整合时所需jar包 mybatis-spring-1.3.1.jar

用于文件上传：



对Excel文件进行处理（按照sheet和表格进行读取）jxl.jar

两个api 包，tomcat中也含有（可不引用）

1. 创建包中的实现类
2. 表现层

在com.controller中建立UserController.java

在com.model中建立User

1. 服务层（业务逻辑层）

在com.service 建立UserService接口

建立com.service.impl 包，建立实现上面接口的类 UserServiceImpl

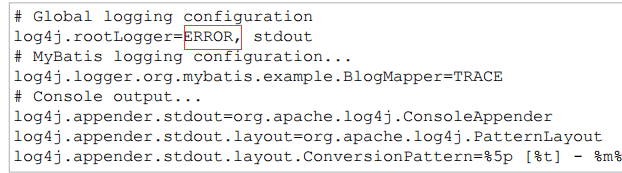
1. 持久层

利用mapper代理的形式，分别建立接口UserMapper.java和UserMapper.xml(起名的四个规范)

六、配置文件

1. log4j.properties

寻找路径：8.1.1.2 Step 2: Configure Log4J（在mybatis写的配置文档pdf的目录结构中）



将文件中的ERROR改为DEBUG方便调试，在myeclipse中需要在mybatis.xml中加载配置才可以驱动日志管理：



1. Mybatis.xml

配置文件路径：

下的第二个配置文件：在此配置数据库的**连接引用**和**定义别名**，另外在myeclipse2016下

需要**配置setting log4j**来驱动日志管理。。

配置别名：

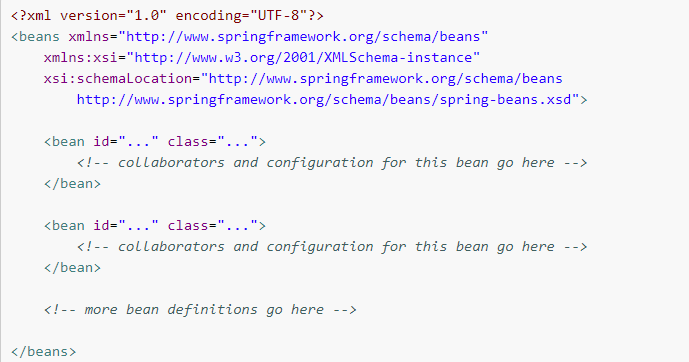
因为采用的mapper代理的方式进行mybatis开发，所以mapper的配置自动加载 不需要配置

1. applicationContext.xml

Spring下的：spring-framework-4.3.6.RELEASE-dist/spring-framework-4.3.6.RELEASE/docs/spring-framework-reference/html/index.html



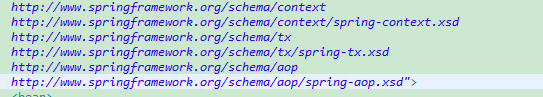
点击此目录，找到他所对应的配置文件：



1. 添加命名空间：

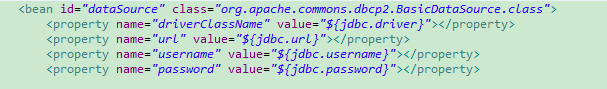
添加context、tx、aop：





1. 引入数据：

引入数据库 :

