**Web作品说明**

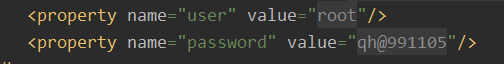
课程结束要求**提交设计程序和作品说明书。**

1. 程序导出为war文件，**一定不要忘记导出源代码，**数据库导出为sql文件，作品说明书为word文档。
2. 将所有文件压缩，压缩文件名为：班级—学号—姓名—系统名称
3. 将压缩文件作为附件发送到邮箱**lyzxf2015@163.com**中。
4. 截止日期2020. 6.15。

小型图书馆管理系统

一、系统开发的意义

* 1. 作品简介及作品说明：

首先这个作品基于ssm的作品设计使用，这里我要说明一下大作业同样使用ssm来完成，防止老师这边环境不同不能运行及可能出现的问题我将录制两个关于大作业和作品的网页内容，并在简单说明一下这些作品的使用，如果老师有时间可以去看一看还有我的数据库是有密码，老师如果不需要可以将bean.xml中的

这两行删除，好的我们开始讲解系统功能

* 1. 功能介绍：

首先我们通过登陆，来到首页 这里我们使用了一个AdminLte前端模板来装饰我们的页面，前端左侧菜单栏如下,当然是一个图书管理系统，这个图书管理系统不止能实现图书的增删改而且能实现访问日志的记录，登陆用户的管理以及权限管理还能模拟借阅和查询功能，登陆修改密码注销登录等功能让我们以下介绍以下这些功能

二、系统的基本功能

2.1 介绍：

图书管理中左侧栏都是我们重点介绍的功能





以及个人查询

2.2图书管理

2.2.1查询：

一点击图书管理就会来到这个页面



2.2.2 分页查询：这里帮助我们查询出来了所有的图书信息，这一页我们实现了分页查询可以通过底下的123就可以等进行分页查询

2.2.3 添加：

点击新这里我们可以新建一个书籍，这样我们就来到这个页面



点击保存他会替我们跳到查询页面



找到最后一页：



会看见我们添加的数据已经添加进来了

2.2.4 更新：

点击修改会进入这么一个页面



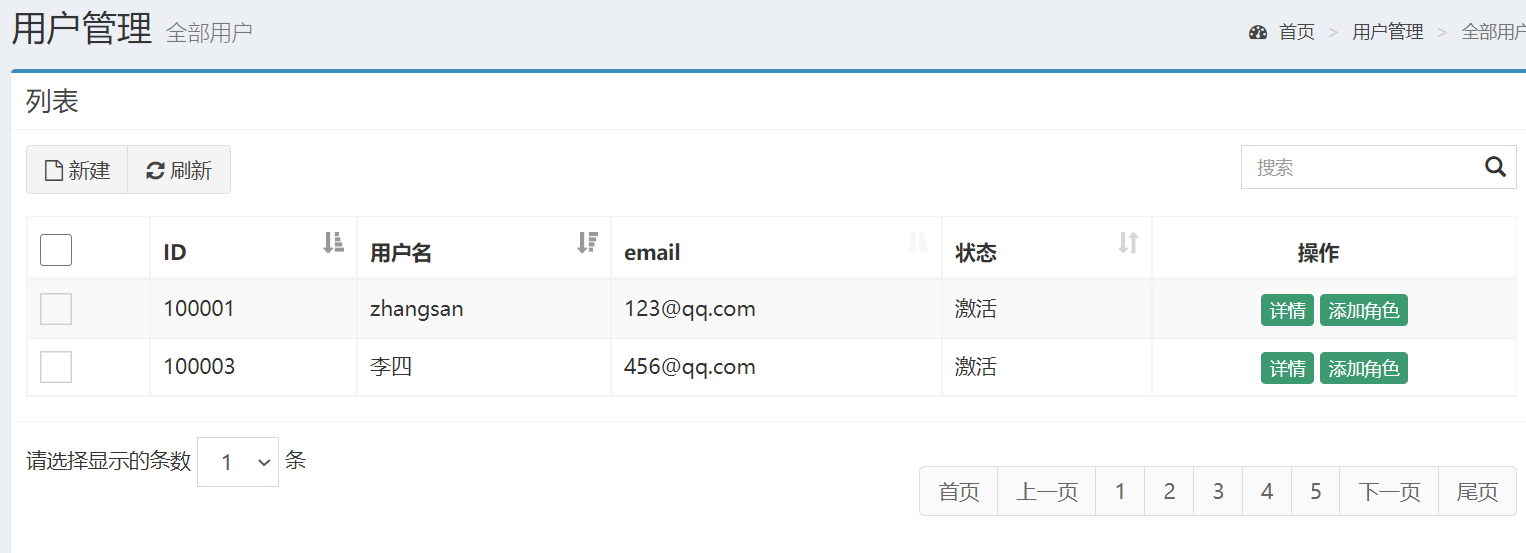
和刚刚添加差不多，只是执行的update但是默认提示了我们之前的数据，这里就不多做演示

