

20小组-张皓捷-19302010021-Lab4实验报告

DevCloud中对Lab4进行项目需求规划（Scrum项目）的截图

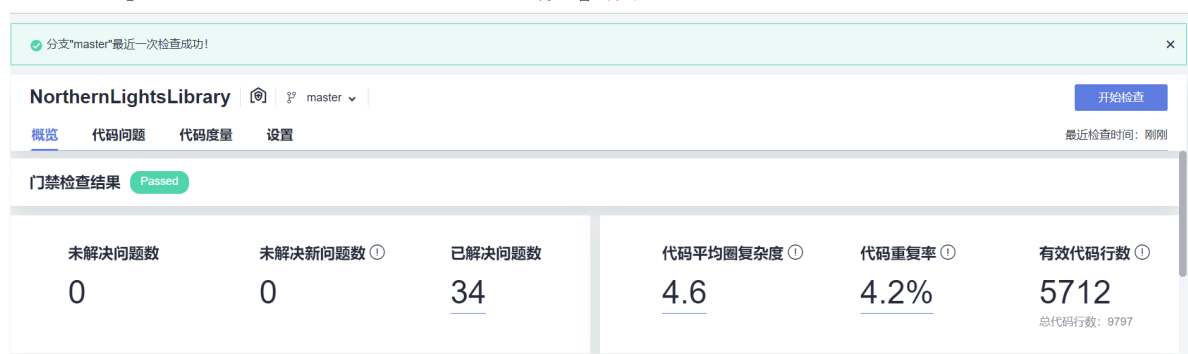


The screenshot shows the DevCloud Backlog interface for the 'NorthernLightsLibrary' project. The 'Backlog' tab is selected, and the filter is set to '全部' (All). The table lists tasks with columns for '处理人' (Assignee), '预计开始日期' (Estimated Start Date), '标题' (Title), '优先级' (Priority), '创建时间' (Creation Time), and '实际工时' (Actual Time). The tasks are as follows:

处理人	预计开始日期	标题	优先级	创建时间	实际工时
张皓捷-193020...	2021/05/11	Feature 系统记录	中	2021/05/11 15:52:03 GMT+0	0.00
张皓捷-193020...	2021/05/11	Feature 罚款的缴纳	中	2021/05/11 15:51:53 GMT+0	0.00
朱亦文-193020...	2021/05/11	Epic 罚款的添加	中	2021/05/11 15:51:40 GMT+0	0.00
朱亦文-193020...	2021/05/11	Feature 预约借阅超期	中	2021/05/11 15:51:02 GMT+0	0.00
朱亦文-193020...	2021/05/11	Feature 读者身份区分	中	2021/05/11 15:50:17 GMT+0	0.00
张皓捷-193020...	2021/05/11	Feature 注册学邮验证实现	中	2021/05/11 15:49:37 GMT+0	0.00
张皓捷-193020...	2021/04/28	Task 在线预约	中	2021/04/13 15:44:15 GMT+0	0.00

At the bottom, it shows '总条数: 15' (Total items: 15) and a pagination control for page 1 of 1.

DevCloud中对Lab4的项目代码使用代码检查的检查结果截图



The screenshot shows the 'Code Check' results for the 'NorthernLightsLibrary' project. The status is 'Passed'. The table displays the following metrics:

未解决问题数	未解决新问题数 ①	已解决问题数	代码平均圈复杂度 ①	代码重复率 ①	有效代码行数 ①
0	0	34	4.6	4.2%	5712

Additional information includes '最近检查时间: 刚刚' (Last check time: Just now) and '总代码行数: 9797' (Total lines of code: 9797).

利用Git对Lab3截止日期前的最后一次commit打tag的截图

```
commit e21a06049895ce3e5a9alc19cf7aade43847aa62 (tag: lab3-real-finish)
Author: 张皓捷 <zhj630985214@gmail.com>
Date: Mon May 3 22:13:09 2021 +0800
```

完成了最终的文档

利用Git给Lab4截止日期前的最后一次commit打tag的截图

```
lab4-finish
PS D:\软件工程\NorthernLightsLibrary> git log
commit 7ad5f976f363fa847b49fbd4fab5ac30e14c0bd5 (HEAD -> lab4-dev, tag: lab4-finish)
Author: 张皓捷 <zhj630985214@gmail.com>
Date: Sun May 23 23:10:49 2021 +0800
```

