

山东农业大学

毕 业 论 文

题目：____博客中台系统____

院 部____信息科学与工程学院____

专业班级____计算机科学与技术 2014-4____

届 次____2018 届____

学生姓名____谢庆贺____

学 号____20145021____

指导教师____吴秋兰（副教授）____

二〇一八年 五月 二十日

博客中台系统

Blog Transfer Platform System

专业 Speciality	计算机科学与技术 Computer Science and Technology
学生 Undergraduate	谢庆贺 Xie Qinghe
指导教师 Supervisor	吴秋兰 Wu Qiulan

目 录

1 引言.....	1
1.1 研究的目的和意义.....	1
1.2 国内外研究现状.....	1
2 需求分析.....	3
2.1 可行性分析.....	3
2.2 业务分析.....	3
2.3 数据库分析.....	4
2.4 性能分析.....	5
2.5 安全性分析.....	5
3 系统设计.....	6
3.1 系统总体规划.....	6
3.2 概要设计.....	6
3.2.1 系统总体功能模块图.....	6
3.2.2 系统的业务流程.....	6
3.3 数据库设计.....	6
4 系统实现.....	11
4.1 研发工具.....	11
4.2 关键技术.....	12
4.3 功能模块实现.....	12
4.3.1 系统核心对接 API 实现-开源中国.....	13
4.3.2 系统核心对接 API 实现-CSDN.....	15
4.3.3 系统核心对接 API 代码实现.....	18
4.3.4 登录页面实现.....	18
4.3.5 文章列表.....	18
4.3.6 用户博客绑定显示页面.....	20
4.3.7 栏目管理.....	21
4.3.8 数据统计.....	22
4.3.9 日志记录.....	23
4.4 系统难点与解决方案.....	24
5 系统测试.....	25
5.1 系统测试的目的与重要性.....	25
5.2 系统测试的目的与重要性.....	25
5.3 具体测试用例及结果.....	28

6 总结.....	29
参考文献.....	30
致谢.....	31

Contents

1 Introduction.....	1
1.1 Purpose and Significance of the Study.....	1
1.2 Research Status at Home and Abroad.....	1
2 Needs Analysis.....	3
2.1 Feasibility Analysis.....	3
2.2 Service Analysis.....	3
2.3 Database Analysis.....	4
2.4 Performance Analysis.....	5
2.5 Security Analysis.....	5
3 System Design.....	6
3.1 Overall System Planning.....	6
3.2 Summary Design.....	6
3.2.1 Overall System Function Block Diagram.....	6
3.2.2 Business Processes of the System.....	6
3.3 Database Design.....	6
4 System Implementation.....	11
4.1 Develop Tools.....	11
4.2 Key Technologies.....	12
4.3 Functional Module Implementation.....	12
4.3.1 System Core Docking API Implementation - Open Source China.....	13
4.3.2 System core docking API implementation -CSDN.....	15
4.3.3 System Core Docking API Code Implementation.....	18
4.3.4 Login Page Implementation.....	18
4.3.5 Article List.....	18
4.3.6 User Blog Binding Display Page.....	20
4.3.7 Section Management.....	21
4.3.8 Data Statistics.....	22
4.3.9 Login Page Implementation.....	23
4.4 System Difficulties and Solutions.....	24
5 System Test.....	25
5.1 Purpose and Importance of System Testing.....	25
5.2 Purpose and Importance of System Testing.....	25
5.3 Specific Test Cases and Results.....	28

6 Summary.....	29
References.....	30
Thanks.....	31

博客中台系统

【摘要】 博客中台系统，就是把目前用户主流博客系统的文章进行整理拉取到中台，提高用户查询与整理效率。根据需求对博客中台系统进行了整体设计，主要包括四个功能模块：第三方博客数据的提取与处理模块、博客管理模块、用户管理模块、图表数据统计模块。根据数据结构特点，设计了系统数据库。为了提高系统扩展性和代码复用，基于 B/S 的 MVC 三层架构体系，以 java、coffeescript 为开发语言，以关系型数据库 mysql 和内存数据库 redis 为数据库，研发了博客中台系统。通过实际应用证明实现博客中台系统的方案是可行的。

【关键词】 博客；中台；oauth2；授权；数据整合

Blog Transfer Platform System

【Abstract】 The platform system of the blog is to organize the articles of the current user mainstream blog system to the middle stage and improve the user's query and collation efficiency. According to the requirements on the blog China system has carried on the overall design, mainly includes four function modules: third-party blog data extraction and processing module, the blog management module, user management module, the chart data statistics module. According to the data structure characteristics, the system database is designed. In order to improve system extensibility and reusability of the code, the MVC based on B/S three-tier architecture system, for the development of language, such as Java, coffeescript to relational database mysql and redis memory database as the database, China developed the blog system. It is feasible to realize the scheme of the platform system through practical application.

【Key words】 Blogs; Transfer platform; Oauth2; Authorization; Data integration

1 引言

1.1 研究的目的是和意义

当今数据爆炸的时代，随着信息互联网的快速发展，数据的整合、分析和垃圾数据的过滤显得尤为重要，大数据分析显得尤其热门，但是数据的分析不是为了分析而去分析，是为了整合资源和提取共有用的商业价值信息，从而创造更好的服务和改变。这才是大数据时代的目的。我国近几年互联网发展非常迅猛，在信息如此多的时代，想要记住所有接触到的知识是不可能也是不现实的，那么就需要借助工具或者辅助平台去帮助人去解决很多问题，让人从劳动中解脱出来，这样也会使人的生产效率变得更高^[1]。在信息技术飞速发展的当今社会，信息技术成为了带动整个社会继续发展的重要一环。能否引进、结合信息技术来实现新的突破、开辟新的道路，成为当前各行各业需要面对和解决的重要问题。现在传统行业也受到很大的影响，研究这个项目的目的不仅仅只是解决这一问题，而是同一类问题的解决思路 and 方案，同时还有技术的支持。多博客系统分析平台，可以使用户整理得到不同平台自己的数据痕迹，大大增加了查阅效率，使用户更加高效的进行生产活动。所以博客中转平台就是方便用户进行自己博客的统计与备忘录，用户不可能都记得自己的所有博客网站写过的文章，博客中转平台的目的就是为用户博客提供中转与备忘录^{[2][3]}。

1.2 国内外研究现状

目前随着平台的多种多样和复杂化，人们有时候不得不去使用多种不同相同功能的平台，而这些平台因为各种原因各有各的好处。比如电商平台京东物流速度快，服务质量好，大型家电在京东购买是首选，但是京东商品种类不如淘宝，对于需求时间不是特别紧迫和一些冷门商品，只能选择去淘宝购买。对于 IT 工作者来说，都要有一颗开源和无私分享的心，所以一般开发者都会去各种论坛与博客社区分享心得体会与技术分享、或者是学习的记录随记，方便以后查看阅读，而互联网博客论坛各种各样，github、CSDN、博客园、开源中国、stackoverflow 等等一些优秀社区，一般都会遇见在不同社区写的文章无法进行统计到一个平台，而国内外还未出现过类似的博客中转平台^[2]，所以作为一款博客中转平台工具，一定会受到用户的喜爱和追捧。随着信息爆炸时代的来临，垃圾信息的筛选和数据的统计变得越来越重要，中台只是一个思想，适用于各种领域，比如零售电商中台、房源中转平台、售后服务中台等等，这些

系统目前在国内外还未形成完成的平台规范，而这恰恰是企业所需要的，抽象出来的这种平台一定会有很大的前景，必定受到企业热爱并且为企业创造价值^[4]。

目前企业级软件市场开始出现大量的整合系统和辅助系统，其实这些系统和中转分析平台类似，只是目前的类似系统虽然在顶级互联网公司存在规范的流程与实现过程，但是都是不对外公开的，包括阿里云都存在企业中台系统，但是只在内部使用，目前来说还未有对外实施出售的打算，所以企业级中台系统的市场是广阔的，并且具体的实施方案可以根据具体的业务逻辑来定制。

2 需求分析

2.1 可行性分析

对于系统的技术要求，首先会支持使用率和用户群体较多的博客系统，如 github、csdn 等社区，因为数据的拉取的可能是用户的隐私数据，需要用户授权才可以进行，由于社区不同，有些社区出于安全考虑可能存在一些不让外部系统进行爬去信息，这些都可以通过技术手段进行克服。

2.2 业务分析

博客中台系统业务分析：

文章管理（有些平台不支持文章增删改）、发表文章（有些平台不支持）、用户账号管理就是用户各个平台的账号密码管理、用户账号角色管理、通过用户角色判断用户是否有操作文章博客的权限、数据统计可以统计用户在各个平台的发表文章数量和活跃度与参与度、通过多种维度的表格进行展示、栏目管理。数据拉取平台，这是一个纯粹的后端博客数据拉取平台、有些平台支持接口拉取、有些平台必须爬取数据、如果是自己内部博客系统，比如开发多平台博客管理平台和博客系统是同一家公司或者机构，就可以通过直接去数据库拉取原始数据。系统分为九个功能。

