编号 2017024420160234l2

南京航空航天大学金城学院

毕业设计

题　目　基于 Hadoop 的 HDFS 分布式云存储系统的设计与实现

学生姓名	桑泽寰
学　　号	2016023412
学　　院	信息工程学院
专　　业	软件工程
班　　级	20170244
指导教师	王姝懿 讲师

二〇二一年五月

南京航空航天大学金城学院

本科毕业设计（论文）诚信承诺书

本人郑重声明：所呈交的毕业设计（论文）（题目：基于Hadoop的 HDFS 分布式云存储系统的设计与实现）是本人在导师的指导下独立进行研究所取得的成果。尽本人所知，除了毕业设计（论文）中特别加以标注引用的内容外，本毕业设计（论文）不包含任何其他个人或集体已经发表或撰写的成果作品。

作者签名：桑泽寰　　2021 年 5 月 27　日

（学号）：　　2016023412

基于 Hadoop 的 HDFS 分布式云存储系统的设计与实现

摘　要

近年来，随着云计算和软件即服务（SaaS）业务的不断发展，互联网应用对高效稳定且可横向快速拓展的分布式文件存储系统的需求越来越高。针对这一需求，国内一些知名的 IDC 厂商纷纷推出了自己的云对象存储系统。本课题依托 Hadoop 的 HDFS 文件存储系统为底层，借助 Vmware Esxi 大型虚拟化平台，尝试自主设计并实现一种多地域、多节点的对象存储系统，同时与阿里云、七牛云实现容灾冗余备份。旨在帮助中小型企业，在充分利用自身已有的服务器资源，尽可能降低企业基建开销的前提下，发展并构建自己的小微型云生态。

本课题在设计和实现的过程中，充分借鉴了现阶段多种常用的高并发数据处理方案和权限约束框架，在此基础上采用前端、后端、异步进程分离的系统模型，在数据库中使用横向权限约束系统中的用户权限，借助阿里云的 DNS 解析 API，腾讯云的验证码和短信服务，对未来实际业务需求下的多语言、多模型。多框架的融合型软件系统的开发设计做出了探索和实践。

关键词：云计算，分布式，云存储，大数据

Design and implementation of HDFS distributed cloud storage system based on Hadoop

Abstract

In recent years, with the continuous development of cloud computing and software as a service (SaaS) business, the demand of Internet applications for efficient, stable and fast expanding distributed file storage system is increasing. In response to this demand, some domestic well-known IDC manufacturers have launched their own cloud object storage system. Relying on Hadoop's HDFS file storage system as the bottom layer and with the help of the large-scale virtualization platform of VMware esxi, this project attempts to design and implement a multi-region and multi-node object storage system independently, and at the same time realizes disaster recovery redundancy backup with Alibaba cloud and Qiliu cloud. In order to help small and medium-sized enterprises to develop and build their own small and micro cloud ecosystem on the premise of making full use of their existing server resources and reducing their infrastructure costs as much as possible.

In the process of design and implementation of this project, based on a variety of commonly used high concurrency data processing schemes and permission constraint framework at this stage, the front-end, back-end and asynchronous process separation system model is adopted. The horizontal permission constraint system is used in the database to restrict the user's permission. With the help of alicloud's DNS resolution API, Tencent cloud's verification code and SMS service, this paper makes exploration and practice on the development and design of multi-language, multi model and multi framework integrated software system under the actual business requirements in the future.

Key Words： Cloud Computing; Distributed; Cloud Storage; Big Data

目　录

第一章 绪论

1.1 研究内容

信息技术驱动的技术企业，通常拥有海量碎片化文件存储的需求。现阶段随着信息技术的发展，数据呈爆炸式增长，图片和视频已经逐渐成为人们记录和分享信息的主要方式，各种互联网应用，催生海量的内容与数据,预计到 2020 年底，全球数据总量将达到 44ZB，其中 80%左右为非结构化数据[1]。这些存储的文件，往往容积不是很大，且不需要进行高速的读写操作。因此用于存储海量数据文件的分布式存储系统应运而生。作为云计算底层核心基础设施,分布式数据存储系统是各种云计算服务的基础,是云计算重要的组成部分[2]。通常情况下，企业需要对这些文件的读写进行鉴权操作，且为避免本地存储分区出现异常，相关数据需要冗余备份。常规的存储系统已经很难满足这种海量文件的存储需求，因此诞生了能够自由拓展的分布式存储系统[3]。本课题的研究内容即在现有的分布式文件存储系统上，进行逻辑层面的业务分离，将实际的文件存储和文件的寻址，鉴权分离成两个逻辑通道，借助于阿里云、七牛云的多云文件备份机制，从而构建分布式的对象云存储系统。

1.2 研究意义

通过对目前主流的云计算企业的技术方案的研究和学习，本课题能够构建一个分布式的云存储系统，为企业实现碎片文件的多地域容错存储。本系统集成了多种语言和环境，从某种程度上更是对一种能够多业务分离、安全稳定的融合性软件系统的构建和探索。

1.3 开发版本迭代

1.V1.0 版本

基于 Python 开发的初始迭代版本。由于 Python 对复杂的 API 接口的逻辑控制问题，基于 Python 开发的初始化版本无法对前端传入的复杂参数进行处理，且基于 Flask 构建的 Python 后端引擎，在代码发生修改的时候需要终止当前进程，重新启动来重新加载当前项目的代码，这会使得当前用户丢失当前的工作进度，系统出现短时不可用的现象。

2.V2.0 版本

本版本放弃了使用 Python 作为后端引擎，改为使用 ThinkPHP 对其进行替代。ThinkPHP 会借助 PHP 的 Exec 函数，拉起 Python 脚本执行来自前端的，不能够被 PHP 直接处理的指令。但由于依赖 Exec 函数，系统没有设计一套异步进程处理机制，导致 PHP 经常会出现线程死锁的现象，因此废弃该版本的进一步开发。

3.V3.0 版本

基于 Redis、ThinkPHP 和 Python 开发的第三代版本。开发了一个任务处理的队列，使用 ThinkPHP 处理来自客户端的具有时效性质的高并发的及时任务，Python 做为异步进程来守听来自 Redis 中的需要异步处理的问题。此版本基本解决了系统的高性能需求，但依然存在任务队列不能够按节点分区，用户的访问权限得不到很好的控制等诸多问题。

4.V 4.0 版本

基于 Redis、ThinkPHP 和 Python 为后端底层，引入 Vue 作为前端的界面的渲染语言，对系统的前端页面进行了全面重写的第四代版本。系统的前端页面不再依赖 ThinkPHP 提供视图渲染，从后端代码中砍掉了 MVC 中的 View 视图。另一方面，系统引入了 Kafka 消息队列，不再使用原有的 Redis 伪消息队列，通过配置 Python 监听 Kafka 中不同分区的数据，实现不同节点、不同任务的后端业务的分离处理。针对 3.0 版本中用户权限不能很好的控制的问题，4.0 版本引入了基于横向权限的新业务框架，系统可以对用户权限进行很好的控制，还可以在系统后台动态的实现对用户权限的修改操作。

1.4 开发环境

1.4.1 服务器环境及各组件版本列表

表 1.1 服务器环境及各组件版本列表

环境/组件名称	版本
CentOS	7.9.2009 x86_64
宝塔面板	7.5.2
Nginx	1.19.6
MYSQL	8.0.20
Redis	6.0.9

表 1. 1 [续]

环境/组件名称	版本
Kafka	2. 8. 0
Hadoop	3. 3. 0
PHP	7. 4. 16
PYTHON	3. 7. 8

1. 4. 2主要开发环境组件介绍

1. 4. 2. 1 后端 API 主语言 PHP

PHP （Pre Hypertext Preprocessor）是一个被广泛使用在各种服务器上的服务器端执行的脚本语言[4]。目前，除 PHP 外，网站后端主要使用 Java 开发。相较而言，使用 PHP 构建的项目不需要进行编译。其开发成本低，执行速度快，可移植性很好，内置了大量丰富的函数库，本课题使用 PHP 7.4.16 作为后端主语言，主要用于处理前端界面传入的不需要很长时间处理的 API 请求，通过与消息队列的巧妙配合，PHP 会将前端传入的长处理任务转发到 Python 中进行解析和处理。