配置数据源：

1. 配置事务管理器：
2. 配置通知：即绑定事务管理，判断哪些方法需要事务管理



各标签的含义：

propagation： 传播行为

REQUIRED ： 表示指定的方法必须在事务的环境中进行，如果环境是非事务环境，就创建一个事务环境

SUPPORTS ： 表示指定的方法执行时候 如果是在事务环境下 那就用事务方式执行，如果是在非事务环境下 方法就在非事务环境中执行

read-only ：只读事务。 对于增删改操作，修改数据库 对数据库的链接要求较高 。 查询：只要读取数据库就行，对数据库的链接要求低





1. 配置切面：



第一个\*:表示返回值类型 com.service 表示要配置的切面的方法所在的包

第二个\*:com.service.\* 表示com.service及其子包下的所有类

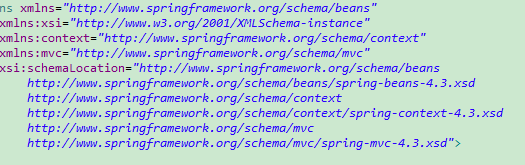
第三个\*:类中的所有方法

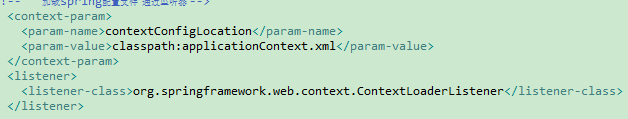
(..) :方法的参数可以是任意的

com.service包及其子包下的所有方法（任意返回值类型 ，任意参数）都设置为切面

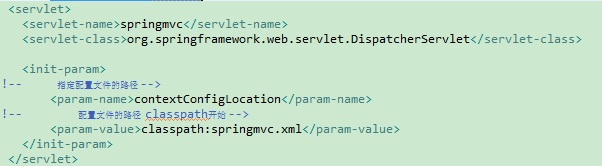
1. 配置持久层：配置mybaits sqlsessionFactroy



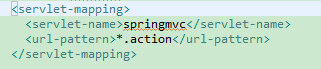
1. 配置注解：
2. Springmvc.xml配置
3. 加载命名空间：
4. 配置视图解析器指定前缀后缀：
5. 配置注解驱动：
6. 通过包扫描的方式进行controller定义 ：
7. 配置web.xml
8. 加载spring配置文件，通过监听器并实现绑定：