2.2.5 订阅：这里买个关子到后面我们的个人我的借阅以下

2.3 用户管理：

2.3.1用户管理

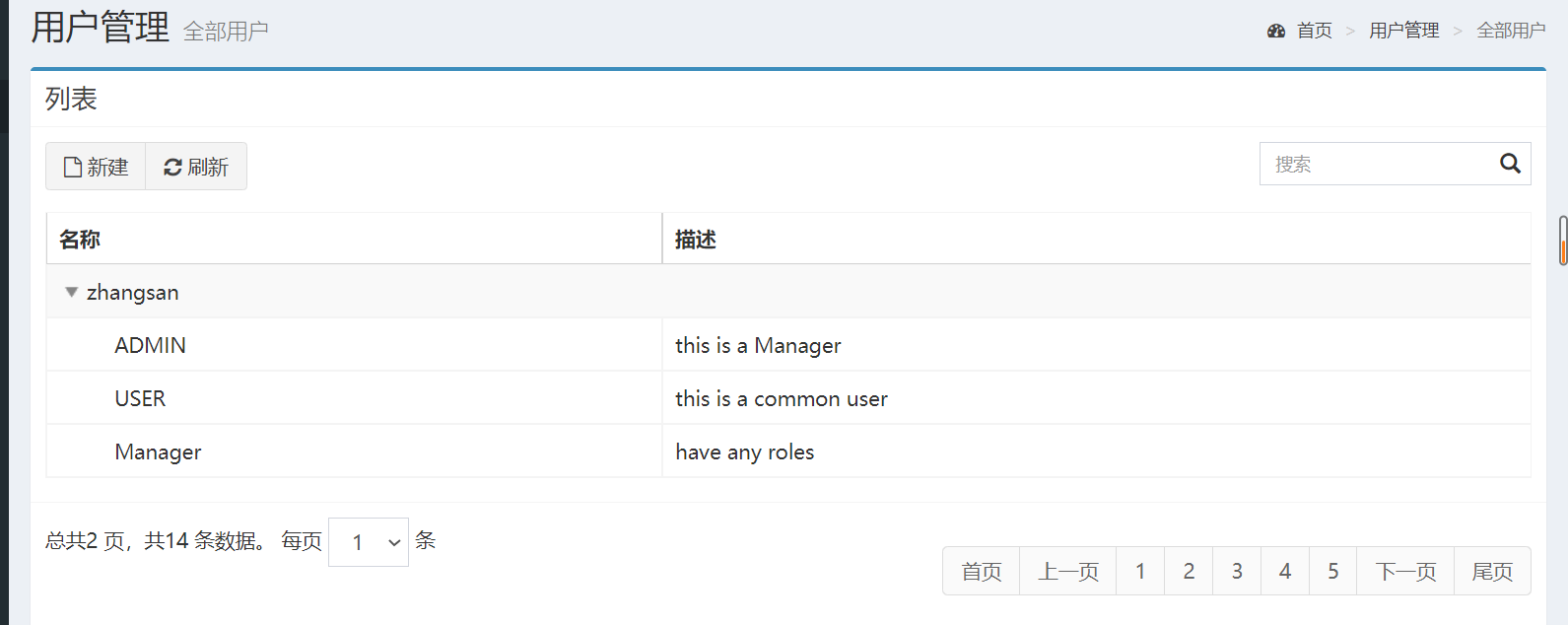
点击用户管理我们来到查询所有用户界面



这里没什么好说的查询出来所有用户的数据

2.3.2 详情：

这个界面就很有意思了



这里我使用adminLte的前端模板中的树表格，其次呢zhangsan这个用户下面是什么，这里我们还有一个表叫角色表，这张表的作用就是表示权限，与用户关联以分配给其相应的权限，这里它们两者之间有一张关联表，这个表显示出来的数据就是通过连接查询查询出来的用户表下所关联的权限

