

毕业设计(论文)

|  |  |
| --- | --- |
| 课题名称 | 数据可视化分析平台 |
|  | ——后台管理系统的设计与实现 |
| 二级学院 | 计算机与软件学院 |
| 专 业 | 软件技术 |
| 班 级 | 软件1851 |
| 学 号 | 1802343124 |
| 姓 名 | 张浩杰 |
| 指导教师 | 许莫淇 |

年 月 日

计算机与软件学院

毕业设计（论文）诚信承诺

我谨在此承诺：本人所写的毕业论文《基于B/S架构的在线数据可视化分析平台——后台管理系统的设计与实现》，系本人独立完成，没有抄袭行为，凡涉及其他作者的观点和材料，均作了注释与说明，若有不实，后果由本人承担。

承诺人（签名）：

年 月 日

摘 要

自从互联网兴起之后，中国互联网行业高速发展，各种新型应用层出不穷，吸引了越来越多的网民。同时，随着应用中数据的海量增长，数据可视化平台凭借着其对于数据处理的灵活性和数据展现的生动性，在数据分析领域的发展前景一片欣欣向荣，各类数据分析可视化平台顺势而生。新的机遇带来了新的挑战，如何对数据可视化平台有效的进行管理成为了一个新的课题。

本文基于开发的数据可视化系统，在现有的数据管理和分析基础上结合数据分析行业的实际需求和业务流程，设计并实现了数据可视化分析平台的后台管理系统。系统基于Java的SpringBoot框架，搭配Vue框架对前端进行展示。

系统根据业务需求划分为四个模块，分别为：用户模块、数据分析处理模块和数据权限模块。本系统基于SpringBoot框架和Vue框架实现了数据可视化分析平台的后台管理功能；管理员登录本系统可以对自己的基础信息进行修改也可以增加新的管理员；在本系统中管理员的主要作用是管理数据可视化分析平台的数据接入类型、平台总处理的数据量分析图和每个使用平台的用户连接、上传数据的次数和处理的数据量。通过分析用户的行为可以及时修改平台中出现的问题以维持开发平台的长久发展。

关键词：数据可视化；后台管理系统；SpringBoot

Abstract

Since the rise of the Internet, China's Internet industry has developed rapidly, and various new applications have emerged one after another, attracting more and more Internet users. At the same time, with the massive growth of data in applications, the data visualization platform, with its flexibility in data processing and vividness of data presentation, has a prosperous development prospect in the field of data analysis, and various data analysis and visualization platforms have emerged. New opportunities have brought new challenges, and how to effectively manage the data visualization platform has become a new topic.

Based on the data visualization system developed in this paper, combined with the actual needs and business processes of the data analysis industry on the basis of existing data management and analysis, the background management system of the data visualization analysis platform is designed and implemented. The system is based on Java's SpringBoot framework and is used with Vue framework to display the front end.

The system is divided into four modules according to business requirements, namely: user module, data analysis and processing module, and data authority module. This system is based on the SpringBoot framework and the Vue framework to realize the background management function of the data visualization analysis platform; the administrator can log in to the system to modify his own basic information or add new administrators; the main role of the administrator in this system is Manage the data access type of the data visualization analysis platform, the analysis graph of the total data volume processed by the platform, and the connection of each user using the platform, the number of data uploads, and the amount of data processed. By analyzing the behavior of users, problems in the platform can be modified in time to maintain the long-term development of the development platform.

Keywords: data visualization; background management system; SpringBoot

目 录

[第1章 绪论 1](#_Toc72612616)

[1.1 课题背景与研究意义 1](#_Toc72612617)

[1.1.1 课题背景 1](#_Toc72612618)

[1.1.2 研究意义 1](#_Toc72612619)

[1.2 研究目标及内容 1](#_Toc72612620)

[1.2.1 研究目标 1](#_Toc72612621)

[1.2.2 研究内容 1](#_Toc72612622)

[1.3 所做工作 1](#_Toc72612623)

[第2章 开发技术与工具 2](#_Toc72612624)

[2.1 开发技术 2](#_Toc72612625)

[2.1.1 SpringBoot 2](#_Toc72612626)

[2.1.2 Spring Data JPA 2](#_Toc72612627)

[2.1.3 MySQL 2](#_Toc72612628)

[2.1.4 Maven 2](#_Toc72612629)