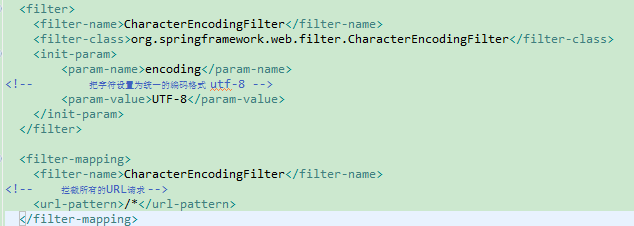
1. 配置中央控制器定义servlet



1. 绑定servlet处理



1. 添加过滤器，处理中文字符问题



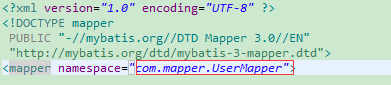
1. 只保留index.jsp用来指定跳转页面：



1. 配置UserMapper.xml

配置文档路径：

遵循mapper代理的形式，命名空间与权限命名保持一致：



七、导入所需外部资源（图片，js）

1. js前台框架 essyui 框架（开源）

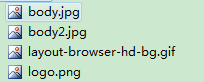
jquery-easyui-1.5.1 essyui

My97DatePicker 日期控件

echarts.min.js 图表处理

jquery-1.11.3.min.js jquery

1. 导入背景图片



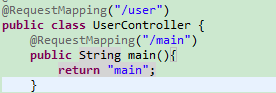
1. 设置跳转路径



八、实现功能

1. 通过controller实现页面跳转

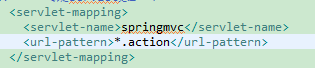
（1）跳转main.jsp



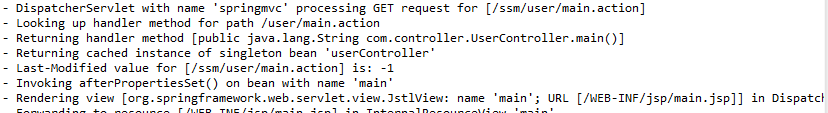
实现原理：在springmvc中增加了视图解析器：



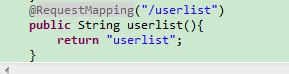