2.3.3 添加角色：

点击添加角色来到此页面



可以通过右边的复选框来给角色添加权限，点击保存就会在连接表中插入一条数据，以关联角色和用户

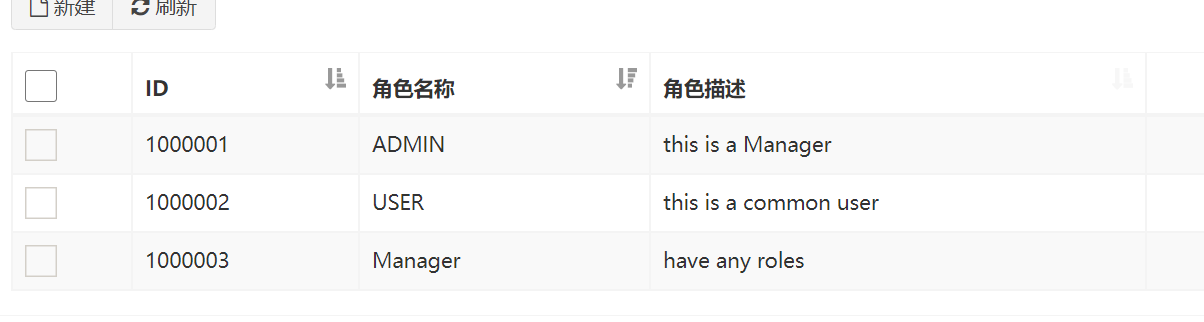
2.3.4 新建



这个没什么好说的就是新建一个用户，这里有一点这个密码会被我在service层中执行加密，所以不是原有的保存密码

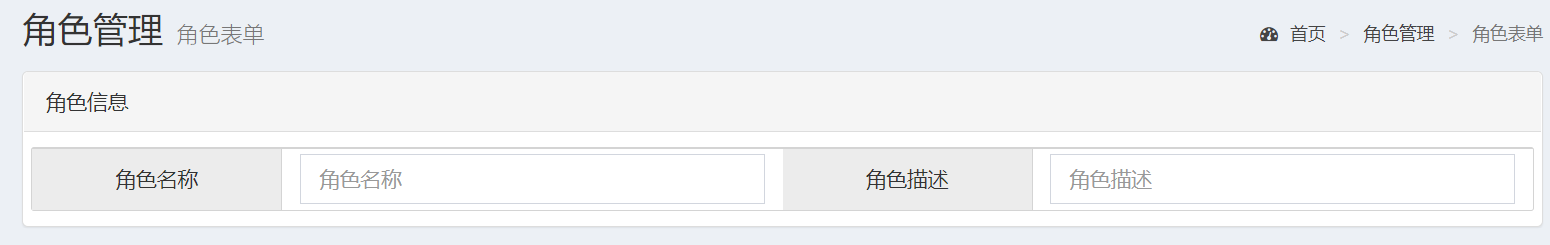
* 1. 角色管理：

2.4.1查询：



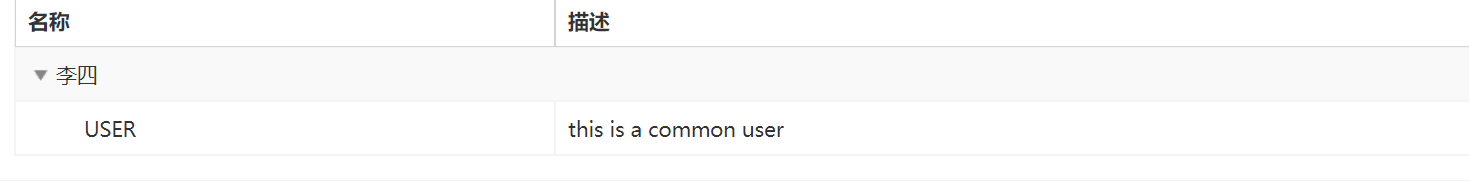
没什么好说的一个findAll方法