更新实验报告

项目各个页面的截图

首页

首页我的账户 ▾管理员

superadmin: admin
邮箱

书名作者名ISBN查询

计算机
科学
丛书

第10版

软件工程

作者: Ian Sommerville

ISBN: 1234567891

《软件工程（原书第10版）》是机械工业出版社出版的图书，该书作者是（英）Ian Sommerville。

Sun May 23 2021 19:56:37 GMT+0800 (中国标准时间) 查看详情

登录页

登录

* 用户名

请输入用户名

* 密码

请输入密码

* 登陆身份

☐ 普通管理员

☐ 超级管理员

☐ 普通读者

登录

重置

游客登录 | 还没有账号? 先注册

注册页

注册

* 用户名

学/工号

* 密码

* 确认密码

* 选择身份

☐ 教师

☐ 本科生

☐ 研究生

邮箱

验证邮箱

* 验证码

提交

重置

游客登录 | 已有账号? 登录

修改密码页

修改密码

* 旧密码

旧密码

* 新密码

新密码

* 确认密码

确认密码

提交

重置

现场还书页

首页

我的账户

管理员

Asuperadmin: admin
邮箱

现场还书

现场借书

现场取书

上传新书

添加副本

操作查询

设置时长

新管理员

* ISBN 1

* 状态 1

请选择

新增书本

现场还书

现场借书页

首页我的账户 ▾管理员

现场还书

现场借书

现场取书

上传新书

添加副本

操作查询

设置时长

新管理员

* ISBN 1

新增书本

* 读者学/工号

立即借书

现场取书页

首页我的账户 ▾管理员

现场还书

现场借书

现场取书

上传新书

添加副本

操作查询

设置时长

新管理员

* 读者学/工号

立即查询

上传新书页

首页我的账户管理员

superadmin: admin 邯郸

现场还书

现场借书

现场取书

上传新书

添加副本

操作查询

设置时长

新管理员

上传封面

将图片拖到此处，或 点击上传

提示：可支持PNG、JPG、BMP，图片大小不超过4M,长边不大于4096像素，请保证识别的部分为图片的主题部分

书本名称

ISBN

出版日期

金额

作者

简介

立即上传

取消

添加副本页

首页我的账户管理员

superadmin: admin 邯郸

现场还书

现场借书

现场取书

上传新书

添加副本

操作查询

设置时长

新管理员

ISBN

校区

副本数量

添加副本

设置用户借阅副本数目/预约借阅时长页

首页我的账户管理员

superadmin: admin 邯郸

	身份	借阅副本数	借阅时长	预约时长	操作
现场还书	teacher	10	29天22时30分0秒	30天0时0分0秒	编辑
现场借书	undergraduate	10	30天0时0分0秒	30天0时0分0秒	编辑
现场取书	postgraduate	10	30天0时0分0秒	30天0时0分0秒	编辑
上传新书					
添加副本					
操作查询					
设置时长					
新管理员					

新管理员页

首页我的账户 ▾管理员

superadmin: admin邯郸

现场还书

现场借书

现场取书

上传新书

添加副本

操作查询

设置时长

新管理员

用户名	Email	操作
<input type="text"/>	<input type="text"/>	<div>新增重置</div>
admin	admin@admin.com	
zhanghaojie	zhj63095214@gmail.com	<div>删除</div>

查看副本页

首页我的账户 ▾管理员

superadmin: admin邯郸

书名	作者	ISBN	所在分馆	状态	借书者	操作
软件工程	Ian Sommerville	1234567891-1	邯郸校区图书馆	可用（在架上）		<div>查看记录</div>
软件工程	Ian Sommerville	1234567891-2	邯郸校区图书馆	可用（在架上）		<div>查看记录</div>
软件工程	Ian Sommerville	1234567891-3	邯郸校区图书馆	可用（在架上）		<div>查看记录</div>

预约页

首页我的账户 ▾

undergraduate: 19302010075

书名	作者	ISBN	所在分馆	状态	借书者	操作
软件工程	Ian Sommerville	1234567891-1	邯郸校区图书馆	可用（在架上）		<div>预约</div>
软件工程	Ian Sommerville	1234567891-2	邯郸校区图书馆	可用（在架上）		<div>预约</div>
软件工程	Ian Sommerville	1234567891-3	邯郸校区图书馆	可用（在架上）		<div>预约</div>