1. 4. 2. 2 异步处理辅语言 Python

Python 是一个非常受欢迎的解释型语言。使用 Pyhon 编写的程序，不仅代码结构优雅，后期阅读维护方便，且开发过程简单，高效。其强大的多线程处理机制，被广泛应用于独立的、大型的项目开发设计中。在本课题中，Python 主要从 kafka 接受来自 PHP 发送的异步处理的消息，并借助本地的线程池对消息进行处理。

1. 4. 2. 3 关系型数据库 MySQL

MySQL 是目前在生活中最常见也是最频繁使用的数据库系统。与目前主流的数据库系统相比，MYSQL 开源、廉价的特性使得其在各种系统中被广泛使用。本项目主要使用 MYSQL 8.0 存储需要长期保存的业务数据，通过 MYSQL 的 Slave 和 Master 机制实现数据业务的读写分离和容灾备份。通过建立索引来大幅度的减少数据的查询时间[5]。

1. 4. 2. 4 NOSQL 数据库 REDIS

Redis 是一种常见的 NoSQL 类型的数据库，其特点是常驻内存，且读写速度快。但其存

在逻辑搜索的索引缺陷。本项目中主要使用 Redis 作为系统临界缓存使用。因为本项目中，用户的身份信息需要在多台服务器和多种语言之间调用，所以本项目中将 Redis 虚拟化成了系统的 Session 存储池，用来跨域存储用户的 Session 数据。

1.4.2.4 消息队列 KAFKA

Kafka 是一种常用的消息队列，其分为消费者和生产者两个部分。本项目中，作为生产者的 PHP 将需要异步处理的任务信息发送到 Kafka 中，在系统后台运行的异步消费 Python 程序，会根据当前 Python 的 线程队列的情况从 Kafka 中读取 指定条数的信息进行消费。

1.4.2.5 堡塔面板

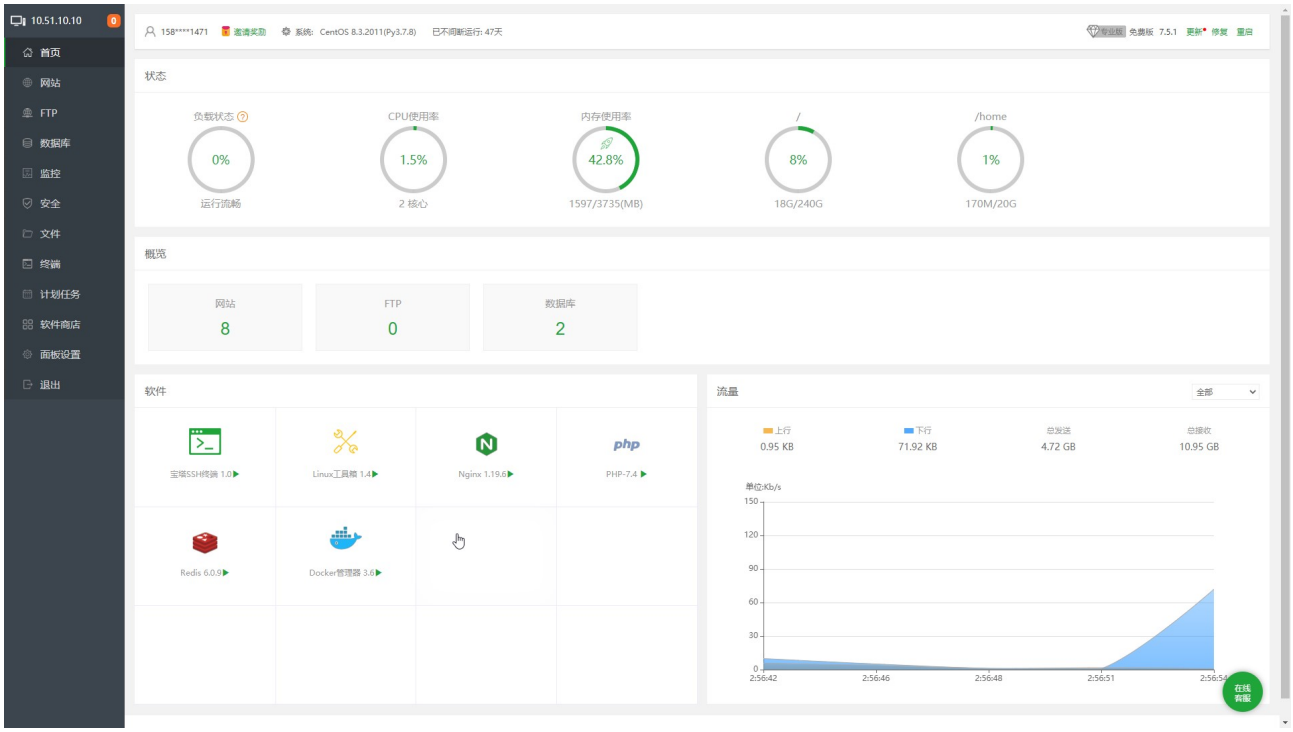

图 1.1 宝塔面板控制页面

本课题在开发过程中，主要使用了堡塔服务器面板对服务器进行基本的操作，运行维护配置。相较于使用图形界面的 Windows 系统而言，本课题使用的无图形化界面 CentOs 系统性能更加稳定，且系统底层占用的性能比较小。部署堡塔服务器面板后，通过堡塔面板的 API 接口功能，后端 API 接口可以通过 POST 访问面板指定 API 接口的形式，对面板中运行的网站，数据库，防火墙，应用程序进行管理和配置。如当用户在本系统中创建新的存储桶资源时，后端 API 接口会调用面板的相关接口，自动创建新的网站，配置相关网站的域名，

SSL 证书等信息，实现通过 API 接口动态管理服务器。图 1.1 和 图 1.2 展示的是宝塔面板的控制页面和面本的 API 功能接口的配置页面。配套用户前台中提供的 SOV 集群管理功能（详见第三章第三小节），开发人员可对运行当前课题的服务器群组进行快速高效的批量管理。