并且在web.xml中加载了监听器：



在控制输出的结果如下：



1. 跳转userlist.jsp页面



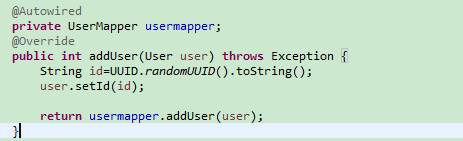
1. 新增功能
2. 在com.mapper中添加接口以及实现方法





1. 在com.service中添加接口以及实现方法

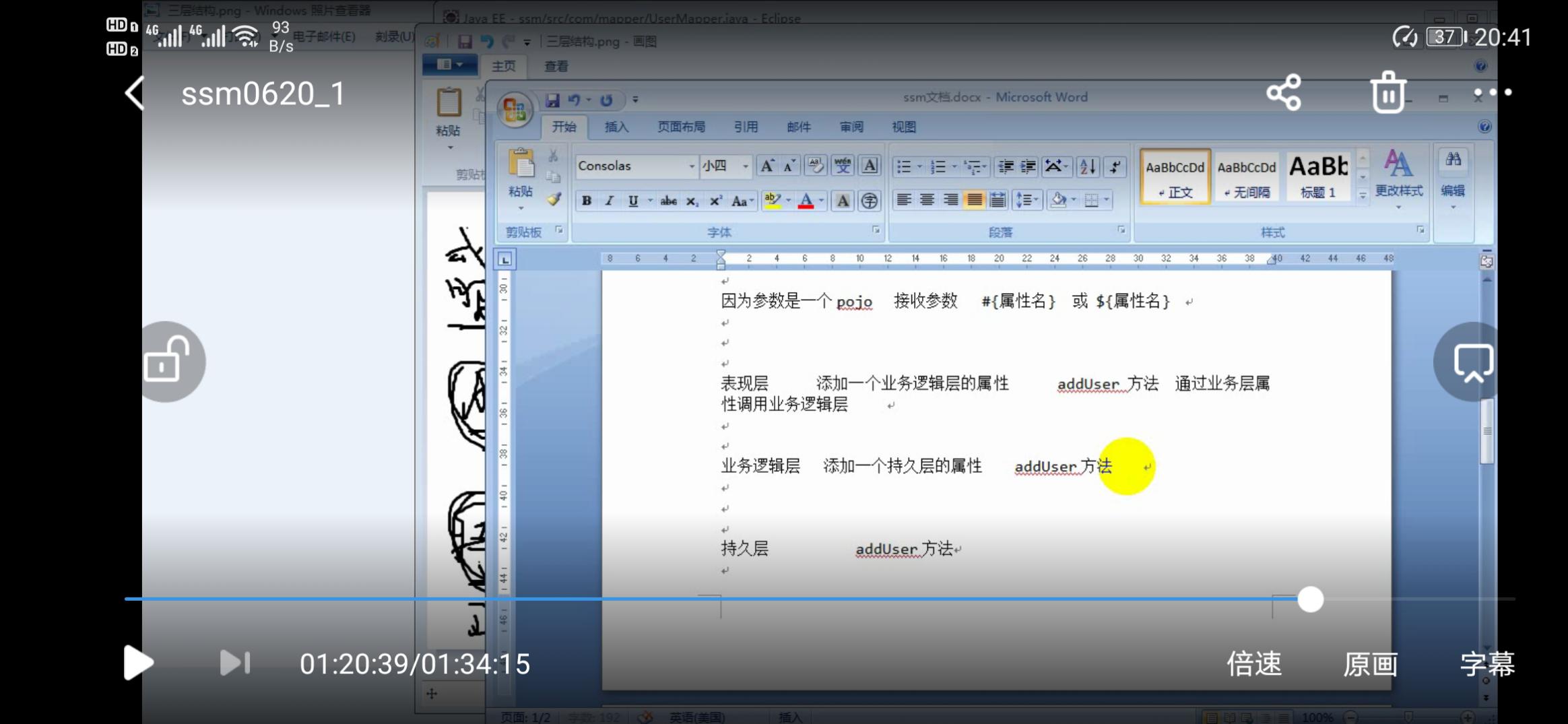




1. 在com.controller中添加新增方法

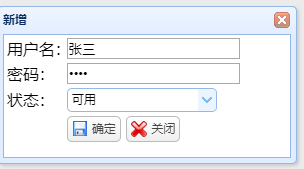


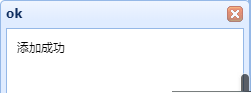
1. 各层之间相互关联，利用注解以及引用的形式



1. 实现添加功能以及数据库显示：

前台





数据库：



1. 通过以上的学习，三层代码的优点：
2. 实现了功能代码的隔离，降低了耦合度（高内聚，低耦合）
3. ds

**实现功能部分在此之后将不再过多阐述，只做简单说明和截图，后台代码实现功能统一展示。**

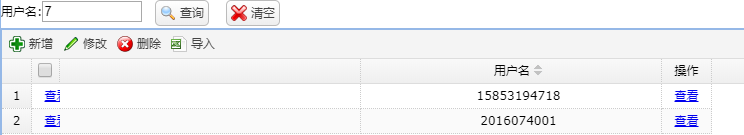
1. Service服务层添加接口的原因

Aop按接口的方式进行编程

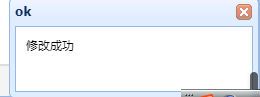
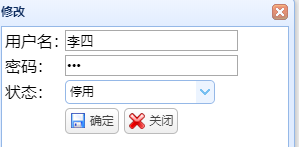
1. 查询功能

