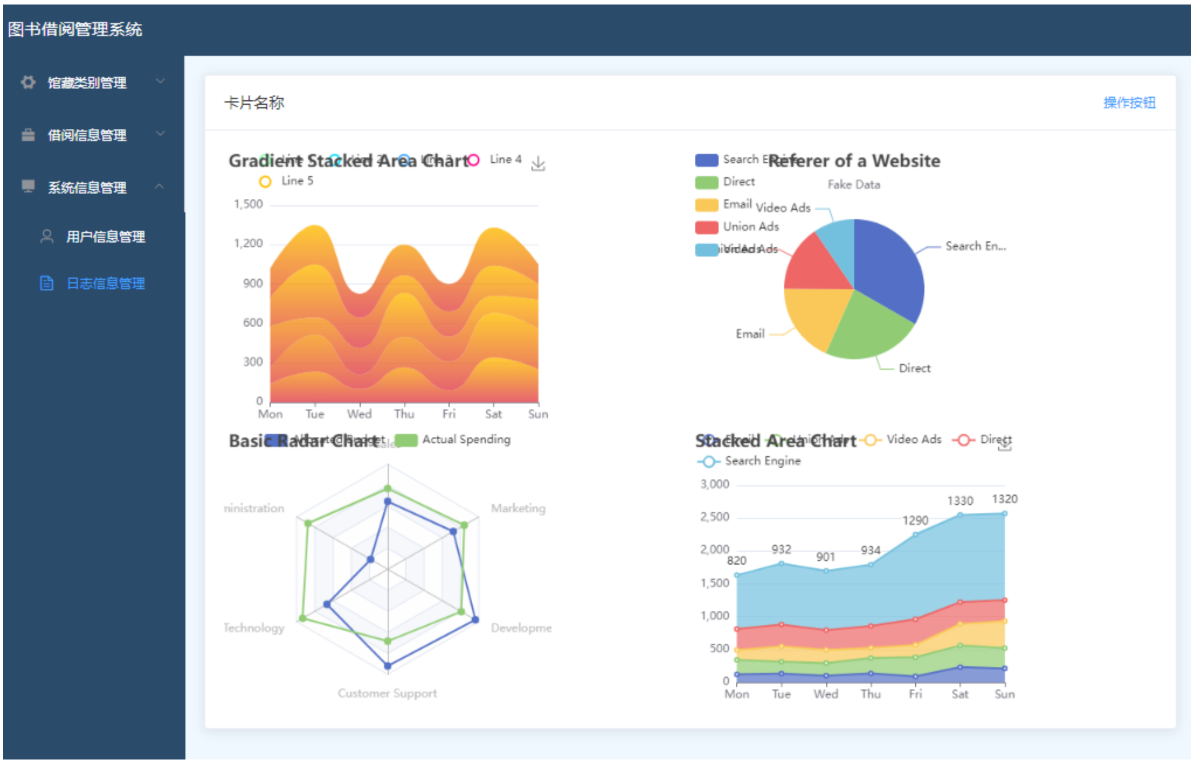
第一章、基于SpringBoot微服务架构的前后端分离图书借阅管理系统设计与实现

```
/**
   该系统主要完成图书借阅用户登录模块、图书类别模块维护、馆藏图书模块维护、图书借阅模块信息维护、
日志信息
维护、用户模块信息维护、借阅数据分析模块查看等信息、该课设课题虽然是一个很老的课设课题，不过该系
统目却采
用目前企业流行的开发模式即前后端分离开发模式，后端采用java语言开发，B/S架构项目，使用mysql作为
数据存储
的数据库、使用mybatisPlus(JDBC封装框架)作为数据持久层技术、使用springboot作为微服务技术、让
开发变得
更加简单、快速、易学；使用shiro作为认证、鉴权；使用swagger技术作为前后端分离开发后端提供前端开
发API文档、
前端使用目前最流行的VUE技术作为前端技术栈、使用饿了么团队研发的基于VUE的Element框架，这样的框
架很多，该
框架简洁、易用、国内企业主流；最重要的是通过本次项目给大家传达灌输目前企业开发主流模式-及前后端分
离开发，
前端开发有前端工程师，后端开发有后端工程师，不过对于学习者而言都要会，这样就会称为全栈工程师，为
后面打下坚实基础。*/
```

1.1.整体效果图



第二章、图书借阅管理系统工程SQL脚本

```
/*
SQLyog Ultimate v9.10
MySQL - 8.0.22 : Database - book
*********************************************************************
```

```
*/
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0
*/;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
CREATE DATABASE /*!32312 IF NOT EXISTS*/`book` /*!40100 DEFAULT CHARACTER SET
utf8
COLLATE utf8_bin */ /*!80016 DEFAULT ENCRYPTION='N' */;
USE `book`;
/*Table structure for table `book` */DROP TABLE IF EXISTS `book`;
CREATE TABLE `book` (
`bid` int NOT NULL AUTO_INCREMENT,
`cateid` int DEFAULT NULL COMMENT '图书类别表外键',
`bnum` varchar(50) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL COMMENT '图书
编
号',
`bname` varchar(50) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL COMMENT '图
书名
称',
`price` int DEFAULT NULL COMMENT '单价',
`author` varchar(50) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL COMMENT '作
者',
`ddesc` text CHARACTER SET utf8 COLLATE utf8_bin COMMENT '简介',
`quantity` int DEFAULT NULL COMMENT '库存数量',
`deleted` int DEFAULT '0' COMMENT '0未删除、1已删除，默认是0',
`createtime` varchar(30) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL
COMMENT
'创建时间',
`modifytime` varchar(30) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL
COMMENT
'修改时间',
PRIMARY KEY (`bid`)
) ENGINE=InnoDB AUTO_INCREMENT=26 DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
/*Data for the table `book` */
insert into
`book`(`bid`,`cateid`,`bnum`,`bname`,`price`,`author`,`ddesc`,`quantity`,`delete
d`,`cre
atetime`,`modifytime`) values (1,2,'1001','红楼梦',100,'曹雪芹','《红楼梦》，中国古代
章回体长
篇小说，中国古典四大名著之一，一般认为是清代作家曹雪芹所著。小说以贾、史、王、薛四大家族的兴衰为
背景，以
富贵公子贾宝玉为视角，以贾宝玉与林黛玉、薛宝钗的爱情婚姻悲剧为主线，描绘了一批举止见识出于须眉之
上的闺阁
佳人的人生百态，展现了真正的人性美和悲剧美，可以说是一部从各个角度展现女性美以及中国古代社会世态
百相的史
诗性著作。',99,0,NULL,'2021-07-27 17:05:26'),(2,1,'1002','我这一辈子',100,'老
舍','《我这一辈
子》是老舍先生创作于1937年抗战前夕的一部著名的中篇小说。描写了一个旧时代普通巡警的坎坷一生。
1937年7月1
日载于《文学》第9卷第1号。《我这一辈子》是老舍第一个创作黄金时期的压轴作品，格外具有挺拔于时代的
进步气
息',80,0,NULL,'2021-07-27 17:06:04'),(3,1,'1003','平凡的世界',100,'路遥','《平凡的世
界》是中
```

```
国作家路遥创作的一部全景式地表现中国当代城乡社会生活的百万字长篇小说。全书共三部。1986年12月首次
出版。\n
该书以中国70年代中期到80年代中期十年间为背景，通过复杂的矛盾纠葛，以孙少安和孙少平两兄弟为中心，
刻画了当
时社会各阶层众多普通人的形象；劳动与爱情、挫折与追求、痛苦与欢乐、日常生活与巨大社会冲突纷繁地交
织在一
起，深刻地展示了普通人在大时代历史进程中所走过的艰难曲折的道路',99,0,NULL,'2021-07-16
14:56:35'),
(4,1,'1004','梦里花落知多少',100,'郭敬明','《梦里花落知多少》是当代作家郭敬明创作的一部长篇
小说。小说
以北京、上海等大都市为背景，讲述了几个年轻人的爱情故事，故事情节曲折，语言幽默生动，大量的引入口
语与通俗
幽默的比喻。 [1] \n主人公是一些即将走出校门的大学生，在成长的过程中，友情、爱情都在经历着蜕变。
小说一经
出版便在市场上取得重大成功，销量上成为2004年全国文学类畅销书第一名。',999,0,'2021-06-25
11:15:37','2021-07-16 14:57:16'),(6,2,'1005','三国演义',100,'罗贯中','三国演义三国演义
三国演义
三国演义',1000,0,'2021-07-15 16:16:51','2021-07-27 17:04:29'),(7,1,NULL,'三国演义
1',99,NULL,NULL,NULL,0,NULL,NULL),(8,1,NULL,'三国演义
2',99,NULL,NULL,NULL,0,NULL,NULL),
(24,NULL,NULL,'23',23,'23','2323',23,NULL,NULL,NULL);
/*Table structure for table `borrow` */
DROP TABLE IF EXISTS `borrow`;
CREATE TABLE `borrow` (
`boid` int NOT NULL AUTO_INCREMENT,
`sid` int DEFAULT NULL,
`bid` int DEFAULT NULL,
`otime` varchar(50) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL,
`itime` varchar(50) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL,
`quantity` int DEFAULT NULL,
`deleted` int DEFAULT '0',
`uid` int DEFAULT NULL,`stat` int DEFAULT '0' COMMENT '0已借出、1已归还',
PRIMARY KEY (`boid`)
) ENGINE=InnoDB AUTO_INCREMENT=35 DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
/*Data for the table `borrow` */
insert into
`borrow`(`boid`,`sid`,`bid`,`otime`,`itime`,`quantity`,`deleted`,`uid`,`stat`)
values
(27,1,1,'2021-07-16 14:56:10','2021-07-27 17:04:26',1,0,1,1),(28,2,2,'2021-07-16
14:56:18','2021-07-27 17:04:27',1,0,1,1),(29,1,6,'2021-07-16 14:56:27','2021-07-
27
17:04:29',1,0,1,1),(30,2,3,'2021-07-16 14:56:35',NULL,1,0,1,0),(31,1,1,'2021-07-
16
14:57:05',NULL,1,0,1,0),(32,1,4,'2021-07-16 14:57:16',NULL,1,0,1,0),
(33,1,1,'2021-07-27
17:05:20','2021-07-27 17:05:26',1,0,1,1),(34,1,2,'2021-07-27
17:06:04',NULL,20,0,1,0);
/*Table structure for table `category` */
DROP TABLE IF EXISTS `category`;
CREATE TABLE `category` (
`cateid` int NOT NULL AUTO_INCREMENT,
`catename` varchar(50) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL COMMENT
'类
别名称',
`deleted` int DEFAULT '0' COMMENT '0未删除、1已删除',
`createtime` varchar(30) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL
COMMENT
'创建时间',
`modifytime` varchar(30) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL
COMMENT
'修改时间',
PRIMARY KEY (`cateid`)
) ENGINE=InnoDB AUTO_INCREMENT=27 DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
/*Data for the table `category` */
insert into `category`(`cateid`,`catename`,`deleted`,`createtime`,`modifytime`)
values (1,'近代小说类',0,NULL,NULL),(2,'古典小说类',0,NULL,NULL),(3,'计算机
类',0,NULL,NULL),
(4,'考研类',0,'2021-06-25 10:55:42','2021-06-25 10:56:49'),(6,'探索
类',0,NULL,'2021-07-13
14:38:18'),(11,'探索类1',0,NULL,'2021-08-09 19:24:28'),(12,'ouio',1,NULL,NULL),
(13,'uip',1,NULL,NULL),(14,'sdgh',1,NULL,NULL),(15,'fjf',1,NULL,NULL),
(16,'adsfsd',1,NULL,NULL),(17,'56768768',1,'2021-07-13 10:20:57',NULL);
/*Table structure for table `logs` */
DROP TABLE IF EXISTS `logs`;
CREATE TABLE `logs` (
`operid` int NOT NULL AUTO_INCREMENT,
`sid` int DEFAULT NULL,
`uid` int DEFAULT NULL COMMENT '用户外键',
`opertime` varchar(50) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL COMMENT
'操
作时间',
`opername` varchar(50) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL COMMENT
'操
作人',
`ip` varchar(50) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL COMMENT '操作
IP',
`methods` text CHARACTER SET utf8 COLLATE utf8_bin COMMENT '操作模块方法',
`ddesc` text CHARACTER SET utf8 COLLATE utf8_bin COMMENT '描述',
PRIMARY KEY (`operid`)
) ENGINE=InnoDB AUTO_INCREMENT=180 DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
/*Data for the table `logs` */insert into
`logs`(`operid`,`sid`,`uid`,`opertime`,`opername`,`ip`,`methods`,`ddesc`) values
(66,NULL,NULL,'2021-07-15 16:22:39','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.backBook','归还图书操作'),
(67,NULL,NULL,'2021-07-15 16:23:10','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.borrowBook','借阅图书操
作'),
(68,NULL,NULL,'2021-07-15
16:23:35','admin','127.0.0.1','com.xls.app.controller.BookController.find','分页查
询图书信
息'),(69,NULL,NULL,'2021-07-15
16:23:36','admin','127.0.0.1','com.xls.app.controller.BookController.find','分页查
询图书信
息'),(70,NULL,NULL,'2021-07-15
16:23:36','admin','127.0.0.1','com.xls.app.controller.BookController.find','分页查
询图书信
息'),(71,NULL,NULL,'2021-07-15
16:23:37','admin','127.0.0.1','com.xls.app.controller.BookController.find','分页查
询图书信
息'),(72,NULL,NULL,'2021-07-15
```

```
16:23:37','admin','127.0.0.1','com.xls.app.controller.BookController.find','分页查
询图书信
息'),(73,NULL,NULL,'2021-07-15
16:23:37','admin','127.0.0.1','com.xls.app.controller.BookController.find','分页查
询图书信
息'),(74,NULL,NULL,'2021-07-15
16:23:37','admin','127.0.0.1','com.xls.app.controller.BookController.find','分页查
询图书信
息'),(75,NULL,NULL,'2021-07-15
16:39:47','admin','127.0.0.1','com.xls.app.controller.BookController.find','分页查
询图书信
息'),(76,NULL,NULL,'2021-07-15
16:49:32','admin','127.0.0.1','com.xls.app.controller.BookController.find','分页查
询图书信
息'),(77,NULL,NULL,'2021-07-15 16:50:09','刘
备','127.0.0.1','com.xls.app.controller.BookController.find','分页查询图书信息'),
(78,NULL,NULL,'2021-07-15 16:50:23','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.backBook','归还图书操作'),
(79,NULL,NULL,'2021-07-15 16:51:01','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.borrowBook','借阅图书操
作'),
(80,NULL,NULL,'2021-07-15 16:51:16','刘
备','127.0.0.1','com.xls.app.controller.BookController.find','分页查询图书信息'),
(81,NULL,NULL,'2021-07-15 16:51:49','刘
备','127.0.0.1','com.xls.app.controller.BookController.find','分页查询图书信息'),
(82,NULL,NULL,'2021-07-15 16:51:53','刘
备','127.0.0.1','com.xls.app.controller.BookController.find','分页查询图书信息'),
(83,NULL,NULL,'2021-07-15 16:52:08','刘
备','127.0.0.1','com.xls.app.controller.BookController.find','分页查询图书信息'),
(84,NULL,NULL,'2021-07-15 16:52:40','刘
备','127.0.0.1','com.xls.app.controller.BookController.find','分页查询图书信息'),
(85,NULL,NULL,'2021-07-15 16:53:18','刘
备','127.0.0.1','com.xls.app.controller.BookController.find','分页查询图书信息'),
(86,NULL,NULL,'2021-07-16
10:53:47','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(87,NULL,NULL,'2021-07-16
10:54:35','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(88,NULL,NULL,'2021-07-16
10:55:23','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(89,NULL,NULL,'2021-07-16
10:56:55','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(90,NULL,NULL,'2021-07-16
10:56:55','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(91,NULL,NULL,'2021-07-16
10:57:36','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(92,NULL,NULL,'2021-07-16
10:57:36','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(93,NULL,NULL,'2021-07-16
```

```
10:59:14','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(94,NULL,NULL,'2021-07-16
10:59:14','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(95,NULL,NULL,'2021-07-
1610:59:45','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLi
keBook','
统计读者喜欢借书信息'),(96,NULL,NULL,'2021-07-16
10:59:45','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(97,NULL,NULL,'2021-07-16
11:00:11','admin','127.0.0.1','com.xls.app.controller.BookController.find','分页查
询图书信
息'),(98,NULL,NULL,'2021-07-16
11:03:03','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(99,NULL,NULL,'2021-07-16
11:03:03','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(100,NULL,NULL,'2021-07-16
11:03:07','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(101,NULL,NULL,'2021-07-16
11:03:07','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(102,NULL,NULL,'2021-07-16
11:04:18','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(103,NULL,NULL,'2021-07-16
11:04:18','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(104,NULL,NULL,'2021-07-16
11:04:20','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(105,NULL,NULL,'2021-07-16
11:04:20','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(106,NULL,NULL,'2021-07-16
11:10:39','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(107,NULL,NULL,'2021-07-16
11:10:39','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(108,NULL,NULL,'2021-07-16
11:11:33','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(109,NULL,NULL,'2021-07-16
11:11:33','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(110,NULL,NULL,'2021-07-16
11:11:35','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(111,NULL,NULL,'2021-07-16
```

```
11:13:11','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(112,NULL,NULL,'2021-07-16
11:13:34','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(113,NULL,NULL,'2021-07-16
11:14:33','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(114,NULL,NULL,'2021-07-16
11:15:28','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(115,NULL,NULL,'2021-07-16
11:17:55','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(116,NULL,NULL,'2021-07-16
11:20:07','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(117,NULL,NULL,'2021-07-16
14:32:42','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(118,NULL,NULL,'2021-07-16
14:37:39','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(119,NULL,NULL,'2021-07-16
14:48:42','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(120,NULL,NULL,'2021-07-16
14:48:58','admin','127.0.0.1','com.xls.app.controller.UsersController.changeStat
s','更改
用户状态'),(121,NULL,NULL,'2021-07-16
14:49:00','admin','127.0.0.1','com.xls.app.controller.UsersController.changeStat
s','更改
用户状态'),(122,NULL,NULL,'2021-07-16 14:53:22','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.backBook','归还图书操作'),
(123,NULL,NULL,'2021-07-16 14:53:25','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.backBook','归还图书操作'),
(124,NULL,NULL,'2021-07-16 14:53:28','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.backBook','归还图书操作'),
(125,NULL,NULL,'2021-07-16 14:53:30','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.backBook','归还图书操作'),
(126,NULL,NULL,'2021-07-16 14:53:37','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.backBook','归还图书操作'),
(127,NULL,NULL,'2021-07-16 14:53:57','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.backBook','归还图书操作'),
(128,NULL,NULL,'2021-07-16 14:53:59','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.backBook','归还图书操作'),
(129,NULL,NULL,'2021-07-16 14:54:00','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.backBook','归还图书操作'),
(130,NULL,NULL,'2021-07-16 14:54:02','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.backBook','归还图书操作'),
(131,NULL,NULL,'2021-07-16
14:54:49','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(132,NULL,NULL,'2021-07-16 14:56:10','张
```