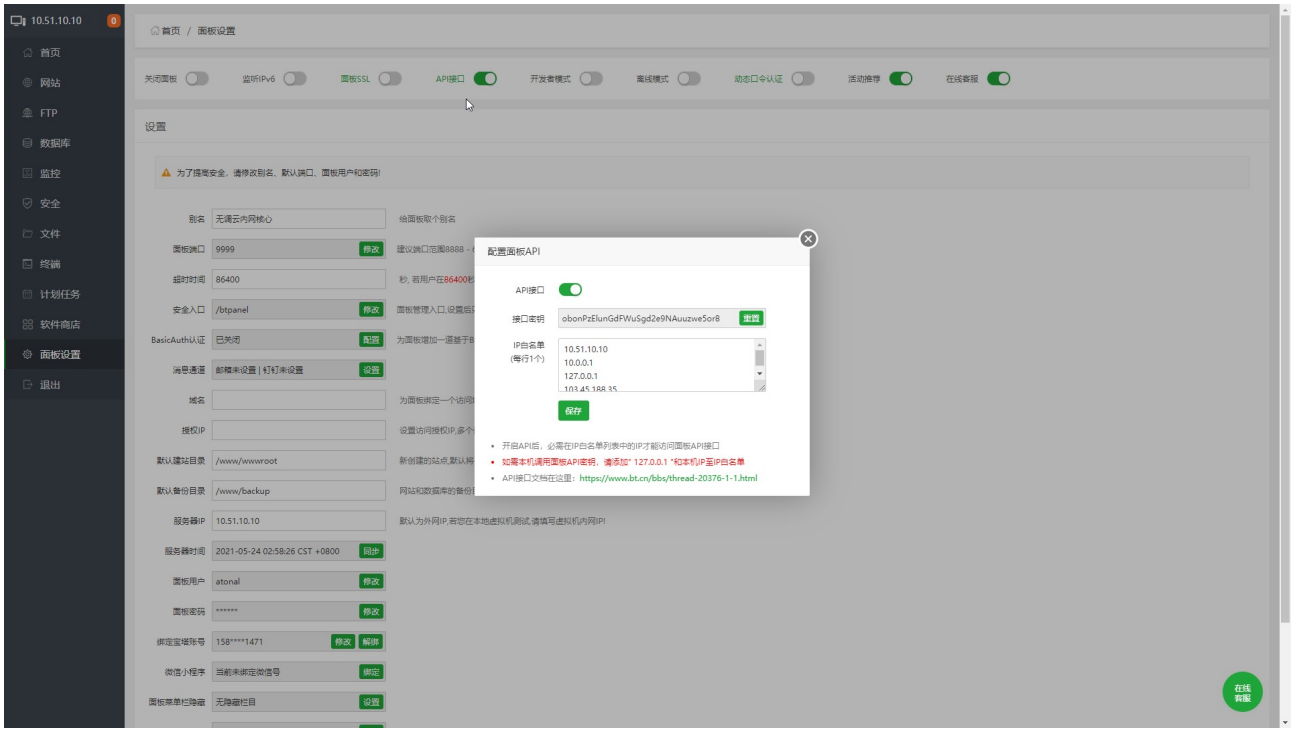

图 1. 2 宝塔面板控制页面

1. 4. 3 Coding DevOps

本课题在开发过程中，主要使用腾讯提供的 Coding DevOps 一站式开发平台进行协同开发。如图 1.3 所示，本课题有两个制品库组成，每个制品库中有多个不同的代码仓库，分别是 前端制品库（前端界面仓库，前端调试工具仓木）、后端制品库（API 接口仓库，对象存储系统仓库，Python 异步消费者仓库）。图 1.4.展示的是项目在开发过程中，提交和推送的迭代日志信息。系统开发过程中，设计的相关的 API 接口的参数和文档则发布在语雀平台中，详见附录《无调云计算平台 API 接口文档 V4. 0》。

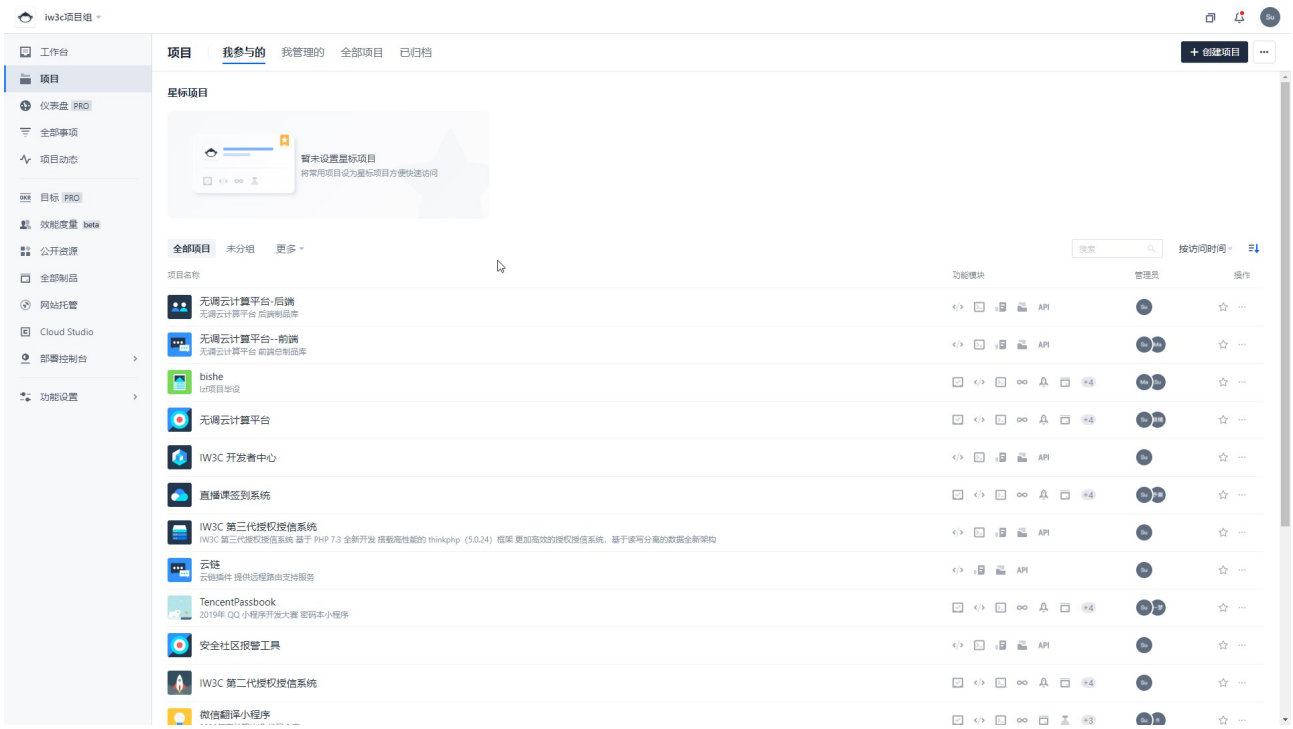

图 1. 3 Coding 制品仓库

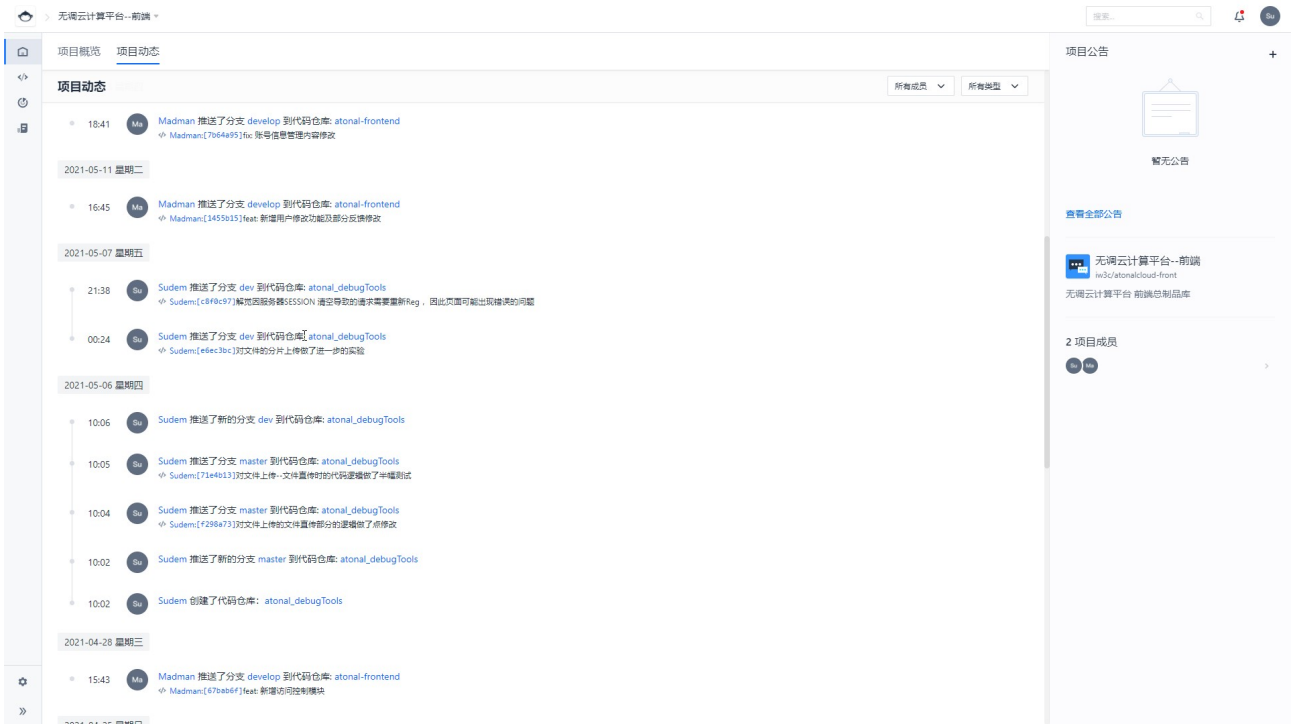

图 1. 4 Git 迭代日志

1.5 开发难点

1. 异步进程处理

和常见业务系统的逻辑不同，本课题中设计的云计算系统需要及时处理大量的数据，如请求日志，文件同步上传等。这些请求通常需要花费数秒来完成。如果将数据处理的过程加到用户的前端请求中实现，就会造成前端的 API 请求过程拥堵，用户需要等待 ThinkPHP 将消息处理结束才能继续执行新的操作。因此，本课题研发了一种能够和 PHP 这种服务端解释型语言进行交互的异步的任务处理程序，通过 Kafka 进行任务的调度，在不阻塞前台运行的情况下异步处理数据，并可根据系统的负载状态动态的调整异步进程的数量。