[2.1.5 Vue 2](#_Toc72612630)

[2.1.6 Vuetify 2](#_Toc72612631)

[2.1.7 V-charts 2](#_Toc72612632)

[2.1.8 Node.js 3](#_Toc72612633)

[2.1.9 Axios 3](#_Toc72612634)

[2.1.10 Vuex 3](#_Toc72612635)

[2.1.11 Vue-router 3](#_Toc72612636)

[2.2 开发工具 3](#_Toc72612637)

[2.2.1 IntelliJ IDEA 3](#_Toc72612638)

[2.2.2 GitHub 3](#_Toc72612639)

[2.2.3 Navicat 3](#_Toc72612640)

[2.2.4 CentOS8 3](#_Toc72612641)

[2.2.5 Visual Studio Code 3](#_Toc72612642)

[第3章 系统需求分析 4](#_Toc72612643)

[3.1 系统总体需求概述 4](#_Toc72612644)

[3.1.1 系统稳定性分析概述 4](#_Toc72612645)

[3.1.2 产品特点分析概述 4](#_Toc72612646)

[3.2 功能性需求分析 4](#_Toc72612647)

[3.2.1 用户模块 4](#_Toc72612648)

[3.2.2 数据分析处理模块 4](#_Toc72612649)

[3.2.3 数据展示模块 4](#_Toc72612650)

[3.2.4 用户行为模块 4](#_Toc72612651)

[第4章 系统概要设计 5](#_Toc72612652)

[4.1 系统整体功能图 5](#_Toc72612653)

[4.2 数据库设计 5](#_Toc72612654)

[4.2.1 用户表设计 5](#_Toc72612655)

[4.2.2 用户行为表设计 5](#_Toc72612656)

[4.3 系统接口设计 5](#_Toc72612657)

[4.3.1 用户模块接口设计 5](#_Toc72612658)

[4.3.2 数据分析处理模块接口设计 5](#_Toc72612659)

[4.3.3 系统权限模块接口设计 5](#_Toc72612660)

[4.4 Vue封装设计 5](#_Toc72612661)

[4.4.1 组件封装设计 5](#_Toc72612662)

[4.4.2 接口封装设计 5](#_Toc72612663)

[第5章 系统详细设计 6](#_Toc72612664)

[5.1 用户模块 6](#_Toc72612665)

[5.1.1 登录功能设计 6](#_Toc72612666)

[5.1.2 新增管理员功能设计 6](#_Toc72612667)

[5.1.3 修改基本信息功能设计 6](#_Toc72612668)

[5.2 数据分析处理模块 6](#_Toc72612669)

[5.2.1 数据库表上传与文件上传次数对比功能（图表展示）设计 6](#_Toc72612670)

[5.2.2 计算指定用户每周上传次数功能设计 6](#_Toc72612671)

[5.3 系统权限模块 6](#_Toc72612672)

[5.3.1 分类展示用户行为功能设计 6](#_Toc72612673)

[5.3.2 根据用户上传次数排序功能设计 6](#_Toc72612674)

[5.3.3 展示所有用户信息功能设计 6](#_Toc72612675)

[5.3.4 监控用户上传数据量功能设计 6](#_Toc72612676)

[5.3.5 控制系统数据源接入类型功能设计 6](#_Toc72612677)

[第6章 系统功能的实现 1](#_Toc72612678)

[6.1 用户模块 1](#_Toc72612679)

[6.1.1 登录功能实现 1](#_Toc72612680)

[6.1.2 新增管理员功能实现 1](#_Toc72612681)

[6.1.3 修改基本信息功能实现 1](#_Toc72612682)

[6.2 数据分析处理模块 1](#_Toc72612683)

[6.2.1 数据库表上传与文件上传次数对比功能（图表展示）实现 1](#_Toc72612684)

[6.2.2 计算指定用户每周上传次数功能实现 1](#_Toc72612685)

[6.3 系统权限模块 1](#_Toc72612686)

[6.3.1 分类展示用户行为功能实现 1](#_Toc72612687)

[6.3.2 根据用户上传次数排序功能实现 1](#_Toc72612688)

[6.3.3 展示所有用户信息功能实现 1](#_Toc72612689)

[6.3.4 监控用户上传数据量功能实现 1](#_Toc72612690)

[6.3.5 控制系统数据源接入类型功能实现 1](#_Toc72612691)