功能截图1：



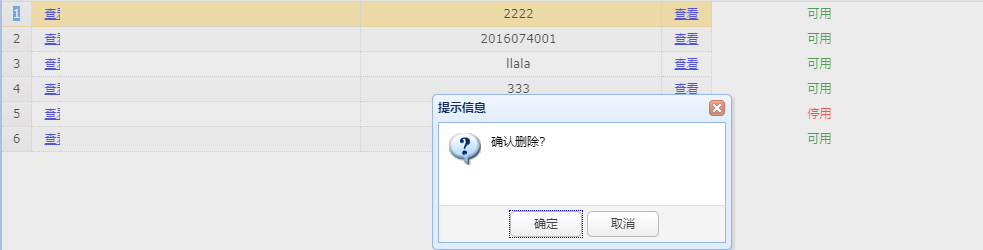
查询功能2（按条件查询）：  
 

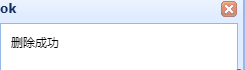
1. 修改功能



数据库中：

1. 删除功能





1. 登录功能

利用数据库查询到的与自己输入的进行判断，是否有词条数据



1. 上述功能实现代码

Usercontroller中：

查询：



修改：



删除：

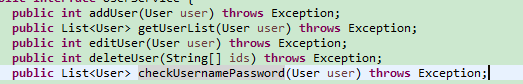


登录：

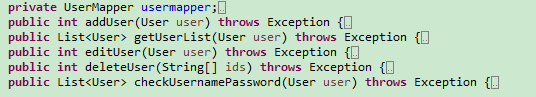


Service中

接口：

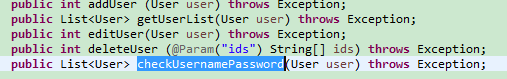


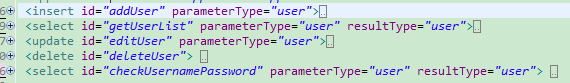
接口实现类：



mapper中

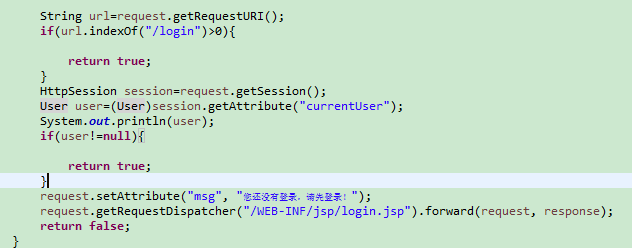
接口：



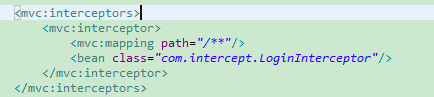
接口实现xml：  


九、功能扩展

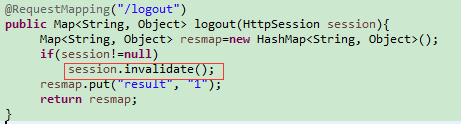
1. 拦截器（拦截未登录不合法跳转）
2. 建立包：
3. 建立过滤器 ：
4. 实现限制功能：



1. 加载到applicationContext.xml中拦截所有请求：



1. 退出功能



**清除session**

1. 限制增改用户名重复问题

思路：在新增或者更新之前通过查询是否有该用户，若有这直接返回resmap，提示错误原因。

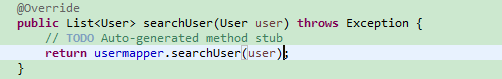
具体实现代码如下：

持久层：

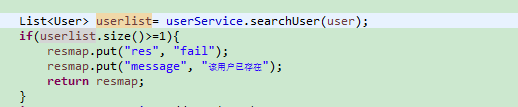




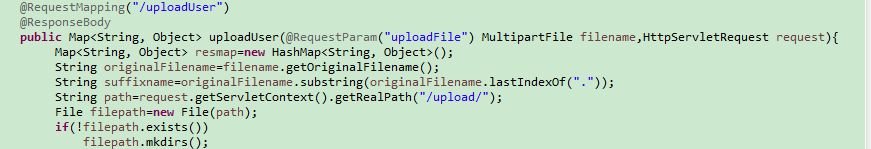
服务层：



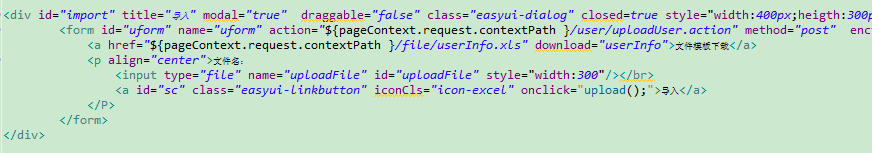
表现层：



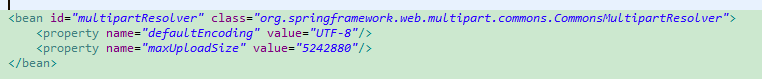
1. 文件上传下载
2. 在表现层获取前端文件名字，并将其重命名



1. 前端以表单的形式获取，其规则为：
2. form表单的method属性设置为post；
3. form表单的enctype属性设置为multipart/form-data；
4. 表单有文件上传输入框 <input type="file" name="filename" />
5. 当form表单的enctype属性为multipart/form-data时，浏览器就会采用二进制流来处理表单数据，服务器端就会对文件上传的请求进行解析处理。Spring MVC通过MultipartResolver实现文件上传功能。MultipartResolver是一个接口对象，需要通过它的实现类CommonsMultipartResolver来完成文件上传。