2.4.2 新建



也是插入一条角色，角色就这两个操作，可能看起来很鸡肋，但是角色在我的系统中是很重要的。

我们回到user的详情



Zhangsan的用户是有三个角色，李四的用户只有一个角色，点击右上角头像，点击注销，我们用李四来登录看看

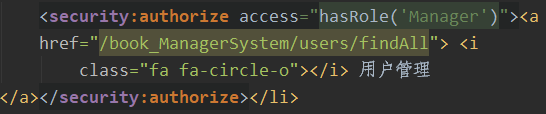




只能看见这两条管理



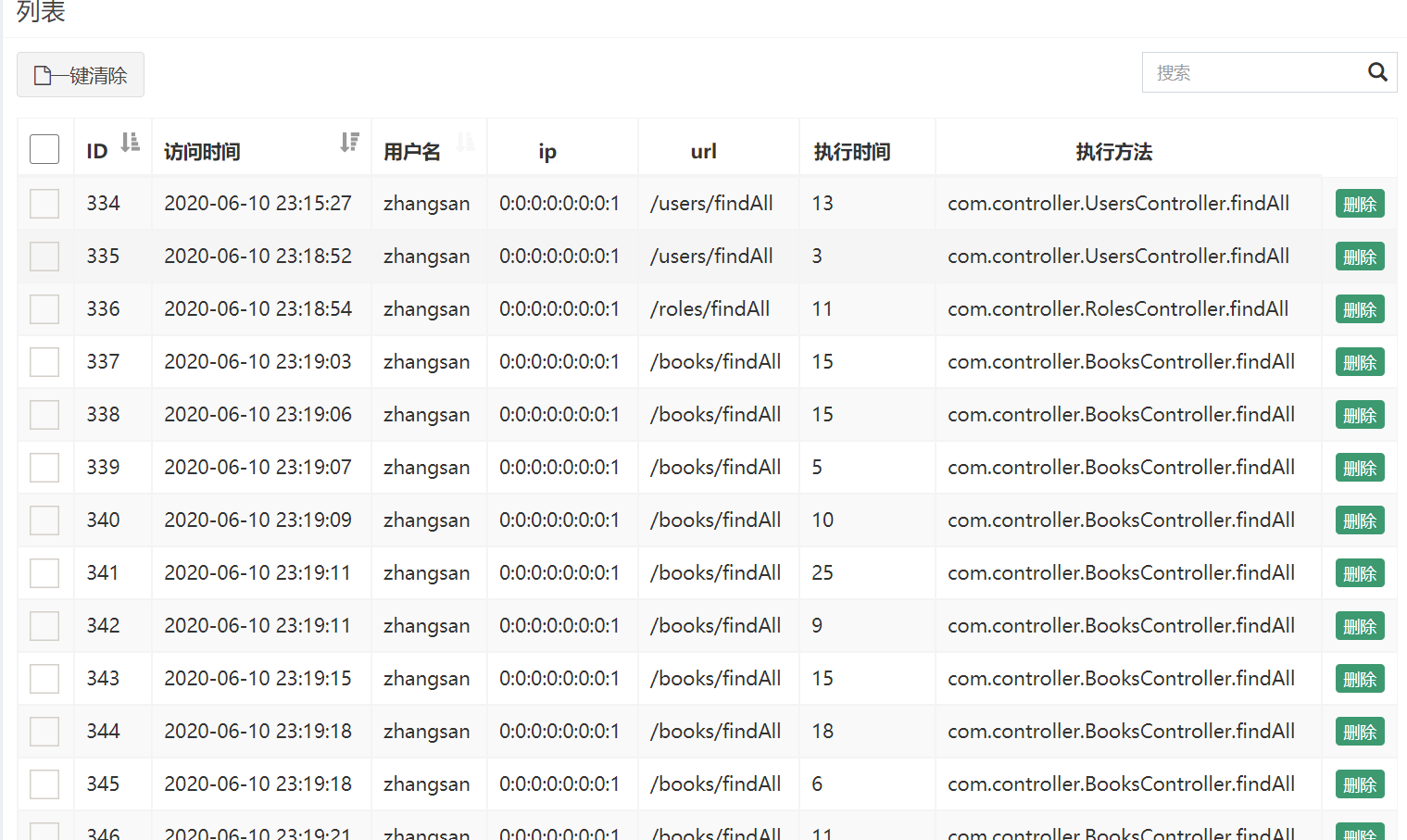
点击图书管理也只有借阅功能，这里我模拟zhangsan为管理员而李四只是一个普通用户，普通用户只需要借阅归还就可以了，至于怎么实现的，得益于Spring-security框架的强大