[第7章 系统测试 2](#_Toc72612692)

[7.1 测试环境准备 2](#_Toc72612693)

[7.2 功能性测试 2](#_Toc72612694)

[7.3 系统部署测试 2](#_Toc72612695)

[第8章 总结和展望 3](#_Toc72612696)

[8.1 总结 3](#_Toc72612697)

[8.2 展望 3](#_Toc72612698)

[致谢 4](#_Toc72612699)

[参考文献 5](#_Toc72612700)

# 绪论

这是一个数据爆炸的时代，每个人在网络上留下的痕迹都成为信息，这些信息被记录，整理，以便创造价值，由此数据分析应运而生，并且流行起来。商业智能便是其中之一，它是一个面向企业的应用，通过报表分析和数据分析，达到帮助企业决策的目的。数据可视化平台的后台管理系统提供了用户管理及用户行为分析等功能，本章将从课题背景与研究意义、研究目标及内容两个角度论述。

## 课题背景与研究意义

### 课题背景

本课题是来自于开发的数据可视化平台；为了保证数据可视化分析平台运行的稳定性；特地为该系统开发了数据可视化分析平台的后台管理系统。

“数据可视化”是当今数据分析领域中发展最快，也最引人注目的领域，目前市场上出现了许多数据可视化平台并且配备相应的后台管理系统，在这种趋势下，我开发的数据可视化分析平台也决定为系统配备后台管理系统，该系统可以友好的管理数据和用户信息；通过柱状图或饼图的形式展现系统的使用量和在市场中的地位。

### 研究意义

本课题开发的系统是数据可视化后台管理，管理员通过登录后台管理首先看到的是当天用户上传文件类型数据量对比；其次管理员有权限控制前台系统的功能如：数据库连接类型、用户是否可以继续接入数据等；通过管理员全局把控系统的走向，可以使系统在运行的过程中更稳定；管理员通过分析用户每日上传的数据量也可以更清晰的了解前台系统的使用功耗。

## 研究目标及内容

### 研究目标

本论文研究的目标是致力于开发一款和前台数据可视化系统相匹配的后台管理系统。该平台管理员可以管理用户的行为，也可以增加和删除管理员的数目。使用该平台的用户可以管理数据源接入和类型，也可以禁止用户进行违规行为的操作；平台首页以柱状图的形式展现出当天用户上传数据量的对比，了解用户使用平台的次数；及时修改平台中需要修改的弊端，维持平台的长久发展。

### 研究内容

本文的研究内容是设计并实现一个数据可视化分析平台的后台管理系统。通过SpringBoot框架实现后台的技术选型，采用JPA实现数据的增删改查，使用Navicat作为数据库管理工具，前端以Vue框架和V-charts结合完成框架搭建和图像的展现研究内容。本文的研究内容主要包括：

1. 对于主流的数据可视化管理系统进行分析，了解后台管理系统的作用，抽出最主要的功能，作为需要开发项目的必备功能。
2. 研究匹配所选后台技术栈的接口管理工具，通过搭建一个统一管理和测试接口的平台，可以清楚接口的开发进度和开发结果。
3. 在项目开发之前封装统一响应体、枚举类和自定义异常，建立代码规范；为后续的项目开发减少冲突。

## 所做工作

为了使系统实现的功能更加完备，系统性能更加稳定，在开发过程中主要做了如下的工作：

1. 确立系统的基本功能；分析开发系统需要具备的开发工具和开发技术，选择可以满足系统稳定性需求的开发技术，在后端的技术栈选择SpringBoot和JPA相搭配，接口文档采用Swagger统一管理；前端选择Vue搭建前端的框架，Vuetify作为页面内显示的组件。
2. 完成项目功能的详细设计和实现；分析每个模块具体需要实现的功能点并通过所选技术栈进行实现，完成前后端功能的联调。
3. 测试系统功能实现是否完善；根据系统实现的具体功能，对每个模块的各个功能进行功能测试，系统只有通过完整的测试，才能确定系统的稳定性和安全性。

# 开发技术与工具

本章作者将介绍在此次后台管理中所涉及到的所有技术与工具。合理使用技术、恰当使用工具，快速地进行开发。

## 开发技术

### SpringBoot

SpringBoot是目前主流的开发框架之一。它是一个大而全的框架，它的设计目的是简化Spring应用的初始搭建，尽可能快速的启动运行Spring应用和尽可能少的配置文件，相当于一个启动Spring项目的工具，集合了一些库，可以快速高效的进行Spring应用的开发。最直观的好处在于简化的统一配置文件、开箱即用的许多依赖、内嵌的Tomcat等。