2. 系统架构设计

目前，互联网信息系统中主要使用由模型，视图，控制器三个部分组成的 MVC 设计框架。与常规设计模型不同的是，本系统采用了 前端、后端、异步线程三个模块分离的系统架构，采用了多种不同的开发语言进行互相配合。系统模块多，功能强，并发性和稳定性好。

3. 网络拓扑的搭建

系统基于 OpenVPN 构建了一套多节点之间虚链路隧道。各节点之间的出口网络使用 IKuai 软路由互联。在基于 IKuai API 的基础上，研发了一套可以在 OpenVPN 直接自动学习路由的动态路由算法。

4. 弹性业务底层

系统的业务底层可以弹性拓展其它功能和组件。系统的核心架构面向未来，可以为在此基础上开发的其它业务做支撑。

1.6 技术架构

1.6.1 前、后、异步分离的组合型工作模式

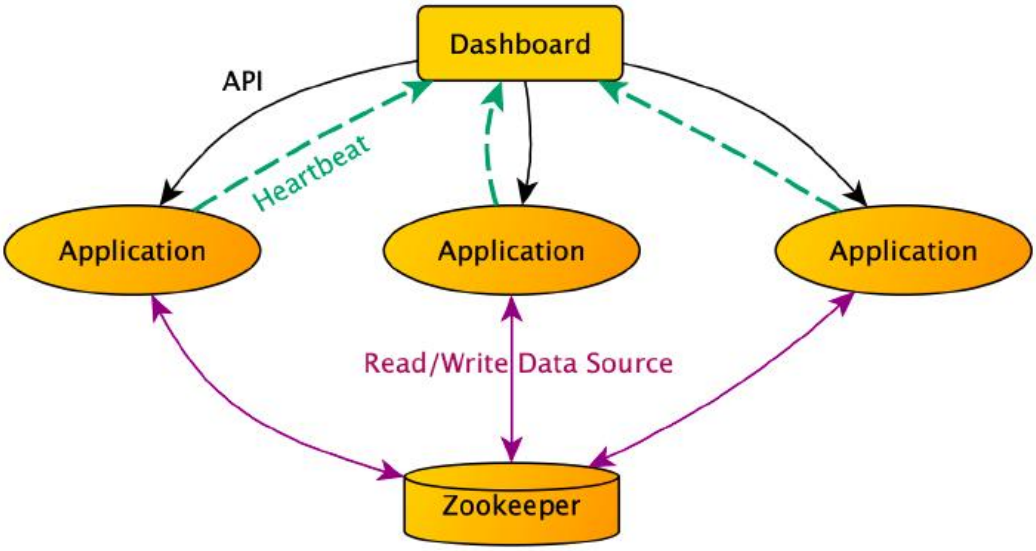

图 1.5 常规 MVC 设计模型

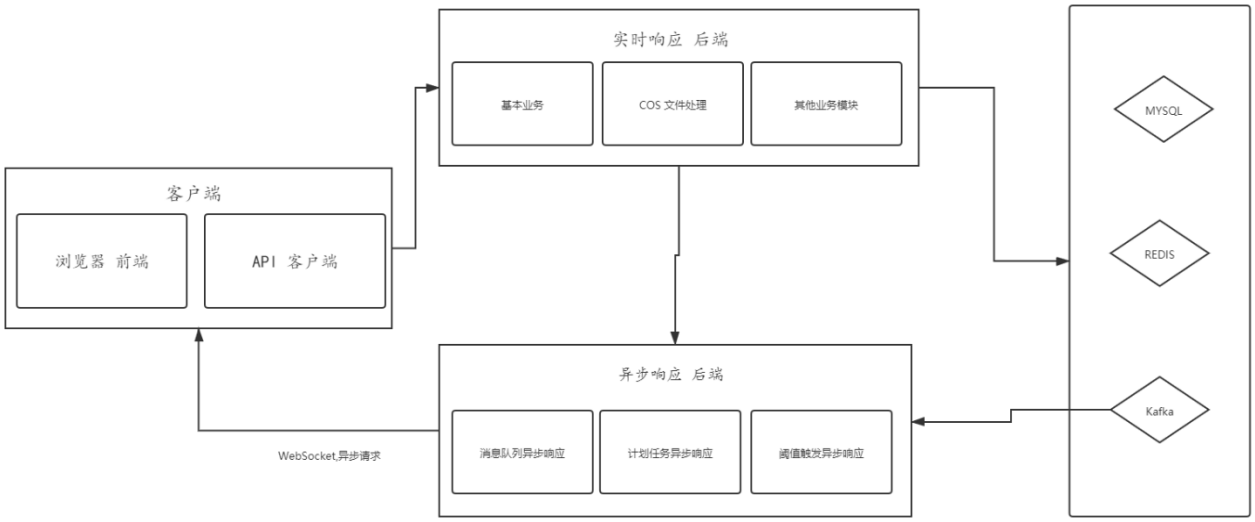

图 1.6 前、后、异步进程分离的组合型工作模式

和常规 MVC 设计模式（图 1.5）由模式-视图-控制器三部分组成不同[6]，如图 1.6.所示，本项目使用的设计模式，将前端，后端，异步分为三个不同的部分，每个后端程序中都

只由模式-控制器两个部分组成，只对逻辑层的应用进行处理，不再渲染视图模型。对视图模型的渲染转交到客户端自行进行渲染。这种设计模式的大大提高了系统的可靠性，系统的各个核心模块均相互对立存在，相互独立运行。即使某个模块出现了故障，不能正常运行，用户也仅会感知到部分功能不可用，不会导致整个系统功能的全部失效。

1.6.2 横向数据权限

在常规的系统中，对数据权限的处理通常有横向和竖向的数据权限两种管理方式。竖向的数据权限，需要对数据进行逻辑的级联，这种鉴权方案可以对系统中的业务进行更加细致的管理，但在查询数据的时候，需要从权限表中去查询对应的权限情况，只适用于数据量不是很多的系统。横向的数据权限则将数据的权限控制直接写在对应的数据行中指定的字段中，系统直接从读出的数据行中字段中去处理数据。这种方式对数据的控制面可能不是非常的细致，但对系统数据库的逻辑级联比较少，适合数据业务量比较的大，但对鉴权要求不是非常高的系统。综合考虑，本项目中的业务数据较多，为了减少数据库的复杂，本项目使用横向数据权限对数据进行处理。

1.6.3 异步消息队列

在高并发的业务系统中，常常有很多需要在后台异步处理的任务。这些任务通常不需要实时的响应，且需要花费较长的时间去完成。因此诞生了异步消息队列的说法。其本质是一个存在于服务器上的队列，在不同的程序之间，可以按照实际需求向这个队列中压入数据（队尾），也可以从队列中取出数据（队头），根据对数据队列的操作方向的不同，将负责插入数据的程序称为生产者，将负责读出数据的程序称作消费者。与常规的队列操作不同的是，消息队列有接受者的逻辑分离，也就是说一条消息可以被多个消费者同时消费，也可以仅被单一消费者消费生成者在生产消息的时候可以设置接收消息的消费者，从而实现不同人物的不同程序执行。

1. 6. 4 RSA 双向非对称加密

RSA 非对称加密算法，是目前业界中最常用、最安全的加密算法之一。本算法由一对数字密钥组成，每组数字密钥之间存在固定的级联关系，不可随意更改。通常情况下，服务器和客户端各持有这对数字密钥的一部分。为了方便称呼，将存放在服务器上不公开的密钥称作私钥，将下发给客户端或公布在互联网上的部分称作为公钥。如图 1.6.4 所示，当本系统

的客户端或 API 程序需要和服务器进行通讯时，客户端或 API 程序首选会从自己内存中读取当前使用的密钥对的版本信息。客户端使用密钥对中的公钥对需要传出的数据进行加密，然后将当前使用的密钥对的版本（特征值）发送给服务器。服务器接收到客户端发送的加密消息后，首先会从数据库中通过特征值取出对应的密钥对，然后用该密钥对的私钥对数据进行解密。服务器进行响应数据时，也会将当前所使用的密钥对的特征值，使用密钥对的私钥加密后的数据发送给客户端，客户端必须持有对应密钥对的公钥才能对数据进行解密操作。[7]。通过这种算法的配合，确保客户端发出的数据，在数据传输过程中，不可读取，更加不可能进行修改。且从服务器响应的数据，在不知道公钥的情况下也无法读取。为了确保安全，项目使用的 RSA 密钥均使用标准由赛门铁壳签发的 SSL 数字证书，且跟随数字证书的有效期定期更换。