只有在前端配上这么一个标签，只有拥有Manager用户的人才能看见这个功能

2.5日志管理：

点击日志管理



看见这么一张图，这么多数据都是什么，这里是将用户访问了我们网站的哪些目录以及这些用户ip是什么访问了哪些方法，假设有用户非法访问我们网站，也可以通过日志观察到（当然这是不可能的），不想看的日志可以删掉，如果感觉一条一条删很累，左上角还有一个一键清除，这里使用了Aop的前置通知和后置通知实现aop日志管理，后面我们在代码中简单说以下。

2.6 我的借阅--借阅-归还功能：

这是我个人最喜欢的功能，点击个人数据我们会看见这么一条数据



在回到图书管理：



我们会看见这本书的状态变为不可借阅，如果此时继续点击右边的借阅会发生什么，会什么也没发生，这句代码已经在相应的service中处理过了，所以此时不会有影响，回到页面点击归还



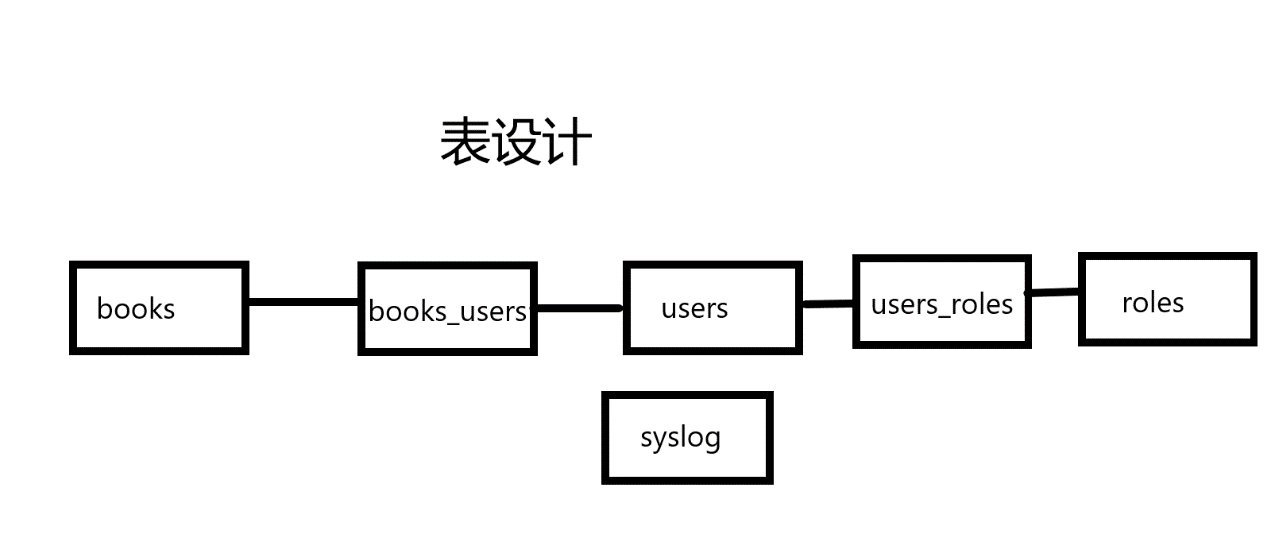
再回来查看以下



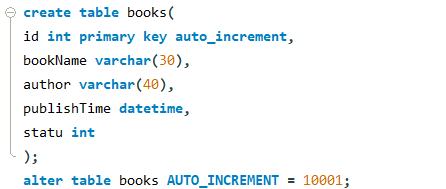
这本书又可以借阅了，重点代码我将在后面的代码介绍中重点介绍功能。

三、数据库设计

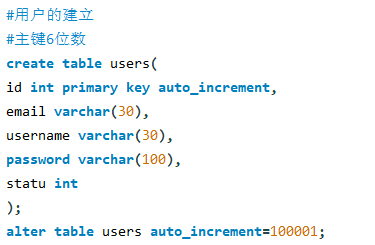
数据库一共有6张表



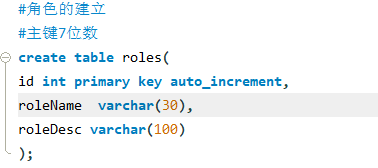
3.1 图书表：最重要的表，用于管理图书



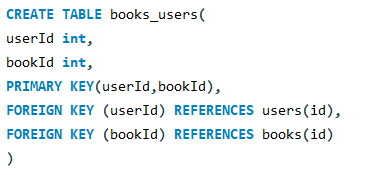
3.2 用户表：用于管理用户数据的表



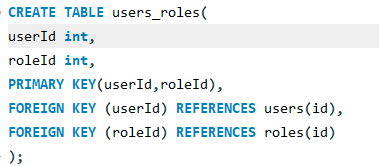
3.3 角色表：管理角色数据的表



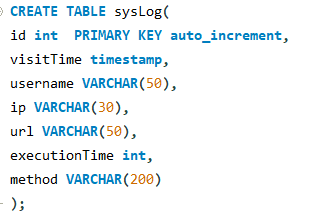
3.4 图书-用户表：用来关联图书和用户的：



3.5 用户-角色表：用来关联用户和角色的：



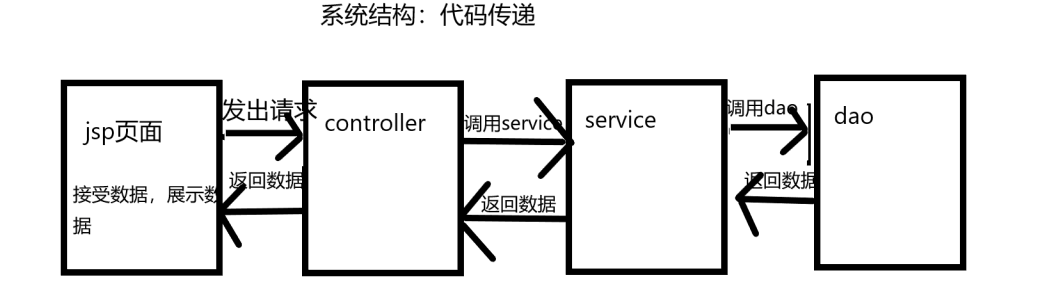
3.6 日志表：记录日志



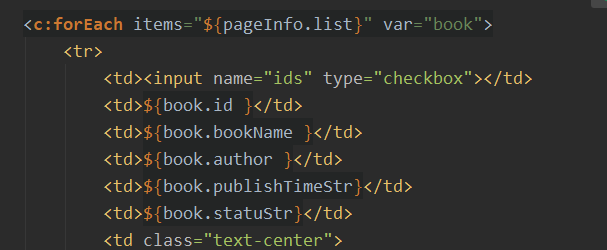
四、系统设计

1.系统的总体结构,系统的流程图等。

如下：



在页面中我们一般使用el表达式和c：foreach标签来取出数据



在代码中一般采用controller-》service-》dao这么一个顺序，有需要处理的逻辑就在service层中处理以下，没有就上面哪个箭头调用过去

2.系统的软件开发环境

操作系统：windows

开发软件：idea

运行软件：tomcat7插件

数据库软件（DBMS）：mysql

五、详细实现

1．系统实现

以表格形式，总结出系统具体的页面和java类，及各页面和java类的功能。

5.1 页面

|  |  |
| --- | --- |
| 页面名称（或java类） | 功能描述 |
| aside.jsp | 侧边栏的页面 |
| book-add.jsp | 添加书籍的页面 |
| book-list.jsp | 展示所有书籍的页面 |
| book-update.jsp | 更新所有书籍的页面 |
| header.jsp | 页面的头部 |
| login.jsp | 登陆页面 |
| main.jsp | 将head和aside包含进来的主页面 |
| mybook-list.jsp | 我的借阅页面 |
| role-add.jsp | 角色添加页面 |
| role-list.jsp | 角色展示页面 |
| syslog-list.jsp | 日志展示页面 |
| user-add.jsp | 用户添加页面 |
| user-role-add.jsp | 给用户添加角色页面 |
| user-show.jsp | 用户展示所有角色页面 |