### Spring Data JPA

Spring Data JPA是Spring大家族中的一员，简化实现基于JPA的存储库。它致力于将重复工作量减少到实际需要的量，减少样板代码的编写，使开发人员可以更好的专注于逻辑代码，改善了数据访问层的实现。

### MySQL

MySQL是一个开源的关系型数据库管理系统，它的主要优势是快速、健壮、易用和免费。它适用于中小型企业和个人用户，占用空间小、上手快速、跨平台等都是它对个人用户的友好体现，对比Oracle，它是免费的，上手更简单的，对比SQL Server只能在Windows操作系统运行，它则可以跨平台，在UNIX、Linux、Windows 和 Mac OS 等操作系统上都能运行。

### Maven

Maven翻译为“专家”“内行”是一个带有依赖管理工具和项目管理工具的构建工具，可以对 Java 项目进行构建和依赖管理。不仅能够帮助我们实现自动化构建，还能抽象构建过程，它是跨平台的，对外提供了一致的操作接口，并且，它提供了中央仓库，能帮助用户自动下载构件。Maven 也可以被用于构建和管理其他项目，例如 C#，Scala，Ruby 和其他语言编写的项目。

### Vue

vue是一个友好的、多用的途且高性能的js框架，它的优势在于学习曲线平缓、使用灵活、运行高效，是一款构建用户界面的响应式和渐进式的框架。无论项目大小，都可以满足开发人员的需求，最简配置，按需引入，使项目渐进式成长，可以帮助我们创建可维护性和可测试性更强的代码库。它专注于视图层，采用自底向上的增量开发设计，可以更好地实现代码复用，减少工作量。

### Vuetify

Vuetify是一个 Vue的UI 库，包含手工制作的Material Design风格的精美材料组件。不需要精通设计技能就可以创建美观的应用程序。Vuetify是一个基于Vue2.0d的，为移动而生的组件框架，是一个渐进式的UI框架；旨在提供整洁、语义化和可复用的组件，使构建应用程序更加方便。

### V-charts

V-charts是一个基于Vue2.0和echarts封装的图表组件。echarts翻译为“商业级数据图表”，是一个基于 JavaScript 的开源可视化图表库。V-charts的设计解决了echarts生成图表时，经常需要做繁琐的配置项的问题，它只需要统一提供一种数据格式，设置简单的配置项，便可生成常见图表。

### Node.js

Node.js 是一个基于 Chrome V8 引擎的 Javascript 运行环境。他并不是一个库，而是类似于一个JS语言解释器。它的非阻塞I/O和事件驱动两项特色，使它可以成为一个优秀的、高性能的服务器端运行环境，是可以让JavaScript运行在服务端的开发平台，它让JavaScript成为一个服务端脚本语言，成为了Web前端开发必不可少的基础设施。

### Axios

Axios 是一个基于 promise 的 HTTP 库，用于HTTP的请求，既可以用于客户端（浏览器），也可以用于服务端（node.js）。

### Vuex

Vuex 是一个专为 Vue.js 应用程序设计的状态管理模式。它采用集中式存储管理应用的所有组件的状态，并以相应的规则保证状态以一种可预测的方式发生变化。它的表现为，把应用中的所有组件的状态剥离出来，在应用外部创建一个全局单例模式，对这个模式采用集中式存储管理。它的设计可以可以进一步实现例如Vue、React这类响应式框架的代码间松耦合的需求。

### Vue-router

Vue-router是一个vue.js下的路由组件，与vue.js深度集成，适合用于构建单页面应用，是一个路径管理系统。Vue的单页面应用是基于路由和组件的，路由则代替了传统的a标签，将路径和组件映射起来，在Vue单页应用中，起到页面之间切换和跳转作用，从组件的角度看，相当于路径之间的切换，也是组件的切换。

## 开发工具

### IntelliJ IDEA

IntelliJ IDEA（Integrated Development Environment） 是一款现代化智能开发环境工具软件，由JetBrains软件公司开发，主要支持 Java、Scala、Groovy 等语言，同时具备支持目前主流的技术和框架，擅长于企业应用、移动应用和 Web 应用的开发。相比于传统的开发工具，它将工作空间的概念转变为Project和Module两个不同的概念，项目工程（Project）是聚合工程或者普通根目录，其中的子工程叫做模块（Module），每个子工程之间相互独立，按需关联，可以使用独立的JDK和Maven