（1）博客用户管理

博客拥有独立的账号进行账号登录、注册、找回密码、修改密码、修改个人详细资料功能。

（2）博客文章管理

游客和用户均可以查看系统所有文章，用户登录完成可以进行发表文章，修改文章，对文章格式进行设计，插入图片、超链接、文字样式修改、文章分类、简介、代码高亮显示、封面图片上传等功能。用户登录以后可以对文章进行评论、博主可以删除文章。文章还拥有模糊搜索功能，可以根据文章内容和文章标题的关键字进行搜索。文章有分类排行、文章阅读排行、文章热门标签统计。

（3）博客个性化定制

对于博客的首页导航栏显示数据，都是保存在数据库当中，可以随时进行更改，站长信息拥有单独的一页，可以随时根据不同需求进行定制。

（4）三方博客账号绑定

可以绑定三方的博客账号进行博客汇总统计，支持的绑定博客网站为 开源中国、CSDN、博客园、fh 博客（我自己开发的一套博客系统）、github。用户绑定账号以后可以选择禁用或者启用，可以修改绑定账号信息，可以解除绑定。

（5）文章汇总管理

用户绑定博客账号后，文章列表就可以自动拉取到用户绑定博客的文章信息，可以查看文章、可以对文章进行备份，对于自己平台的文章，可以进行编辑

（6）栏目管理

可以对不同博客系统拉取到的文章进行分类，可以新建栏目、修改栏目、删除栏目、查看栏目、搜索栏目

（7）对维度数据统计

可以统计用户在不同博客平台发表的文章动态数据图，支持数据视图、折线图、柱状图显示，还可以对图形进行下载保存，可以实时监控自己在所有平台发表的文章数据动态。

（8）数据拉取平台

因为不同博客系统总是千奇百怪、不能对接一个博客系统就去开发一套适配代码，所以才自己开发了一套博客系统，用于数据拉取的标准，后期如果平台做到很大，就可以让对接方遵循这套标准进行开发，数据拉取平台，可以定时的去博客系统数据库拉取用户博客。

（9）日志记录

因为使用本系统用户是为了简化、提高效率，所以力求操作界面简单、易于操作和理解，符合用户理解性和操作习惯，不能存在二义性，为防止用户误操作，必须添加日志记录，可以随时进行数据还原。

2.3 数据库分析

数据库设计的好坏直接影响到项目的性能好坏，是系统的关键，优秀的数据库设计必须遵循第三范式规范，减少数据冗余，避免数据异常维护，节约存储空间，高效的数据访问^[5]。

数据库表主要有以下几部分：

（1）文章表（主键，标题，文章内容，md 文档，文章简介，所属用户，文章标签，文章发表时间，文章修改时间，文章状态，文章额外信息，文章外链链接 url，文章来源）。

（2）文章-标签表（主键，文章 id，标签 id）。

（3）文章栏目表（主键，文章栏目名称，文章栏目说明，文章栏目创建

时间，(4) 文章栏目修改时间，文章栏目权重，文章栏目状态)。

(5) 评论表 (主键，文章 id，用户 id，评论父楼层，评论子楼层，评论时间，是否删除状态，评论内容，是否是经典评论)。

(6) 文章统计表 (主键，统计时间，统计数据量，统计所属用户)

(7) 用户角色表 (主键，角色唯一 key，角色数据范围，角色名称，角色说明，角色资源，创建时间，修改时间)。

(8) 用户表 (主键，用户名，用户昵称，用户性别，用户类型，密码，真实姓名，邮箱，手机号，用户状态，用户数据资源类型，创建时间，修改时间)。

(9) 角色-用户表 (主键，用户 id，角色 id，角色 key，创建时间，修改时间)。

(10) 用户三方博客授权表 (主键，用户 id，账号来源，账号来源所属唯一 id，昵称，用户文件 url，用户头像，用户授权 token，三方秘钥，刷新 token，有效时间戳，创建时间，修改时间，额外信息)。

(11) 标签表 (主键，标签名称，标签说明)。

2.4 性能分析

数据存储系统和模块运行系统需要在内网中进行操作，减少带宽所带来的不确定性故障和带宽开支，数据库采用主从数据库、采用 nosql 进行数据缓存，提高系统并发和系统稳定性。系统发版升级必须使用户毫无察觉，可以随时进行系统发版。可以保证系统 7*24 小时运行^[6]。系统要分为线上处理和离线处理两种方式，使用分布式微服务架构，使热门数据和冷门数据分离与缓存机制，提高系统并发的能力^{[7] [8]}。

2.5 安全性分析

系统需要保持稳定高效的运行，也要保证系统安全性，对于用户敏感操作，都要通过日志来记录，可以随时进行数据恢复，采用 mysql 主从数据库，可以保证数据永不丢失，用户密码相关信息采用 MD5 加盐进行不可逆加密^[9]，这就意味着脱库也无法得到用户密码，对数据传输进行 DES 对称加密，在无法知道私钥的情况下，无法进行数据的解密、对私钥进行定期修改，防止私钥因为各种原因泄露带来的风险^[10]。数据库进行热备份，这就意味着，只要机器不坏，永远不用停机。

3 系统设计

3.1 系统总体规划

博客中台功能分为：文章管理、用户管理、栏目管理、数据统计，文章发表，用户个人信息修改，接入第三方博客系统，博客模糊搜索，用户注册，登录。

3.2 概要设计

3.2.1 系统总体功能模块图

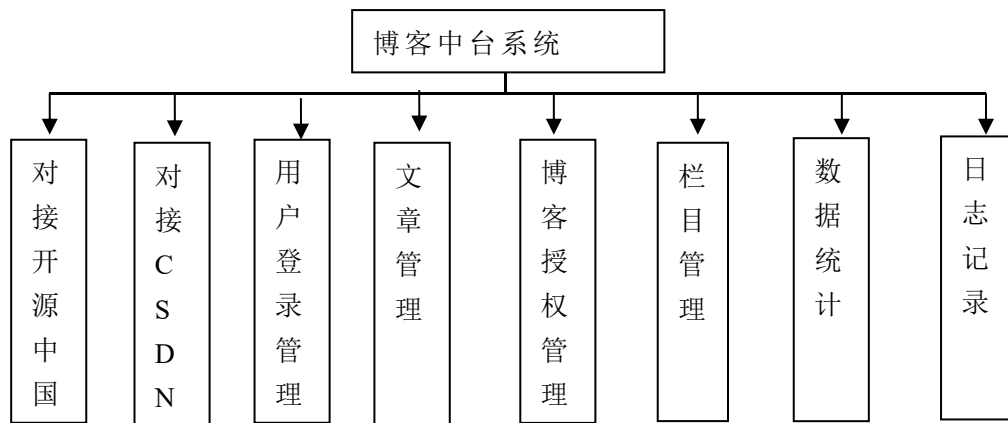


图 3.1 系统组织架构概览

3.2.2 系统的业务流程

用户通过注册账号，注册完成进行登录认证，用户登录成功，可以进行三方博客账号绑定，绑定三方账号成功，系统就可以自动拉取到三方博客信息，用户可以发表文章，修改文章，对文章进行存档，可以修改栏目，解除账号绑定，多维度实时显示用户博客发表动态数据统计图。

3.3 数据库设计

(1) 博客文章表

博客文章表主要用于文章存储，主键唯一标识文章，用户主键用于关联博客属于某个用户，文本内容使用 html 页面进行存储，前端展示直接使用插件展示 html 文本即可。其他字段是一些扩展字段，具体结构如表 3.1 所示。

表 3.1 bg_article 博客文章表

字段	数据类型	注释	类型
id	int(11)	主键	主键
user_id	int(11)	用户 id	
title	varchar(255)	博客标题	
htmlContent	text	html 源码	
summary	text	概览	
cid	int(11)	分类 id	MUL
uid	int(11)	用户 id	MUL
publishDate	datetime	发布时间	
editTime	datetime	修改时间	
state	int(6)	0 表示草稿箱，1 表示已发表，2 表示已删除，字段可扩展	
source	varchar(40)	来源：oschina、csdn、github、official、cnblogs	

(2) 文章标签文章关联表

文章标签文章关联表映射文章与标签的关联关系，具体结构如表 3.2 所示。

表 3.2 bg_article_tags 文章标签文章关联表

字段	数据类型	注释	类型
id	int(11)	主键	主键
aid	int(11)	文章 id	MUL
tid	int(11)	标签 id	MUL

(3) 文章分类表

文章分类表主要用于文章分类管理，具体结构如表 3.3 所示。

表 3.3 bg_category 文章分类

字段	数据类型	注释	类型
id	int(11)	主键	主键
cateName	varchar(64)	分类名称	
date	date	创建时间	

(4) 文章评论表

文章评论表用于存储文章的评论信息，具体结构如表 3.4 所示。

表 3.4 bg_comments 文章评论表

字段	数据类型	注释	类型
id	int(11)	主键	主键
aid	int(11)	文章 id	MUL
content	text	评论内容	
publishDate	datetime	发表时间	
parentId	int(11)	-1 表示正常回复，其他值表示是评论的回复	MUL
uid	int(11)	用户 id	MUL