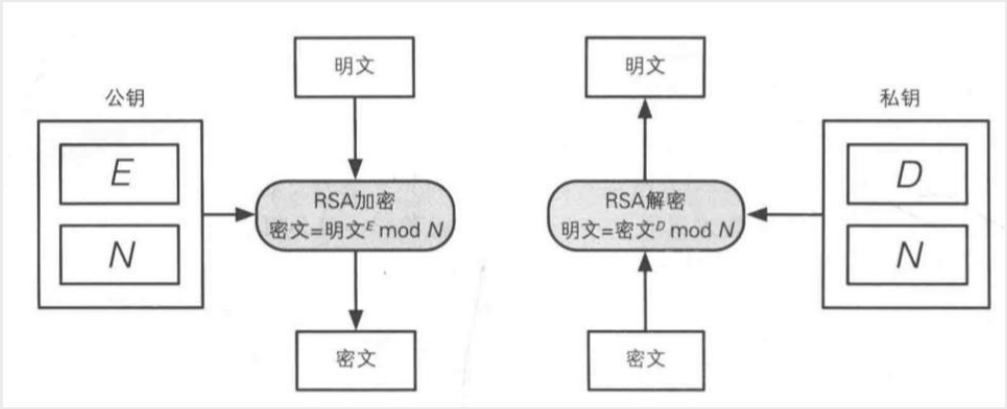

图 1.7 RSA 双向加密过程

1.6.5 基于 HDFS 的分布式文件存储

HDFS 分布式文件系统，是 Apache 基金会的 Hadoop 分布式系统基础架构中一个组件[8]。其本质是一个高度容错的分布式系统，当 HDFS 运行时，会为数据块创建多个副本，从而使得数据能够在不同集群中进行快速且可靠的数据访问[8]。HDFS 系统主要由名字节点（NameNode）、数据节点（DataNode）、数据块（block）这三种元素组成[8]。名字节点（NameNode）主要负责管理文件系统命名空间，调整客户端访问文件的权限，同时决定了数据块到数据节点的映射关系(图 1.8 所示为 HDFS 自带的 WEB 控制台中的管理界面)，