### GitHub

Git是一个版本控制工具，Github是一个用git做版本控制的项目托管平台。版本控制最主要的功能便是追踪文件的变更，包括什么时候、什么人修改了文件的什么内容等信息真实地了记录下来。每一次文件的改变，文件的版本号都将增加。除了记录版本变更外，版本控制的另一个重要功能是并行开发。

### Navicat Premium

Navicat是一个可多重连接的数据库管理工具，它可以连接到Oracle、MySQL、SQL Server、PostgreSQL、SQLite、和MariaDB数据库，让数据库管理更加方便。Navicat Premium结合了其它Navicat系列中所有成员的功能，可满足目前数据库管理系统的使用功能，包括事件、存储过程、视图、触发器、函数等。

### CentOS8

CentOS8是CentOS Linux发行版中的一个版本，是一个稳定的，可预测的，可管理的和可复制的操作平台，该平台源自Red Hat Enterprise Linux（RHEL）。类似于Windows操作系，CentOS8是Linux操作系统，可以在其中进行开发，运行和部署。

### Visual Studio Code

Visual Studio Code是一个可以免费使用的、轻量级的、简化高效的代码编辑器，这个编辑器来自online editor Monaco，由坐镇苏黎世的Eric Gamma率领团队打造，用typescript写就。同时支持调试、任务执行，版本管理等开发操作。它的设计目的就是为开发者们提供一个快速的编码-编译-调试的开发工具。

# 系统需求分析

后台管理人员经过对整个项目的了解以及研究之后，将对后台管理中的系统需求分析做出如下概述。为后续的人员管理、系统管理提供更大的便捷。

## 系统总体需求概述

### 系统稳定性分析概述

对于一个软件而言，系统稳定性主要在于数据交互的稳定。

我们在此项目中使用MySQL技术作数据持久化操作。MySQL数据库本身不仅具有体积小、安装简单、易于维护的优点，而且还具有性能卓越服务稳定，很少出现异常宕机的特点。因此，我们在项目中拿MySQL数据库做数据持久化操作。与此搭配使用Spring Data JPA 进行数据的读取操作。Spring Data JPA 在一些简单的查询中非常方便，有利于快速开发，可节省大量代码。为我们的后台管理系统项目的敏捷开发提供便利。

使用Spring Boot 做后端框架开发接口。代码的目录层级明确划分，在Spring Boot 的基础上，也可实现接口的快速开发。对于整个代码的而言，可读性很强。也方便后续的迭代开发。

### 产品特点分析概述

该系统有如下几个特点：

* 1. 提供 Swagger 接口文档：当项目中存在大量的接口时，没有一个明确的文档来描述接口信息，是什么不明确的行为，因此，我们使用 Swagger 来做接口说明文档，为开发提供便捷。
  2. 用户行为分析：在此次的后台管理系统中，我们不仅对一些常规的权限进行管理，而且加入了对用户的行为进行分析的功能。如果数据大量的产生，便可通过这个分析，去对用户进行一个大概方向的预判。

## 功能性需求分析

### 用户模块

1. 后台管理员登录。采用常规账号密码进行登录。
2. 新增管理员功能。超级管理员新增管理员，实现多人管理系统。
3. 管理员查看所有用户基本信息功能。可根据某些字段进行排序操作。
4. 管理员修改用户基本信息功能。修改密码、昵称、头像等的基本信息操作。

### 系统数据分析处理模块

数据库表文件与各类文件上传次数的对比功能。系统会自动分析统计并绘制出数据库表文件与其他各类文件上传次数的对比统计图。绘制图表为：条形图和南丁格尔玫瑰图等两类图表。

### 系统权限模块

管理员修改系统中被允许连接的数据源功能。例如，允许系统连接MySQL数据库、PostgreSQL 数据库，不允许连接 Oracle 数据库的功能。并且可以适当修改某个连接对象的基本信息（封面图、名称、是否禁用）。