用户信息页

首页

我的账户

undergraduate: 19302010075

基本信息

已借书籍

预约书籍

操作记录

用户名	身份	信誉积分
19302010075	undergraduate	100

用户记录页

首页

我的账户

undergraduate: 19302010075

基本信息

已借书籍

预约书籍

操作记录

预约记录

借阅记录

还书记录

罚款记录

日期	用户名	ISBN
暂无数据		

软件设计文档

另附，位于此报告的最后。

产品使用指南

IP地址及端口

ip地址：47.101.35.239 端口80

如何使用管理员账号

系统默认自带了一个超级管理员账号。账号名为admin，密码默认为admin，登录成功后可自行修改密码。

可以使用超级管理员账号添加其他的（普通）管理员。账号名可以在添加管理员时自己设置，密码默认为111111，登录后可以自行修改密码。

组员的分配情况

张皓捷，朱亦文：后端
项心叶，周思瑜：前端

小组的实验过程记录，遇到的问题以及解决方案

实验过程记录

91d5979 (HEAD -> master, origin/mastet, origin/master, origin/HEAD) Merge branch 'lab4-dev'

f002bdd (lab4-dev) Merge branch 'backend' into lab4-dev

a27167f (origin/backend, backend) 修正了FileNotFoundException没有继承接口FileNotFoundException的问题

ea8a2f7 修改了上次push的错误...

2bc2326 实验报告框架上传

6eb8cce 按照华为云的要求解决问题

dbf00b0 Merge branch 'backend'

ea56f33 再次尝试使用465端口发邮件

f691b1f (origin/lab4-dev) Merge branch 'backend' into lab4-dev

6ae468a 邮件发送端口改成465

b5f9bb5 Merge branch 'frontend' into lab4-dev

adfd0d 合并backend分支

eb32a8e 配置了后端生产环境的配置文件

a3e9ae1 (origin/frontend, frontend) 关于提醒的前端显示问题

3a0285d 在批量借书的时候的批量信息里添加了具体的uniquebookmark，而不是只显示这本书

40842ed 问题1, 2, 3

e8776fe 将已支付的罚款缴纳按钮设置为disable

29af28d 修复了邮件发送提醒失败问题

b32f06a 更改了管理员查询罚款界面的金额显示问题

7276efe 修复了查询操作记录页面的显示问题

b57f468 更改了查询记录页面的显示bool

86ac11f 新增了预约和借阅的到期显示 即罚款缴纳按钮的函数实现

1834741 修正了后端/admin/record没有遵守规范的问题

2492851 增加了罚款缴纳按钮

5f39663 修复了设置时长的一些问题（input输入校验仍未完成）

97689f1 显示用户罚款的时候添加了fineID

78eafbb Merge branch 'frontend' of codehub.devcloud.cn-north-4.huaweicloud.com:2021rjgc_d20xz00001/NorthernLightsLibrary into frontend

8e7f492 后端添加了批量借阅的接口

c90e659 修改了罚款金额显示 修改了设置时长的get问题

79c9980 修改了罚款金额显示 修改了设置时长的get问题

88a09b2 修正了message的显示问题。。。

a1c4046 Merge branch 'backend_temp' into backend

9e54379 修正了postgraduate不能预约书的问题，添加了预约到期但管理员还没取消时，不能借书的限制

15fc591 修改了message无法正常显示的问题

bf30ce2 修改了身份认证

e32f2f2 新增了成功显示信息

2be1d5e 增加了价格必须大于0的验证

363e1cb 修改了undergraduate显示问题

2a3142e 隐藏了login中的测试代码

66c22af 修改了身份设置，将student改成了teacher, undergraduate, postgraduate

f7fa377 (backend_temp) 修改了配置文件的数据库密码

330ab1d 修改了管理员界面的两个图标

7ae7ae4 将发邮件的按钮由三个改为一个

4348654 Merge branch 'frontend' of codehub.devcloud.cn-north-4.huaweicloud.com:2021rjgc_d20xz00001/NorthernLightsLibrary into frontend