5.2 Controller包

|  |  |
| --- | --- |
| 页面名称（或java类） | 功能描述 |
| BookController | 用于拦截所有图书管理相关的页面处理实现跳转拦截，重定向，调用service方法等 |
| RolesController | 用于拦截所有角色管理相关的页面处理实现跳转拦截，重定向，调用service方法等 |
| SysLogAop | 这个类比较特殊，使用前置和后置通知，加强controller中的方法 |
| SysLogController | 用于拦截所有日志管理相关的页面处理实现跳转拦截，重定向，调用service方法等 |
| UsersController | 用于拦截所有用户管理相关的页面处理实现跳转拦截，重定向，调用service方法等 |

5.3 service包：

Service分为接口和实现类，实现类就是实现了这些接口的方法，所以我们重点介绍一下实现类

|  |  |
| --- | --- |
| 页面名称（或java类） | 功能描述 |
| BookServiceImpl | 主要作用调用dao层方法IBooksDao实现相应操作,并对controller方法的请求加以逻辑处理 |
| RolesServiceImpl | 主要作用调用dao层方法IRolesDao实现相应操作 |
| SysLogServiceImpl | 主要作用调用dao层方法ISysLogDao实现相应操作 |
| UserServiceImpl | 主要作用调用dao层方法IUsersDao实现相应操作 |

5.4 dao层：

|  |  |
| --- | --- |
| 页面名称（或java类） | 功能描述 |
| IBooksDao | 调用相关sql语句，实现对数据库books表的增删改查 |
| IUsersDao | 调用相关sql语句，实现对数据库Users表的增删改查 |
| IRolesDao | 调用相关sql语句，实现对数据库Roles表的增删改查 |
| ISyslogsDao | 调用相关sql语句，实现对数据库Syslog表的增删改查 |

5.5 domain

|  |  |
| --- | --- |
| 页面名称（或java类） | 功能描述 |
| Books | 建立Books表实现与数据库的字段值一一对应，封装数据库的字段的数据，以此增删改查 |
| Books\_Users | 这里有一次比较麻烦的连接查询建立连接表的实体类，对应books\_users表的字段 |
| Roles | 建立Roles表实现与数据库的字段值一一对应，封装数据库的字段的数据，以此增删改查 |
| UserInfo | 建立Users表实现与数据库的字段值一一对应，封装数据库的字段的数据，以此增删改查,这个表叫这个名字的原因是Users在spring-security已经有这么一个类，为了防止重名，而带来可能发生的错误 |
| SysLog | 建立SysLog表实现与数据库的字段值一一对应，封装数据库的字段的数据，以此增删改查 |

5.6 Utils帮助类:

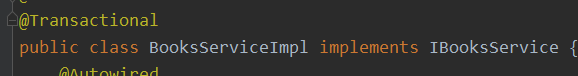
|  |  |
| --- | --- |
| 页面名称（或java类） | 功能描述 |
| DateUtils | 实现对时间的格式转换，将字符串封装成对应的时间，和将时间封装成对应的字符串格式 |
| BcryptPasswordEncoderUtils | 加密帮助类，因为登陆密码是加密的，事先需要有一个账户，通过这个类可以算出加密后的密码是什么，以此插入一条数据进数据库 |

2．具体实现示例

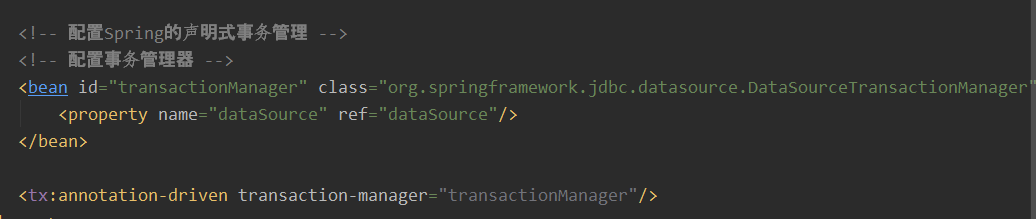
以某个自己比较满意的页面为例，给出部分具体的实现代码。

5.7 订阅查询功能：

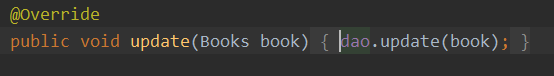
这个功能是我个人最满意的功能，因为这个功能成功体现了spring的事务管理的功能在此之前，每次进行service层的代码编写时都会在service层类上加上这么一个注解