# 系统概要设计

在本章中，作者将从宏观角度对后台开发中的系统概要设计做出如下概述。包括系统整体功能图、数据库设计、系统接口设计、Vue组件以及接口进行统一封装。

## 系统整体功能图

至目前为止，此次后台管理系统的整体功能图如下图所示：

图

## 数据库设计

整体E-R图如下图所示

图

### 用户表设计

1. 用户实体。用户唯一标识、账号、密码、角色id、居住地、性别、头像、昵称、角色、是否登录、创建时间、更新时间、是否禁用。
2. 实体属性图如图下所示。

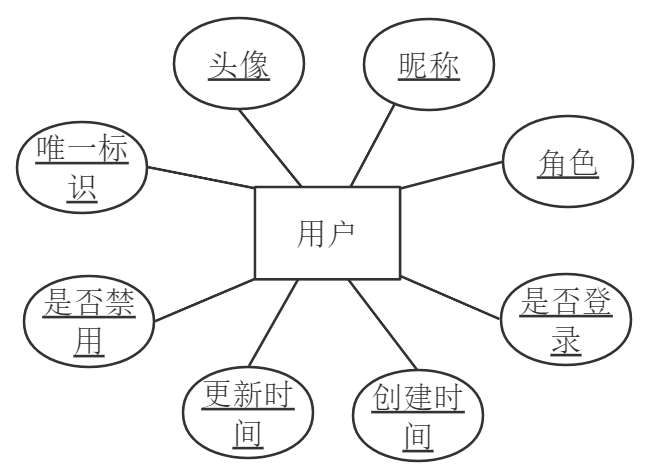


图 4.1 用户实体属性图

### 用户行为表设计

1. 用户行为实体。行为唯一标识、名称、用户唯一标识、类型、父级编码等。
2. 实体属性图如下图所示。

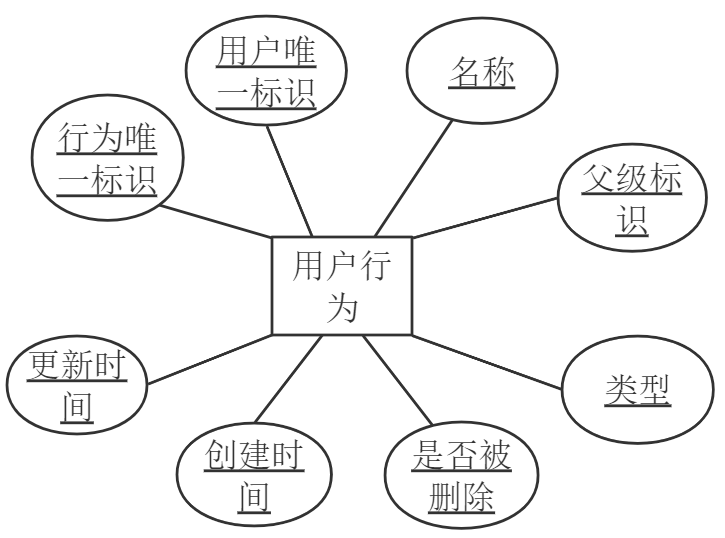


图 4.2 用户行为实体属性图

## 系统接口设计

### 用户模块接口设计

1. 后台管理员登录接口

请求参数如下表所示

图 4.3 后台管理员登录接口请求参数表

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 说明 |
| account | string | 账号 |
| password | string | 密码 |

返回参数如下表所示

图 4.4 后台管理员登录接口返回参数表

|  |  |  |
| --- | --- | --- |
| 状态码 | 状态信息 | 返回参数 |
| 1 | 成功 | 用户基本信息 |
| 20002 | 密码错误 | null |
| 20009 | 用户不存在 | null |

1. 新增管理员接口

请求参数如下表所示

图 4.5 新增管理员接口请求参数表

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 说明 |
| account | string | 账号 |
| password | string | 密码 |
| avatar | string | 头像 |
| nickname | string | 昵称 |

返回参数如下表所示

图 4.6 新增管理员接口返回参数表

|  |  |  |
| --- | --- | --- |
| 状态码 | 状态信息 | 返回参数 |
| 1 | 成功 | 用户基本信息 |
| 20010 | 用户已经存在 | null |

1. 查询用户的基本信息接口

此接口暂时不需请求参数

返回参数如下表所示

图 4.7 查询用户的基本信息接口返回参数表

|  |  |  |
| --- | --- | --- |
| 状态码 | 状态信息 | 返回参数 |
| 1 | 成功 | 用户基本信息数组 |
| 50001 | 数据未找到 | null |