b780b25 将发送邮件的按钮由三个改为一个

e9b2a02 修改了网页图标和网页标题
b8ef7f9 修改并测试了管理员发邮件提醒用户的功能
cb8c461 按照前端的要求修改了setUserConfiguration的接口
2ca0026 修改了网页图标和网页标题
d358b9e 修改了网页图标和网页标题
4577b77 Merge remote-tracking branch 'origin/frontend' into frontend
abc74b0 增加了状态的提交
079cf61 Merge branch 'backend_temp' into backend
e7cd31c 完成了批量借阅的界面
2b6ad20 修改了登录时传入的参数
dcc276c 修改并测试了获得用户的各种记录，获得一本书下面所有的记录
49e909c 基本完成了TimeSet类(还有验证天时分秒的数字范围没写)
edcedb7 预约/借阅超期，罚款未缴纳邮件提醒功能完成
fb0ac51 完善了管理员查询操作记录类AdminRecord类
1e75e71 Merge remote-tracking branch 'origin/frontend' into frontend
d5a26b5 修改了注册界面，增加了身份选项以及邮箱验证
dee1d8b 完善了管理员查看每个副本历史记录 CopyRecord类
208b935 完善了用户类中显示自己的操作记录的UserRecord类
434cd7f 管理员查看副本历史记录功能完成
6b58ab6 用户&管理员查询用户系统记录功能完成
133eed5 修复了一些小问题，进行了测试
2254139 用户查看借阅/预约/归还历史记录功能完成
faf03d4 修正了还书时选择丢失的时候没有还书记录的问题
15fe768 完成了付罚款的功能
658e39a Merge branch 'backend' of codehub.devcloud.cn-north-4.huaweicloud.com:2021rjgc_d20xz00001/NorthernLightsLibrary into backend
653932d 完成了显示所有罚款的功能
9e79de3 罚款和还书的系统记录插入功能完成
f84fc3c 完成了设置用户最大借阅数目，最长借阅时间，最大预约时间的功能
d9225b0 测试超管设置用户最大借阅时长等三个属性
f8fd65f 超级管理员设置用户最大借阅/预约时间和副本数量功能完成
22056f1 增加系统记录相关的实体类
817448a 新增了几个类的大纲，修改了router
c2b387f 把/admin/receiveBookFromUser的接口的返回改成了List，需要麻烦前端改一下
57169e1 zhj重构了还书的业务逻辑，实现了lab4中还书可以设置状态丢失等等的需求
1f2f3f7 后端添加了罚款类
acfee5 为图书添加了价格属性
bf52c21 完成了借书时的deadline
c0b5e67 增加了预约的时候自动增加deadline的功能
009e8d4 完成了读者身份区分
5c508a4 上传了周二下午的讨论记录
27e5d35 zhj完成了对注册功能验证码的支持
15382ea 创建了EmailUtils类，但里面什么都没有，只是象征一下自己开始写lab4了

遇到的问题及解决方案

后端

问题1：一开始在自己机器上发邮件没问题，部署到服务器后发现发邮件会提示“cannot connect to server smtp.163.com:25”，最后发现是阿里云把25端口禁用了，改用465端口发，就好了

问题2：还书时“过期”、“没过期”、“损坏”、“丢失”、“完好”排列组合一下可能产生很多if-else，解决方法见设计文档的Package transaction

前端

问题1：在批量还书的界面遇到了一些问题。当只需要动态生成一个表项的时候，前端代码写起来还是比较简单的。但是当需要动态生成两个表项（即还书的ISBN和还书的状态）的时候遇到了一些问题。如果只放在一个表项里，那验证会有问题，如果放在表单里，那么会生成一个表单数组，表单的验证也同样会有问题。于是，最后把两个表项放在了一个 `<div>` 的块元素里，动态增减 `<div>`，就解决了。

问题2：在设置时长界面遇到了一些问题。总结构是表单内嵌套表格，而表格的某一行的表项是表单的表项，可以修改并且上传。我们发现使用element ui中自带的表项似乎无法正常显示，于是还是使用了原始html的元素，但是在验证表单时遇到了问题：表单根本无法验证！于是我们使用了一些小技巧，使得用户无法输入违法的表单，于是避开了表单验证。

问题3：在设置时常界面还遇到了不存在想要的组件的问题。我们理想的组件是可以滚动的，有小时分秒四个空格的可以滚动的选择器。但是很不幸，element ui没有提供这样的组件，于是我们手写了一个，将小时分秒变为4个input的输入框，并规定只能往输入框中输入数字。还通过js使得用户输入了过大的时间（比如25小时）会自动给天进一。

问题4：在前后端商定接口文档的时候，没有讲明传输数据的格式，导致前端在显示后端发来的数据的时候一开始错误显示了。

问题5：一开始没有很明确这个lab的需求，没有将各种提示放在一封邮件中发送，后来明确需求之后改正了

每个小组成员单独的实验总结

张皓捷的小组实验总结

百感交集，不知道写什么好，感谢大家一起完成这个lab。

朱亦文的小组实验总结

本次lab的任务对于后端来说不算少，但由于种种原因我也没能很早开工，谢谢同样负责后端的组长最后在功能调试以及完善上给我的帮助，希望最后一个lab可以有更充足的时间进行测试。