@Transaction注解实现了事务控制，这也是spring框架神奇之处之一，只要你配置了相应的事务管理器，并开启注解配置



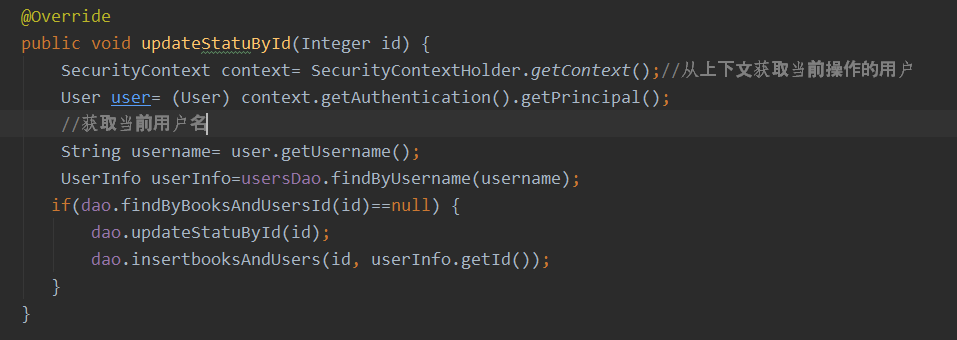
但这个注解一致都配置着，一直没有什么用，但借阅查询功能使得它起了作用



一般的service层如上面一样只需要调用dao层一句代码即可，此时事务管理基本上起不到什么作用做一次操作

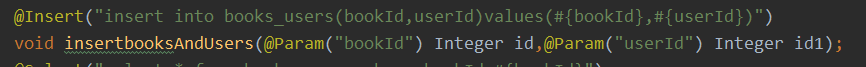
我们看看订阅查询功能

订阅：

service

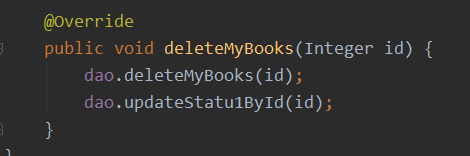
dao：

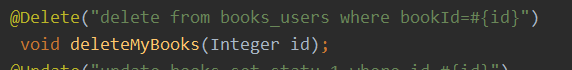


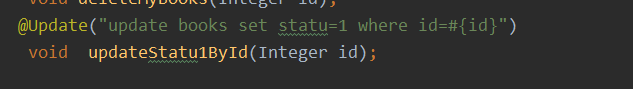


订阅首先要将图书的状态设为0即不可借阅，后再往连接表中插入一条数据，以关联我们的数据，这意味着发一次请求即一个controller请求，一次service调用，那么对应的是两次dao层调用，在初次的调试时我的第二句插入代码即插入出现了异常，spring的事务管理非常好的处理了这一点，立马回滚我的service层方法，使得数据的操作撤销，如果没有使用事务管理，将会看到书的状态被设为不可借阅，但我的借阅中又没有这本书，这就是一次误操作，唯一的方法就是打开数据库自己再把图书的状态设置为可借阅状态。