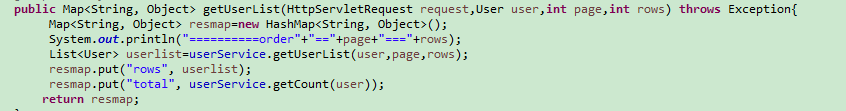
1. springmvc.xml中配置文件上传解析器



1. 分页功能
2. 前端利用easyUI的pagination 显示分页：

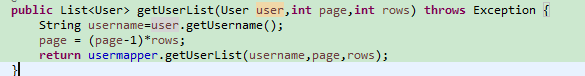


1. 后台获取前台的页码数和每页现实的行数，后台静参数写到形参中会自动获取，这应该是easyui的特别之处吧。



1. 分别在服务层和持久层编写实现代码

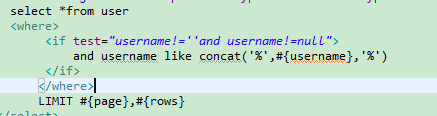
服务层：（接口不再赘述）



提两点：

1. 每次的起始行应为（page-1）\*rows，这是因为第一次limit起始行为page-1才是我们所需要的起始地点。
2. 因为在mapper.xml若在接口中写了param注解，在mapper.xml中利用包扫描的形式的user将不会被识别，我就在服务层中直接把user对象的username属性传到了mapper中，再利用注解的形式进行书写，就识别到username啦。