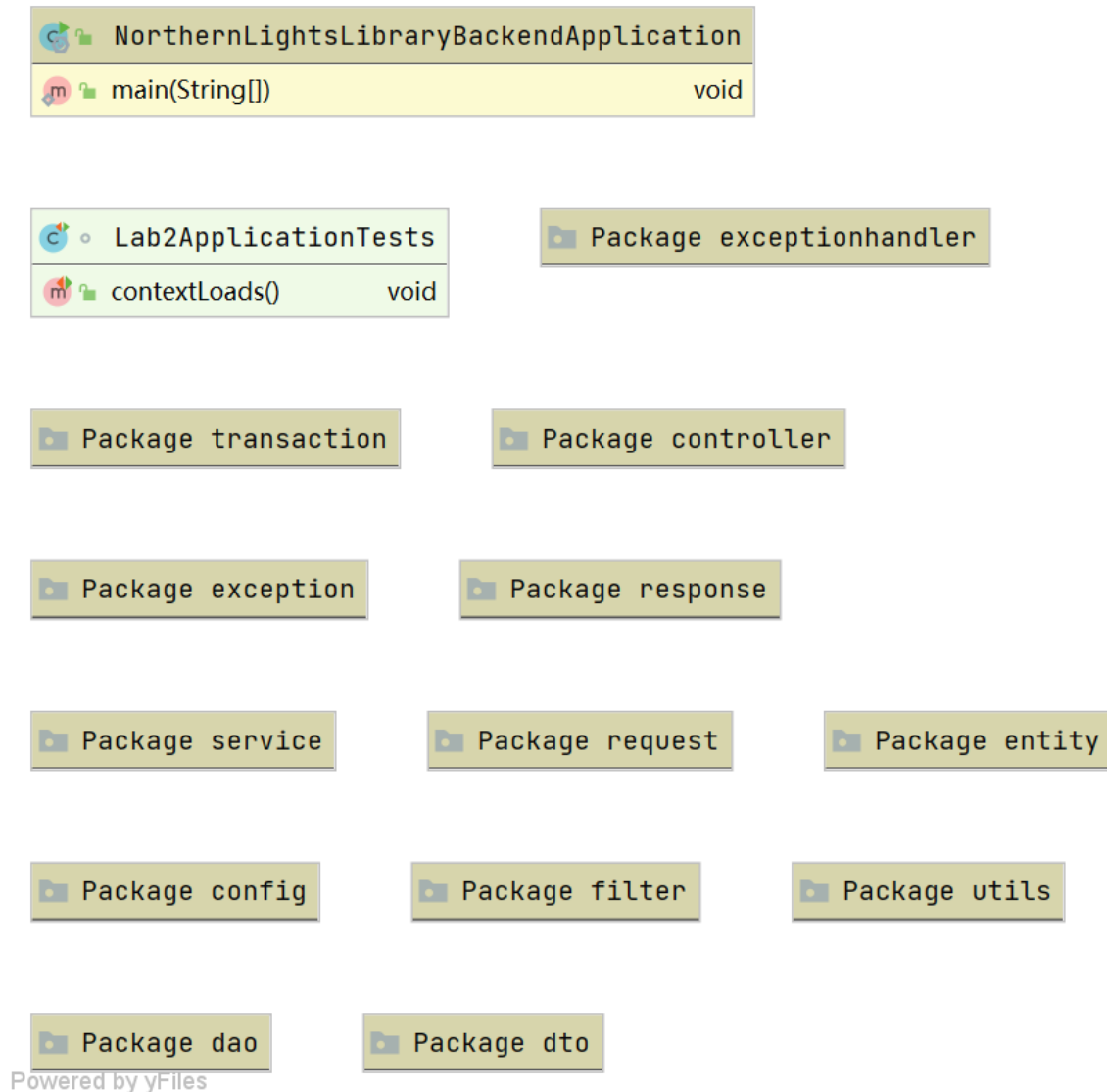
项心叶的小组实验总结

首先感谢组长，很有耐心地将前端的问题逐条整理在文档里，很大地方便了前端找bug。这次lab前端任务不多，但是时间仍旧比较紧。我堆到了ddl前夕写差点来不及.....感谢前端的另一位同学承担了很大的任务量，让我们的项目得以及时完成。