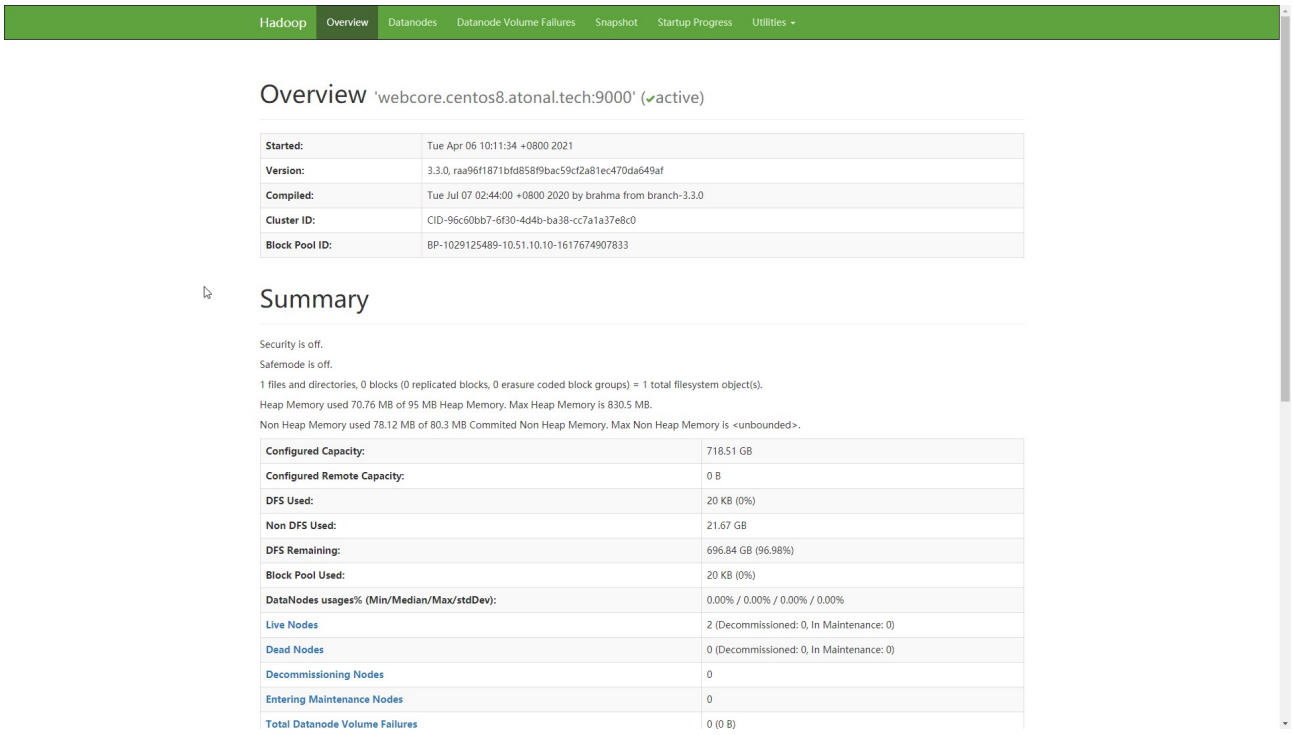

图 1.8 HDFS 控制面板（NameNode）

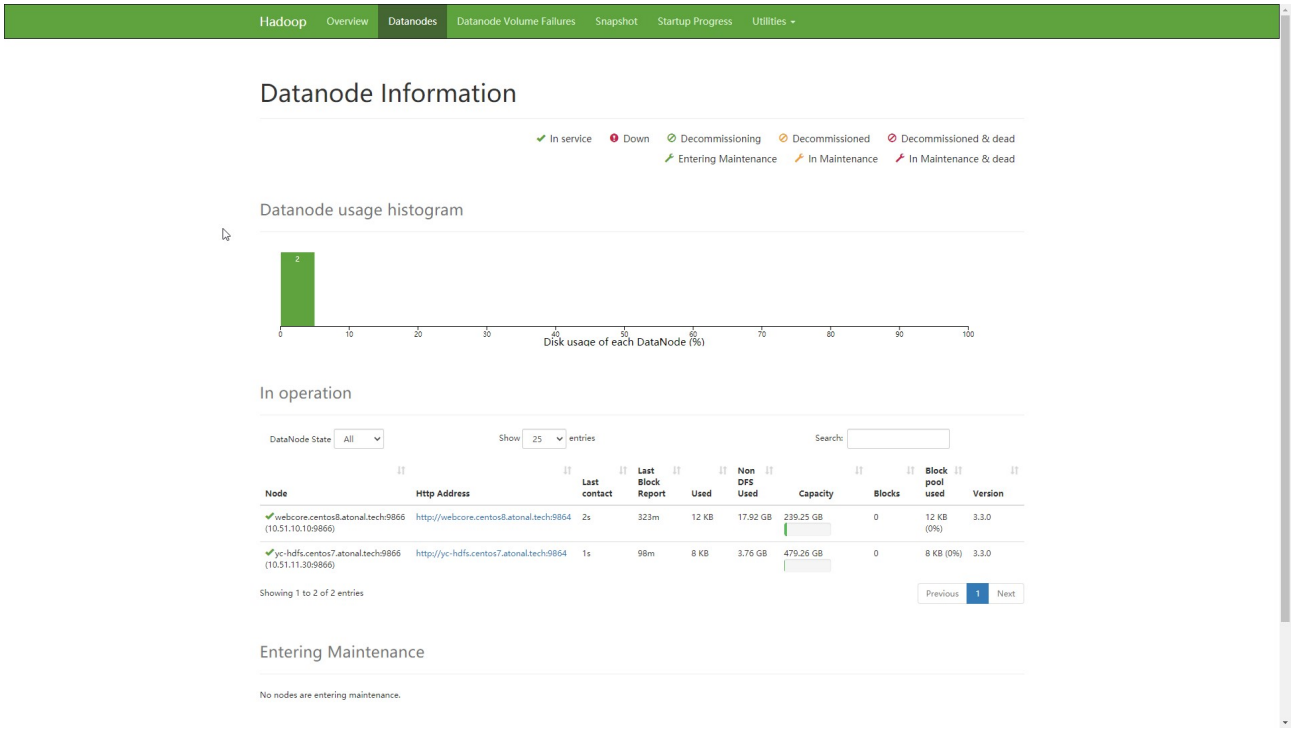

图 1.9 HDFS 控制面板（DataNode）

数据节点（DataNode）(图 1.9)为名字节点的从节点，负责在客户端请求时执行文件系

统的读写操作[8]。数据库（block）是 HDFS 读写数据的基本单位[8]。HDFS 系统中的文件会被划分成一个或者多个端，每个段单独存储在数据块中[8]。

1.6.6 对象文件存储

在常规的文件存储系统中，文件的逻辑位置（包含读取权限）和文件的实际数据位置之间是一个并行的关系。即文件存储系统既需要负责对文件的逻辑位置的索引判断，同时也需要负责文件的实际数据位置之间的读取和写入操作。对象文件系统则是对这种常规文件系统的一种改进。文件的逻辑位置和文件的数据位置被分成两个独立的模块。即针对文件的逻辑操作，由逻辑服务器（逻辑程序）独立的完成，此过程中，逻辑服务器并不关心文件的实际存储位置，即使数据服务器发生了故障或部分故障，也不会影响逻辑文件的索引。同样数据服务器也不关心文件的逻辑权限问题，其只负责对数据的读取和写入。在这种新结构的文件存储系统中，每一个文件均为一个数据对象，均可独立的对齐进行权限控制操作。

1.7 网络拓扑

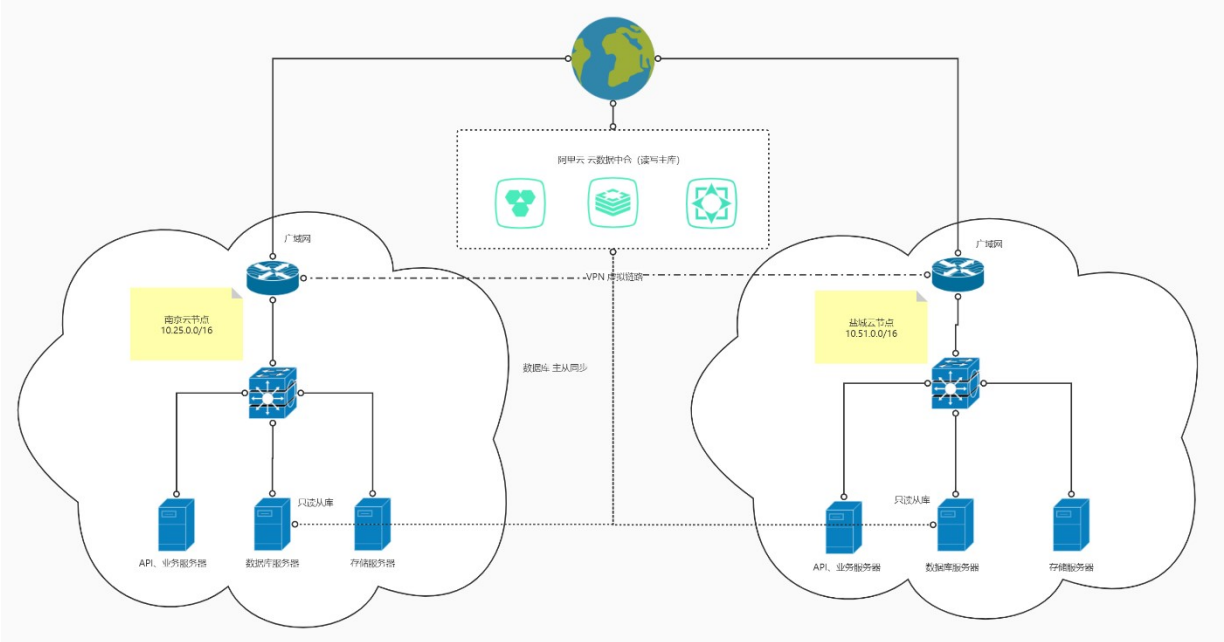

图 1.10 系统网拓扑结构图

图 1. 11 IKuai 智能软路由

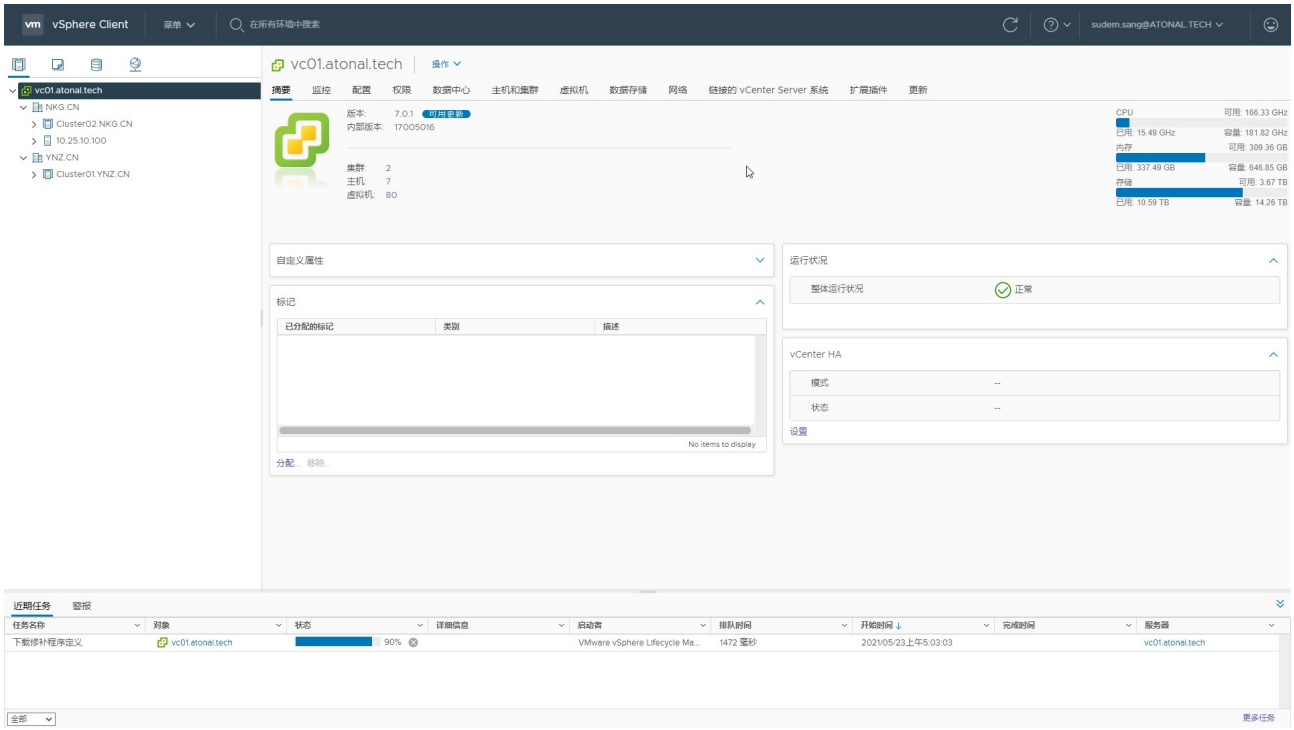

图 1. 12 VCenter 虚拟化管理中心

如图 1.10.所示，系统主要由盐城，南京两大数据中心组成，各数据中心之间通过 OpenVPN 配合 IKuai 智能软路由（图 1.11 所示） 互联互通。各节点中的数据中心均有多台使用 VSphere Client 虚拟化的服务器组成，这些服务器连接到 VCenter 中，由 VCenter 进行统一的集中控制管理（图 1.12 所示）。每个数据中心节点中均拥有自己的独立的数据从属服务器，用于系统中数据的高性能读取。数据中心服务器与暴露在公网中的云上数据中心级联，当服务器发生数据写入时，会向处于公网中的数据中心发起写入请求，数据中心服务器处理数据更新后，通过 MySQL 的 SLAVE 主从备份机制，更新写入的数据到各分部的数据中心节点中。每个数据中心节点中均存有云上数据中心数据的备份（仅 MYSQL 数据），当云上数据中心发生灾难时，系统可自动选举新的云下数据中心成为容灾节点，继承云上数据中心的职责向系统中的其他数据节点提供同步读写分离业务。