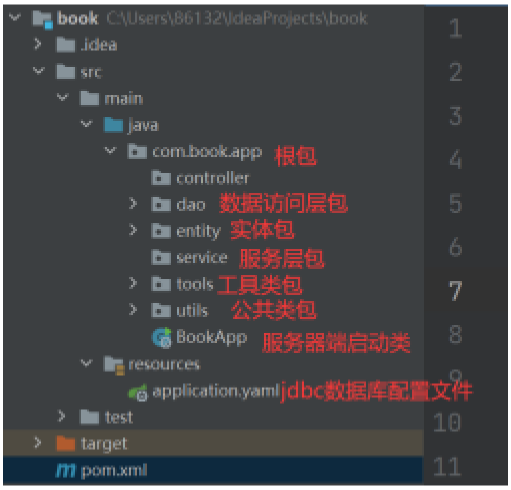
```
飞','127.0.0.1','com.xls.app.controller.BorrowController.borrowBook','借阅图书操
作'),
(133,NULL,NULL,'2021-07-16 14:56:18','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.borrowBook','借阅图书操
作'),
(134,NULL,NULL,'2021-07-16 14:56:27','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.borrowBook','借阅图书操
作'),
(135,NULL,NULL,'2021-07-16 14:56:35','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.borrowBook','借阅图书操
作'),
(136,NULL,NULL,'2021-07-16
14:56:45','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(137,NULL,NULL,'2021-07-16 14:57:05','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.borrowBook','借阅图书操
作'),
(138,NULL,NULL,'2021-07-16 14:57:16','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.borrowBook','借阅图书操
作'),
(139,NULL,NULL,'2021-07-16
14:57:25','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(140,NULL,NULL,'2021-07-16
14:58:22','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(141,NULL,NULL,'2021-07-16
14:58:29','admin','127.0.0.1','com.xls.app.controller.BookController.find','分页查
询图书信
息'),(142,NULL,NULL,'2021-07-17
11:18:00','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(143,NULL,NULL,'2021-07-17
11:41:50','admin','127.0.0.1','com.xls.app.controller.BookController.find','分页查
询图书信
息'),(144,NULL,NULL,'2021-07-17
11:42:19','admin','127.0.0.1','com.xls.app.controller.BookController.find','分页查
询图书信
息'),(145,NULL,NULL,'2021-07-17
11:44:17','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(146,NULL,NULL,'2021-07-17
13:59:53','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(147,NULL,NULL,'2021-07-19
17:00:54','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(148,NULL,NULL,'2021-07-19
17:01:07','admin','127.0.0.1','com.xls.app.controller.UsersController.changeStat
s','更改
用户状态'),(149,NULL,NULL,'2021-07-19
17:01:10','admin','127.0.0.1','com.xls.app.controller.UsersController.changeStat
s','更改
用户状态'),(150,NULL,NULL,'2021-07-27
```

```
17:03:03','admin','127.0.0.1','com.xls.app.controller.BookController.find','分页查
询图书信
息'),(151,NULL,NULL,'2021-07-27
17:03:30','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(152,NULL,NULL,'2021-07-27 17:04:26','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.backBook','归还图书操作'),
(153,NULL,NULL,'2021-07-27 17:04:27','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.backBook','归还图书操作'),
(154,NULL,NULL,'2021-07-27 17:04:29','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.backBook','归还图书操作'),
(155,NULL,NULL,'2021-07-27 17:05:20','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.borrowBook','借阅图书操
作'),
(156,NULL,NULL,'2021-07-27 17:05:26','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.backBook','归还图书操作'),
(157,NULL,NULL,'2021-07-27 17:05:57','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.borrowBook','借阅图书操
作'),
(158,NULL,NULL,'2021-07-27 17:06:04','张
飞','127.0.0.1','com.xls.app.controller.BorrowController.borrowBook','借阅图书操
作'),
(159,NULL,NULL,'2021-07-27 17:06:40','刘
备','127.0.0.1','com.xls.app.controller.BookController.find','分页查询图书信息'),
(160,NULL,NULL,'2021-07-27 17:07:21','刘
备','127.0.0.1','com.xls.app.controller.BookController.find','分页查询图书信息'),
(161,NULL,NULL,'2021-07-27
17:16:12','admin','127.0.0.1','com.xls.app.controller.BookController.find','分页查
询图书信
息'),(162,NULL,NULL,'2021-07-27
17:16:42','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(163,NULL,NULL,'2021-07-27
17:20:00','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(164,NULL,NULL,'2021-07-30
14:38:08','admin','127.0.0.1','com.xls.app.controller.BookController.find','分页查
询图书信
息'),(165,NULL,NULL,'2021-07-30
14:39:22','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(166,NULL,NULL,'2021-08-02
14:21:24','admin','127.0.0.1','com.xls.app.controller.BorrowController.countLike
Book','
统计读者喜欢借书信息'),(167,NULL,NULL,'2021-08-02
14:21:28','admin','127.0.0.1','com.xls.app.controller.BookController.find','分页查
询图书信
息'),(168,NULL,NULL,'2021-08-12 11:08:27','管理
员','127.0.0.1','com.xls.app.controller.UsersController.update','更新用户信息'),
(169,NULL,NULL,'2021-08-12 11:09:15','管理
员','127.0.0.1','com.xls.app.controller.UsersController.update','更新用户信息'),
(170,NULL,NULL,'2021-08-12 11:10:24','管理
员','127.0.0.1','com.xls.app.controller.UsersController.update','更新用户信息'),
(171,NULL,NULL,'2021-08-12 11:10:27','管理
员','127.0.0.1','com.xls.app.controller.UsersController.update','更新用户信息'),
(172,NULL,NULL,'2021-08-12 11:10:40','管理
员','127.0.0.1','com.xls.app.controller.UsersController.update','更新用户信息'),
(173,NULL,NULL,'2021-08-12 11:11:39','管理
员','127.0.0.1','com.xls.app.controller.UsersController.update','更新用户信息'),
(174,NULL,NULL,'2021-08-12 11:12:56','管理
员','127.0.0.1','com.xls.app.controller.UsersController.update','更新用户信息'),
(175,NULL,NULL,'2021-08-12 11:13:12','管理
员','127.0.0.1','com.xls.app.controller.UsersController.update','更新用户信息'),
(176,NULL,NULL,'2021-08-12 14:16:45','管理
员','127.0.0.1','com.xls.app.controller.UsersController.update','更新用户信息'),
(177,NULL,NULL,'2021-08-12 14:16:46','管理
员','127.0.0.1','com.xls.app.controller.UsersController.update','更新用户信息'),
(178,NULL,NULL,'2021-08-12 16:17:12','管理
员','127.0.0.1','com.xls.app.controller.UsersController.update','更新用户信息'),
(179,NULL,NULL,'2021-08-12 16:17:13','管理
员','127.0.0.1','com.xls.app.controller.UsersController.update','更新用户信息');
/*Table structure for table `users` */
DROP TABLE IF EXISTS `users`;
CREATE TABLE `users` (
`uid` int NOT NULL AUTO_INCREMENT,
`password` varchar(200) COLLATE utf8_bin DEFAULT '000000',
`uname` varchar(50) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL,
`images` text CHARACTER SET utf8 COLLATE utf8_bin,
`stats` tinyint(1) DEFAULT '1' COMMENT '0表示启用、1禁用状态',
`deleted` int DEFAULT '1',
`createtime` varchar(30) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL,
`modifytime` varchar(30) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT
NULL,PRIMARY KEY (`uid`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
/*Data for the table `users` */
insert into
`users`(`uid`,`password`,`uname`,`images`,`stats`,`deleted`,`createtime`,`modify
time`)
values (1,'000000','张
飞','https://img1.baidu.com/it/u=3960132582,584653904&fm=15&fmt=auto&gp=0.jpg',1,
0,NULL,
'2021-08-12 14:16:46'),(2,'000000','李
逵','https://img2.baidu.com/it/u=661539886,3900694712&fm=26&fmt=auto&gp=0.jpg',1,
0,NULL,
'2021-08-12 14:16:45'),
(3,'000000','admin','https://img1.baidu.com/it/u=3960132582,584653904&fm=15&fmt=
auto&gp
=0.jpg',1,0,NULL,NULL),(4,'000000','刘
备','https://img1.baidu.com/it/u=3960132582,584653904&fm=15&fmt=auto&gp=0.jpg',1,
0,'2021
-06-24 09:27:02','2021-08-12 16:17:13'),(5,'000000','嬴
政','https://img1.baidu.com/it/u=2179921797,2025214210&fm=26&fmt=auto&gp=0.jpg',1
,0,'202
1-06-24 14:48:53','2021-06-26 09:29:31');
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */
```

第三章、图书借阅管理系统工程目录结构

3.1工程目录结构

```
开发工具 idea工具、jdk版本1.8,工程目录如图
```



```
book C:\Users\86132\IdeaProjects\book
  .idea
  src
    main
      java
        com.book.app 根包
          controller
          dao 数据访问层包
          entity 实体包
          service 服务层包
          tools 工具类包
          utils 公共类包
          BookApp 服务器端启动类
      resources
        application.yamljdbc数据库配置文件
    test
  target
  pom.xml
```

3.2工程maven配置文件

```
<dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
        </dependency>
        <dependency>
            <groupId>org.mybatis.spring.boot</groupId>
            <artifactId>mybatis-spring-boot-starter</artifactId>
            <version>2.2.0</version>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-aop</artifactId>
        </dependency>
        <dependency>
            <groupId>com.alibaba</groupId>
            <artifactId>easyexcel</artifactId>
            <version>2.2.6</version>
        </dependency>

        <dependency>
            <groupId>io.springfox</groupId>
            <artifactId>springfox-boot-starter</artifactId>
            <version>3.0.0</version>
        </dependency>
        <dependency>
            <groupId>com.github.xiaoymin</groupId>
            <artifactId>swagger-bootstrap-ui</artifactId>
            <version>1.9.6</version>
        </dependency>
    </dependencies>
    <!--    打包使用-->
    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
```

第四章、图书借阅管理系统服务器端开发

4.1.馆藏图书实体类设计与实现

```
package com.book.app.bean;
import com.alibaba.excel.annotation.ExcelIgnore;
import com.alibaba.excel.annotation.ExcelProperty;
import io.swagger.annotations.ApiModel;
import io.swagger.annotations.ApiModelProperty;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import java.io.Serializable;

@Data
@AllArgsConstructor
@NoArgsConstructor
@ApiModel
public class Book  implements Serializable {
    @ExcelIgnore
    @ApiModelProperty(hidden = true)
    private Integer bid;
    @ApiModelProperty(name = "bnum",value = "图书编号",
            example = "BNS1001",dataType ="string",required = true )
    @ExcelProperty(value = "图书编号")
    private String bnum;
    @ApiModelProperty(name = "bname",value = "图书名称",
            example = "《等风来》",dataType ="string",required = true )
    @ExcelProperty(value = "图书名称")
    private String bname;
    @ApiModelProperty(name = "price",value = "单价",
            example = "99",dataType ="integer",required = true )
    @ExcelProperty(value = "单价")
    private Integer price;
    @ApiModelProperty(name = "author",value = "作者",
            example = "无名",dataType ="string",required = true )
    @ExcelProperty(value = "作者")
    private String author;
    @ApiModelProperty(name = "ddesc",value = "一部励志书籍",
            example = "一部励志书籍",dataType ="string",required = true )
    @ExcelProperty(value = "简介")
    private String ddesc;
    @ApiModelProperty(name = "quantity",value = "馆藏数量",
            example = "100",dataType ="integer", required = true )
    @ExcelProperty(value = "数量")
    private Integer quantity;
    @ApiModelProperty(hidden = true)
    @ExcelIgnore
    private String createTime;
    @ApiModelProperty(hidden = true)
    @ExcelIgnore
    private String modifyTime;
    @ApiModelProperty(hidden = true)
    @ExcelIgnore
    private Integer deleted;//删除标记 0未删除  1 已删除
    @ApiModelProperty(name = "category",example = "category",required = true)
    @ExcelIgnore
    private Category category;//图书类别外键
}
```

4.2.图书类别实体类设计与实现

```
package com.book.app.bean;

import io.swagger.annotations.ApiModel;
import io.swagger.annotations.ApiModelProperty;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

import java.io.Serializable;

@Data
@AllArgsConstructor
@NoArgsConstructor
@ApiModel
public class Category implements Serializable {
    @ApiModelProperty(example = "44")
    private Integer cateid;
    @ApiModelProperty(value = "图书类别名称",required = true,dataType = "string",
            example = "古典文学")
    private String catename;
    @ApiModelProperty(value = "创建时间",required = true,dataType = "string",
            example = "2022-06-27 14:00:00")
    private String ctime;
    @ApiModelProperty(value = "修改时间",required = true,dataType = "string",
            example = "2022-06-27 14:00:00")
    private String mtime;
    @ApiModelProperty(hidden = true)
    private Integer del;//0未删除 1已删除
    //统计分类数量
    @ApiModelProperty(value = "统计分类值",required = true,dataType = "int",
            example = "10")
    private  Integer value;
    @ApiModelProperty(value = "统计分类名称",required = true,dataType = "string",
            example = "古典文学")
    private String name;
}
```

4.3.图书借阅实体类设计与实现

```
package com.book.app.bean;
import io.swagger.annotations.ApiModel;
import io.swagger.annotations.ApiModelProperty;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import java.io.Serializable;
@Data
@AllArgsConstructor
@NoArgsConstructor
@ApiModel
public class Borrow  implements Serializable {
    @ApiModelProperty(hidden = true)
    private Integer boid;
    @ApiModelProperty(name ="book",value = "图书引用对象",example = "book",
            required = true)
    //借阅图书
    private Book book;
    @ApiModelProperty(hidden = true)
    //借出时间
    private String otime;
    @ApiModelProperty(hidden = true)
    // 归还时间
    private String itime;
    @ApiModelProperty(name ="quantity",value = "借阅数量",example = "1",
            required = true)
    //借出数量
    private Integer quantity;
    @ApiModelProperty(name ="user",value = "借阅人引用对象",example = "user",
            required = true)
    private Users user;
    @ApiModelProperty(hidden = true)
    private Integer stats; //0已借出 1已归还
}
```

4.4.用户实体类设计与实现

```
package com.book.app.bean;
import io.swagger.annotations.ApiModel;
import io.swagger.annotations.ApiModelProperty;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import java.io.Serializable;
@Data
@AllArgsConstructor
```

```
@NoArgsConstructor
@ApiModel
public class Users implements Serializable {
    @ApiModelProperty(hidden = true)
    private Integer uid;
    @ApiModelProperty(value = "用户名称",
            required = true,dataType = "string",example = "张三")
    private String uname;
    @ApiModelProperty(value = "密码",required = true,
            dataType = "string",
            example = "000000")
    private String password;
    @ApiModelProperty(value = "头像",required = true,
            dataType = "string",
            example = "http://xxxxx.ping")
    private String images;
    @ApiModelProperty(hidden = true)
    private Integer stats;//0启用 1 禁用
    @ApiModelProperty(hidden = true)
    private  Integer deleted;
    @ApiModelProperty(hidden = true)
    private String createtime;
    @ApiModelProperty(hidden = true)
    private String modifytime;
}
```

4.5.操作日志实体设计

```
package com.book.app.bean;
import io.swagger.annotations.ApiModel;
import io.swagger.annotations.ApiModelProperty;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import java.io.Serializable;
@Data
@AllArgsConstructor
@NoArgsConstructor
@ApiModel
public class Logs implements Serializable {
    @ApiModelProperty(hidden = true)
    private Integer operid;
    @ApiModelProperty(name ="opertime",value = "操作时间",
            required = true,example = "2022-06-27")
    private String opertime;//操作时间
    @ApiModelProperty(name ="opername",value = "操作人",
            required = true,example = "张三")
    private String opername;
    @ApiModelProperty(name ="ip",value = "操作IP",
            required = true,example = "127.0.0.1")
    private String ip;
    @ApiModelProperty(name ="methods",value = "操作时间",
            required = true,example = "save...")
    private String methods;
    @ApiModelProperty(name ="ddesc",value = "操作方法描述",
            required = true, example = "desc",dataType = "string")
    private String ddesc;
    @ApiModelProperty(name ="times",value = "统计来访人员次数,为了报表显示",
            required = true,example = "2022-06-27",dataType = "integer")
    private  Integer times;
}
```