周思瑜的小组实验总结

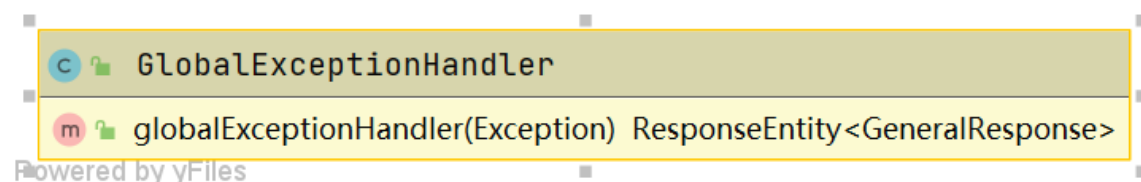
前端的嵌套字典赋值问题比较烦，搞了很久，这次lab新学了一个el-button-radio可以切换页面，其余搞得东西其实和上次比较相像，因此前端的任务量其实还行。

整体项目结构



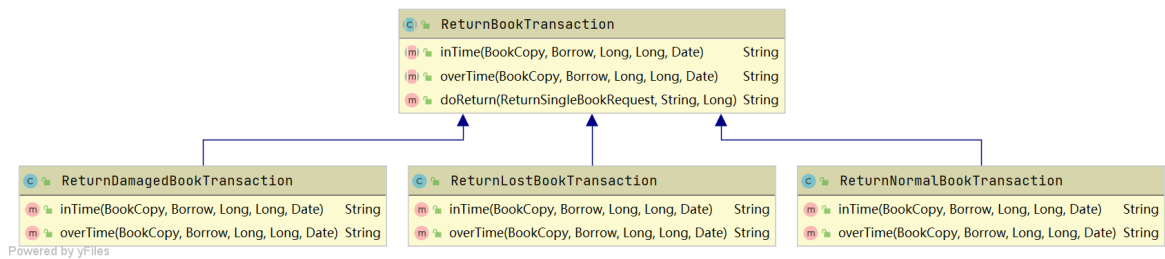
Package exceptionhandler

异常处理类的包。没什么特别的设计，就是一个全局的异常处理类



Package transaction

目前只有关于return book(还书)的transaction。uml图如下



这样设计的考虑是还书的时候，书本的状态有“完好”，“损坏”，“丢失”三种状态。可能需要对每种状态进行判断，进行不一样的处理，这样需要写很多if-else。

因此抽象出三种Transaction类，分别是ReturnDamagedBookTransaction，ReturnLostBookTransaction，ReturnNormalBookTransaction，都继承自抽象基类ReturnBookTransaction。

在业务逻辑中，通过反射调用，得到对应的transaction。并调用其doReturn方法

```
//初始化状态到类名的哈希表
HashMap<String, String> statusToClassName = new HashMap<>();
statusToClassName.put("ok",
    "com.example.lab2.transaction.returnbook.ReturnNormalBookTransaction");
statusToClassName.put("damaged",
    "com.example.lab2.transaction.returnbook.ReturnDamagedBookTransaction");
statusToClassName.put("lost",
    "com.example.lab2.transaction.returnbook.ReturnLostBookTransaction");


try {
    //反射创建对应的transaction
    Class<?> clz =
        Class.forName(statusToClassName.get(returnSingleBookRequest.getStatus()));
    ReturnBookTransaction returnBookTransaction =
        (ReturnBookTransaction) clz.getConstructor().newInstance();


    //给反射创建的transaction注入bean
    autowireCapableBeanFactory.autowireBean(returnBookTransaction);


    //尝试还书
    String res = returnBookTransaction.doReturn(returnSingleBookRequest,
        admin, adminLibraryID);
    return res;
} catch (ClassNotFoundException | NoSuchMethodException |
    InstantiationException | IllegalAccessException | InvocationTargetException e) {
    throw new RuntimeException(e.getMessage());
}
```


doReturn是在抽象基类中实现的，主要是进行三种还书状态共有的逻辑。之后doReturn方法根据是否超期调用各自的inTime和overTime方法（这两个方法都在子类中被实现）。