第二章 系统设计

2.1 需求分析

2.1.1 系统安全需求

现阶段，网络安全问题日益突出，如何安全稳定的存储用户的数据， 是所有互联网软件系统开发时不可避免的难题。本项目试图构建的云存储系统，是用户在日常生产生活中不可避免的重要组成部分。根据文献[7]可知，现阶段互联网中主要存在的漏洞/安全威胁主要为(1)SQL 注入，(2)失效的身份认证，(3)敏感数据泄露，(4)XML 外部实体，(5)失效的访问控制，（6）安全配置错误，（7）跨站脚本（XSS），（8）不安全反序列化，（9）使用已知的漏洞组件，（10）日志和监控不足。这就系统在设计时，必须能够对基于互联网络传输的数据进行加密处理，系统必须要具备识别并屏蔽上述常见安全威胁的能力同时系统还因考虑的社会工程学的原理，对一些常用的社工渗透技巧能够进行有效的防范。

2.1.2 高并发处理需求

作为云存储系统，随着用户业务量的增长，对存储系统的业务调用频率也会呈指数的提升。这就要求系统需要有处理高并发并行请求的能力，且系统的代码逻辑要简单高效，同时系统的模型接口可以跟随用户的请求量的拓展进行弹性的扩展和缩减。系统能够自动根据用户请求的业务层次不同，对请求的任务做分级响应。高优先级别的请求及时响应，低优先级别的请求置入后台队列异步响应。

2.1.3 动态权限配置需求

在用户的实际生产生活中，通常会对数据文件的存储拥有多种不同程度的数据权限的控制要求，这些要求并非一种定制化的原则，即用户通常会按照自己的意愿去配置用户的业务权限。这就要求系统在设计时必须具备动态权限配置和控制的能力。系统不仅能够针对不同的用户赋予不同的请求权限，同时还得使得这些被赋予的权限能够根据用户自己的意愿去个性化动态的配置。

2.1.4 多地域分布式需求

由于运营商网络的因素影响，用户存储在系统中的文件在实际的业务请求中，通常会出现传输速度慢，或者请求时间长，请求不稳定等问题的困扰。这就要系统在开发设计的时候必须要有支持多地域分布式存储的能力。系统存储的数据能够在多个不同的地域中进行部署，根据用户的实际业务需求需要，能够让用户能够选择最佳的运行地域。由于单机设备的存储空间受到 BIOS 主板和服务器空间因素的影响，所以需要系统能够对文件进行分布式的存储，能够动态去增加存储总量。

2.1.5 高可用容灾冗余需求

云计算时代，稳定是第一生产力。系统的核心业务必须具备一定的冗余容灾机制。如当部分存储节点因各种不可预计的突发因素发生故障的时候能够及时的容灾冗余，采取一系列的技术手段，以保证用户的正常业务请求。

2.1.6 面向未来的拓展支持需求

第三次工业革命以来，信息技术的发展日新月益。用户在未来可能会有越来越多新的功能需求。为了最大限度的减少用户的成本，尽可能的利用现有的资源，这就要求系统在设计的时候能够未雨绸缪，为用户可能存在的新需求留下拓展的空间。能够在最小修改现有资源，代码的基础上最大化的支持各种新功能的加入。

2.2 系统模块

2.2.1 系统前台模块

系统前台模块为系统基本信息的展示模块，主要是对系统中相关产品信息的展示，以及产品活动的介绍推荐。主要由菜单列表，首页产品信息列表，首页轮播图列表，首页页脚合作商 Logo 信息列表这四个部分组成。相关展示信息的内容可以在系统的业务中台模块中进行动态的修改和配置。在下一个阶段，系统前台模块还将能够对系统中的产品的具体操作文档进行统一的展示，用户可以在查阅相关的技术文档后，基于系统的 API 自行开发应用。

2.2.2 用户前台模块

用户前台页面是用户在登录以后对系统中的业务逻辑进行修改和控制的界面。用户可以在该模块下对当前用户账户中所开通的产品业务进行进一步的管理和控制。以本项目的融合云存储系统为例，用户可以采用前台页面的融合云存储菜单对存储资源的存储桶进行配置。

既可配置存储桶所拥有的访问权限和同步策略，也可以对存储桶的访问域名、前端请求时对外暴露的 header、允许的请求方式、请求 header，以及跨域资源和 Reffer 防盗链等进行配置和管理。

2.2.3 业务中台模块

业务中台模块是系统真正意义上的后台管理模块。该模块仅管理人员可以访问。管理员可以在系统的业务中台中对系统中注册的用户信息进配置。如可以指定系统中每一用户所属的用户角色信息，指定不同的用户角色可以访问的模块、接口、路由菜单等具体的权限，配置系统在运行时所依赖的环境变量，修改系统允许直接从前端访问的域名地址信息，修改系统的前台模块中展示的内容。

2.2.4 前端调试工具

前端调试工具模块是一个独立的模块。系统开发人员和其它开发者可借助前端调试工具调试加密请求，对系统的逻辑进行测试。同时可查阅系统在运行过程中产生的请求日志和错误日志，并对错误信息进行评估和解决问题。开发者还可以在调试工具中融合云存储系统所支持的分片上传和文件直传功能。

2.2.5 服务器集群控制工具

如图 2.1 所示，基于宝塔面板的 API 开发的服务器集群控制工具可以很直观清晰地监控集群中所有服务器的运行状态信息，并能借助塔面板对运行异常的服务器进行管理和控制，方便管理员及时排查问题和处理错误。当系统后台监控到服务器运行异常时，会自动向用户发送设备运行异常的邮件告警信息。本系统支持使用无调云的账号直接登录，也可以在无调云系统中，通过单点登录到系统。

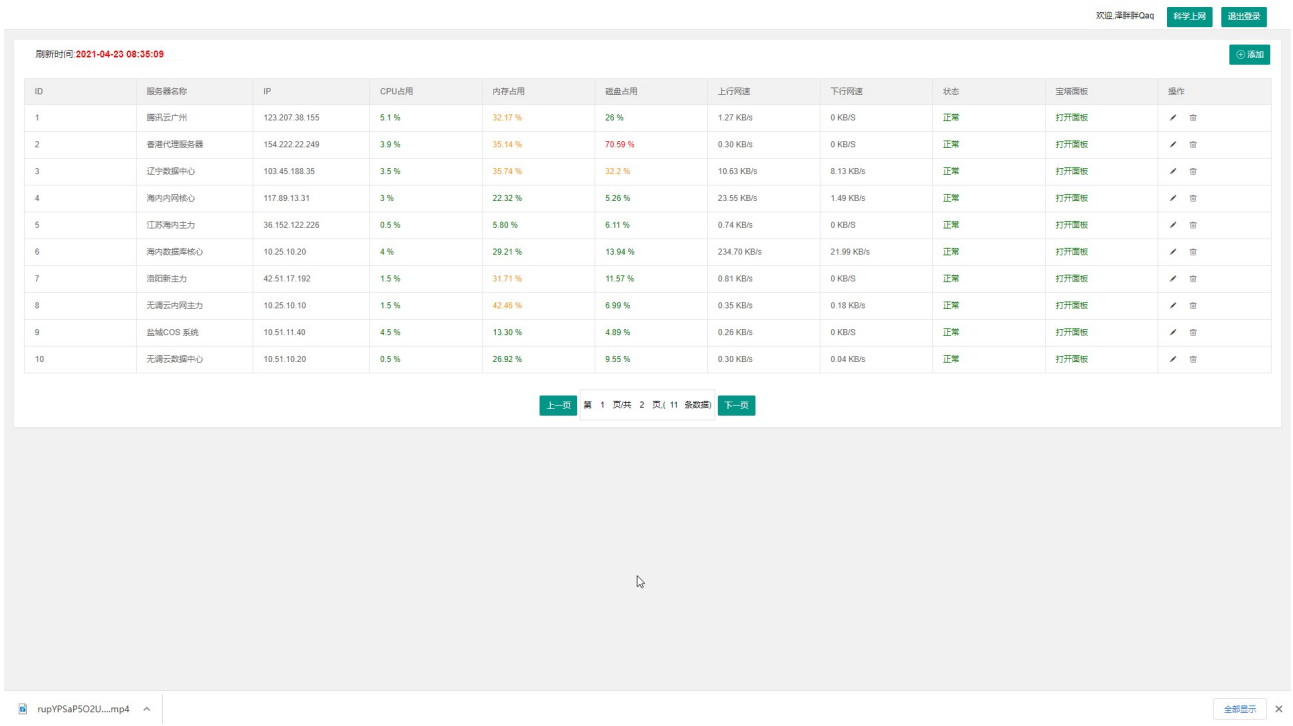

图 2.1 服务器集群控制工具

2.3 数据库设计

2.3.1 NoSQL 数据库 Redis

非关系型 NoSQL 数据库 Redis 常驻系统内存。和关系型数据库 MySQL 相比，Redis 读取速度快，可设置数据存储时间，且提供多个独立数据分区。如图 2.2 所示，本课题 Redis 中主要存储系统在运行过程中产生的一些临时数据，和当前已经登录的用户的会话缓存资源。图中展示的是当前已经登录到系统中的用户的会话信息，主要内容为当前用户的 UserId、 登录地址、个人基本信息等。