第五章、图书借阅管理系统DAO层设计与实现

5.1.馆藏图书DAO设计与实现

```
package com.book.app.dao;
import com.book.app.bean.Book;
import org.apache.ibatis.annotations.Param;
import org.springframework.stereotype.Repository;
import java.util.List;
@Repository
public interface BookDao {

    List<Book> listPage(@Param("page")int page,
                        @Param("size")int size,
                        @Param("bname")String bname);

    int count(@Param("bname") String bname);

    int save(Book book);

    int remove(List<Integer> list);

    int  update(Book book);

    List<Book> list(Book book);

    Book get(Integer bid);

    void changStore(Book book);

}
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
        PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
        "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.book.app.dao.BookDao">

    <resultMap id="bookMapper" type="com.book.app.bean.Book">
        <id property="bid" column="bid"/>
        <result property="bnum" column="bnum"/>
        <result property="bname" column="bname"/>
        <result property="ddesc" column="ddesc"/>
        <result property="price" column="price"/>
        <result property="author" column="author"/>
        <result property="quantity" column="quantity"/>
        <result property="createTime" column="createTime"/>
        <result property="modifyTime" column="modifyTime"/>
        <association property="category" javaType="com.book.app.bean.Category">
            <id property="cateid" column="cateid"/>
            <result property="catename" column="catename"/>
        </association>
    </resultMap>
    <insert id="save">
        insert into book(bnum, bname, price, author, ddesc,
                         createtime, quantity, cateid) value (#{bnum},
            #{bname},
            #{price},
            #{author},
            #{ddesc},now(),
            #{quantity},
            #{category.cateid})
    </insert>

    <update id="update">
        update book
        set bnum=#{bnum},
            bname=#{bname},
            price=#{price},
            author=#{author},
            ddesc=#{ddesc},
            modifytime=now(),
            quantity=#{quantity},
            cateid=#{category.cateid}
        where bid = #{bid}
    </update>

    <update id="remove">
        update book set deleted=1 where bid in
        <foreach item="item" index="index" collection="list"
                 open="(" separator="," close=")">
            #{item}
        </foreach>
    </update>
    <update id="changStore">
        update book
        set quantity=#{quantity}
        where bid = #{bid}
    </update>

    <select id="listPage" resultMap="bookMapper">
        select * from book b ,category c
        <where>
            <if test="bname!=null and bname!=''">
                b.bname like '%${bname}%'
            </if>
            and b.cateid=c.cateid and b.deleted=0
        </where>
        limit #{page},#{size}
    </select>

    <select id="count" resultType="java.lang.Integer">
        select count(*) from book b ,category c
        <where>
            <if test="bname!=null and bname!=''">
                b.bname like '%${bname}%'
            </if>
            and b.cateid=c.cateid and b.deleted=0
        </where>
    </select>
    <select id="list" resultType="com.book.app.bean.Book">
        select *
        from book b
    </select>
    <select id="get" resultType="com.book.app.bean.Book">
        select *
        from book
        where bid = #{bid}
    </select>

</mapper>
```

5.2.馆藏图书类别DAO设计与实现

```
package com.book.app.dao;
import com.book.app.bean.Category;
import org.apache.ibatis.annotations.Param;
import org.springframework.stereotype.Repository;
import java.util.List;
@Repository
public interface CategoryDao {
    /**
     *
     * @param page 分页起始数据
     * @param size 每页显示多少条记录
     * @return  List 返回集合时间
     */
    List<Category> listPage(@Param("page") int page,
                            @Param("size") int size,
                            @Param("cateName")String cateName);

    /**
     * 统计分页记录数
     * @return
     */
    int count( @Param("cateName")String cateName);

    int save(Category category);

    int remove(List<Integer> cids);
    //修改
    int  update(Category category);

    List<Category> list();
    List<Category>countTypesBooks();
}
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
        PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
        "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.book.app.dao.CategoryDao">
    <insert id="save">
        insert into category(catename, createtime) value (#{catename},now())
    </insert>
    <!--    批量删除-->
    <update id="remove">
        update category set deleted=1 where cateid in
        <foreach item="item" index="index" collection="list"
                 open="(" separator="," close=")">
            #{item}
        </foreach>
    </update>
    <update id="update">
        update category
        set catename=#{catename},
            modifytime=now()
        where cateid = #{cateid}
    </update>
    <resultMap id="categoryMapper" type="com.book.app.bean.Category">
        <id property="cateid" column="CATEID"></id>
        <result property="catename" column="CATENAME"/>
        <result property="ctime" column="createtime"/>
        <result property="mtime" column="MODIFYTIME"/>
    </resultMap>

    <select id="listPage" resultMap="categoryMapper">
        select * from category
        <where>
            <if test="cateName!=null and cateName!=''">
                cateName like '%${cateName}%'
            </if>
            and deleted=0
        </where>
        limit #{page},#{size}
    </select>
    <select id="count" resultType="java.lang.Integer">
        select count(*)
        from category
        <where>
            <if test="cateName!=null and cateName!=''">
                cateName like '%${cateName}%'
            </if>
            and deleted=0
        </where>
    </select>
    <select id="list" resultType="com.book.app.bean.Category">
```

```
        select *
        from category
        where deleted = 0
    </select>
    <select id="countTypesBooks" resultType="com.book.app.bean.Category">
        select c.catename name, count(c.cateid) as 'value'
        from book b
                 left join category c on b.cateid = c.cateid
        GROUP BY c.catename

    </select>
</mapper>
```

5.3.图书借阅DAO设计与实现

```
package com.book.app.dao;
import com.book.app.bean.Borrow;
import org.apache.ibatis.annotations.Param;
import org.springframework.stereotype.Repository;
import java.util.List;
@Repository
public interface BorrowDao {

    List<Borrow> listPage(@Param("page") int page,
                          @Param("size")int size,
                          @Param("uname")String uname,
                          @Param("bname")String bname);

    int count(@Param("uname")String uname,
              @Param("bname")String bname);

    int borrow(Borrow borrow);

    Borrow getBorrow(@Param("boid")int boid);

    int backBooks(Borrow borrow);

    List<Borrow> listPageHistoryBorrow(@Param("page")int page,
                                       @Param("size")Integer size,
                                       @Param("uname")String uname,
                                       @Param("bname")String bname);

    int countHistoryBorrow(@Param("uname")String uname,
                           @Param("bname")String bname);
}
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
        PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
        "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.book.app.dao.BorrowDao">

    <resultMap id="borrowMapper" type="com.book.app.bean.Borrow">
```

```
        <id property="boid" column="boid"/>
        <result property="quantity" column="quantity"/>
        <result property="itime" column="itime"/>
        <result property="otime" column="otime"/>
        <result property="stats" column="stat"/>
        <association property="user" javaType="com.book.app.bean.Users">
            <id property="uid" column="uid"/>
            <result column="uname" property="uname"/>
        </association>
        <association property="book" javaType="com.book.app.bean.Book">
            <id property="bid" column="bid"/>
            <result column="bname" property="bname"/>
        </association>

    </resultMap>
    <insert id="borrow">
        insert into borrow(uid, bid, quantity, otime) value (#{user.uid},#
{book.bid},#{quantity},now())
    </insert>
    <update id="backBooks">
        update borrow set itime=now(),stat=#{stats} where boid=#{boid}
    </update>
    <select id="listPage" resultMap="borrowMapper">
        select * from borrow bo ,book b ,users u
        <where>
            <if test="bname!=null and bname!=''">
                b.bname like '%${bname}%'
            </if>
            <if test="uname!=null and uname!=''">
                and u.uname=#{uname}
            </if>
            and bo.bid=b.bid and bo.uid=u.uid and bo.deleted=0 and stat=0
        </where>

        limit #{page},#{size}
    </select>
    <select id="count" resultType="java.lang.Integer">
        select count(*) from borrow bo ,book b ,users u
        <where>
            <if test="bname!=null and bname!=''">
                b.bname like '%${bname}%'
            </if>
            <if test="uname!=null and uname!=''">
                and u.uname=#{uname}
            </if>
            and bo.bid=b.bid and bo.uid=u.uid and bo.deleted=0 and stat=0
        </where>

    </select>
    <select id="getBorrow" resultMap="borrowMapper">
        select *
        from borrow bo,
             book b,
             users u
        where boid = #{boid}
```

```
          and bo.bid = b.bid
          and bo.uid = u.uid
          and bo.deleted = 0
          and stats = 0
    </select>
    <select id="countHistoryBorrow" resultType="java.lang.Integer">
        select count(*) from borrow bo ,book b ,users u
        <where>
            <if test="bname!=null and bname!=''">
                b.bname like '%${bname}%'
            </if>
            <if test="uname!=null and uname!=''">
                and u.uname=#{uname}
            </if>
            and bo.bid=b.bid and bo.uid=u.uid and bo.deleted=0 and stat=1
        </where>
    </select>

    <select id="listPageHistoryBorrow" resultMap="borrowMapper">
        select * from borrow bo ,book b ,users u
        <where>
            <if test="bname!=null and bname!=''">
                b.bname like '%${bname}%'
            </if>
            <if test="uname!=null and uname!=''">
                and u.uname=#{uname}
            </if>
            and bo.bid=b.bid and bo.uid=u.uid and bo.deleted=0 and stat=1
        </where>

        limit #{page},#{size}
    </select>
</mapper>
```

5.4.用户DAO设计与实现

```
package com.book.app.dao;

import com.book.app.bean.Users;
import org.apache.ibatis.annotations.Param;
import org.springframework.stereotype.Repository;
import java.util.List;
@Repository
public interface UsersDao {


    List<Users> listPage(@Param("page") Integer page,
                         @Param("size")Integer size,
                         @Param("uname")String uname);

    int count(@Param("uname")String uname);

    int save(Users users);

    int remove(List<Integer> list);
```

```java
    int update(Users users);

    Users  login(@Param("uname")String uname);
}
```

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
        PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
        "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.book.app.dao.UsersDao">
    <insert id="save">
        insert into users(uname, password, images, createtime)
            value (#{uname},#{password},#{images},now())
    </insert>
    <update id="remove">
        update users set deleted=1 where uid in
        <foreach item="item" index="index" collection="list"
                 open="(" separator="," close=")">
            #{item}
        </foreach>
    </update>
    <update id="update">
        update users
        set uname=#{uname},
            password=#{password},
            images=#{images},
            modifytime=now()
        where uid = #{uid}
    </update>

    <select id="listPage" resultType="com.book.app.bean.Users">
        select * from users
        <where>
            <if test="uname!=null and uname!=''">
                uname like '%${uname}%'
            </if>
            deleted=0
        </where>
        limit #{page},#{size}
    </select>
    <select id="count" resultType="java.lang.Integer">
        select count(*) from users
        <where>
            <if test="uname!=null and uname!=''">
                uname like '%${uname}%'
            </if>
            deleted=0
        </where>
    </select>
    <select id="login" resultType="com.book.app.bean.Users">
        select  * from users where  uname=#{uname} and  deleted=0 and stats=0
    </select>
```

```
</mapper>
```

5.5.日志DAO设计与实现

```
package com.book.app.dao;

import com.book.app.bean.Logs;
import org.apache.ibatis.annotations.Param;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public interface LogsDao {

    int save(Logs logs);

    List<Logs> listPage(@Param("page") int page, @Param("size")int size,
                        @Param("opername")String opername);

    int count(@Param("opername")String opername);
    //统计来访用户次数
    List<Logs> countAccessTimes();

}
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
        PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
        "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.book.app.dao.LogsDao">
    <insert id="save">
        insert into logs(opername, opertime, ip, methods, ddesc)
            value (#{opername},now(),#{ip},#{methods},#{ddesc})
    </insert>
    <select id="listPage" resultType="com.book.app.bean.Logs">
        select * from logs
        <where>
            <if test="opername!=null and opername!=''">
                opername   like '%${opername}%'
            </if>
        </where>
          limit #{page},#{size}
    </select>
    <select id="count" resultType="java.lang.Integer">
        select count(*) from logs
        <where>
            <if test="opername!=null and opername!=''">
                opername   like '%${opername}%'
            </if>
        </where>
    </select>
    <select id="countAccessTimes" resultType="com.book.app.bean.Logs">
```

```
        select  opername ,count(*) as times from logs GROUP BY  opername
    </select>
</mapper>
```

第六章、图书借阅管理系统Service层设计与实现

6.1.馆藏图书业务层设计与实现

```
package com.book.app.service;

import com.book.app.bean.Book;

import java.util.List;
//馆藏图书接口
public interface BookService {

    List<Book>listPageBooks(int page,int size,String bname);

    int countBooks(String bname);

    boolean saveBook(Book book);

    boolean removeBook(List<Integer> list);

    boolean updateBook(Book book);

    List<Book> exportBooks(Book book);

    Book getBookById(int bid);

    void changStore(Book book);
}
```

```
package com.book.app.service.impl;

import com.book.app.bean.Book;
import com.book.app.dao.BookDao;
import com.book.app.service.BookService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;
//馆藏图书接口实现类
@Service
public class BookServiceImpl implements BookService {
    @Autowired
    private BookDao bookDao;

    @Override
    public List<Book> listPageBooks(int page, int size, String bname) {
        return bookDao.listPage((page-1)*size,size,bname);
    }

    @Override
    public int countBooks(String bname) {
        return bookDao.count(bname);
    }

    @Override
    public boolean saveBook(Book book) {
        return bookDao.save(book)>0;
    }

    @Override
    public boolean removeBook(List<Integer> list) {
        return bookDao.remove(list)>0;
    }

    @Override
    public boolean updateBook(Book book) {
        return bookDao.update(book)>0;
    }

    @Override
    public List<Book> exportBooks(Book book) {

        return bookDao.list(book);
    }

    @Override
    public Book getBookById(int bid) {
        return bookDao.get(bid);
    }

    @Override
    public void changStore(Book book) {
        bookDao.changStore(book);
    }
}
```

6.2.图书类别业务层设计与实现

```
package com.book.app.service;

import com.book.app.bean.Category;

import java.util.List;
//类别业务层接口
public interface CategoryService {

    List<Category> listPageCategory(int page , int size, String cateName);

    int countCategory(String cateName);

    boolean saveCategory(Category category);

    boolean removeBatchCategory(List<Integer> cids);

    boolean updateCategory(Category category);

    List<Category> listCategory();

    List<Category>countTypesBooks();
}
```

```
package com.book.app.service.impl;

import com.book.app.bean.Category;
import com.book.app.dao.CategoryDao;
import com.book.app.service.CategoryService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.List;
//类别业务层实现类
@Service
public class CategoryServiceImpl implements CategoryService {
    @Autowired
    private CategoryDao categoryDao;
    @Override
    public List<Category> listPageCategory(int page, int size, String cateName)
{
        return categoryDao.listPage((page-1)*size,size,cateName);
    }

    @Override
    public int countCategory(String cateName) {
        return categoryDao.count(cateName);
    }

    @Override
    public boolean saveCategory(Category category) {
        return categoryDao.save(category)>0;
    }

    @Override
    public boolean removeBatchCategory(List<Integer> cids) {
        return categoryDao.remove(cids)>0;
    }

    @Override
    public boolean updateCategory(Category category) {
        return categoryDao.update(category)>0;
    }

    @Override
    public List<Category> listCategory() {
        return categoryDao.list();
    }

    @Override
    public List<Category> countTypesBooks() {
        return categoryDao.countTypesBooks();
    }
}
```

6.3.图书借阅业务层设计与实现

```
package com.book.app.service;

import com.book.app.bean.Book;
import com.book.app.bean.Borrow;
import java.util.List;
//借阅业务层接口
public interface BorrowService {

    List<Borrow> listPageBorrow(int page,
                          int size,
                          String uname,
                          String bname);

    int countBorrow(String uname,
              String bname);

    boolean borrowBooks(Borrow borrow, Book book);

    boolean backBooks(int boid);

    int countHistoryBorrow(String uname, String bname);

    List<Borrow> listPageHistoryBorrow(Integer page, Integer size, String uname, String bname);
}
```

```
package com.book.app.service.impl;

import com.book.app.bean.Book;
import com.book.app.bean.Borrow;
import com.book.app.bean.Users;
import com.book.app.dao.BorrowDao;
import com.book.app.service.BookService;
import com.book.app.service.BorrowService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;
//借阅业务层实现类
@Service
@Transactional
public class BorrowImplService implements BorrowService {
    @Autowired
    private BorrowDao borrowDao;
    @Autowired
    private BookService bookService;
    @Override
    public List<Borrow> listPageBorrow(int page, int size, String uname, String bname) {
        return borrowDao.listPage((page-1)*size,size,uname,bname);
    }

    @Override
    public int countBorrow(String uname, String bname) {
        return borrowDao.count(uname,bname);
    }

    @Override
    public boolean borrowBooks(Borrow borrow, Book book) {
        book=bookService.getBookById(book.getBid());
        if(borrow.getQuantity()<=book.getQuantity()){
            //馆藏数量减去借阅数量，重新更新馆藏数量
            book.setQuantity(book.getQuantity()-borrow.getQuantity());
            //更新借阅馆藏数量
            bookService.changStore(book);
        }else{
            return false;
        }
        //添加借阅图书ID
        borrow.setBook(book);
        return borrowDao.borrow(borrow)>0;
    }

    @Override
    public boolean backBooks(int boid) {
        Borrow borrow=getBorrow(boid);
        Book book=bookService.getBookById( borrow.getBook().getBid());
        //最终归还数量是库存数量加上借阅数
        book.setQuantity(book.getQuantity()+borrow.getQuantity());
        bookService.changStore(book);
        borrow.setStats(1);//更改借阅状态
        return borrowDao.backBooks(borrow)>0;
    }

    @Override
    public int countHistoryBorrow(String uname, String bname) {
        return borrowDao.countHistoryBorrow(uname,bname);
    }

    @Override
    public List<Borrow> listPageHistoryBorrow(Integer page, Integer size, String uname, String bname) {
        return borrowDao.listPageHistoryBorrow((page-1)*size,size,uname,bname);
    }


    public Borrow getBorrow(int boid) {

        return borrowDao.getBorrow(boid);
    }
```

```
}
```

6.4.用户业务层设计与实现

```
package com.book.app.service;
import com.book.app.bean.Users;
import java.util.List;
//业务层接口
public interface UserService {

    List<Users> listPageUsers(Integer page, Integer size, String uname);

    int countUsers(String uname);

    boolean saveUsers(Users users);

    boolean removeUsers(List<Integer> list);

    boolean updateUsers(Users users);

    Users login(String uname);
}
```

```
package com.book.app.service.impl;

import com.book.app.bean.Users;
import com.book.app.dao.UsersDao;
import com.book.app.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;
//用户业务层实现类
@Service
public class UserServiceImpl implements UserService {
    @Autowired
    private UsersDao usersDao;

    @Override
    public List<Users> listPageUsers(Integer page, Integer size, String uname) {
        return usersDao.listPage((page-1)*size,size,uname);
    }

    @Override
    public int countUsers(String uname) {
        return usersDao.count(uname);
    }

    @Override
    public boolean saveUsers(Users users) {
        return usersDao.save(users)>0;
    }
    @Override
    public boolean removeUsers(List<Integer> list) {
        return usersDao.remove(list)>0;
    }