归还也是同样：





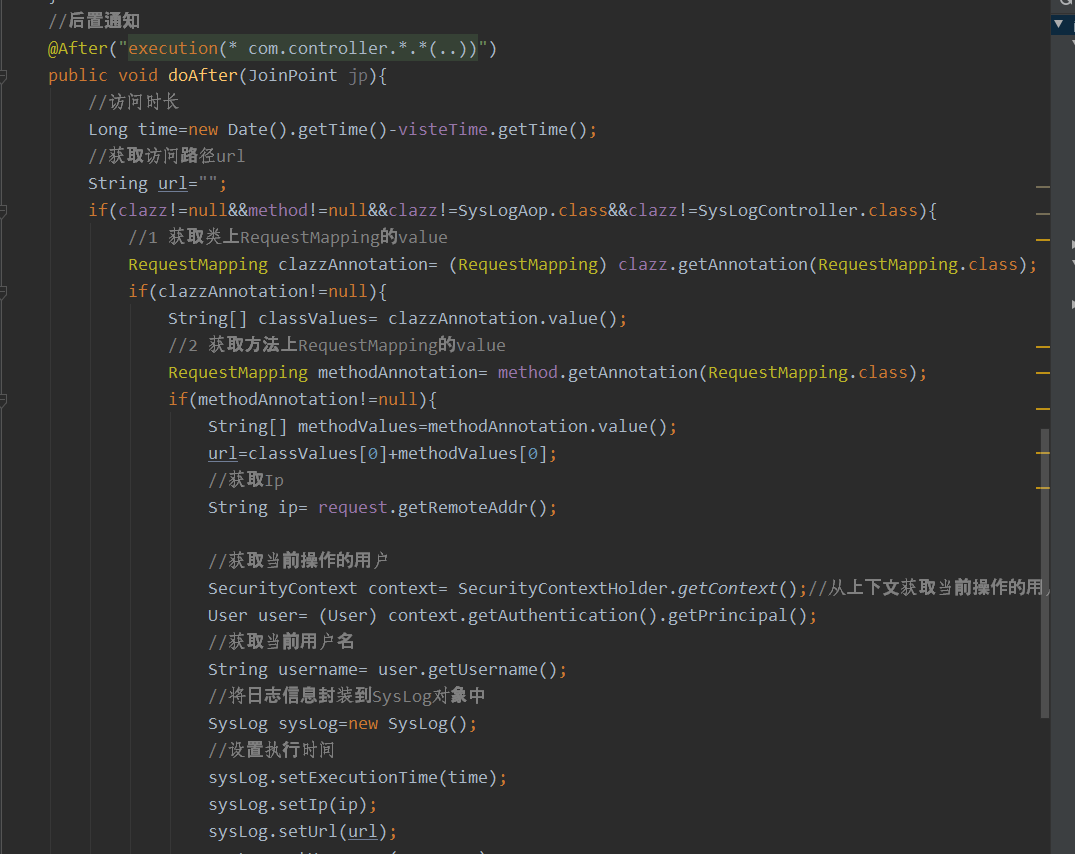


两句代码，同样需要事务控制回滚

5.8 日志管理：

日志管理我们通过aop实现了日志管理，首先我要说明一点，这里的许多代码不是由我独立完成的，因为之前的练习中已经跟着教课的老师敲了一边，这里我直接拿过来用了：





这里用到了大量的反射技术去获取我们需要的数据，这可以说是最考验反射知识内容掌握程度的代码了

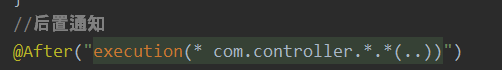
我们重点说明一下这几个注解



Aspect配置这是接下来这个类是一个切面

Component组件注入，将这个类交给spring去管理，这也就是为什么@Autowired注解能自动注入的原因了





Spring有两大功能：一个是ioc依赖注入，另一个是aop面向切面编程：

切面个人理解就是通知和切入点的结合。

@Before表示接下来这个方法要配置一个前置通知，并且指定给所有com包下的controller类配置前置通知，即切入点就是com.controller中的方法，作用就是在方法之前通过类似动态代理这样的技术将方法在执行之前切一刀放到方法的最前面

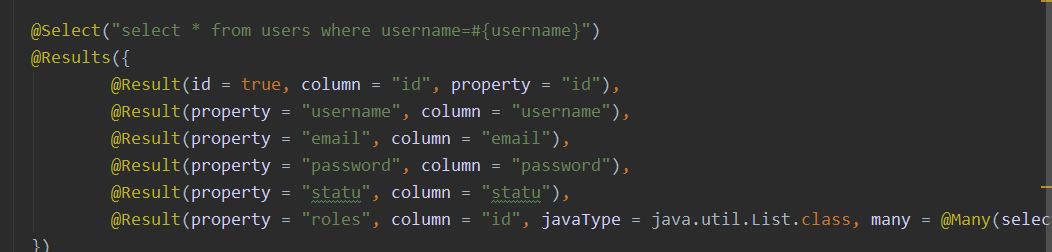
@After表示接下来这个方法要配置一个后置通知，并且指定给所有com包下的controller类配置后置通知，即切入点就是com.controller中的方法，作用就是在方法之前通过类似动态代理这样的技术将方法在执行的最后面切一刀放到方法的最后面。

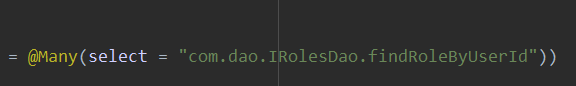
上面合起来的作用就是：在你执行任何请求controller执行跳转，重定向之前都会执行一次前置通知上配置的方法，在方法执行最后，会执行后置通知中的方法，这样就可以不更改原有代码的基础上更改现有的代码，实现日志的添加。

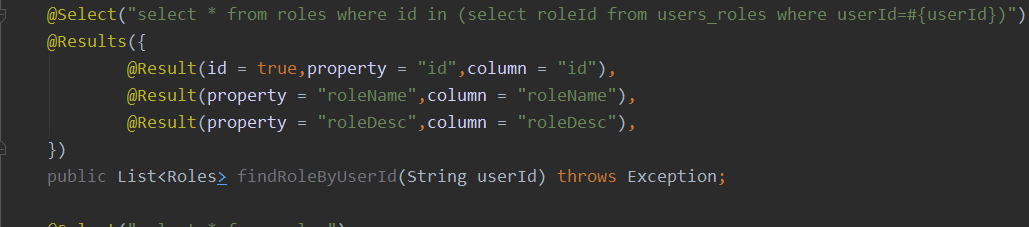
5.9 连接查询：

这里介绍一个连接查询，就是用户详情的查询界面

权限查看



重点说一下最后这句@Result这里的意思是根据这张表的id封装roles数据，通过IrolesDao中的方法findRoleByUserId查询结果去封装我们的roles



这里就将上一个需要的id传给了当前这个方法当作userid做一次子查询将查询的结果再封装回roles

六、总结

这个系统的特点就是有一个借阅归还功能和日志管理功能，这两个功能比较有意思。其次在技术上只能说还有不少需要学习的地方，毕竟框架难点在框架设计和源码上而非在使用上，使用的话，现在技术更新的飞快，这个框架可以学会用，下个框架就又要重新学，所以个人认为，了解一个框架的设计思想和源码相比使用更重要，再比如后面还没有接触的spring boot等，只能说学习可能是永无止境的把。

七、附录

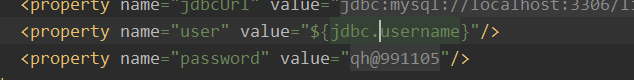
在插入完成数据库的表之后，默认有一个登陆的账号和密码

1

用户名：zhangsan

密码：123456

1. 就是大作业的设计我不是按照实验要求来做的，是根据当前这个图书管理系统改出来的，如果老师有需要更改可以再通知我
2. 对这两个作业额外增加了一个说明视频，老师有空可以看看
3. 如果老师不需要登陆数据库的用户名和密码



把jdbcurl下的这两句话删除掉