(5) 数据统计表

数据统计表用于博客定时任务统计信息记录，具体结构如表 3.5 所示。

表 3.5 bg_pv 数据统计

字段	数据类型	注释	类型
id	int(11)	主键	主键
countDate	date	统计时间	
pv	int(11)	数量	
uid	int(11)	用户 id	MUL

(6) 角色表

角色表用于用户角色管理存储，具体结构如表 3.6 所示。

表 3.6 bg_roles 角色表

字段	数据类型	注释	类型
id	int(11)	主键	主键
name	varchar(32)	角色名称	

(7) 角色用户关联表

角色用户关联表存储角色与用户的对应关系，具体结构如表 3.7 所示。

表 3.7 bg_roles_user 角色用户关联表

字段	数据类型	注释	类型
id	int(11)	主键	主键

rid	int(11)	角色 id	MUL
uid	int(11)	用户 id	MUL

(8) 三方账号绑定表

三方账号绑定表用于记录用户绑定的三方博客信息，具体结构如表 3.8 所示。

表 3.8 bg_social_connections 三方账号绑定记录

字段	数据类型	注释	类型
id	bigint(20)	主键 id	主键
user_id	bigint(20)	用户 id	MUL
provider_id	varchar(255)	三方账号来源	
provider_user_id	varchar(255)	服务提供商返回的唯一标识 (openid)	MUL
display_name	varchar(200)	名称	
profile_url	varchar(1024)	文件路径	
image_url	varchar(1024)	头像 url	
access_token	varchar(1024)	token	
secret	varchar(1024)	密钥	
refresh_token	varchar(1024)	刷新 token	
expire_time	bigint(20)	access_token 的有效期	
created_at	datetime	创建时间	
updated_at	datetime	修改时间	

(9) 标签表

标签表主要用于记录博客的标签分类，具体结构如表 3.9 所示。

表 3.9 bg_tags 标签

字段	数据类型	注释	类型
id	int(11)	主键	主键
tagName	varchar(32)	标签名称	UNI

(10) 用户表

用户表主要用于注册用户的个人账号信息，主键是用户的唯一标识信息，用于唯一的标识用户不可重复，用户名称用户用户登录账号，也是全局唯一的，不可重复，密码属性一般长度在 6 到 20 位之间，因为密码是用 MD5 加盐加密的最后得到一个长度小于 256 的字符串。所以密码设置长度不能是密码本身的长度。其他字段是一些扩展字段，具体结构如表 3.10 所示。

表 3.10 bg_user 用户表

字段	数据类型	注释	类型
id	int(11)	主键	主键
username	varchar(64)	用户名	
nickname	varchar(64)	昵 称	
password	varchar(255)	MD5 加密暗文密码	
enabled	tinyint(1)	锁 定	
email	varchar(64)	邮 箱	
userface	varchar(255)	用户主页	
regTime	datetime	注册时间	

4 系统实现

4.1 研发工具

(1) IntelliJ IDEA

IntelliJ IDEA 是最优秀的 java 语言开发集成工具，在业界被公认为最好的 java 开发工具之一，支持各种插件的安装，是由 jeBrains 公司开发的产品。

(2) mysql

mysql 世界三大主流数据库之一，具有高可扩展性，性能落后于 sqlserver 和 oracle，但是其开源的特性，使 mysql 成为关系型数据库榜首。mysql 的优秀性能不可忽略。mysql 目前是 oracle 开源的项目，mysql 是目前最受欢迎的开源数据库，并在关系型数据库排名第一，ubuntu 刚开始底层也是采用 mysql，后来因为 mysql 被 oracle 收购可能存在毕源的风险而换成了 MariaDB，MariaDB 和 mysql 同样都是由 Monty 进行开发，alisql 同样是对 mysql 的改进与优化，mysql 为开源界做出了巨大贡献，mySQL/ MariaDB 可以无痕切换，无需任何配置，所有在 mysql 存在商业化的倾向下，很多企业级开发都生产环境都换成了 MariaDB，mysql 与 MariaDB 性能相差可以忽略不计。

(3) Git

Git 是由 linux 为了 linux 系统开发的一套开源的分布式的强大的版本控制系统，可以高效的管理项目版本控制。

(4) Sequel Pro

Sequel Pro 是一个快速开源的，易于使用的 Mac 数据库管理应用程序，用于处理 MySQL 数据库的图形界面系统，可以高效的处理 mysql 数据，可让您直接访问本地和远程服务器上的 mysql 数据库^[11]。

(5) Visual Studio Code

Visual Studio Code 是一个轻量级但功能强大的源代码编辑器，可在桌面上运行，可用于 Windows，MacOS 和 Linux。它具有内置的对 JavaScript，TypeScript 和 Node.js 的支持，并且具有丰富的其他语言（如 C++，C#，Java，Python，PHP，Go）和运行时（如 .NET 和 Unity）的扩展生态系统^[12]。这是一款微软开源的优秀的前端开发者工具，作为本项目的前端开发使用，可以很好的与 git 进行集成，安装可扩展性插件。

(6) maven

maven 是一个优秀的 apache 旗下的开源项目，是一款软件项目版本控制工

具，基于项目对象模型（POM）的概念，maven 可以从中央仓库拉取 jar 进行项目的搭建，已经逐渐替代 svn，通过搭建 maven 私有仓库，可以使用自己仓库的任意代码，一般公司为了安全性烤炉都会去搭建自己的 maven 仓库。

4.2 关键技术

（1）oauth 2.0

oauth2 取代了 2006 年创建的原始 OAuth 协议所做的工作。OAuth 2.0 专注于简化客户开发人员，同时为 Web 应用程序，桌面应用程序，手机和客厅设备提供特定的授权流程。此规范及其扩展正在 IETF OAuth 工作组内开发。目前 qq、微信、github、google、fa、facebook、微博账号开放平台均遵循 oauth2 协议。OAuth 2.0 授权框架支持第三方 应用程序获取对 HTTP 服务的有限访问权限 通过编排审批交互来代表资源所有者 在资源所有者和 HTTP 服务之间，或者通过允许第三方应用程序以自己的名义获得访问权限。

（2）Spring Boot

SpringBoot 可以轻松创建独立的，生产级的基于 Spring 的应用程序，可以独立运行。大多数 Spring Boot 应用程序只需要很少的 Spring 配置，完全可以替代传统的 spring+springmvc 模式的架构，直接嵌入 Tomcat，Jetty（无需部署 WAR 文件），可以直接达成 jar 进行运行，尽可能自动配置 Spring

（3）mybatis

mybatis 是支持自定义的 java 数据持久化框架，可以自定义 sql，动态扩展 sql。消除了几乎所有的 JDBC 代码以及手动设置的参数，它以接近原生 JDBC 的性能，完成底层数据库操作，把数据写入到数据库当中。

（4）dubbo

dubbo 是阿里巴巴开源的一个分布式服务治理 RPC 框架，目前最为流行的服务治理框架为 dubbo 和 springcloud，springcloud 生态比 dubbo 更为完善，模块也更为丰富，维护与更新也更为活跃与频繁，但是 dubbo 经受了国内许多互联网开发的成功案例，性能略高于 springcloud，使用也更为方便^[13]。

（5）nginx

nginx 是一个反向代理服务器，目前国内主流网站均采用 nginx 最为反向代理服务器^[14]。

4.3 功能模块实现

对于对接的系统，首先需要去该系统官方申请开发者账号，一般主流有影响力的博客网站都会有专门的开发者平台。为了安全和用户隐私考虑，最后用

户授权采用 oauth2 授权码模式进行认证，可以使用户无需在本系统输入第三方系统的账号密码即可认证登录授权成功，提高系统的粘合度和使用门槛。对接系统有多重多样，有 github、csdn、开源中国、博客园等主流博客系统。这里只拿对接开源中国和 csdn 进行示例。

4.3.1 系统核心对接 api 实现-开源中国

(1) 用户授权

url 引导用户导向开源中国登录授权页面通过 url 导向，用户跳转到开源中国登录页面进行授权。用户授权完成，跳转到自定义页面。用户授权页面如图 4.1 所示。



图 4.1 开源中国登录授权页面

(2) 通过 code 换取用户授权凭证 token

通过得到的 code 和 state 防伪造参数，通过 code 和 state 进行认证。返回 token 信息如图 4.2 所示。

```
{
  "access_token": "cb2285cc-33a8-435b-84c4-60beb5f1a339",
  "refresh_token": "5a8ca6eb-155f-40cf-a959-cbe1acd27721",
  "uid": 3855775,
  "token_type": "bearer",
  "expires_in": 604799
}
```