    @Override
    public boolean updateUsers(Users users) {
        return usersDao.update(users)>0;
    }

    @Override
    public Users login(String uname) {
        return usersDao.login(uname);
    }
}
```

6.5.日志业务层设计与实现

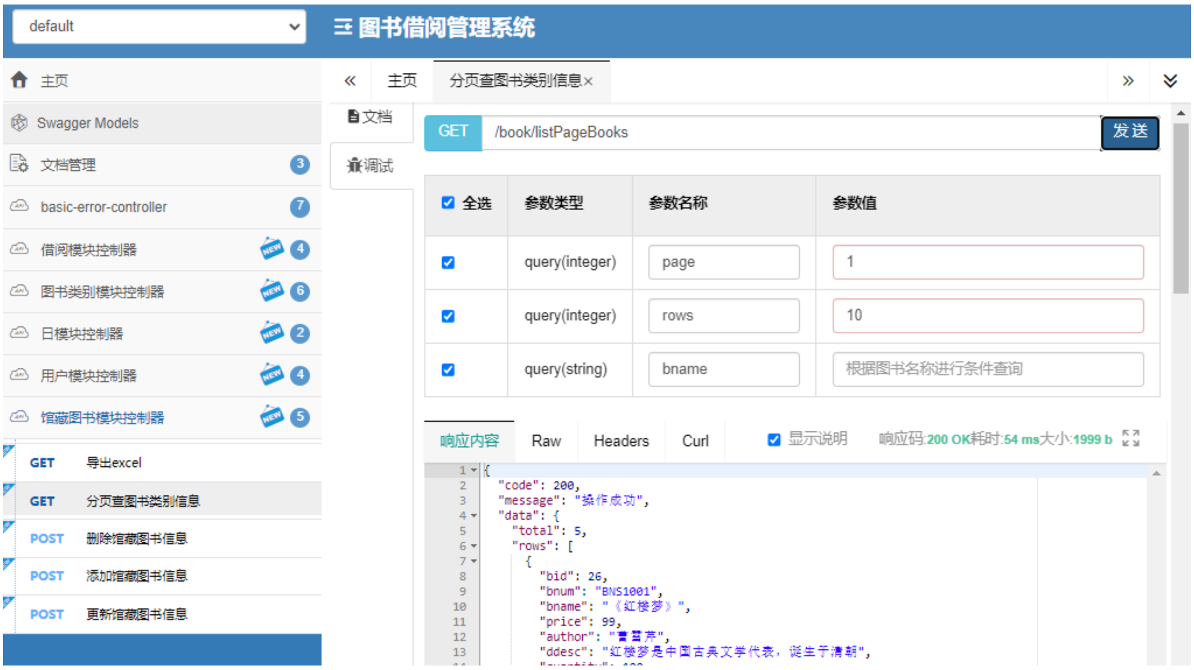
```
package com.book.app.service;

import com.book.app.bean.Logs;

import java.util.List;
//日志业务层接口
public interface LogsService {

    boolean saveLogs(Logs logs);

    List<Logs> listPageLogs(int page, int size,
                        String opername);

    int countLogs(String opername);

    List<Logs> countAccessTimes();

}
```

```
package com.book.app.service.impl;

import com.book.app.bean.Logs;
import com.book.app.dao.LogsDao;
import com.book.app.service.LogsService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;
//日志业务层实现类
@Service
public class LogsServiceImpl implements LogsService {
    @Autowired
    private LogsDao logsDao;
    @Override
    public boolean saveLogs(Logs logs) {
        return logsDao.save(logs)>0;
    }

    @Override
    public List<Logs> listPageLogs(int page, int size, String opername) {
        return logsDao.listPage((page-1)*size,size,opername);
    }

    @Override
    public int countLogs(String opername) {
        return logsDao.count(opername);
    }

    @Override
    public List<Logs> countAccessTimes() {
        return logsDao.countAccessTimes();
    }
}
```

第七章、图书借阅管理系统Controller层设计与实现

7.1.馆藏图书控制器类设计与实现

```
package com.book.app.controller;

import com.alibaba.excel.EasyExcel;
import com.book.app.bean.Book;
import com.book.app.bean.Category;
import com.book.app.service.BookService;
import com.book.app.tools.CommonResult;
import io.swagger.annotations.Api;
import io.swagger.annotations.ApiImplicitParam;
import io.swagger.annotations.ApiImplicitParams;
import io.swagger.annotations.ApiOperation;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import javax.servlet.http.HttpServletResponse;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

@Api(tags = "馆藏图书模块控制器")
@RestController
@RequestMapping("/book")
public class BookController {
    @Autowired
    private BookService bookService;

    @ApiOperation("分页查图书类别信息")
    @ApiImplicitParams({
            @ApiImplicitParam(name = "page",
                    value = "分页的起始时间", required = true, example = "1"),
            @ApiImplicitParam(name = "rows", value = "每页显示记录数",
                    required = true, example = "10"),
            @ApiImplicitParam(name = "bname", value = "根据图书名称进行条件查询",
                    required = false, example = "古典文学"),
    })
    @GetMapping("/listPageBooks")
    public CommonResult listPageBooks(Integer page, Integer rows, String bname)
{
        Map<String, Object> map = new HashMap<>();
        map.put("total", bookService.countBooks(bname));
        map.put("rows", bookService.listPageBooks(page, rows, bname));
        return CommonResult.success(map);
    }

    @ApiOperation("添加馆藏图书信息")
    @PostMapping("/saveBook")
    public CommonResult saveBook(Book book, Category category) {
        book.setCategory(category);
        return CommonResult.success(bookService.saveBook(book));
    }

    @ApiOperation("更新馆藏图书信息")
    @PostMapping("/updateBook")
    public CommonResult updateBook(Book book, Category category) {
        book.setCategory(category);
        return CommonResult.success( bookService.updateBook(book));
    }

    @ApiOperation("删除馆藏图书信息")
    @PostMapping("/removeBook")
    public CommonResult removeBook(@RequestParam("bids[]") List<Integer> list) {
        return CommonResult.success(bookService.removeBook(list));
    }

    @ApiOperation("导出excel")
    @GetMapping("/excelExport")
    public CommonResult excelExport(HttpServletResponse response, Book book) {
        List<Book> list = bookService.exportBooks(book);
        String strUrl = "C:\\Users\\Administrator\\Desktop\\" + "图书馆藏信息" +
System.currentTimeMillis() + ".xlsx";
        EasyExcel.write(strUrl, Book.class).sheet("模板").doWrite(list);
        return CommonResult.success(true);
    }



}
```



GET /book/listPageBooks

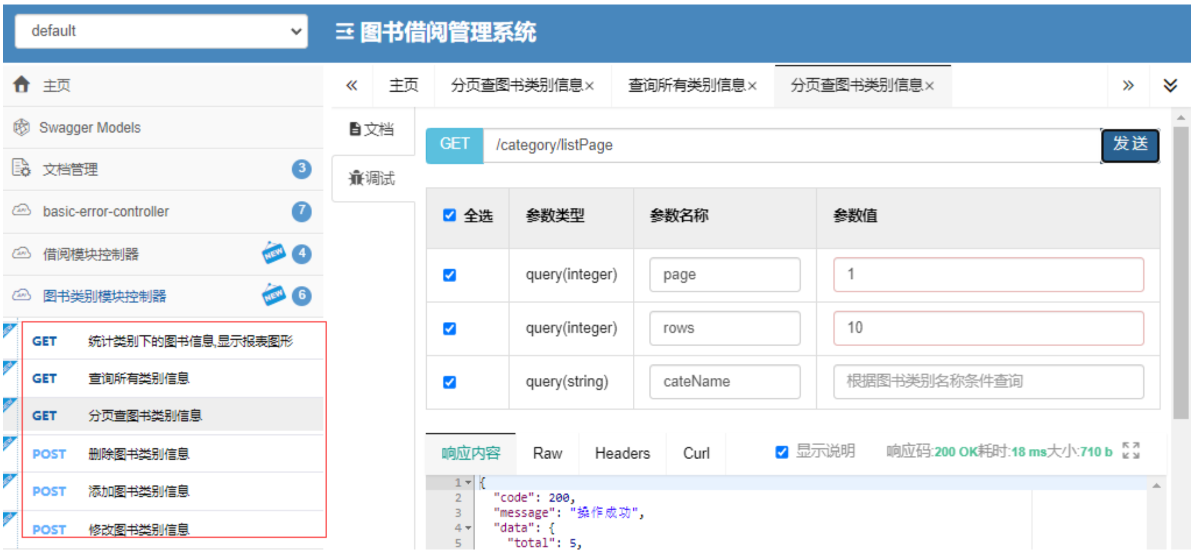
☑ 全选	参数类型	参数名称	参数值
☑	query(integer)	page	1
☑	query(integer)	rows	10
☑	query(string)	bname	根据图书名称进行条件查询

响应码:200 OK耗时:54 ms大小:1999 b

```
{
  "code": 200,
  "message": "操作成功",
  "data": {
    "total": 5,
    "rows": [
      {
        "bid": 26,
        "bnum": "BNS1001",
        "bname": "《红楼梦》",
        "price": 99,
        "author": "曹雪芹",
        "ddesc": "红楼梦是中国古典文学代表，诞生于清朝",
```

7.2.图书类别控制器类设计与实现

```java
package com.book.app.controller;

import com.book.app.bean.Category;
import com.book.app.service.CategoryService;
import com.book.app.tools.CommonResult;
import io.swagger.annotations.Api;
import io.swagger.annotations.ApiImplicitParam;
import io.swagger.annotations.ApiImplicitParams;
import io.swagger.annotations.ApiOperation;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.HashMap;
import java.util.List;
import java.util.Map;
@Api(tags = "图书类别模块控制器")
@RestController
@RequestMapping("/category")
public class CategoryController {
    @Autowired
    private CategoryService categoryService;
     @ApiOperation("分页查图书类别信息")
     @ApiImplicitParams({
             @ApiImplicitParam(name = "page",
                     value = "分页的起始时间",required = true,example = "1"),
             @ApiImplicitParam(name = "rows",value = "每页显示记录数",
                     required = true,example = "10"),
             @ApiImplicitParam(name = "cateName",value = "根据图书类别名称条件查询",
                     required = false,example = "古典文学"),
     })
    @GetMapping("/listPage")
    public CommonResult listPage(int page, int rows, String cateName){
        Map<String,Object> map=new HashMap<>();
        map.put("total",categoryService.countCategory(cateName));
        map.put("rows",categoryService.listPageCategory(page,rows,cateName));
        return CommonResult.success(map);
    }

    @ApiOperation(value = "添加图书类别信息")
    @PostMapping("/saveCategory")
    public  CommonResult saveCategory(@RequestBody Category category){
        return CommonResult.success(categoryService.saveCategory(category));
    }


    @ApiOperation(value = "修改图书类别信息")
    @PostMapping("/updateCategory")
    public  CommonResult updateCategory(@RequestBody Category category){
        return CommonResult.success(categoryService.updateCategory(category));
    }

    @ApiOperation(value = "删除图书类别信息")
    @PostMapping("/removeBatchCategory")
    public  CommonResult removeBatchCategory(@RequestParam("cids[]")
List<Integer> cids){
        return CommonResult.success(categoryService.removeBatchCategory(cids)) ;
    }

    @ApiOperation(value = "查询所有类别信息")
    @GetMapping("/listCategory")
    public  CommonResult listCategory(){
        return CommonResult.success(categoryService.listCategory());
    }

    @ApiOperation(value = "统计类别下的图书信息,显示报表图形")
    @GetMapping("/countTypesBooks")
    public  CommonResult countTypesBooks(){
        return CommonResult.success(categoryService.countTypesBooks()) ;
    }
}
```



GET /category/listPage

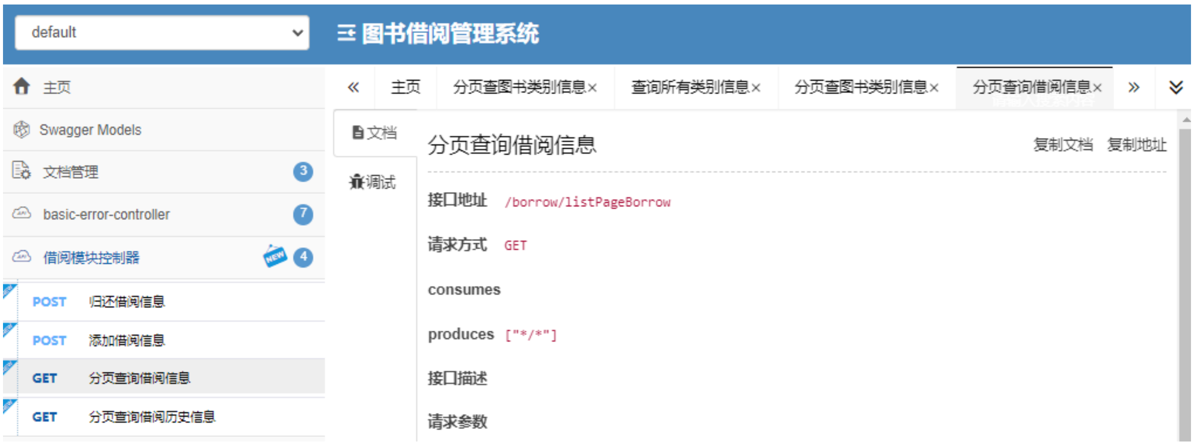
☑ 全选	参数类型	参数名称	参数值
☑	query(integer)	page	1
☑	query(integer)	rows	10
☑	query(string)	cateName	根据图书类别名称条件查询

响应码:200 OK耗时:18 ms大小:710 b

```
{
  "code": 200,
  "message": "操作成功",
  "data": {
    "total": 5,
```

7.3.图书借阅控制器类设计与实现

```java
package com.book.app.controller;

import com.book.app.bean.Book;
import com.book.app.bean.Borrow;
import com.book.app.service.BorrowService;
import com.book.app.tools.CommonResult;
import io.swagger.annotations.Api;
import io.swagger.annotations.ApiImplicitParam;
import io.swagger.annotations.ApiImplicitParams;
import io.swagger.annotations.ApiOperation;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.HashMap;
import java.util.Map;

@RestController
@RequestMapping("/borrow")
@Api(tags = "借阅模块控制器")
public class BorrowController {
    @Autowired
    private BorrowService borrowService;


    @ApiOperation("分页查询借阅信息")
    @ApiImplicitParams({
            @ApiImplicitParam(name = "page",
                    value = "分页的起始时间",required = true,example = "1"),
            @ApiImplicitParam(name = "rows",value = "每页显示记录数",
                    required = true,example = "10"),
            @ApiImplicitParam(name = "uname",value = "根据借阅人名称查询",
                    example = "张三"),
            @ApiImplicitParam(name = "bname",value = "根据借阅图书名称查询",
                    example = "《等风来》"),
    })
    @GetMapping("/listPageBorrow")
    public CommonResult listPageBooks(Integer page, Integer rows,
                                      String uname, String bname) {
        Map<String, Object> map = new HashMap<>();
        map.put("total", borrowService.countBorrow(uname, bname));
        map.put("rows", borrowService.listPageBorrow(page, rows, uname, bname));
        return CommonResult.success(map);
    }
    @ApiOperation("分页查询借阅历史信息")
    @ApiImplicitParams({
            @ApiImplicitParam(name = "page",
                    value = "分页的起始时间",required = true,example = "1"),
            @ApiImplicitParam(name = "rows",value = "每页显示记录数",
                    required = true,example = "10"),
            @ApiImplicitParam(name = "uname",value = "根据借阅人名称查询",
                    example = "张三"),
            @ApiImplicitParam(name = "bname",value = "根据借阅图书名称查询",
                    example = "《等风来》"),
    })
    @GetMapping("/listPageHistoryBorrow")
    public CommonResult listPageHistoryBorrow(Integer page,
                                                     Integer rows, String uname,
String bname) {
        Map<String, Object> map = new HashMap<>();
        map.put("total", borrowService.countHistoryBorrow(uname, bname));
        map.put("rows", borrowService.listPageHistoryBorrow(page, rows, uname,
bname));
        return CommonResult.success(map);
    }


    @ApiOperation("添加借阅信息")
    @PostMapping("/borrowBooks")
    public CommonResult borrowBooks(@RequestBody Borrow borrow, @RequestBody
Book book) {
        return CommonResult.success(borrowService.borrowBooks(borrow, book));
    }
    @ApiOperation("归还借阅信息")
    @ApiImplicitParams({
            @ApiImplicitParam(name = "boid",value = "归还图书信息ID",
                    example = "1"),
    })
    @PostMapping("/backBooks")
    public CommonResult backBooks(Integer boid) {
        return CommonResult.success(borrowService.backBooks(boid));
    }

}
```



分页查询借阅信息

接口地址 /borrow/listPageBorrow

请求方式 GET

consumes

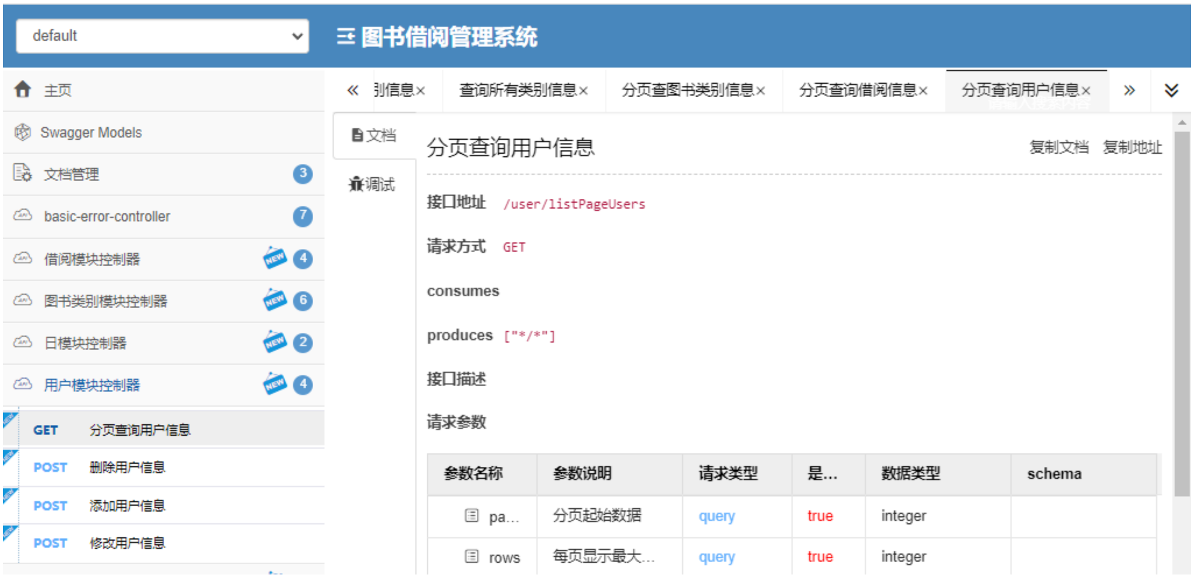
produces ["*/*"]

接口描述

请求参数

7.4.用户控制器类设计与实现