Package controller


 InitializeControllerTest


 SuperAdminController


 InitializeController


 NormalUserController


 AuthControllerTest

 ReserveController

 BorrowController

 SearchController



 AdminController


 AuthController


Powered by yFiles


没有什么特别的设计。


Package exception


  UploadException

 Package notfound

 Package bookcopy

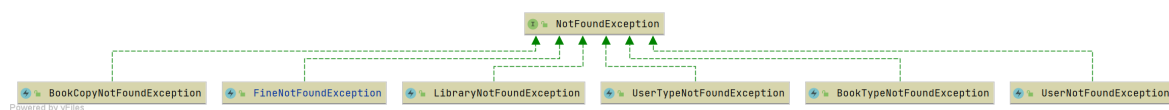
 Package reserve

 Package borrow

 Package auth




Powered by yFiles

为各种类型的错误都创建了自己的Exception类，在对应的包下。其中，notfount包中存储的是



定义一个接口NotFoundException，这样全局异常处理类就可以检查，当发现抛出的是NotFoundException时就将http状态码设置成404，否则就设置成400

Package response

 UserInfoResponse GeneralResponse Package search

Powered by yFiles

是lab3中为某些返回值比较复杂的接口专门创建的response对象，lab4就没有新增和修改过。但是这个包里的东西似乎和DTO类有一些重叠，具体应该是写DTO还是Response也有些含糊不清，未来可能考虑修改。

Package service

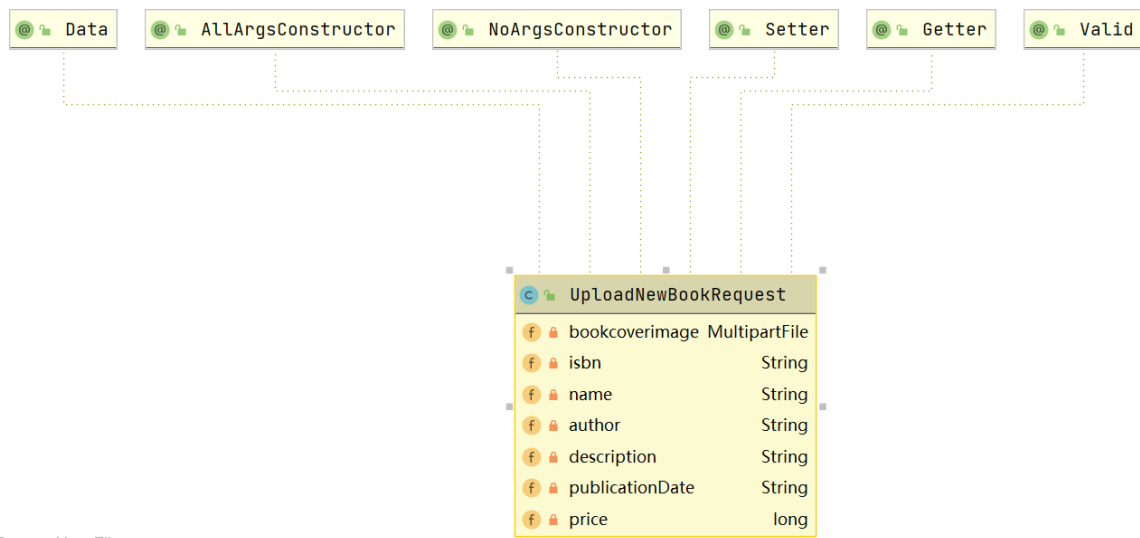
和controller差不多，一种类型的service一个类，还有其对应的单元测试类，没有什么特别的设计