图 4.2 开源中国返回 token 信息

(3) 通过 accessToken 换取用户文章列表
文章列表信息如图 4.3 所示。

```
{
  "author": "繁华-",
  "id": 1809672,
  "authorid": 3855775,
  "title": "测试文章-标题",
  "body": "<h2>mysql简单sql优化与索引命中</h2> , </p>
<p>参考地址: <a href='\"http://zhuanlan.51cto.com/art/201707/545580.htm\"' rel='\"nofollow\">链接</a>
</p>",
  "pubDate": "2018-05-09 17:08:44.0",
  "favorite": 0,
  "url": "https://my.oschina.net/u/3855775/blog/1809672",
  "relatives": [
    {
      "title": "文章标题自动加“图”字",
      "url": "https://my.oschina.net/zhumofeng/blog/403786"
    },
    {
      "title": "标题测试",
      "url": "https://my.oschina.net/yeyeyeid/blog/841227"
    },
    {
      "title": "测试博客标题 ",
      "url": "https://my.oschina.net/u/2299201/blog/361981"
    },
    {
      "title": "标题",
      "url": "https://my.oschina.net/u/1785519/blog/1563024"
    }
  ],
  "commentCount": 0,
  "notice": {
    "referCount": 0,
    "replyCount": 0,
    "msgCount": 0,
    "fansCount": 0
  }
}
```

图 4.3 开源中国返回文章列表信息

(4) 通过 accessToken 和文章 id 获取文章详细信息

文章详细信息如图 4.4 所示。

```

{
  "author": "繁华-",
  "id": 1809672,
  "authorid": 3855775,
  "title": "测试文章-标题",
  "body": "<h2>mysql简单sql优化与索引命中</h2> , </p>
<p>参考地址: <a href='\"http://zhuanlan.51cto.com/art/201707/545580.htm\"' rel='\"nofollow\">链接</a>
</p>",
  "pubDate": "2018-05-09 17:08:44.0",
  "favorite": 0,
  "url": "https://my.oschina.net/u/3855775/blog/1809672",
  "relatives": [
    {
      "title": "文章标题自动加“图”字",
      "url": "https://my.oschina.net/zhumofeng/blog/403786"
    },
    {
      "title": "标题测试",
      "url": "https://my.oschina.net/yeyeyeid/blog/841227"
    },
    {
      "title": "测试博客标题 ",
      "url": "https://my.oschina.net/u/2299201/blog/361981"
    },
    {
      "title": "标题",
      "url": "https://my.oschina.net/u/1785519/blog/1563024"
    }
  ],
  "commentCount": 0,
  "notice": {
    "referCount": 0,
    "replyCount": 0,
    "msgCount": 0,
    "fansCount": 0
  }
}

```

图 4.4 开源中国文章信息详情

至此就拿到用户的博客信息，系统可以对拿到的数据在用户授权的范围内进行各种存储操作。

4.3.2 系统核心对接 api 实现-CSDN

(1) 用户授权

url 引导用户导向 CSND 登录授权页面通过 url 导向，用户跳转到开源 CSDN 登录页面进行授权，需要用户输入正确的用户名密码进行认证。因为 CSDN 开发的登录流程也是采用 oauth2 规范，但是却缺少了 state 参数进行参数的防伪造参数的校验。这样无形中增加了改平台账号密码的安全性，对于这种 api 接口，就是存在设计不合理性，所以对接方对于这种接口要单独在 url 中添加自己的防伪造标识，可以大大提高三方认证的安全性。在这里，也对 CSDN 发送了邮件让他们进行接口的改进与修复，希望他们能对接口进行尽快的改进，因为对

于这么大的一个平台来说，这种情况是不允许发生的。CSDN 用户授权页面如图 4.5 所示。



图 4.5 CSDN 授权登录页

用户授权完成，跳转到自定义页面。

(2) 通过 code 换取用户授权凭证 token

通过得到的 code 进行认证, 因为不存在 state 参数。CSDN 返回信息如图 4.6 所示。

```
{
  "access_token": "66be05d77c77478180612f0aff4c5c68",
  "expires_in": 86400,
  "username": "qinghe_"
}
```

图 4.6 CSDN 返回 token

(3) 通过 accessToken 换取用户文章列表

CSDN 返回博客列表信息如图 4.7 所示。



图 4.7 CSDN 返回文章列表

(4) 通过文章列表的文章 url 地址获取用户文章详细信息。因为 CSDN 没有提供文章详细信息接口，所以如果想要获取到文章详细信息，需要抓取博客页面，对抓取到的 html 页面进行分析。CSDN 文章详情页面如图 4.8 所示。

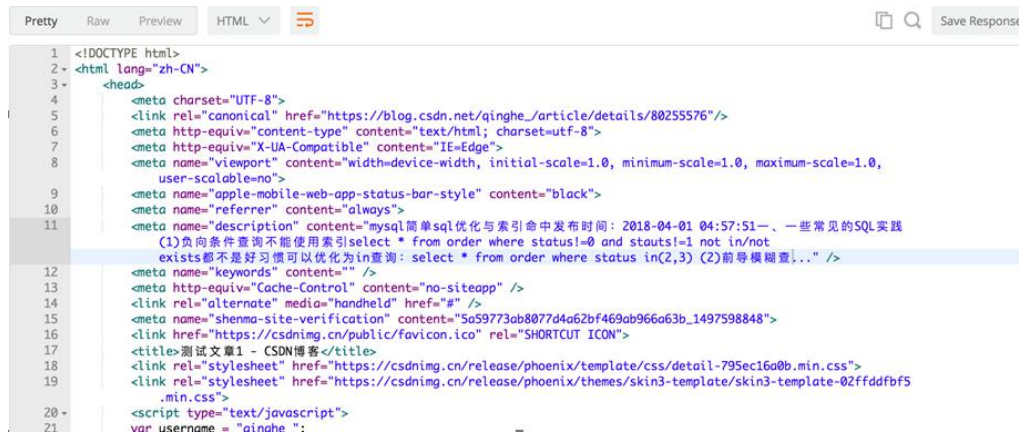


图 4.8 CSND 文章详情页面抓取

通过对 html 文件进行解析,可以拿到用户完整的博客信息,然后提取信息,进行文章数据存储。

4.3.3 系统核心对接 api 代码实现

因为对接博客系统太多,这里只拿开源中国的对接核心代码进行展示。这也是本系统代码最重要的组成部分,首先规定好对接 API 格式。接口功能分为:获取应用级 token、授权码模式获取个人级 token、刷新 token、根据个人级 token 获取用户个人信息、根据个人级 token 获取文章列表。

(1) 获取应用级 token

应用级 token 用于开发者的凭证信息,可以通过应用级 token 获取调用接口权限。传递参数有应用 id、应用密钥、请求方式、回调地址,通过 `HttpRequest.get()` 请求来进行接口调用。

(2) 授权码模式获取个人级 token。

传递参数有应用 id、应用密钥、请求方式、回调地址、获取的 code,通过 `HttpRequest.get()` 请求来进行接口调用。

(3) 刷新 token。

传递参数有应用 id、应用密钥、请求方式、回调地址、获取的旧的刷新 token,通过 `HttpRequest.post()` 请求来进行接口调用。

(4) 根据个人级 token 获取用户个人信息。接口实现代码关键部分:

传递参数有用户个人级 token,通过 `HttpRequest.get()` 请求来进行接口调用。

(5) 根据个人级 token 获取文章列表。

传递参数有用户个人级 token,通过 `HttpRequest.post()` 请求来进行接口调用。

4.3.4 登录页面实现

登录模块前端对于用户名密码进行提示和格式校验,避免后端不必要的请求导致服务器频繁请求,密码使用 MD5 加盐进行验证,表单加密提交,对于重复的请求,添加时间戳校验,用户信息写入 spring security 框架。登录页面如图 4.9 所示。

4.3.5 文章列表

文章列表可以查看用户所有来源的文章,对于官网来源的文章直接可以进行本系统显示,对于其他来源文章,则会跳转到来源渠道的网址 url。可以修改新增删除官网来源的文章,文章修改使用 md 文档书写,可以大大提高写作效率,md 优点就是简洁使用简单,一般用于写作。可以同步不同渠道来源的博

客，可以对删除的文章进行存档到回收站，防止用户误操作。可以进行文章发表，目前只支持发表到官网。文章列表页面如图 4.10 所示。文章详情页面如图 4.11 所示。文章修改页面如图 4.12 所示。文章同步页面如图 4.13 所示。