```
package com.book.app.controller;

import com.book.app.bean.Users;
import com.book.app.service.UserService;
import com.book.app.tools.CommonResult;
import io.swagger.annotations.Api;
import io.swagger.annotations.ApiImplicitParam;
import io.swagger.annotations.ApiImplicitParams;
import io.swagger.annotations.ApiOperation;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

@RestController
@RequestMapping("/user")
@Api(tags = "用户模块控制器")
public class UserController {
    @Autowired
    private UserService userService;

    @ApiImplicitParams({
            @ApiImplicitParam(name = "page",
                    value = "分页起始数据",
                    dataType = "integer",
                    required = true, example = "1"),
            @ApiImplicitParam(name = "rows",
                    value = "每页显示最大记录条数",
                    dataType = "integer",
                    required = true, example = "10"),
            @ApiImplicitParam(name = "uname",
                    value = "根据用户名称进行条件查询"
                    , example = "张三"),
    })
    @ApiOperation("分页查询用户信息")
    @GetMapping("/listPageUsers")
    public CommonResult listPageUsers(Integer page, Integer rows, String uname)
{
        Map<String, Object> map = new HashMap<>();
        map.put("total", userService.countUsers(uname));
        map.put("rows", userService.listPageUsers(page, rows, uname));
        return CommonResult.success(map);
    }

    @ApiOperation("添加用户信息")
    @PostMapping("/saveUsers")
    public CommonResult saveUsers(Users users) {
        return CommonResult.success(userService.saveUsers(users));
    }

    @ApiOperation("删除用户信息")
    @PostMapping("/removeUsers")
    public CommonResult removeUsers(@RequestParam("uids[]") List<Integer> list)
{
        return  CommonResult.success(userService.removeUsers(list)) ;
    }

    @ApiOperation("修改用户信息")
    @PostMapping("/updateUsers")
    public CommonResult updateUsers(Users users) {
        return CommonResult.success(userService.updateUsers(users))  ;
    }
}
```



分页查询用户信息

接口地址 /user/listPageUsers

请求方式 GET

consumes

produces ["*/*"]

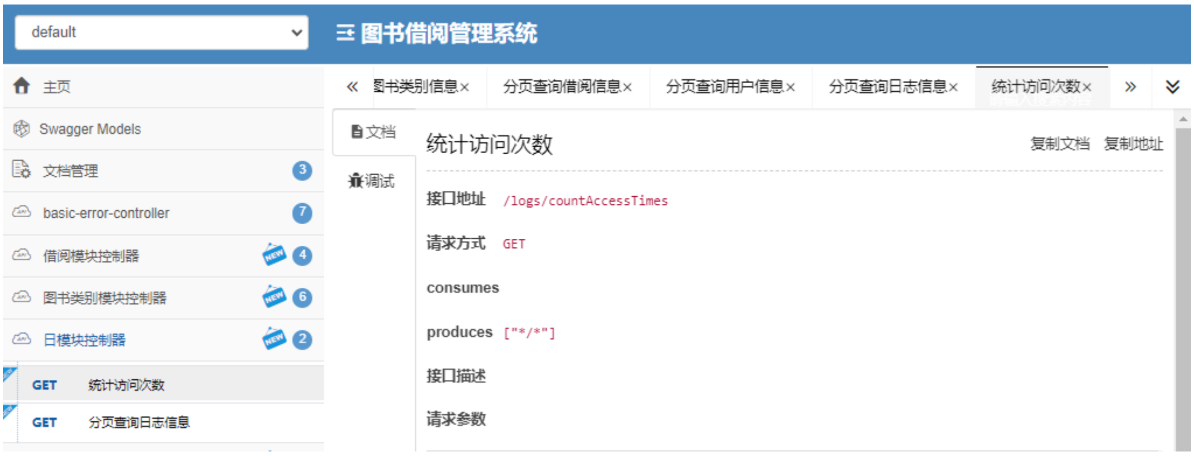
接口描述

请求参数

参数名称	参数说明	请求类型	是...	数据类型	schema
pa...	分页起始数据	query	true	integer	
rows	每页显示最大...	query	true	integer	

7.5.用户控制器类设计与实现

```java
package com.book.app.controller;

import com.book.app.service.LogsService;
import com.book.app.tools.CommonResult;
import io.swagger.annotations.Api;
import io.swagger.annotations.ApiImplicitParam;
import io.swagger.annotations.ApiImplicitParams;
import io.swagger.annotations.ApiOperation;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.HashMap;
import java.util.Map;

@RestController
@RequestMapping("/logs")

@Api(tags = "日模块控制器")
public class LogsController {
    @Autowired
    private LogsService logsService;

    @ApiOperation("分页查询日志信息")
    @ApiImplicitParams({
            @ApiImplicitParam(name = "page",
                    value = "分页的起始时间",required = true,example = "1"),
            @ApiImplicitParam(name = "rows",value = "每页显示记录数",
                    required = true,example = "10"),
            @ApiImplicitParam(name = "opername",value = "根据操作人进行查询",
                    example = "张三"),
    })
    @GetMapping("/listPageLogs")
    public CommonResult listPageLogs(int page, int rows, String opername) {
        Map<String, Object> map = new HashMap<>();
        map.put("total", logsService.countLogs(opername));
        map.put("rows", logsService.listPageLogs(page, rows, opername));
        return CommonResult.success(map);
    }
    @ApiOperation("统计访问次数")
    @GetMapping("/countAccessTimes")
    public CommonResult countAccessTimes() {
        return CommonResult.success(logsService.countAccessTimes());
    }
}
```



统计访问次数

接口地址 /logs/countAccessTimes

请求方式 GET

consumes

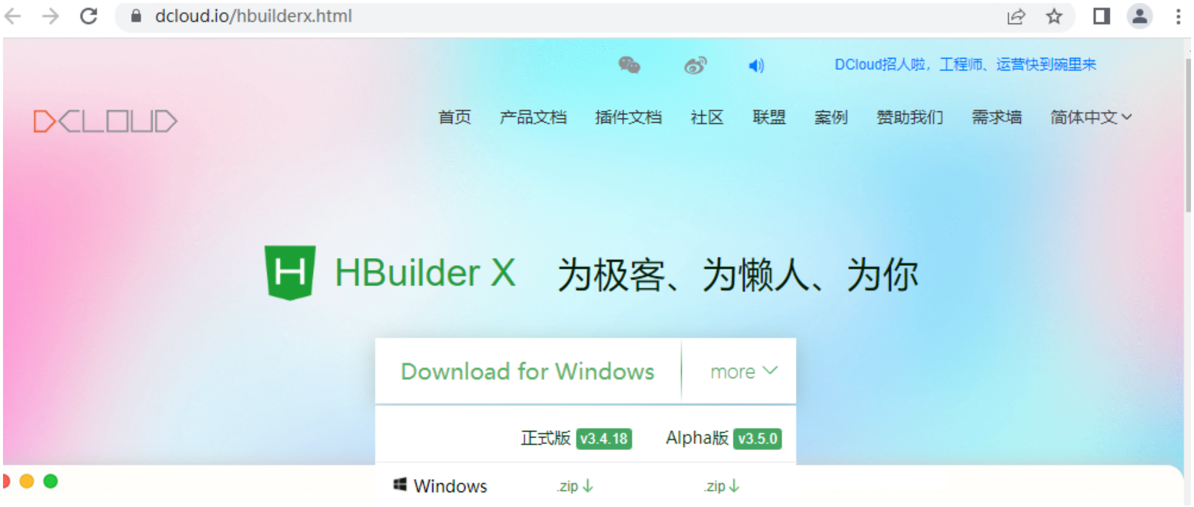
produces ["*/*"]

接口描述

请求参数

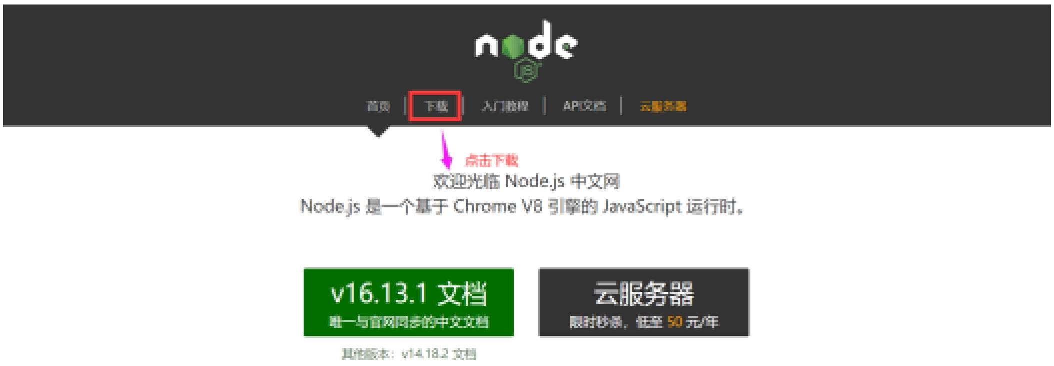
第八章、图书借阅管理系统客户端设计与实现

8.1.hbuilder开发工具下载与安装


dcloud.io/hbuilderx.html
DCloud招人啦，工程师、运营快到碗里来
DCLOUD
首页 产品文档 插件文档 社区 联盟 案例 赞助我们 需求墙 简体中文
HBuilder X 为极客、为懒人、为你
Download for Windows
more
正式版 v3.4.18
Alpha版 v3.5.0
Windows .zip .zip

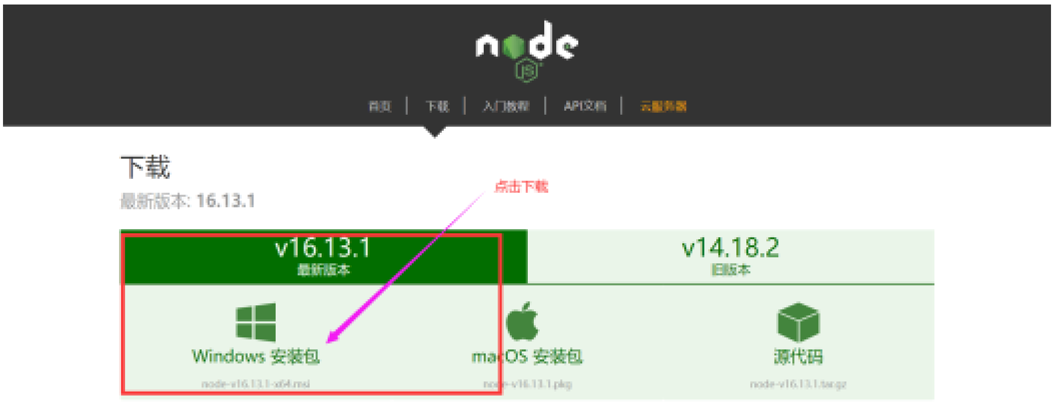
8.2.node服务器下载与安装

```
1.3.1.node.js服务器安装  访问网址:http://nodejs.cn/
```



欢迎光临 Node.js 中文网

Node.js 是一个基于 Chrome V8 引擎的 JavaScript 运行时。



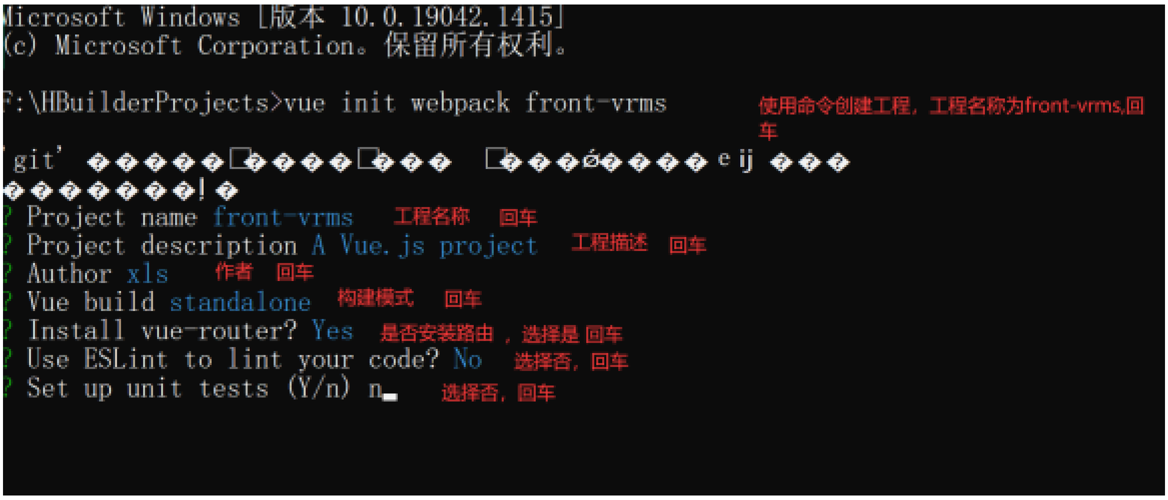
下载

最新版本: 16.13.1

8.3.vueCli安装

```
/**npm简介
node.js服务器安装 访问网址:http://nodejs.cn/9.3.vueCli简介
NPM是随同NodeJS一起安装的包管理工具，能解决NodeJS代码部署上的很多问题，常见的使用场景有以下几种：
允许用户从NPM服务器下载别人编写的第三方包到本地使用。
允许用户从NPM服务器下载并安装别人编写的命令行程序到本地使用。
允许用户将自己编写的包或命令行程序上传到NPM服务器供别人使用。
由于新版的nodejs已经集成了npm，所以之前npm也一并安装好了。
同样可以通过输入 "npm -v" 来测试是否成功安 装。命令如下，出现版本提示表示安装成功*/
```

```
/**
Vue CLI 是一个基于 Vue.js 进行快速开发的完整系统，通过 @vue/cli 快速创建前端工程
配置阿里加速地址： npm install -g cnpm --registry=https://registry.npm.taobao.org
安装命令 npm install -g @vue/cli
查看版本命令vue --version
使用webpack创建项目
步骤1 进入要创建项目的盘符或者文件夹输入如下命令: vue init webpack 工程名称
*/
```



```
licrosoft Windows [版本 10.0.19042.1415]
(c) Microsoft Corporation。保留所有权利。

F:\HBuilderProjects>vue init webpack front-vrms

'git' ������□�����□���  □�����ó���� e ij ���
��������! �
? Project name front-vrms
? Project description A Vue.js project
? Author xls
? Vue build standalone
? Install vue-router? Yes
? Use ESLint to lint your code? No
? Set up unit tests (Y/n) n
```

使用命令创建工程，工程名称为front-vrms,回车

工程名称 回车

工程描述 回车

作者 回车

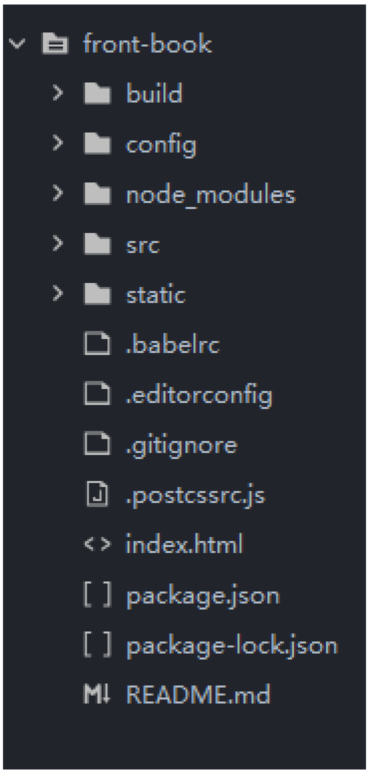
构建模式 回车

是否安装路由，选择是 回车

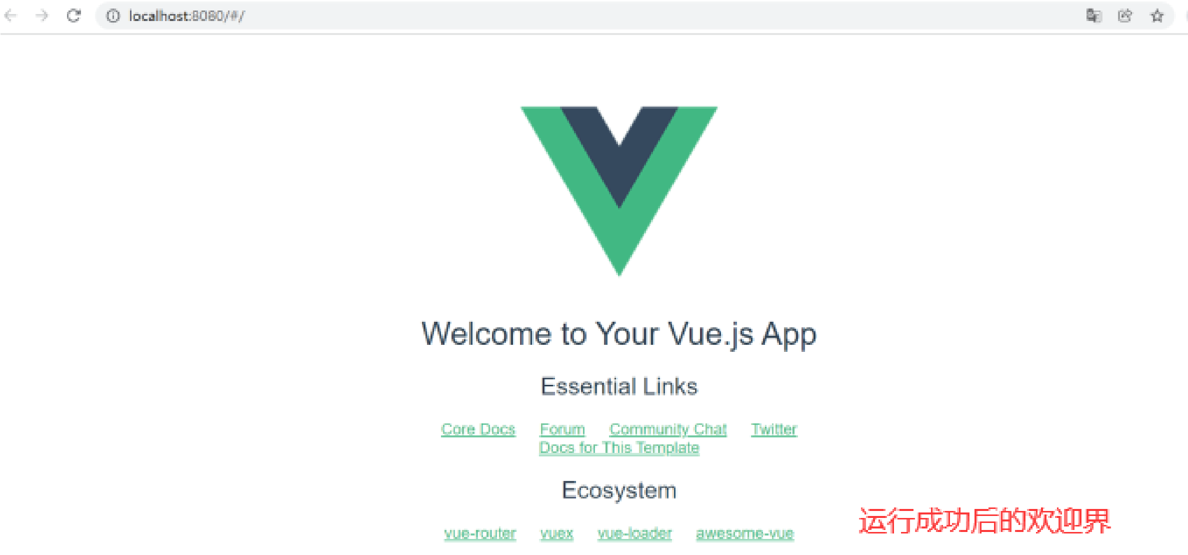
选择否，回车

选择否，回车

8.4客户端工程目录结构