1. 获取用户基本行为信息接口

此接口暂时不需请求参数

返回参数如下表所示

图 4.8 获取用户基本行为信息接口返回参数表

|  |  |  |
| --- | --- | --- |
| 状态码 | 状态信息 | 返回参数 |
| 1 | 成功 | 用户基本行为信息数组 |
| 50001 | 数据未找到 | null |

1. 根据用户连接数据库次数降序排列用户信息接口

此接口暂时不需请求参数

返回参数如下表所示

图 4.9 根据用户连接数据库次数降序排列用户信息接口返回参数表

|  |  |  |
| --- | --- | --- |
| 状态码 | 状态信息 | 返回参数 |
| 1 | 成功 | 用户基本信息数组 |

1. 根据用户上传文件数量降序排列用户信息接口

此接口暂时不需请求参数

返回参数如下表所示

图 4.10 根据用户上传文件数量降序排列用户信息接口返回参数表

|  |  |  |
| --- | --- | --- |
| 状态码 | 状态信息 | 返回参数 |
| 1 | 成功 | 用户基本信息数组 |

1. 修改用户信息接口

请求参数如下表所示

图 4.11 修改用户信息接口请求参数表

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 说明 |
| account | string | 账号 |
| avatar | string | 头像 |
| hometown | string | 居住地 |
| nickname | string | 昵称 |
| sex | string | 性别 |

返回参数如下表所示

图 4.12 修改用户信息接口返回参数表

|  |  |  |
| --- | --- | --- |
| 状态码 | 状态信息 | 返回参数 |
| 1 | 成功 | 用户基本信息 |
| 200012 | 修改失败 | null |

### 数据分析处理模块接口设计

查询数据上传次数分析接口：

1. 此接口暂时不需请求参数
2. 返回参数如下表所示

图 4.13 查询数据上传次数分析接口返回参数表

|  |  |  |
| --- | --- | --- |
| 状态码 | 状态信息 | 返回参数 |
| 1 | 成功 | 数据类型绘图对象数组 |

### 系统权限模块接口设计

1. 管理员新增数据接入类型接口

请求参数如下表所示

图 4.14 管理员新增数据接入类型接口请求参数表

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 说明 |
| cover | string | 封面图 |
| name | string | 名称 |
| isDisabled | int | 是否禁用 |

返回参数如下表所示

图 4.15 管理员新增数据接入类型接口返回参数表

|  |  |  |
| --- | --- | --- |
| 状态码 | 状态信息 | 返回参数 |
| 1 | 成功 | 数据接入对象信息 |
| 50003 | 数据已存在 | null |

1. 查询所有允许接入的数据类型接口

此接口暂时不需请求参数

返回参数如下表所示

图 4.16 查询所有允许接入的数据类型接口返回参数表

|  |  |  |
| --- | --- | --- |
| 状态码 | 状态信息 | 返回参数 |
| 1 | 成功 | 数据接入对象信息数组 |
| 50001 | 数据未找到 | null |

1. 管理员修改数据接入对象的基本信息接口

请求参数如下表所示

图 4.17 管理员修改数据接入对象的基本信息接口请求参数表

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 说明 |
| cover | string | 封面图 |
| isDisabled | int | 是否禁用 |
| name | string | 名称 |

返回参数如下表所示

图 4.18 管理员修改数据接入对象的基本信息接口返回参数表

|  |  |  |
| --- | --- | --- |
| 状态码 | 状态信息 | 返回参数 |
| 1 | 成功 | 数据接入对象 |

## Vue封装设计

在整个项目的实现过程中，全局观要强，对于一些共性的东西，可以单独提出去做成一个组件，方便后续的重复使用。

### 组件封装设计

1. 封装 HTML 代码



图 4.19 封装HTML代码图

1. Vuex中定义提示框数组



图 4.20 Vuex中定义提示框数组图

1. 设计计时器函数，使弹窗组件自动消失



图 4.21 计时器函数实现图

### 接口封装设计

与 Vue 配合使用的接口请求工具是 Axios。在项目中，会出现数次前后端数据交互的现象，那么 Axios 的重复使用必定会产生众多重复的代码。因此，将 Axios 的常用方法进行封装是必须要做的。封装的详细过程如下：