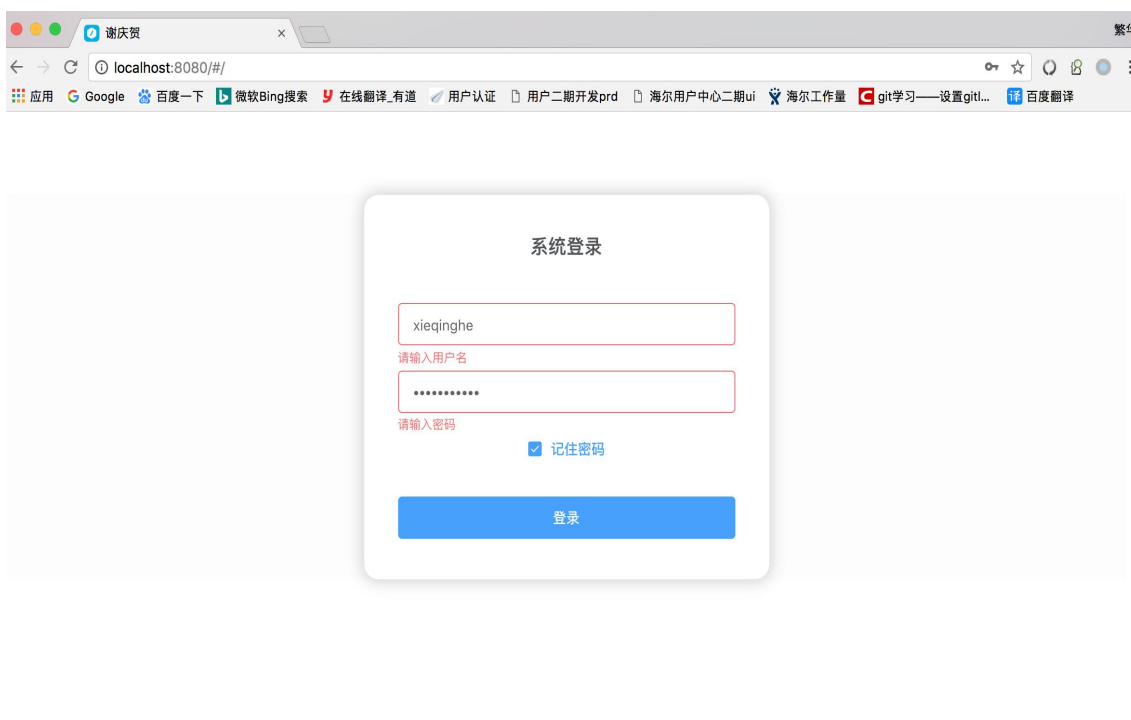


图 4.9 登录页面

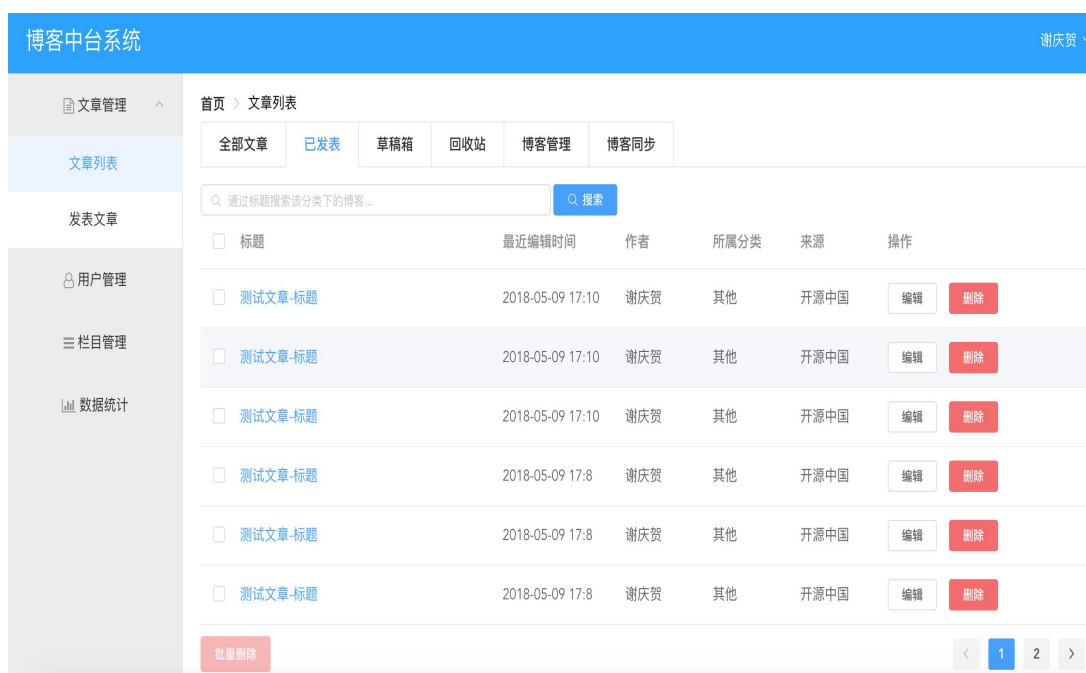


图 4.10 文章列表页面



图 4.11 文章详情页面



图 4.12 文章修改页面



图 4.13 博客同步

4.3.6 用户博客绑定显示页面

用户可以进行三方账号绑定，绑定完成以后就可以拉取三方系统的文章信

息。如图，可以显示绑定的账号，可以对账号进行禁用或者修改、删除。可以随时解除绑定。用户授权页面如图 4.14 所示。



图 4.14 用户授权管理页面

绑定三方账号使用高度抽象封装，规定好接口，对于对接系统只需要实现一套接口即可，无需修改代码，使系统无限可扩展性和 code 解耦，如图，通过对多个类进行封装集成，需要使用哪一个类即可取出来用，别的开发者不必关心具体实现过程。