```
front-book
  build
  config
  node_modules
  src
  static
  .babelrc
  .editorconfig
  .gitignore
  .postcssrc.js
  index.html
  package.json
  package-lock.json
  README.md
```



localhost:8080/#/

Welcome to Your Vue.js App

Essential Links

Core Docs Forum Community Chat Twitter
Docs for This Template

Ecosystem

vue-router vuex vue-loader awesome-vue

运行成功后的欢迎界

第九章、图书借阅管理系统工程前端架构搭建

9.1.Element概述

```
Element，一套为开发者、设计师和产品经理准备的基于 Vue 2.0 的桌面端组件库，官网：
https://element.eleme.cn/#/zh-CN
```

9.2.Element开发环境构建

```
//开发环境安装进入工程目录运行如下命令: npm i element-ui -S
//在main.js中引入如下命令
import ElementUI from 'element-ui';
import 'element-ui/lib/theme-chalk/index.css';
Vue.use(ElementUI);
```

9.3.axios概述

```
//axios是vue框架提倡使用的ajax库
//axios安装 进入工程目录使用 npm install axios -S命令安装
//main.js中引用
import axios from 'axios'
Vue.prototype.$http=axios
```

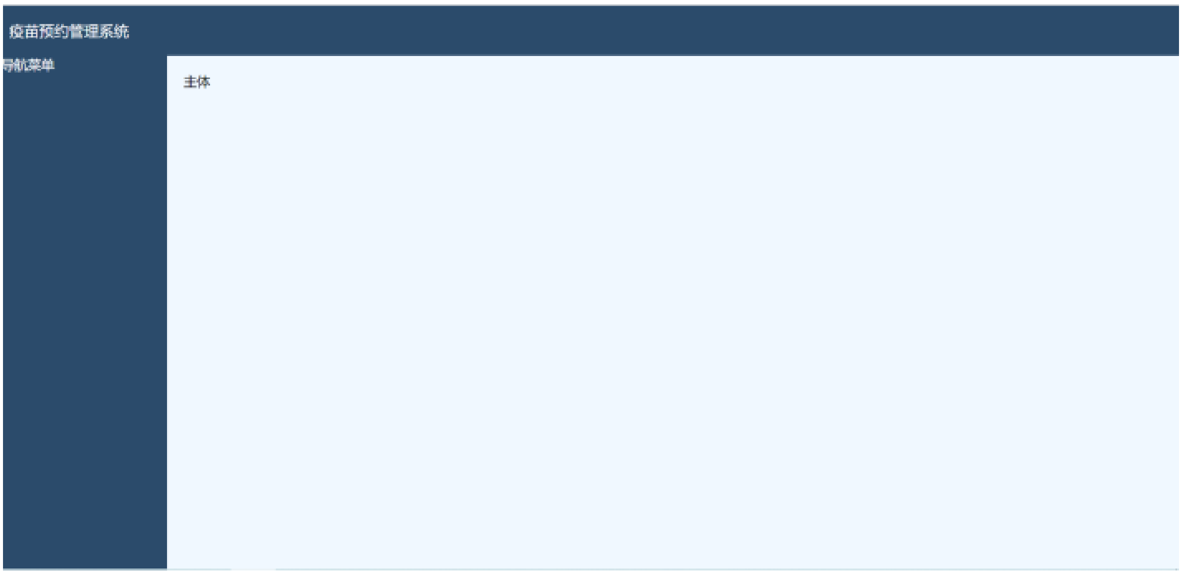
9.4.样式引入

```
//在工程静态资源中编写main.css
html,
body,
#app {
margin: 0px;
height: 100%;
padding: 0px;
}
/
/在main.js中引入
import './assets/css/main.css'
```

9.5.局部组件Main.vue创建

```
<template>
<el-container>
<el-header>疫苗预约管理系统</el-header>
<el-container>
<el-aside width="200px">导航菜单</el-aside>
<el-main>主体</el-main>
</el-container>
</el-container>
</template>
<script>
export default {
data() {
return {}
}
}
```

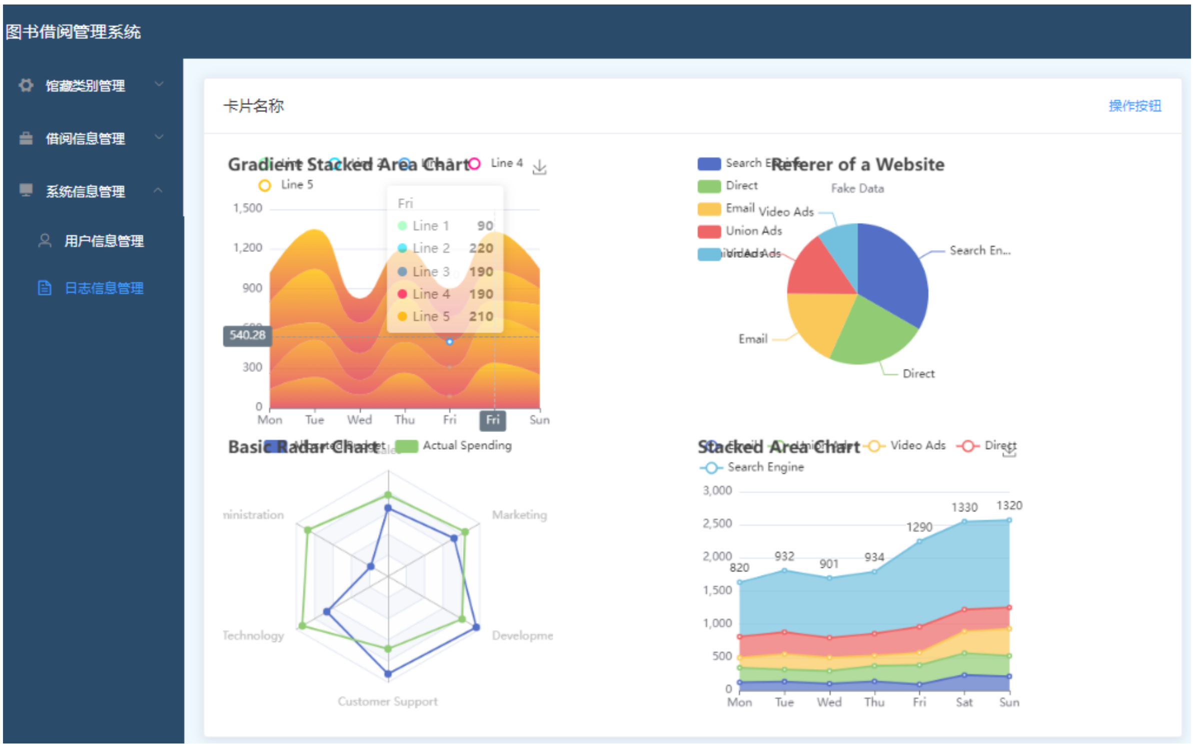
```
</script>
<style scoped>
.el-header {
background-color: #2B4B6B;
display: flex;
justify-content: space-between;
padding-left: 10px;
align-items: center;
color: white;
font-size: 18px;
} .
el-aside {
color: white;
background-color: #2B4B6B;
} .
el-main {
background-color: aliceblue;
} .
el-container {
height: 100%;}
</style>
```



9.6.导航菜单设计与实现

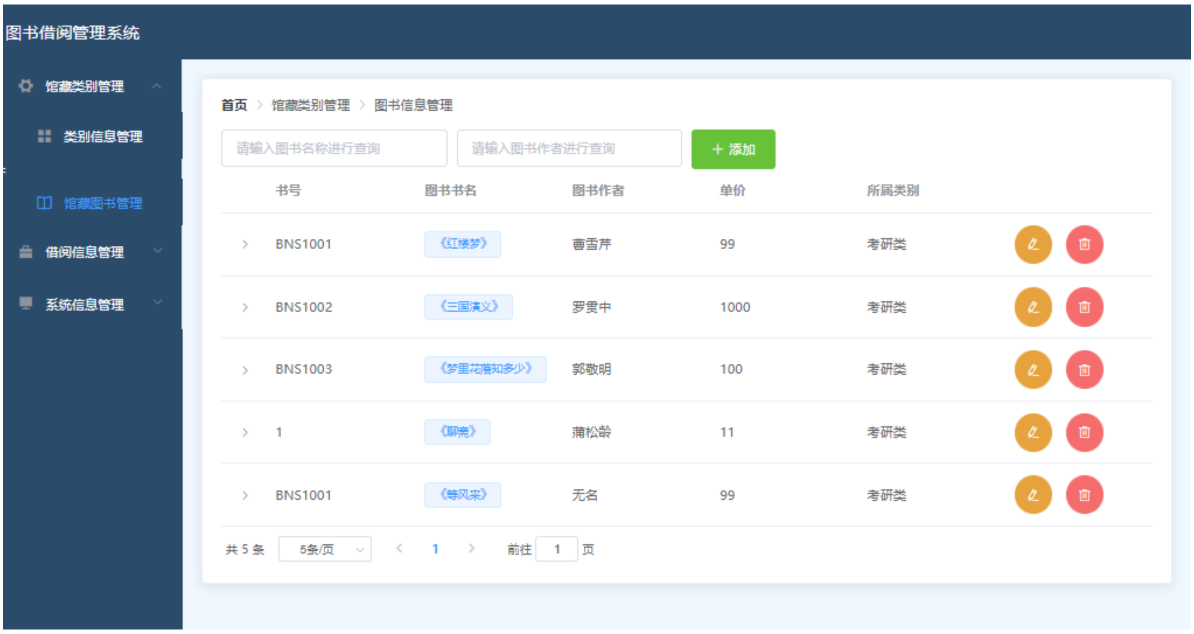
```
//在路由(router)文件夹中的index.js中编写如下代码
import Vue from 'vue'
import Router from 'vue-router'
import Main from '../components/Main.vue'
import Category from '../components/Category.vue'
import Book from '../components/Book.vue'
import Logs from '../components/Logs.vue'
import Borrow from '../components/Borrow.vue'
import Users from '../components/Users.vue'
import Welcome from '../components/Welcom.vue'
Vue.use(Router)

export default new Router({
  routes: [
    {
      path: '/',
      name: 'Main',
      component: Main,
      redirect : '/welcome',
      children:[
        {
          path:'/welcome',
          name: Welcome,
          component: Welcome
        },
        {
        path: '/category',
        name: 'Category',
        component: Category,
      },{
        path: '/book',
        name: 'book',
        component: Book,
      },{
        path: '/borrow',
        name: 'Borrow',
        component: Borrow,
      },{
        path: '/users',
        name: 'Users',
        component: Users,
      },{
        path: '/logs',
        name: 'Logs',
        component: Logs,
      }]
    }
  ]
})
```

```
<template>
  <el-container>
    <el-header>图书借阅管理系统</el-header>
    <el-container>
      <el-aside width="200px">
        <el-menu router background-color="#2B4B6B" text-color="#fff"
         active-text-color="#409EFF" unique-opened>
         <el-submenu index="3">
           <template slot="title">
             <i class="el-icon-s-tools"></i>
             <span>馆藏类别管理</span>
           </template>
           <el-menu-item index="/category">
             <i class="el-icon-menu"></i>
             <span slot="title">类别信息管理</span>
           </el-menu-item>=
           <el-menu-item index="/book">
             <i class="el-icon-reading"></i>
             <span slot="title">馆藏图书管理</span>
           </el-menu-item>
         </el-submenu>
         <el-submenu index="2">
           <template slot="title">
             <i class="el-icon-s-cooperation"></i>
             <span>借阅信息管理</span>
           </template>
           <el-menu-item index="/borrow">
             <i class="el-icon-files"></i>
             <span slot="title">借阅信息管理</span>
           </el-menu-item>
         </el-submenu>
          <el-submenu index="1">
            <template slot="title">
              <i class="el-icon-s-platform"></i>
              <span>系统信息管理</span>
            </template>
            <el-menu-item index="/users">
              <i class="el-icon-user"></i>
              <span slot="title">用户信息管理</span>
            </el-menu-item>
            <el-menu-item index="/logs">
              <i class="el-icon-document"></i>
              <span slot="title">日志信息管理</span>
            </el-menu-item>
          </el-submenu>
        </el-menu>
      </el-aside>
      <el-main>
        <!-- 路由占位符 -->
        <router-view/>
      </el-main>
    </el-container>
  </el-container>
</template>

<script>
  export default {
    data() {
      return {

      }
    }
  }
</script>

<style scoped>
  .el-header {
    background-color: #2B4B6B;
    display: flex;
    justify-content: space-between;
    padding-left: 10px;
    align-items: center;
    color: white;
    font-size: 18px;
  }

  .el-aside {
    color: white;
    background-color: #2B4B6B;
  }

  .el-main {
    background-color: aliceblue;
  }

  .el-container {
    height: 100%;
  }
</style>
```



第十章、图书借阅管理系统前端各个UI模块设计与实现

10.1.馆藏图书模块UI设计与实现

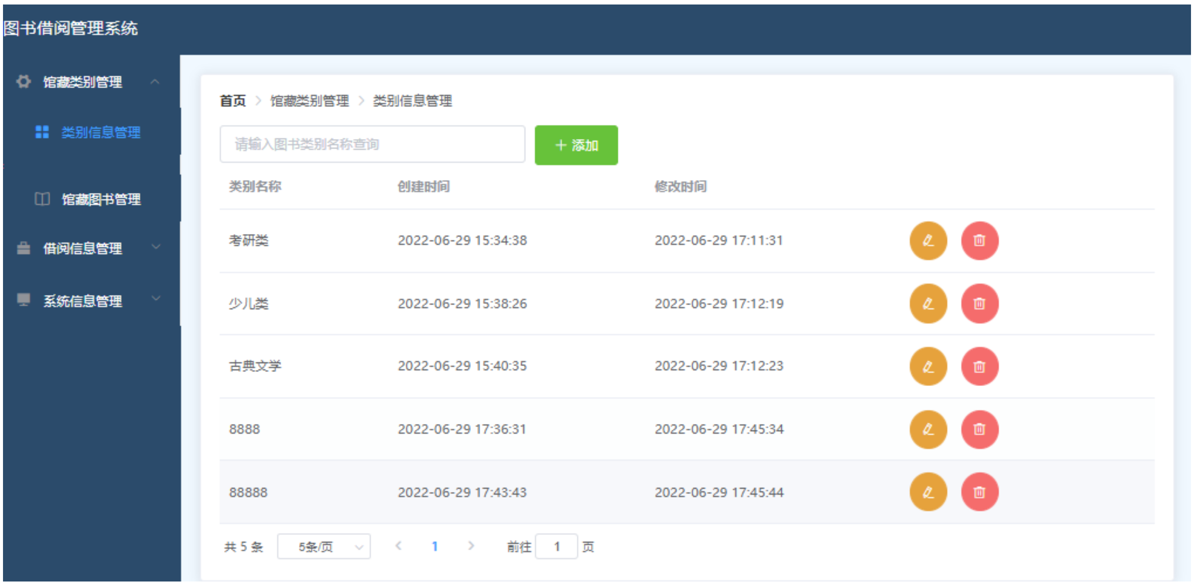


	书号	图书书名	图书作者	单价	所属类别	
>	BNS1001	《红楼梦》	曹雪芹	99	考研类	
>	BNS1002	《三国演义》	罗贯中	1000	考研类	
>	BNS1003	《梦里花落知多少》	郭敬明	100	考研类	
>	1	《聊斋》	蒲松龄	11	考研类	
>	BNS1001	《等风来》	无名	99	考研类	

```
<template>
  <!-- 卡片组件 -->
  <el-card>
    <!-- 面包屑组件 -->
    <el-breadcrumb separator-class="el-icon-arrow-right">
      <el-breadcrumb-item :to="{ path: '/welcome' }">首页</el-breadcrumb-item>
      <el-breadcrumb-item>馆藏类别管理</el-breadcrumb-item>
      <el-breadcrumb-item>图书信息管理</el-breadcrumb-item>
    </el-breadcrumb>
    <el-row :gutter="10">
      <el-col :span="6">
        <el-input clearable @change="listPage()" v-model="book.bname"
placeholder="请输入图书名称进行查询"></el-input>
      </el-col>
      <el-col :span="6">
        <el-input clearable @change="listPage()" v-model="book.author"
placeholder="请输入图书作者进行查询"></el-input>
      </el-col>
      <el-col :span="12">
        <el-button icon="el-icon-plus" @click="openAddDialog" type="success">添加
</el-button>
      </el-col>
    </el-row>
    <!-- 表格组件 -->
    <el-table :data="tableData">
      <el-table-column type="expand" prop="ddesc">
        <template slot-scope="scope">
          <el-tag>简介: {{scope.row.ddesc}}</el-tag>
        </template>
      </el-table-column>
      <el-table-column prop="bnum" label="书号">
      </el-table-column>
      <el-table-column prop="bname" label="图书书名">
        <template slot-scope="scope">
          <el-popover trigger="hover" placement="top">
            <p>图书简介: {{ scope.row.ddesc }}</p>
            <p>库存数量: {{ scope.row.quantity }}</p>
            <div slot="reference" class="name-wrapper">
```

```
              <el-tag size="medium">{{ scope.row.bname }}</el-tag>
            </div>
          </el-popover>
        </template>
      </el-table-column>
      <el-table-column prop="author" label="图书作者">
      </el-table-column>

      <el-table-column prop="price" label="单价">
      </el-table-column>

      <el-table-column prop="category" label="所属类别">
        <template slot-scope="scope">
          {{scope.row.category.catename}}
        </template>
      </el-table-column>
      <el-table-column lable="操作">
        <!-- 作用域插槽 -->
        <template slot-scope="scope">
          <el-tooltip content="点击修改" placement="bottom">
            <el-button type="warning" @click="openEditDialog(scope.row)"
icon="el-icon-edit" circle></el-button>
          </el-tooltip>
          <el-tooltip content="点击删除" placement="bottom">
            <el-button type="danger" @click="remove(scope.row)" icon="el-icon-
delete" circle></el-button>
          </el-tooltip>
        </template>
      </el-table-column>
    </el-table>
    <!-- 分页组件定义 -->
    <el-pagination @size-change="handleSizeChange" @current-
change="handleCurrentChange" :current-page="pager.page"
      :page-sizes="[5, 10, 15, 20]" :page-size="pager.size" layout="total,
sizes, prev, pager, next, jumper"
      :total="pager.total">
    </el-pagination>