1. 创建一个 Axios 对象

生成全局的 Axios 对象，用来全局配置请求的baseURL、responseType、headers。这样做的优点在于：在将来的任何一次请求中，都会自动拼接baseURL，不仅方便全局统一切换管理URL，而且在请求接口的过程中，也能少些URL中前半部分的重复代码。以此类推，还有请求中的响应类型和请求头的设置，都可实现全局的统一设置、切换和管理。



图 4.22 Axios对象的创建信息图

1. 配置请求拦截

所有的网络请求都会先执行这个方法，我们可以在它里面为请求添加一些自定义的内容，也可减少代码的重复使用，方便全局统一管理。



图 4.23 配置强求拦截图

1. 配置响应拦截

所有的网络请求返回数据之后都会先执行此方法，此处可以根据服务器返回的状态码做相应的处理。



图 4.24 配置响应拦截图

1. 封装 get 方法

利用上文的变量 instance 变量（Axios类型），调用 get 方法，传入两个参数，一个指定接口的URL，一个指定调用的参数。

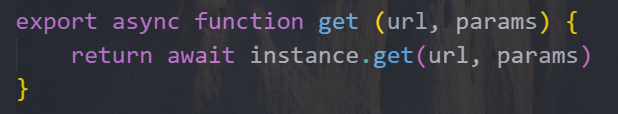


图 4.25 封装 get 方法图

1. 封装 post 方法

利用上文的变量 instance 变量（Axios类型），调用 post 方法，传入两个参数，一个指定接口的URL，一个指定调用的参数。

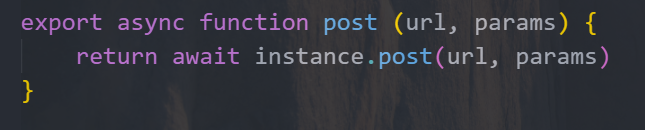


图 4.26 封装 post 方法图

1. 封装 put 方法

利用上文的变量 instance 变量（Axios类型），调用 put 方法，传入两个参数，一个指定接口的URL，一个指定调用的参数。

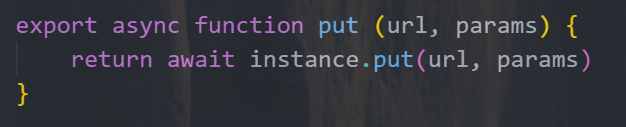


图 4.27 封装 put 方法图

至此，项目中有关前端请求接口的封装就结束了。我们在后续的使用中只需要调用此文件中对应的 get、post、put 方法即可。可以在项目中省略很多重复代码，体现出编程中的封装思想。

# 

# 系统详细设计

本章将根据上文的系统需求分析、系统概要设计，对后台管理系统中的功能按照具体的模块展开详细的描述。包括用户模块、数据分析处理模块、系统权限模块。

## 用户模块

### 登录功能设计

### 新增管理员功能设计

### 修改基本信息功能设计

## 数据分析处理模块

### 数据库表上传与文件上传次数对比功能（图表展示）设计

### 计算指定用户每周上传次数功能设计

## 系统权限模块

### 分类展示用户行为功能设计

### 根据用户上传次数排序功能设计

### 展示所有用户信息功能设计

### 监控用户上传数据量功能设计

### 控制系统数据源接入类型功能设计

# 系统功能的实现

系统功能的实现，是对一个项目的基本要求。在本章中，作者将从客户的角度出发，详细介绍各个功能的操作步骤和功能效果。

## 用户模块

### 登录功能实现

### 新增管理员功能实现

### 修改基本信息功能实现

## 数据分析处理模块

### 数据库表上传与文件上传次数对比功能（图表展示）实现

### 计算指定用户每周上传次数功能实现

## 系统权限模块

### 分类展示用户行为功能实现

### 根据用户上传次数排序功能实现

### 展示所有用户信息功能实现

### 监控用户上传数据量功能实现

### 控制系统数据源接入类型功能实现

# 系统测试

为给客户提供最佳的体验，作者对本后台管理系统做了大量测试工作，在本章中，作者将详细接受此系统中的核心功能测试结果。

## 测试环境准备

## 功能性测试

## 系统部署测试

# 总结和展望

对自己项目有很好的认知以及独特的借鉴是对自己的负责。在本章节中，作者将会进行总结以及表述出自己对当前后台管理系统的展望。

## 总结

## 展望

# 致谢

# 参考文献