持久层：



在接口中：



十、学生信息管理

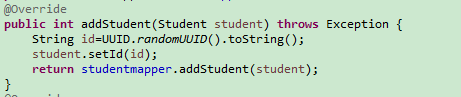
1. 新增功能

在com.mapper中添加接口以及实现方法

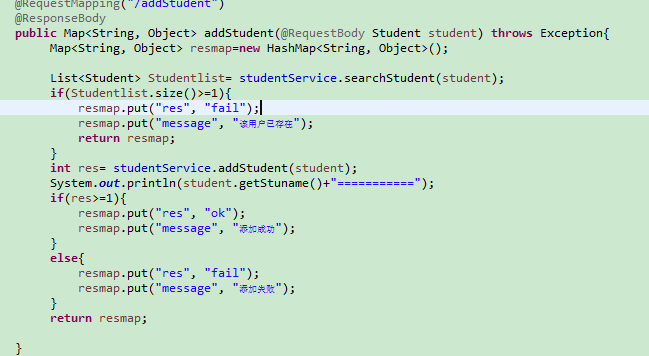
 

在com.service中添加接口以及实现方法



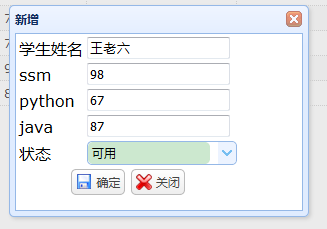


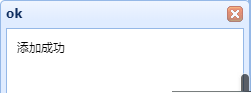
在com.controller中添加新增方法



实现添加功能以及数据库显示：

前台



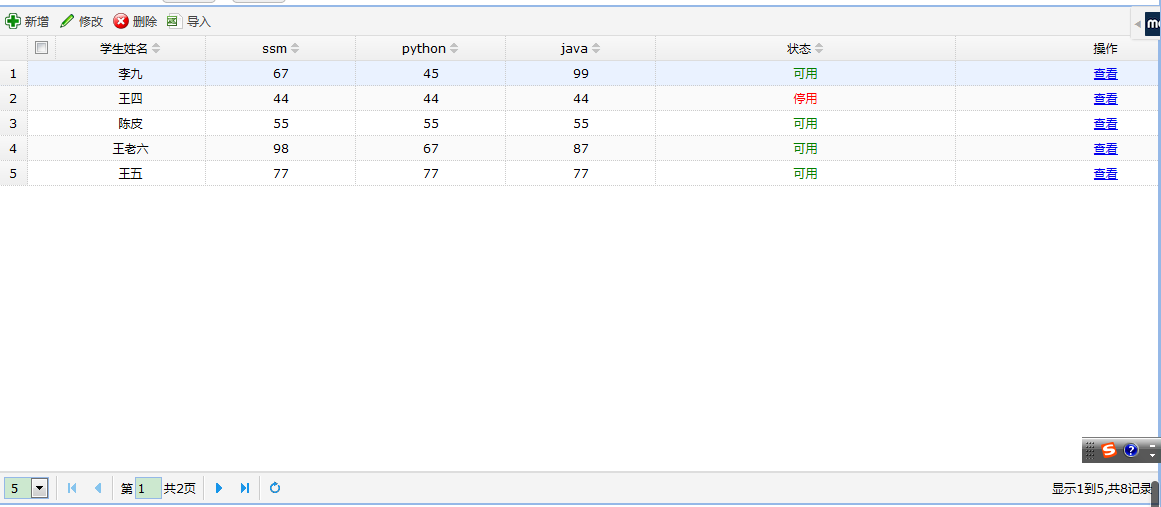


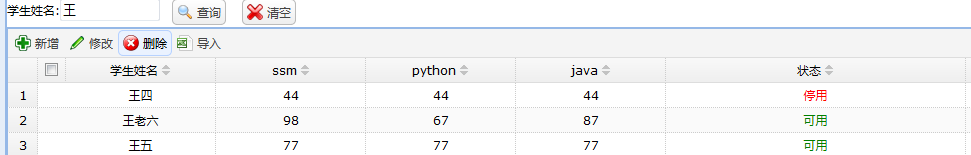
数据库：



1. 查询功能

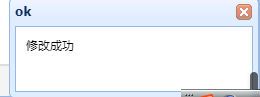
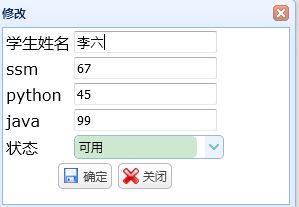
功能截图1：