    <!-- 添加对话框 -->
    <el-dialog title="图书类别添加" :show-close="false" :close-on-click-
modal="false" :visible.sync="addDialogFlag">
      <el-form :model="book" :rules="rules" ref="book">
        <el-form-item label="类别名称" prop="catename">
          <el-input v-model="book.catename" placeholder="请输入类别名称"></el-
input>
        </el-form-item>
      </el-form>
      <div slot="footer" class="dialog-footer">
        <el-button @click="addDialogFlag=!addDialogFlag">取 消</el-button>
        <el-button type="primary" @click="save">保存</el-button>
      </div>
    </el-dialog>

    <!-- 修改对话框 -->
    <el-dialog title="图书类别修改" :show-close="false" :close-on-click-modal="false" :visible.sync="editDialogFlag">

      <div slot="footer" class="dialog-footer">
        <el-button @click="editDialogFlag=!editDialogFlag">取 消</el-button>
        <el-button type="primary" @click="update">保存</el-button>
      </div>
    </el-dialog>

  </el-card>



</template>

<script>
  export default {
    data() {
      return {
        rules: {
          catename: [{
            required: true,
            message: '请输入类别名称',
            trigger: 'blur'
          }]
        }, //定义表单验证规则
        editDialogFlag: false, //控制修改对话框显示
        book: { //封装数据
          bid: 0,
          bname: '',
          author: '',
          price: '',
          quantity: '',
          ddesc: '',
          cateid: 0
        },
        addDialogFlag: false, //控制对话框是否显示
        cateName: '', //条件查询
        tableData: [],
        pager: { //分页封装对象
          page: 1,
          size: 5,
          total: 0
        }
      }
    },
    //该属性是vue对象中的成员变量，作用编写用户自己定的方法
    methods: {
      openAddDialog() {
        this.addDialogFlag = !this.addDialogFlag; //将打开添加对话框标记改成true
      },
      openEditDialog(row) {
        this.category = row;
        console.info(this.category)
        this.editDialogFlag = !this.editDialogFlag;
      },
      update() {
        this.$refs['category'].validate(valid => {
          if (valid) {
            this.$http.post('http://127.0.0.1/category/update', this.category)
              .then(res => {
                if (res.data.code == 200) {
                  this.$message({
                    message: res.data.message,
                    type: 'success'
                  });
                  this.listPage();
                  this.editDialogFlag = !this.editDialogFlag;
                }
              }).catch(error => {
                this.$message.error('错了哦，网络异常');
              })
          }
        })

      },
      save() {
        // let parmas = new URLSearchParams();
        // parmas.append("catename", this.category.catename)
        this.$refs['category'].validate(valid => {
          if (valid) {
            this.$http.post('http://127.0.0.1/category/add', this.category)
              .then(res => {
                if (res.data.code == 200) {
                  this.$message({
                    message: res.data.message,
                    type: 'success'
                  });
                  this.listPage();
                  this.addDialogFlag = !this.addDialogFlag;
                }
              }).catch(error => {
                this.$message.error('错了哦，网络异常');
              })
          }

        })

      },
      remove(row) {
        this.$http.post('http://127.0.0.1/category/remove?cateid=' + row.cateid)
          .then(res => {
            if (res.data.code == 200) {
              this.$message({
                message: res.data.message,
                type: 'success'
              });
              this.listPage();
            }
          }).catch(error => {
            this.$message.error('错了哦，网络异常');
          })
      },

      handleCurrentChange(val) {
        this.pager.page = val;
        this.listPage();
      },
      handleSizeChange(val) {
        this.pager.size = val;
        this.listPage(); //重新调用查询方法

      },
      //用户定义的前端请求方法
      listPage() {
        this.$http.get('http://127.0.0.1/book/listPage', {
          params: {
            page: this.pager.page,
            size: this.pager.size,
            bname: this.book.bname,
            author: this.book.author
          }
        }).then(res => {

          this.tableData = res.data.data.data;
          this.pager.total = res.data.data.total
        }).catch(error => {
          this.$message.error('错了哦，网络异常');
        })
      }

    },
    //vue声明周期函数
    mounted() {
      this.listPage();
    }
  }
</script>

<style>
  .el-breadcrumb {
    padding-bottom: 20px;
  }

  .el-pagination {
    padding-top: 10px;
  }
</style>
```

10.2.图书类别模块UI设计与实现



类别名称	创建时间	修改时间
考研类	2022-06-29 15:34:38	2022-06-29 17:11:31
少儿类	2022-06-29 15:38:26	2022-06-29 17:12:19
古典文学	2022-06-29 15:40:35	2022-06-29 17:12:23
8888	2022-06-29 17:36:31	2022-06-29 17:45:34
88888	2022-06-29 17:43:43	2022-06-29 17:45:44

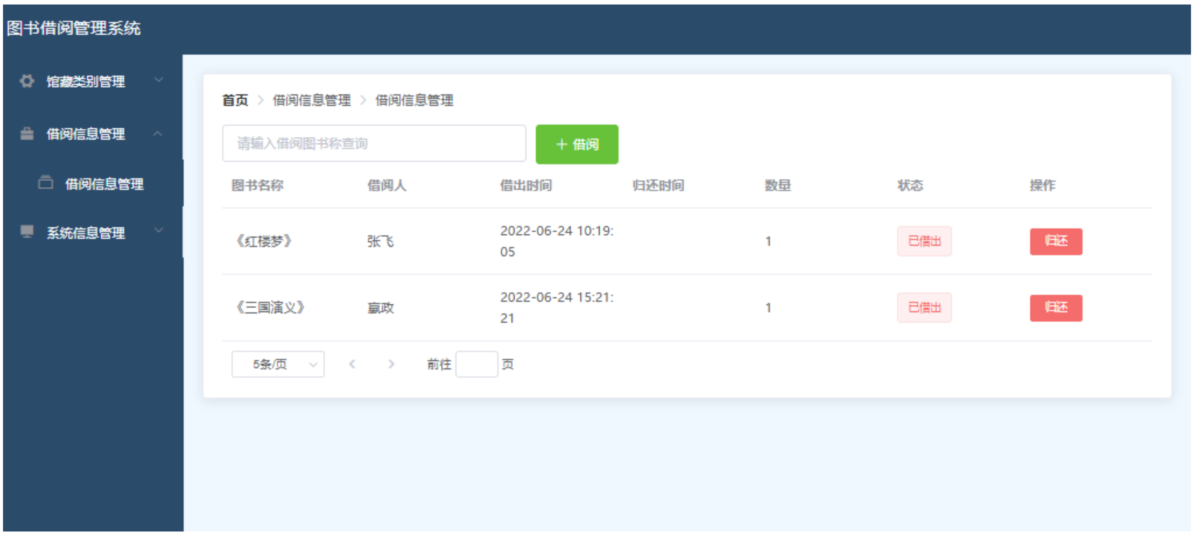
```
<template>
  <!-- 卡片组件 -->
  <el-card>
    <!-- 面包屑组件 -->
    <el-breadcrumb separator-class="el-icon-arrow-right">
      <el-breadcrumb-item :to="{ path: '/welcome' }">首页</el-breadcrumb-item>
      <el-breadcrumb-item>馆藏类别管理</el-breadcrumb-item>
      <el-breadcrumb-item>类别信息管理</el-breadcrumb-item>
    </el-breadcrumb>
    <el-row :gutter="10">
      <el-col :span="8">
        <el-input clearable @change="listPage()" v-model="cateName"
placeholder="请输入图书类别名称查询"></el-input>
      </el-col>
      <el-col :span="12">
        <el-button icon="el-icon-plus" @click="openAddDialog" type="success">添加
</el-button>
      </el-col>
    </el-row>
    <!-- 表格组件 -->
    <el-table :data="tableData" style="width: 100%">
      <el-table-column prop="catename" label="类别名称" width="180">
      </el-table-column>
      <el-table-column prop="createtime" label="创建时间">
      </el-table-column>
      <el-table-column prop="modifytime" label="修改时间">
      </el-table-column>
      <el-table-column lable="操作">
        <!-- 作用域插槽 -->
        <template slot-scope="scope">

          <el-tooltip content="点击修改" placement="bottom">
            <el-button type="warning" @click="openEditDialog(scope.row)"
icon="el-icon-edit" circle></el-button>
          </el-tooltip>
          <el-tooltip content="点击删除" placement="bottom">
```

```
            <el-button type="danger" @click="remove(scope.row)" icon="el-icon-delete" circle></el-button>
          </el-tooltip>
        </template>
      </el-table-column>
    </el-table>
    <!-- 分页组件定义 -->
    <el-pagination @size-change="handleSizeChange" @current-change="handleCurrentChange" :current-page="pager.page"
      :page-sizes="[5, 10, 15, 20]" :page-size="pager.size" layout="total, sizes, prev, pager, next, jumper"
      :total="pager.total">
    </el-pagination>

    <!-- 添加对话框 -->
    <el-dialog title="图书类别添加" :show-close="false" :close-on-click-modal="false" :visible.sync="addDialogFlag">
      <el-form :model="category" :rules="rules" ref="category">
        <el-form-item label="类别名称" prop="catename">
          <el-input v-model="category.catename" placeholder="请输入类别名称"></el-input>
        </el-form-item>
      </el-form>
      <div slot="footer" class="dialog-footer">
        <el-button @click="addDialogFlag=!addDialogFlag">取 消</el-button>
        <el-button type="primary" @click="save">保存</el-button>
      </div>
    </el-dialog>

    <!-- 修改对话框 -->
    <el-dialog title="图书类别修改" :show-close="false" :close-on-click-modal="false" :visible.sync="editDialogFlag">
      <el-form :model="category" :rules="rules" ref="category">
        <el-form-item label="类别名称" prop="catename">
          <el-input v-model="category.catename" placeholder="请输入类别名称"></el-input>
        </el-form-item>
      </el-form>
      <div slot="footer" class="dialog-footer">
        <el-button @click="editDialogFlag=!editDialogFlag">取 消</el-button>
        <el-button type="primary" @click="update">保存</el-button>
      </div>
    </el-dialog>

  </el-card>




</template>

<script>
  export default {
    data() {
      return {
        rules: {
          catename: [{
            required: true,
            message: '请输入类别名称',
            trigger: 'blur'
          }]
        }, //定义表单验证规则
        editDialogFlag: false, //控制修改对话框显示
        category: { //封装数据
          cateid: 0,
          catename: ''
        },
        addDialogFlag: false, //控制对话框是否显示
        cateName: '', //条件查询
        tableData: [],
        pager: { //分页封装对象
          page: 1,
          size: 5,
          total: 0
        }
      }
    },
    //该属性是vue对象中的成员变量，作用编写用户自己定的方法
    methods: {
      openAddDialog() {
        this.addDialogFlag = !this.addDialogFlag; //将打开添加对话框标记改成true
      },
      openEditDialog(row) {
        this.category = row;
        console.info(this.category)
        this.editDialogFlag = !this.editDialogFlag;
      },
      update() {
        this.$refs['category'].validate(valid => {
          if (valid) {
            this.$http.post('http://127.0.0.1/category/update', this.category)
              .then(res => {
                if (res.data.code == 200) {
                  this.$message({
                    message: res.data.message,
                    type: 'success'
                  });
                  this.listPage();
                  this.editDialogFlag = !this.editDialogFlag;
                }
              }).catch(error => {
                this.$message.error('错了哦，网络异常');
              })
          }
        })

      },
      save() {
        // let parmas = new URLSearchParams();
        // parmas.append("catename", this.category.catename)
        this.$refs['category'].validate(valid => {
```

```
        if (valid) {
          this.$http.post('http://127.0.0.1/category/add', this.category)
            .then(res => {
              if (res.data.code == 200) {
                this.$message({
                  message: res.data.message,
                  type: 'success'
                });
                this.listPage();
                this.addDialogFlag = !this.addDialogFlag;
              }
            }).catch(error => {
              this.$message.error('错了哦，网络异常');
            })
        }

      })

    },
    remove(row) {
      this.$http.post('http://127.0.0.1/category/remove?cateid=' + row.cateid)
        .then(res => {
          if (res.data.code == 200) {
            this.$message({
              message: res.data.message,
              type: 'success'
            });
            this.listPage();
          }

        }).catch(error => {
          this.$message.error('错了哦，网络异常');
        })
    },

    handleCurrentChange(val) {
      this.pager.page = val;
      this.listPage();
    },
    handleSizeChange(val) {
      this.pager.size = val;
      this.listPage(); //重新调用查询方法

    },
    //用户定义的前端请求方法
    listPage() {
      this.$http.get('http://127.0.0.1/category/listPage', {
        params: {
          page: this.pager.page,
          size: this.pager.size,
          cateName: this.cateName
        }
      }).then(res => {

        this.tableData = res.data.data.data;
```

```
          this.pager.total = res.data.data.count
        }).catch(error => {
          this.$message.error('错了哦，网络异常');
        })
      }

    },
    //vue声明周期函数
    mounted() {
      this.listPage();
    }
  }
</script>

<style>
  .el-breadcrumb {
    padding-bottom: 20px;
  }

  .el-pagination {
    padding-top: 10px;
  }
</style>
```

10.3.借阅模块UI设计与实现



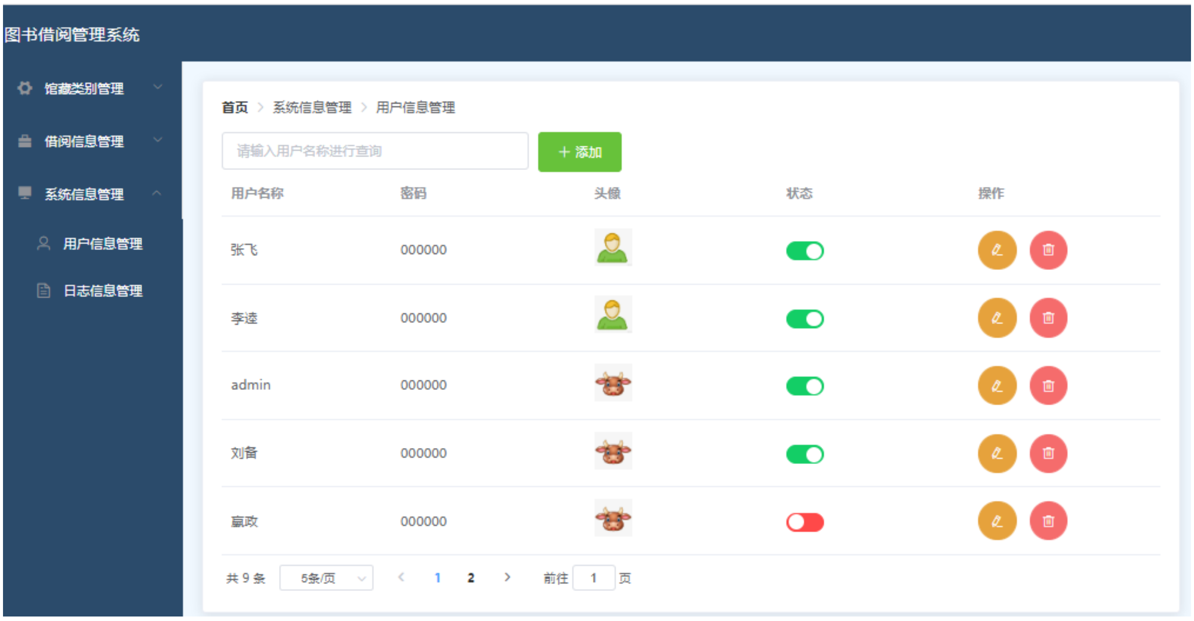
图书名称	借阅人	借出时间	归还时间	数量	状态	操作
《红楼梦》	张飞	2022-06-24 10:19:05		1	已借出	归还
《三国演义》	嬴政	2022-06-24 15:21:21		1	已借出	归还

```
<template>
  <!-- 卡片组件 -->
  <el-card>
    <!-- 面包屑组件 -->
    <el-breadcrumb separator-class="el-icon-arrow-right">
      <el-breadcrumb-item :to="{ path: '/welcome' }">首页</el-breadcrumb-item>
      <el-breadcrumb-item>借阅信息管理</el-breadcrumb-item>
      <el-breadcrumb-item>借阅信息管理</el-breadcrumb-item>
    </el-breadcrumb>
    <el-row :gutter="10">
      <el-col :span="8">
        <el-input clearable @change="listPage()" v-model="borrow.book.bname"
placeholder="请输入借阅图书称查询"></el-input>
```

```
      </el-col>
      <el-col :span="12">
        <el-button icon="el-icon-plus" @click="openAddDialog" type="success">借阅
</el-button>
      </el-col>
    </el-row>
    <!-- 表格组件 -->
    <el-table :data="tableData">
      <el-table-column prop="book" label="图书名称">
        <template slot-scope="scope">
          {{scope.row.book.bname}}
        </template>
      </el-table-column>
      <el-table-column prop="users" label="借阅人">
        <template slot-scope="scope">
          {{scope.row.users.uname}}
        </template>
      </el-table-column>
      <el-table-column prop="otime" label="借出时间">
      </el-table-column>
      <el-table-column prop="itime" label="归还时间">
      </el-table-column>
      <el-table-column prop="quantity" label="数量">
      </el-table-column>
      <el-table-column prop="quantity" label="状态">
        <template slot-scope="scope">
          <el-tag type="success" v-show="scope.row.stat==1">已归还</el-tag>
          <el-tag type="danger" v-show="scope.row.stat==0">已借出</el-tag>
        </template>
      </el-table-column>
      <el-table-column label="操作">
        <!-- 作用域插槽 -->
        <template slot-scope="scope">
          <el-button size="mini" type="danger" v-show="scope.row.stat==0">归还
</el-button>
        </template>
      </el-table-column>
    </el-table>
    <!-- 分页组件定义 -->
    <el-pagination @size-change="handleSizeChange" @current-
change="handleCurrentChange" :current-page="pager.page"
      :page-sizes="[5, 10, 15, 20]" :page-size="pager.size" layout="total,
sizes, prev, pager, next, jumper"
      :total="pager.total">
    </el-pagination>
  </el-card>