4.3.7 栏目管理

可以对栏目进行管理, 栏目管理可以关联更新。栏目列表页面如图 4.15 所示。栏目修改页面如图 4.16 所示。

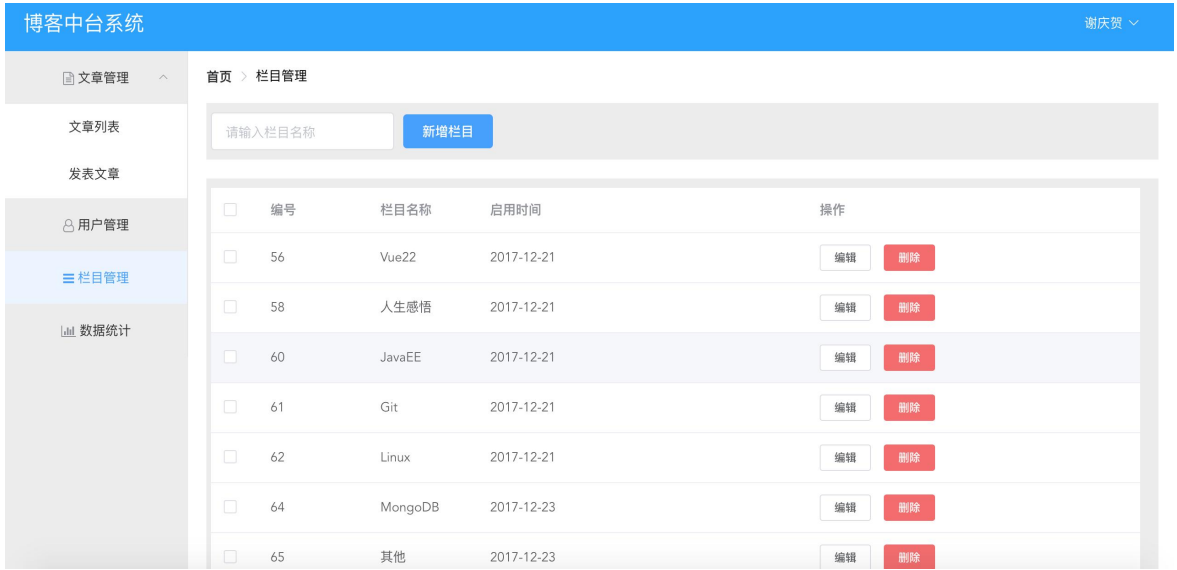


图 4.15 栏目列表页面

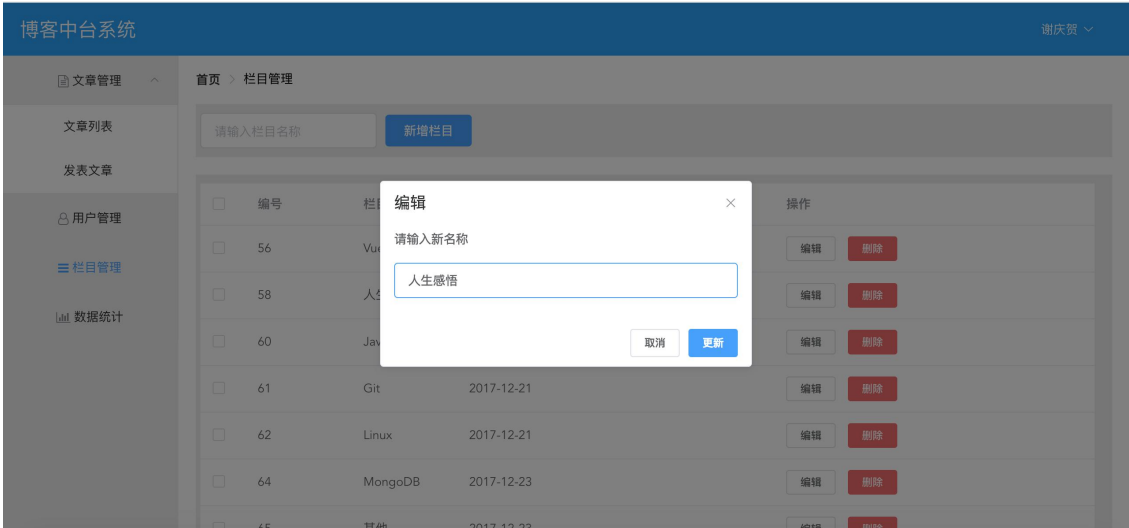


图 4.16 栏目修改页面

4.3.8 数据统计

登数据统计可以统计用户发表博客动态数据，可以随时查看自己发表博客的折线图、柱状图、饼图、数据视图，最后提供报表图片生成，下载报表数据功能。数据统计折线图页面如图 4.17 所示。数据统计柱状图页面如图 4.18 所示。数据统计数据视图页面如图 4.19 所示。

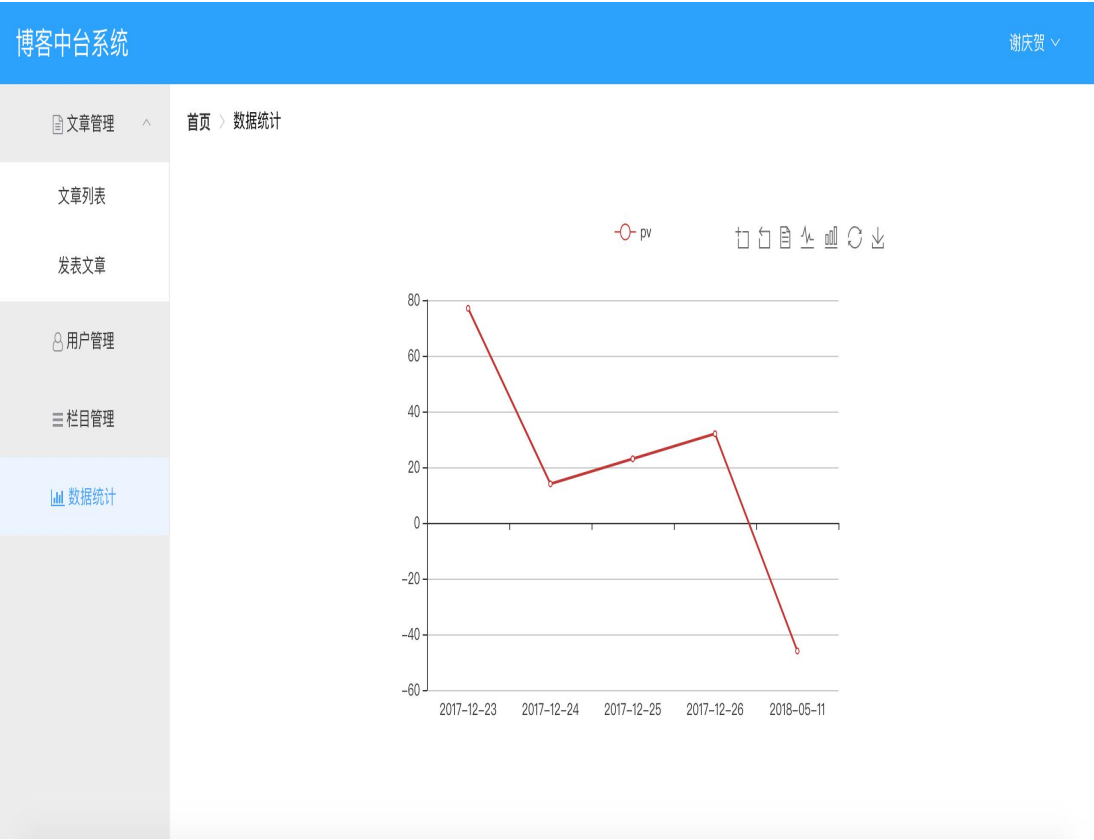


图 4.17 数据统计折线图

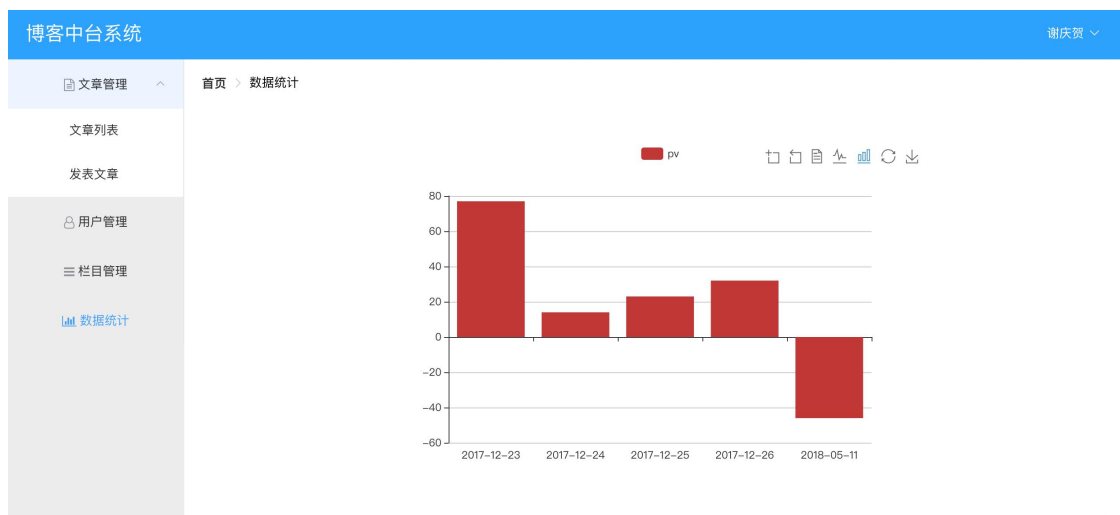


图 4.18 数据统计柱状图

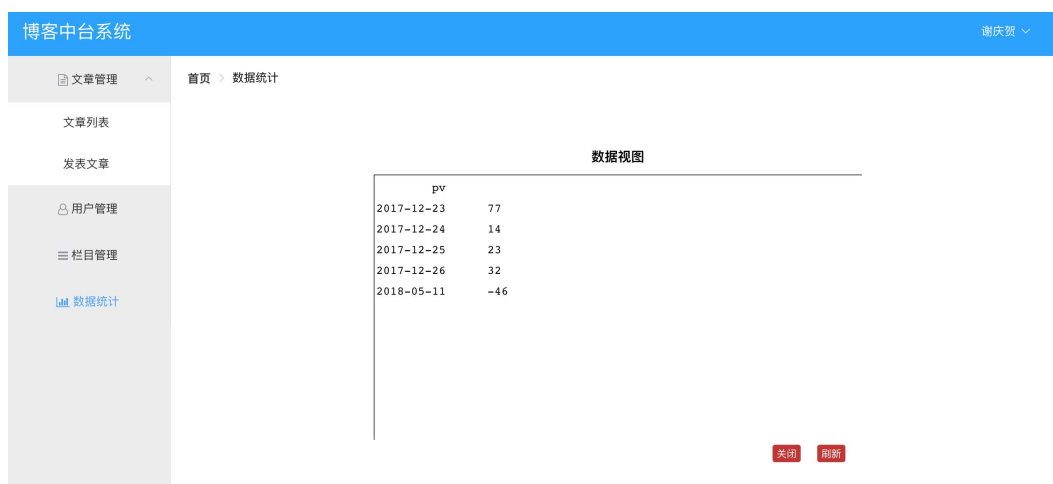


图 4.19 数据统计数据视图

4.3.9 日志记录

系统操作必须提供日志记录，日志记录原则是，关键业务操作，比如删除、增加、更新操作，尤其是审计功能、一定要添加日志记录，用户查询操作一般无需日志记录，但是对于用户敏感信息的查询，必要的时候也要留有痕迹，出现数据安全性问题可以追踪踪迹，对于数据库操作，一般需要 info 方式记录日志，方便排查 sql 语句与 bug 修复、sql 优化调优。对于异常日志：如果是系统异常或者数据库异常，一般也要记录，为 error 级别。对于业务违规操作，一般也要记录日志，方便 bug 调试，为 error 级别。对于 java 的日志记录，最后选择使用 log4j 进行日志记录，log4j 是 apache 的一个重要开源项目，使用 log4j 和 spring 集成特别容易，并且对代码无任何侵入性。Log4j 可以把日记打印到控制台当中，并且对日志进行保存，每天保存一个日志文件进行存档，log4j 核心配