查询功能2（按条件查询）：  
 

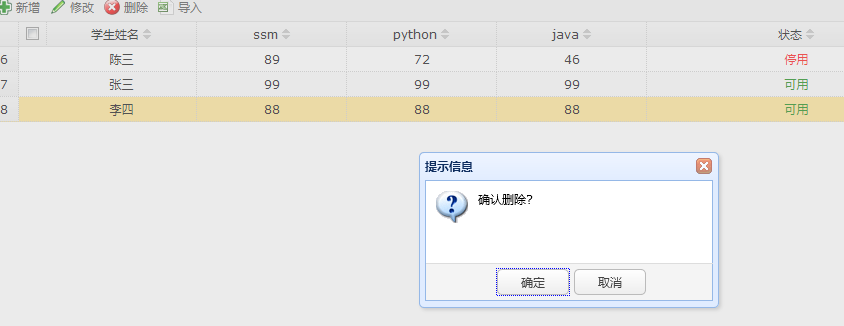
1. 修改功能

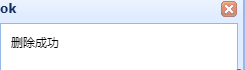
将李九改为李六



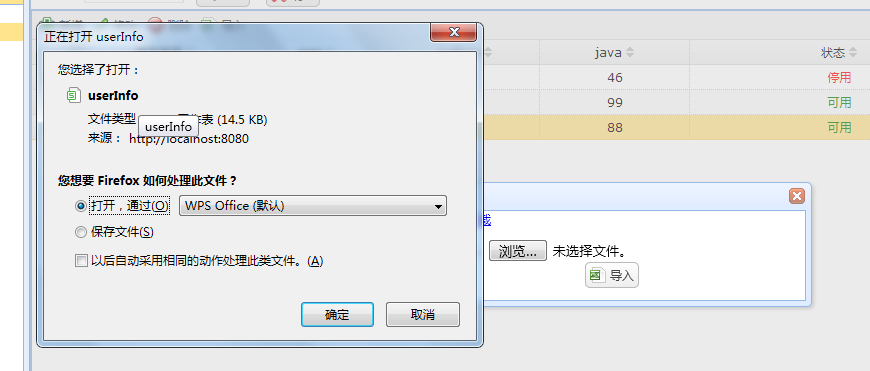
数据库中：

1. 删除功能





1. 文件上传下载



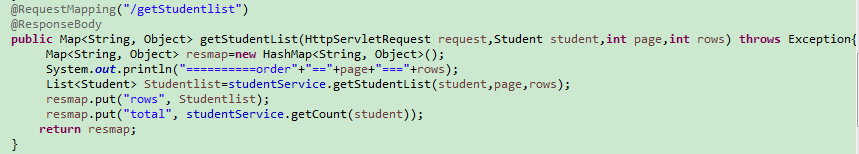
6.分页功能



7.上该管理结合了五、六的全部功能实现

Studentcontroller中：

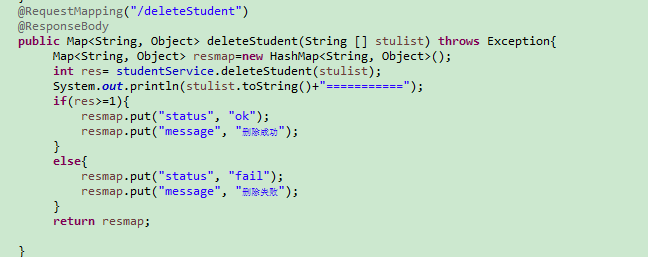
查询：



修改：



删除：

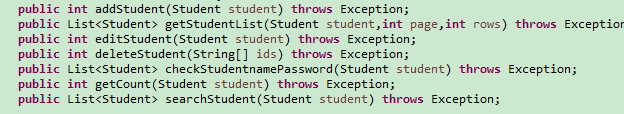


文件上传：

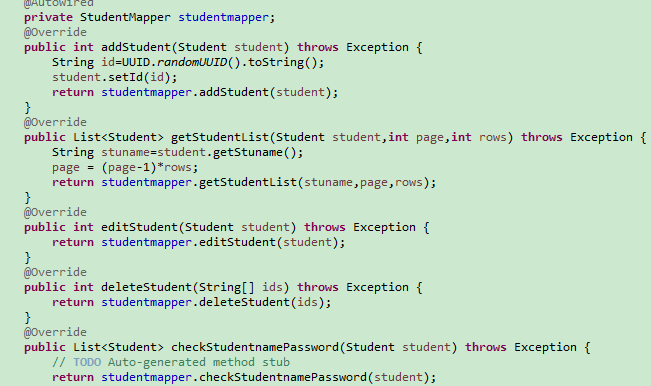


Service中

接口：

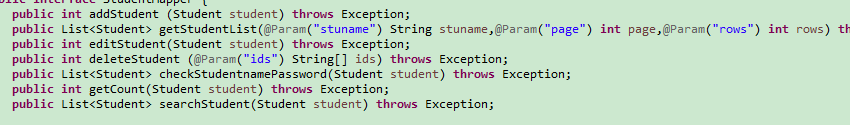


接口实现类：



mapper中

接口：



接口实现xml：  


十一、总结

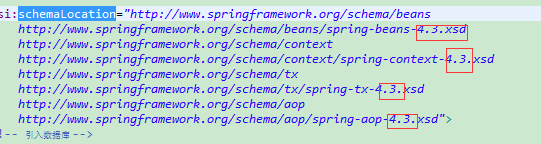
1. 有关配置文件

（1）日志管理问题，需在mybatis.xml下做相关的配置，要注意所放位置，在properties

标签之后，配置为：



（2）在配置spring和springmvc时，由于高版本问题，需要将配置的schemaLocation书写对应的版本：



1. 有关注解

在resquestMapping中区分大小写，否则调用不到指定的方法。

在Controller中返回值为String类的的方法，即为调用跳转页面的方法。

1. 有关引包

自己的经常犯的一个错误，复制权限命名时经常会复制.java的权限命名，在大多数情况下复制的应该为.class命名。

1. 有关web服务的理解

在项目发布后，我们所运行的的项目就在tomcat下了，其中的绝对路径指的是tomcat下的路径。

1. 有关前端easyui

还需要多看多学，前台掌握薄弱，对于json和ajax的传值和定义方式需要多了解多运用。