</template>

<script>
  export default {
    data() {
      return {
```

```
      rules: {
        catename: [{
          required: true,
          message: '请输入类别名称',
          trigger: 'blur'
        }]
      }, //定义表单验证规则
      editDialogFlag: false, //控制修改对话框显示
      category: { //封装数据
        cateid: 0,
        catename: ''
      },
      borrow: {
        boid: 0,
        otime: '',
        itime: '',
        quantity: 0,
        book: {
          bid: 0,
          bname: ''
        },
        users: {
          uid: 0,
          uname: ''
        }
      },
      addDialogFlag: false, //控制对话框是否显示
      cateName: '', //条件查询
      tableData: [],
      pager: { //分页封装对象
        page: 1,
        size: 5,
        total: 0
      }
    }
  },
  //该属性是vue对象中的成员变量，作用编写用户自己定的方法
  methods: {
    openAddDialog() {
      this.addDialogFlag = !this.addDialogFlag; //将打开添加对话框标记改成true
    },
    openEditDialog(row) {
      this.category = row;
      console.info(this.category)
      this.editDialogFlag = !this.editDialogFlag;
    },

    handleCurrentChange(val) {
      this.pager.page = val;
      this.listPage();
    },
    handleSizeChange(val) {
      this.pager.size = val;
      this.listPage(); //重新调用查询方法
      },
      //用户定义的前端请求方法
      listPage() {
        this.$http.get('http://127.0.0.1/borrow/listPage', {
          params: {
            page: this.pager.page,
            size: this.pager.size,
            bname: this.borrow.book.bname
          }
        }).then(res => {

          this.tableData = res.data.data.data;
          this.pager.total = res.data.data.count
        }).catch(error => {
          this.$message.error('错了哦，网络异常');
        })
      }

    },
    //vue声明周期函数
    mounted() {
      this.listPage();
    }
  }
</script>

<style>
  .el-breadcrumb {
    padding-bottom: 20px;
  }

  .el-pagination {
    padding-top: 10px;
  }
</style>
```

10.4.用户模块UI设计与实现



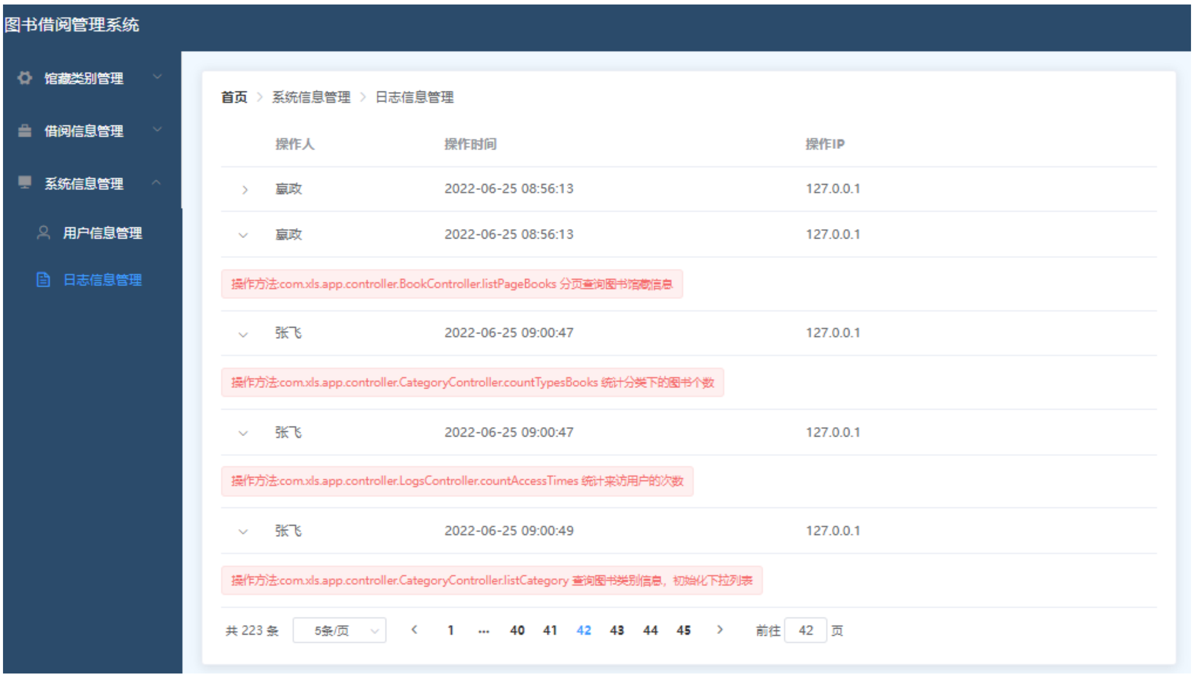
用户名称	密码	头像	状态	操作
张飞	000000		●	
李逵	000000		●	
admin	000000		●	
刘备	000000		●	
嬴政	000000		○	

共 9 条 5条/页 < 1 2 > 前往 1 页

```
<template>
  <!-- 卡片组件 -->
  <el-card>
    <!-- 面包屑组件 -->
    <el-breadcrumb separator-class="el-icon-arrow-right">
      <el-breadcrumb-item :to="{ path: '/welcome' }">首页</el-breadcrumb-item>
      <el-breadcrumb-item>系统信息管理</el-breadcrumb-item>
      <el-breadcrumb-item>用户信息管理</el-breadcrumb-item>
    </el-breadcrumb>
    <el-row :gutter="10">
      <el-col :span="8">
        <el-input clearable @change="listPage()" v-model="uname"
         placeholder="请输入用户名称进行查询"></el-input>
      </el-col>
      <el-col :span="12">
        <el-button icon="el-icon-plus" @click="openAddDialog" type="success">添加</el-button>
      </el-col>
    </el-row>
    <!-- 表格组件 -->
    <el-table :data="tableData" style="width: 100%">
      <el-table-column prop="uname" label="用户名称" width="180">
      </el-table-column>
      <el-table-column prop="password" label="密码">
      </el-table-column>
      <el-table-column prop="images" label="头像">
        <template slot-scope="scope">
          <img  :src="scope.row.images" style="width: 40px;height: 40x;"/>
        </template>
      </el-table-column>
      <el-table-column prop="stats" label="状态">
        <template slot-scope="scope">
          <el-switch
            v-model="scope.row.stats"
            :active-value="0"
            :inactive-value="1"
            active-color="#13ce66"
            inactive-color="#ff4949"  @change="changStats(scope.row)">
          </el-switch>
        </template>
      </el-table-column>
      <el-table-column label="操作">
        <!-- 作用域插槽 -->
        <template slot-scope="scope">

          <el-tooltip content="点击修改" placement="bottom">
            <el-button type="warning" @click="openEditDialog(scope.row)" icon="el-icon-edit" circle></el-button>
          </el-tooltip>
          <el-tooltip content="点击删除" placement="bottom">
            <el-button type="danger" @click="remove(scope.row)" icon="el-icon-delete" circle></el-button>
          </el-tooltip>
        </template>
      </el-table-column>
    </el-table>
    <!-- 分页组件定义 -->
    <el-pagination @size-change="handleSizeChange" @current-
change="handleCurrentChange" :current-page="pager.page"
      :page-sizes="[5, 10, 15, 20]" :page-size="pager.size" layout="total,
sizes, prev, pager, next, jumper"
      :total="pager.total">
    </el-pagination>

    <!-- 添加对话框 -->
    <el-dialog title="用户添加" :show-close="false" :close-on-click-modal="false"
:visible.sync="addDialogFlag">
      <el-form :model="users" :rules="rules" ref="users">
        <el-form-item label="用户名称" prop="catename">
          <el-input v-model="users.uname" placeholder="请输入用户名称"></el-input>
        </el-form-item>
        <el-form-item label="用户密码" prop="catename">
          <el-input v-model="users.password" placeholder="请输入用户密码"></el-
input>
        </el-form-item>
        <el-form-item label="用户头像" prop="catename">
          <el-input v-model="users.images" placeholder="请输入用户头像"></el-input>
        </el-form-item>
      </el-form>
      <div slot="footer" class="dialog-footer">
        <el-button @click="addDialogFlag=!addDialogFlag">取 消</el-button>
        <el-button type="primary" @click="save">保存</el-button>
      </div>
    </el-dialog>

    <!-- 修改对话框 -->
    <el-dialog title="用户修改" :show-close="false" :close-on-click-modal="false"
:visible.sync="editDialogFlag">
      <el-form :model="users" :rules="rules" ref="users">
       <el-form-item label="用户名称" prop="catename">
         <el-input v-model="users.uname" placeholder="请输入用户名称"></el-input>
       </el-form-item>
       <el-form-item label="用户密码" prop="catename">
         <el-input v-model="users.password" placeholder="请输入用户密码"></el-
input>
       </el-form-item>
       <el-form-item label="用户头像" prop="catename">
         <el-input v-model="users.images" placeholder="请输入用户头像"></el-input>
       </el-form-item>
      </el-form>
      <div slot="footer" class="dialog-footer">
        <el-button @click="editDialogFlag=!editDialogFlag">取 消</el-button>
        <el-button type="primary" @click="update">保存</el-button>
      </div>
    </el-dialog>

  </el-card>
```

```
</template>

<script>
  export default {
    data() {
      return {
        uname:'',
        rules: {
          uname: [{
            required: true,
            message: '请输入用户名称',
            trigger: 'blur'
          }],
          password: [{
            required: true,
            message: '请输入密码',
            trigger: 'blur'
          }],
          images: [{
            required: true,
            message: '请输入头像连接地址',
            trigger: 'blur'
          }]
        }, //定义表单验证规则
        editDialogFlag: false, //控制修改对话框显示
        users: { //封装数据
          uid: 0,
          uname: '',
          images:'',
          password:''
        },
        addDialogFlag: false, //控制对话框是否显示
        tableData: [],
        pager: { //分页封装对象
          page: 1,
          size: 5,
          total: 0
        }
      }
    },
    //该属性是vue对象中的成员变量，作用编写用户自己定的方法
    methods: {
      changStats(row){ //改变用户状态
        this.users=row;

 this.$http.post('http://127.0.0.1/users/changStats',this.users).then(res=>{
          if (res.data.code == 200) {
            this.$message({
              message: res.data.message,
              type: 'success'
            });
            this.listPage();
          }

        })
      },
      openAddDialog() {
        this.addDialogFlag = !this.addDialogFlag; //将打开添加对话框标记改成true
      },
      openEditDialog(row) {
        this.users = row;
        this.editDialogFlag = !this.editDialogFlag;
      },
      update() {
        this.$refs['users'].validate(valid => {
          if (valid) {
            this.$http.post('http://127.0.0.1/users/update', this.users)
              .then(res => {
                if (res.data.code == 200) {
                  this.$message({
                    message: res.data.message,
                    type: 'success'
                  });
                  this.listPage();
                  this.editDialogFlag = !this.editDialogFlag;
                }
              }).catch(error => {
                this.$message.error('错了哦，网络异常');
              })
          }
        })

      },
      save() {
        this.$refs['users'].validate(valid => {
          if (valid) {
            this.$http.post('http://127.0.0.1/users/save', this.users)
              .then(res => {
                if (res.data.code == 200) {
                  this.$message({
                    message: res.data.message,
                    type: 'success'
                  });
                  this.listPage();
                  this.addDialogFlag = !this.addDialogFlag;
                }
              }).catch(error => {
                this.$message.error('错了哦，网络异常');
              })
          }

        })

      },
      remove(row) {
        this.$http.post('http://127.0.0.1/users/remove?uid=' + row.uid)
          .then(res => {
            if (res.data.code == 200) {
              this.$message({
                message: res.data.message,
        type: 'success'
       });
       this.listPage();
      }

     }).catch(error => {
      this.$message.error('错了哦，网络异常');
     })
   },

   handleCurrentChange(val) {
    this.pager.page = val;
    this.listPage();
   },
   handleSizeChange(val) {
    this.pager.size = val;
    this.listPage(); //重新调用查询方法

   },
   //用户定义的前端请求方法
   listPage() {
    this.$http.get('http://127.0.0.1/users/listPage', {
     params: {
      page: this.pager.page,
      size: this.pager.size,
      uname:this.uname
     }
    }).then(res => {

     this.tableData = res.data.data.data;
     this.pager.total = res.data.data.total
    }).catch(error => {
     this.$message.error('错了哦，网络异常');
    })
   }

  },
  //vue声明周期函数
  mounted() {
   this.listPage();
  }
 }
</script>

<style>
 .el-breadcrumb {
  padding-bottom: 20px;
 }

 .el-pagination {
  padding-top: 10px;
 }
</style>
```

10.5.日志模块UI设计与实现



图书借阅管理系统

首页 > 系统信息管理 > 日志信息管理

操作人	操作时间	操作IP
嬴政	2022-06-25 08:56:13	127.0.0.1
嬴政	2022-06-25 08:56:13	127.0.0.1
操作方法:com.xls.app.controller.BookController.listPageBooks 分页查询图书馆藏信息		
张飞	2022-06-25 09:00:47	127.0.0.1
操作方法:com.xls.app.controller.CategoryController.countTypesBooks 统计分类下的图书个数		
张飞	2022-06-25 09:00:47	127.0.0.1
操作方法:com.xls.app.controller.LogsController.countAccessTimes 统计来访用户的次数		
张飞	2022-06-25 09:00:49	127.0.0.1
操作方法:com.xls.app.controller.CategoryController.listCategory 查询图书类别信息，初始化下拉列表		

共 223 条 5条/页 < 1 ... 40 41 42 43 44 45 > 前往 42 页

```
<template>
  <!-- 卡片组件 -->
  <el-card>
    <!-- 面包屑组件 -->
    <el-breadcrumb separator-class="el-icon-arrow-right">
      <el-breadcrumb-item :to="{ path: '/welcome' }">首页</el-breadcrumb-item>
      <el-breadcrumb-item>系统信息管理</el-breadcrumb-item>
      <el-breadcrumb-item>日志信息管理</el-breadcrumb-item>
    </el-breadcrumb>
    <!-- 表格组件 -->
    <el-table :data="tableData" >
      <el-table-column type="expand">
        <template slot-scope="scope">
         <el-tag type="danger">操作方法:{{scope.row.methods}} {{scope.row.ddesc}}</el-tag>
        </template>
      </el-table-column>
      <el-table-column prop="opername" label="操作人" width="180">
      </el-table-column>
      <el-table-column prop="opertime" label="操作时间">
      </el-table-column>
      <el-table-column prop="ip" label="操作IP">
      </el-table-column>

    </el-table>
    <!-- 分页组件定义 -->
    <el-pagination @size-change="handleSizeChange" @current-change="handleCurrentChange" :current-page="pager.page"
      :page-sizes="[5, 10, 15, 20]" :page-size="pager.size" layout="total, sizes, prev, pager, next, jumper"
      :total="pager.total">
    </el-pagination>
  </el-card>

</template>
```

```
<script>
  export default {
    data() {
      return {
        rules: {
          catename: [{
            required: true,
            message: '请输入类别名称',
            trigger: 'blur'
          }]
        }, //定义表单验证规则
        editDialogFlag: false, //控制修改对话框显示
        category: { //封装数据
          cateid: 0,
          catename: ''
        },
        addDialogFlag: false, //控制对话框是否显示
        cateName: '', //条件查询
        tableData: [],
        pager: { //分页封装对象
          page: 1,
          size: 5,
          total: 0
        }
      }
    },
    //该属性是vue对象中的成员变量，作用编写用户自己定的方法
    methods: {

      handleCurrentChange(val) {
        this.pager.page = val;
        this.listPage();
      },
      handleSizeChange(val) {
        this.pager.size = val;
        this.listPage(); //重新调用查询方法

      },
      //用户定义的前端请求方法
      listPage() {
        this.$http.get('http://127.0.0.1/logs/listPage', {
          params: {
            page: this.pager.page,
            size: this.pager.size,
          }
        }).then(res => {

          this.tableData = res.data.data.data;
          this.pager.total = res.data.data.count
        }).catch(error => {
          this.$message.error('错了哦，网络异常');
        })
      }
    },
    //vue声明周期函数
    mounted() {
      this.listPage();
    }
  }
</script>

<style>
  .el-breadcrumb {
    padding-bottom: 20px;
  }

  .el-pagination {
    padding-top: 10px;
  }
</style>
```

第十一章、自选项目

10.1计算机类知识问答系统 设计与实现

```
系统架构：Mysql+MybatisPlus/Mybatis+Shiro+Springboot+Swagger+Element
要求：
1、分组讨论独自完成需求分析
2、根据需求独自设计数据
3、根据数据库独立搭建架构
4、根据架构自行完成后端编码
5、独自设计设计完成前端编码工作
6、主要完成热点问题讨论分析功能
周期：7-10天
```

10.2.学生就业信息分析平台设计与实现

```
学生就业信息分析平台
系统架构：Mysql+MybatisPlus/Mybatis+Shiro+Springboot+Swagger+Element
要求：1、分组讨论独自完成需求分析
2、根据需求独自设计数据
3、根据数据库独立搭建架构
4、根据架构自行完成后端编码
5、独自设计设计完成前端编码工作
6、主要实现就业方向、行业、专业数据分析核心业务
周期：10-20天
```

10.3.学生就业信息分析平台设计与实现

```
系统架构：Mysql+MybatisPlus/Mybatis+Shiro+Springboot+Swagger+Element
要求：1、分组讨论独自完成需求分析
2、根据需求独自设计数据
3、根据数据库独立搭建架构
4、根据架构自行完成后端编码
5、独自设计设计完成前端编码工作
6、实现核心业务随机生成试题（单选）、能考试，能自动阅卷
周期：10-20天
```

10.4.儿童成长智慧树平台设计与实现

```
系统架构：Mysql+MybatisPlus/Mybatis+Shiro+Springboot+Swagger+Element
要求：1、分组讨论独自完成需求分析
2、根据需求独自设计数据
3、根据数据库独立搭建架构
4、根据架构自行完成后端编码
5、独自设计设计完成前端编码工作
6、实现核心业务随机生成试题（单选）、能考试，能自动阅卷
周期：10-20天
```