Powered by yFiles

Package request

基本上对于每个后端的接口，都创建了一个类来对应前端发来的数据。比如



Powered by yFiles

由于这部分也没有什么特别的设计，其他的uml图就不放了。

Package entity



为每个实体类创建了一个对象。

Package config filter utils dao

config: 是Spring Security和邮箱的设置类

filter: 是Spring Security的过滤器

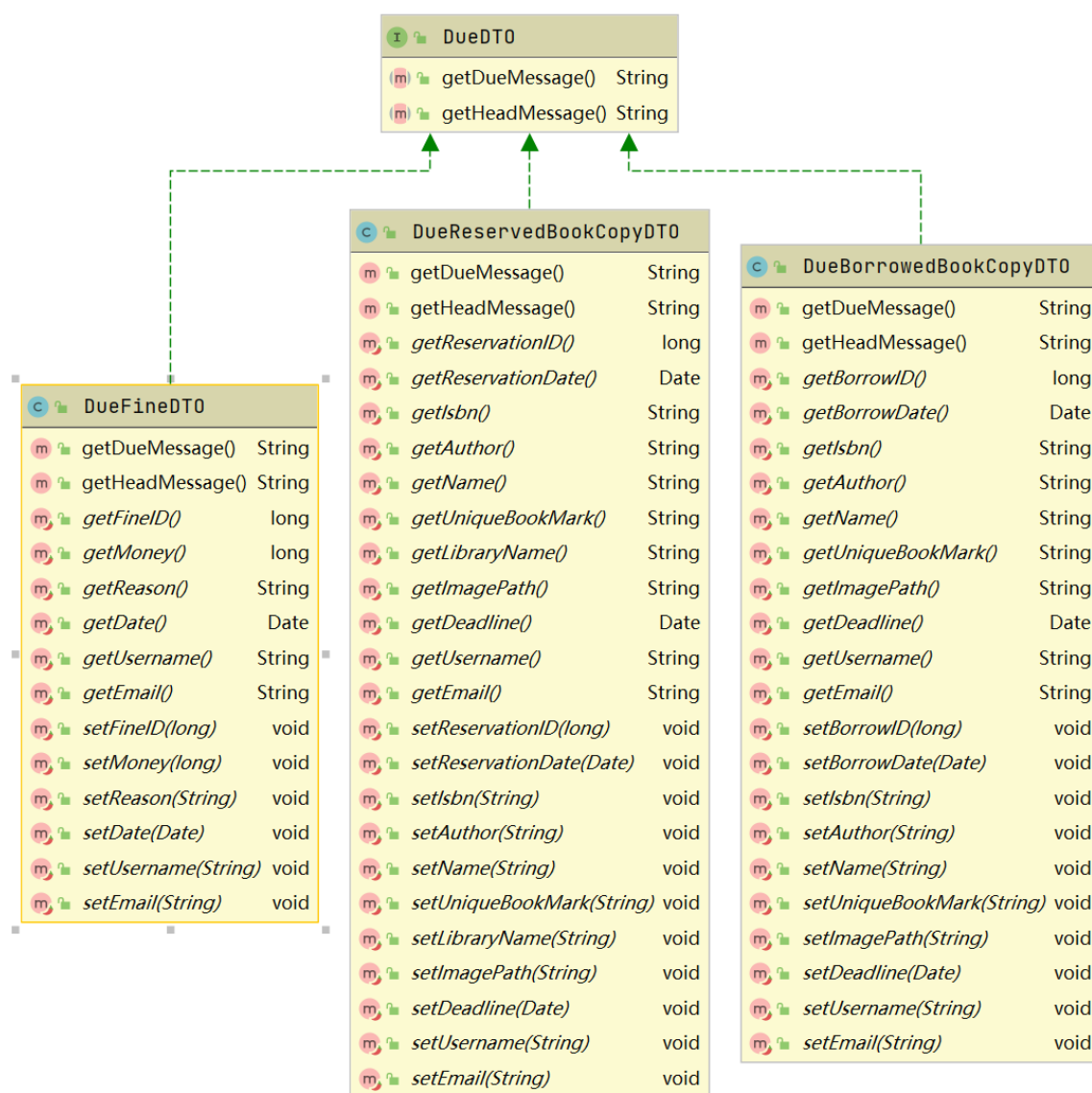
utils: 是一些实用方法的封装

dao: 为每个实体类创建了一个Spring JPA的Repository，放在dao里面。

以上四个包由于确实没什么特别的设计，就不放uml图了

Package dto

是数据传输类的包



Powered by yFiles

其中due包有一些设计，其他的都是简单的数据传输对象

DueDTO类设计的动机是管理员发邮件催用户还书和交罚款的时候需要找到对应的Reservation和Fine和Borrow，然后将它们按用户分类，然后给每个用户发送对应相关的信息。

一开始的设计是找出List,List和List，然后再去查找用户，分别遍历这三个list，生成信息以后再发邮件，这样感觉比较繁琐。

于是想出抽象出DueDTO，即“过期的对象”这种东西，然后DueBorrowedBookCopyDTO，DueFineDTO和DueReservedBookCopyDTO分别继承它，直接从数据库连表查询，在DTO里封装用户的信息和过期对象的信息。每个对象都实现getDueMessage()方法，比如返回一个

你由于借阅JavaScript高级程序设计（第四版）9787115545381-7损坏罚款遭受的罚款64.50元还未支付，请尽快支付

把生成具体过期信息的逻辑转移到转移到DueDTO内部去，而不是在外层循环里拼接字符串。