置文件为，项目使用 lombok 进行日志打印，使代码更加简洁，使 model 层不必再可以去写 get set 方法，减少无用代码，lombok 生成一个日志字段。日志记录如图 4.20 所示。

```
JDBC Connection [com.mysql.jdbc.JDBC4Connection@1ee5cad3] will be managed by Spring
=> Preparing: SELECT * FROM user WHERE username=?
=> Parameters: xieqinghe(String)
<== Columns: id, username, nickname, password, enabled, email, userface, regTime
<== Row: 7, xieqinghe, 谢庆贺, 202cb962ac59075b964b07152d234b70, 1, , https://timgsa.baidu.com/timg?image&quality=80&size=b9999_10000&sec=15
<== Total: 1
Releasing transactional SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@11bb6802]
Fetched SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@11bb6802] from current transaction
=> Preparing: SELECT r.* FROM roles r,roles_user ru WHERE r.`id`=ru.`rid` AND ru.`uid`=?
=> Parameters: 7(Long)
<== Columns: id, name
<== Row: 2, 普通用户
<== Row: 1, 超级管理员
<== Row: 5, 测试角色3
<== Total: 3
Releasing transactional SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@11bb6802]
Transaction synchronization committing SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@11bb6802]
Transaction synchronization deregistering SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@11bb6802]
Transaction synchronization closing SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@11bb6802]
Sun May 20 17:10:34 CST 2018 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5.5.4
Creating a new SqlSession
Registering transaction synchronization for SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@7a542536]
Creating a new SqlSession
Registering transaction synchronization for SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@636b8abb]
JDBC Connection [com.mysql.jdbc.JDBC4Connection@1ee5cad3] will be managed by Spring
=> Preparing: SELECT count(*) FROM article WHERE state=? AND uid=? AND title LIKE concat('%',?, '%')
JDBC Connection [com.mysql.jdbc.JDBC4Connection@34517007] will be managed by Spring
=> Preparing: SELECT count(*) FROM article WHERE uid=? AND title LIKE concat('%',?, '%')
=> Parameters: 7(Long), (String)
=> Parameters: 1(Integer), 7(Long), (String)
<== Columns: count(*)
<== Row: 10
```

图 4.20 日志记录

4.4 系统难点与解决方案

在开发过程当中，难免遇见一些意想不到的问题，遇到很多意想不到和意料当中的问题。对接三方博客系统官方帮助文档缺失，api 存在天然的 bug，开发者账号申请繁琐有些系统审核严格，对接三方系统接口出现莫名其妙的 bug，最难的还是不同系统之间的适配与高可扩展性，怎样才能开发一套在理论上可以无限扩展的系统，在对接不同博客平台当中，设计出一套通用架构是很困难的，因为设计架构和 spring social 开源架构相似，于是通过对 spring social 源码的阅读，学到了很多可扩展性的设计方案，借鉴这个方案设计出自己的独立系统，使其可以进行很好的扩展，并且性能也有很大提高。在对接 oauth2 流程的时候，因为 code 只有一次使用机会，而如果对于原理不太明白则会出现问题。

5 系统测试

软件测试和软件调试时开发当中特别重要的一部分，因为测试直接影响到用户体验与系统健壮性，通过白盒测试和黑盒测试，可以暴露出系统的漏洞。通过设计合理和严谨的测试用例，在最大程度保证测试覆盖率，才能更好的进行系统化测试。

5.1 系统测试的目的与重要性

测试的目的是为了发现尽可能多的缺陷，不是为了说明软件中没有缺陷。成功的测试在于发现了迄今尚未发现的缺陷。所以测试人员的职责是设计这样的测试用例，它能有效地揭示潜伏在软件里的缺陷。确保产品像设计的那样工作，正确实现了所设计的需求，没有实现设计之外的需求，性能可靠，满足期望的兼容性，确保在上线日前达到上线标准，持续追踪项目进度，把控各个开发阶段的产品质量。从测试工程师的角度分析：发现程序中的错误、降低代码出错风险、保证代码质量。从公司的角度分析：降低公司成本^[15]。

5.2 系统测试的目的与重要性

项目后端测试往往是白盒测试更为重要，选择对 service 进行 mock 工具进行断言测试，设计合理的测试用例，mock 是一个专门针对 java 程序的工具包。如：一个合理的 service 服务测试用例如下。

```
@Mock
BrandCacher brandCacher;
@InjectMocks
private LinkPointsCommonReadServiceImpl linkPointsCommonReadService;
@Mock
private ConfigCenter configCenter;
@Before
public void init() {
    MockitoAnnotations.initMocks(this);
}
@Test
public void testFindAllAccountMapping() {
```



```

when(configCenter.get("points.user.brand")).thenReturn("[{\"key\":\"01\",\"value\":"
+ "\"开源中心\",\"code\":\"A\"},{\"key\":\"02\",\"value\":\"项目中心\","
+ "\"code\":\"B\"}]");
Response<List<LinkAccountMapping>> resp7 =
linkPointsCommonReadService.findAllAccountMapping(1L, "01");
Assert.assertTrue(resp7.isSuccess());
when(configCenter.get("points.user.brand")).thenReturn("[{\"key\":\"01\",\"value\":"
+ "\"开源中心\",\"code\":\"A\"}]");
Response<List<LinkAccountMapping>> resp =
linkPointsCommonReadService.findAllAccountMapping(1L, "01");
Assert.assertTrue(resp.isSuccess());
Assert.assertEquals(1, resp.getResult().size());
}

```

对于 dao 层的测试看似简单，但是不能直接通过连接开发的数据库进行测试，一般 java 项目当中使用 H2 数据库进行对 dao 层进行测试，h2 数据库是一个基于内存的微型数据库，专门用来进行测试，不适合于生产级别的系统使用，配置 h2 数据库只需要写好建表语句和插入语句，然后通过配置文件进行数据读取即可。一个合理的 dao 层测试用例为：

```

@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(classes = EmbeddedPersistenceConfig.class)
public class LinkPointRulesDaoTest {
    @Autowired
    private LinkPointRulesDao linkPointRulesDao;
    LinkPointRules mock() {
        Random r = new Random();
        LinkPointRules linkPointRules = new LinkPointRules();
        linkPointRules.setChannelId(1L);
        linkPointRules.setAppKey(1000001L);
        linkPointRules.setBrandKey("01");
        linkPointRules.setRuleShortName("123");
        linkPointRules.setFixRandom(1);
        linkPointRules.setLevel(1);
    }
}

```

```

linkPointRules.setNumberYear(2);
linkPointRules.setScore(55L);
linkPointRules.setStatus(1);
linkPointRules.setRuleKey("1000001");
return linkPointRules;
}

@Test
public void testCreate() throws Exception {
    LinkPointRules linkPointRules = mock();
    Assert.assertTrue(linkPointRulesDao.create(linkPointRules));
}
}

```

编写好测试用例，即可使用 idea 工具进行测试用例的调试与运行。如果测试用例断言与运行结果不同，就代表代码出现问题，则会出现红色提示，测试用例运行失败如图 5.1 所示。

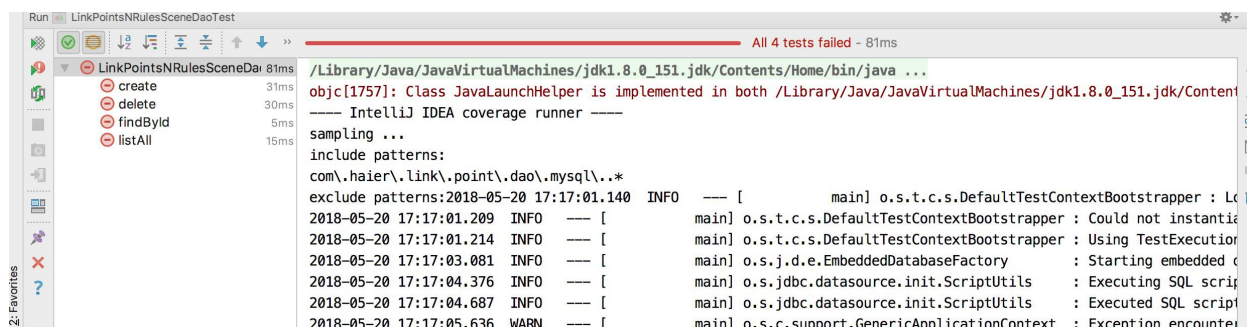


图 5.1 测试用例运行失败

如果断言与运行预期结果相同，代码无 bug，则会出现绿色标志，测试用例运行成功如图 5.2 所示。

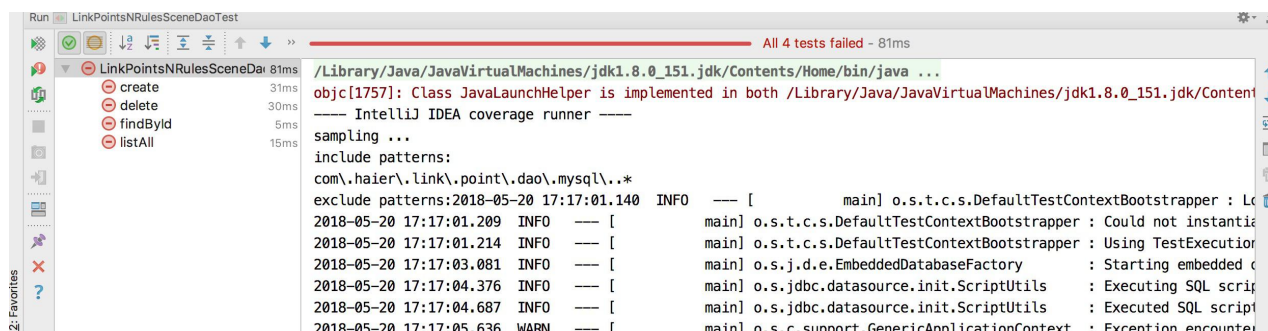


图 5.2 测试用例运行成功

通过测试用例还可以看到测试用例运行时间，通过对测试用例运行时间的分析，可以分析出运行时间耗时较多的代码块，可以专门进行优化。最后编写好所有模块的测试用例，就可以通过 idea 工具运行所有测试用例，idea 会自动统计各个类的覆盖率、类方法的覆盖率、代码行数的覆盖率，很多大型企业，对于软件测试覆盖率都有严格的审查，一般都要求不低于代码覆盖率的 60%。如图是对各个类的覆盖率统计，对于一些特殊的类，是不必编写测试用例的，所以会出现 0%的情况，最终项目的测试覆盖率都达到了 66%，已经超过标准，所以是合格的测试用例。测试用例用例覆盖率统计分析如图 5.3 所示。

Element	Class, %	Method, %	Line, %
BatchPointsInitServiceImpl	0% (0/3)	0% (0/11)	0% (0/91)
LinkChannelBudgetReadServiceImpl	100% (1/1)	100% (3/3)	93% (55/59)
LinkChannelBudgetWriteServiceImpl	100% (1/1)	100% (7/7)	100% (54/54)
LinkPointRulesReadServiceImpl	100% (1/1)	100% (3/3)	84% (21/25)
LinkPointRulesWriteServiceImpl	100% (1/1)	100% (3/3)	77% (42/54)

图 5.3 测试用例覆盖率统计分析

5.3 具体测试用例及结果

测试用例 ID	场景	测试步骤	预期结果
TC1	初始化登录页面	从登录页面访问	登录页面能够正常显示
TC2	跳过登录直接访问数据页面	不进行的登录直接内容页面	系统提示 403 权限不够
TC3	账号密码进行登录	输入正确的账号密码登录	登录成功，进入等用户主页面
TC4	查询文章	查询第一页文章	查询成功显示
TC5	查询文章	查询文章详情	查询成功
TC6	查询文章	查询从第三方获取的文章	跳转到三方登录页
TC7	文章发布	编写好文章进行发布	发布成功
TC8	修改文章	对文章进行修改	修改成功
TC9	删除文章	对文章进行删除	删除成功
TC10	同步文章	对文章进行同步	同步成功
TC11	绑定三方博客账号	绑定三方博客账号	绑定成功

6 总结

在项目开发过程当中，遇到了很多预想不到的问题和困难，在申请开发者账号就用去了两天的时间，通过对多个博客系统的开发者帮助文档阅读与测试，验证可行性方案，可行性验证通过以后才进行代码的设计，代码编写速度就明显加快。一个项目的成型，前期 PRD 蓝图规划尤为重要它是项目的根本，后期根据不同功能选型不同技术进行实现，这个过程是不太难的。

（1）接口规范设计

在开发过程当中，遇到很多预料到和没有事先预料的事情。在对接博客园系统时，官方 api 本身就存在 bug，不能去让对接方去主动修改 bug，所以要通过额外的辅助手段来替代某个存在问题的接口。最难的还是不同对接方系统之间的适配与高可扩展性之间做出选择，需要开发一套优秀的高可扩展的系统，上升到框架的层面。于是使用职责链模式和工厂模式相结合，参考 springmvc 流程的代码设计，设计了一套可以优秀扩展的系统。在系统开发过程中，学到了很多可扩展性的设计方案，在以后工作过程中如果遇到类似的问题，可以很容易去进行解决。

（2）对接三方博客平台

在最终查阅各个对接平台提供的对接 API 时，对接开源中国系统时在 postman 模拟器中测试接口没有问题，但是在代码中对接接口却发生 403 用户无权限问题。最终通过查询请求头，发现必须传递浏览器标识信息；最后通过模拟 google 浏览器标识请求头信息，再进行 API 的请求，就不会存在用户无权限的错误。这些问题都是事先无法预料的，官方对接文章也没有给出说明，在这种情况下，如果不了解 HTTP 请求头信息规则，是无法想到去模拟终端进行请求的。所以不经意间的基础知识，却可以提高解决问题的效率。平常一定要积累知识库，而不是等到出现问题的时候找不到解决方案再去学习。注重学习知识的不断积累，知识库不是一朝一夕进行积累的，而是通过持久的不断的主动学习日积月累的。

参考文献

- [1] 赵彦云. 对大数据统计设计的思考[J]. 统计研究, 2015, 32(6):3-10
- [2] 余明辉, 张良均. Hadoop big data development[J]. 人民邮电出版社, 2018. 30-42
- [3] 设计原本:计算机科学巨匠 Frederick P. Brooks 的反思: Design of design: essays from a computer scientist[M]. 机械工业出版社, 2013
- [4] George Fairbanks. 恰如其分的软件架构[M]. 华中科技大学出版社, 2013
- [5] Silva M D D, Tavares H L. Redis Essentials[J]. Packt Publishing, 2015
- [6] Carlson J L. Redis in Action[J]. Media.johnwiley.com.au, 2013
- [7] 李慧. 浅谈 Dubbo 分布式服务系统的应用[J]. 中国多媒体与网络教学学报(电子版), 2017(6)
- [8] 高洪岩. Java 并发编程:核心方法与框架[M]. 机械工业出版社, 2016
- [9] Rivest R. The MD5Message-Digest Algorithm[J]. Rfc, 1992, 473(10):492-492
- [10] DouglasR.Stinson 道格拉斯 R.斯廷森. 密码学原理与实践[M]. 电子工业出版社, 2016
- [11] CharlesBell, MatsKindahl, LarsThalmann. 高可用 MySQL[M]. 电子工业出版社, 2015
- [12] 张红实, 周志川. 浅谈 Java 虚拟机体系结构和垃圾回收技术[J]. 当代教育实践与教学研究:电子刊, 2017(1)
- [13] 李林锋. 分布式服务框架原理与实践[M]. 电子工业出版社, 2016
- [14] Nedelcu C. Nginx HTTP Server[M]. Packt Publishing, 2015
- [15] 张静, 寇峰, 陈井泉. 软件测试[M]. 中国水利水电出版社, 2016

致谢

经过了一个多月的努力，《博客中台系统》终于全部完成。因为很早去出去实习，所以在实习公司就开始准备毕业设计该如何去写，如何才能写的优秀，在这期间很感谢实习期间带我的两个师父张斌和郭润坤，他们教会了我很多东西，毕业设计所用到的技术积累，基本都是他们那时候带我所交给我的东西，包括 spring 源码的分析，mysql 使用心得等等，有些东西是不经过工作经验的积累无法进行总结得出的结论，他们全部无条件的教授给我，并且郭老师还对我的毕业论文进行了可行性分析和实现方案供我参考，在最后的实习结束以后他也尽力帮我争取更高的工资，别人不欠你什么东西，对我好的人，我一定会从心里一直记得他们。因为工作的原因，平常没有时间回到学习，很感谢吴秋兰老师对我们的理解与支持。吴秋兰老师为了我们论文能够正常进行，还专门整理注意事项让我们更好的去写论文，真的很感谢老师的帮助与支持。

四年的大学时光，收获的不仅仅是学习的知识，还有做人的道理与学习的能力，学会了独立思考能力和端正的思想。大学四年也认识了很多良师益友，我们共同进步，共同努力，感谢遇到的大学的每一个让我受益良多的人，大学的时光总是那么快就要度过，在以后的工作生活当中我也要更加努力。