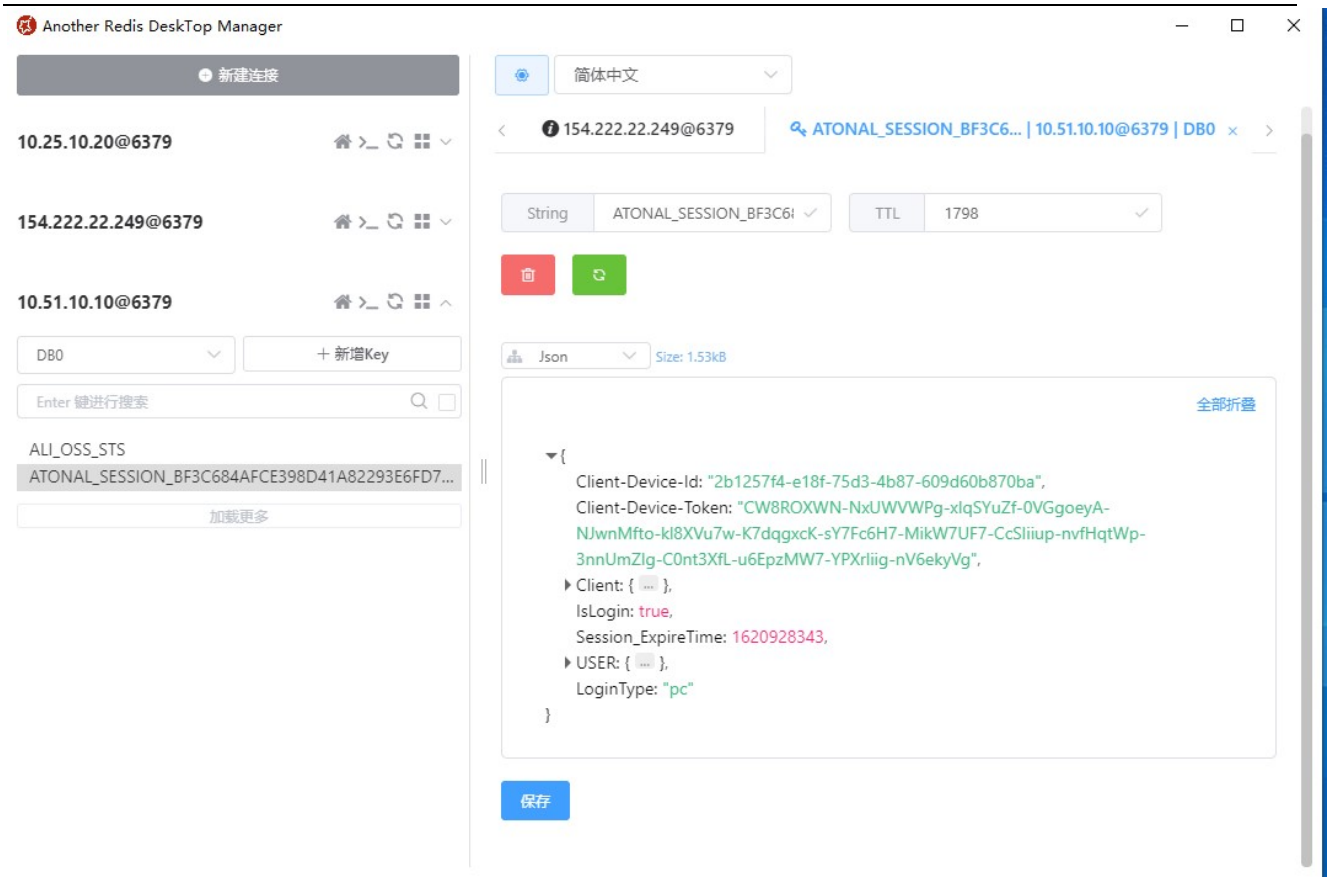

图 2.2 Redis 数据库存储的用户 SESSION 数据

2.3.2 关系型数据库 MySQL

如图 2.4 所示，MySQL 中存储了系统在运行过程中所需要使用的日志、用户及业务信息。主要分为以下四类：1.用户账户类；2.系统基础表；3.临时缓存表；4.具体业务表。图 2.3 主要展示了这四大模块中的表接口的基本关系图。参考第一章的第 7 小节所述，本系统中的数据库采用主从读写分离方案。当应用程序有写入的需求是，会自动请求位于阿里云端的数据中仓（主库）进行读写，主库通过 BinLog 日志的形式将需要同步的操作同步到各数据中心中的 SLAVE 从节点中去。当应用程序有读出请求时，则查询本地的低延迟从库，不仅能够减少对磁盘的同步读写 IO 请求，还可以大大应用层到数据中心之间的网络传输数据延迟。

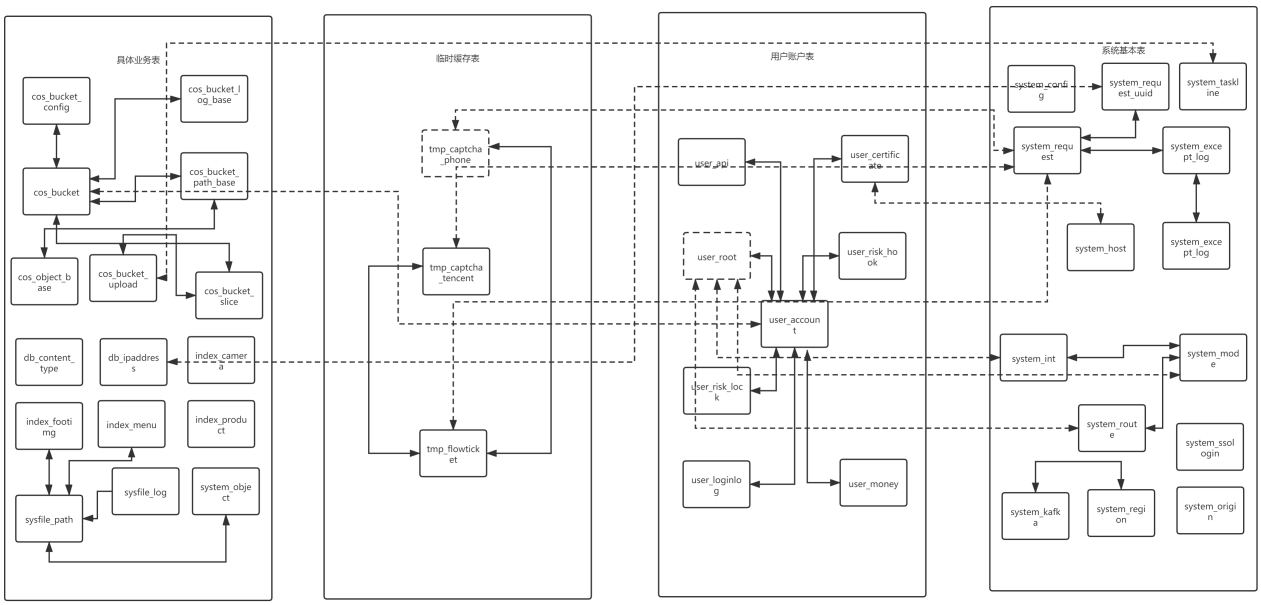

图 2.3 数据库表结构关系图

下面具体介绍每张表的字段和用途。

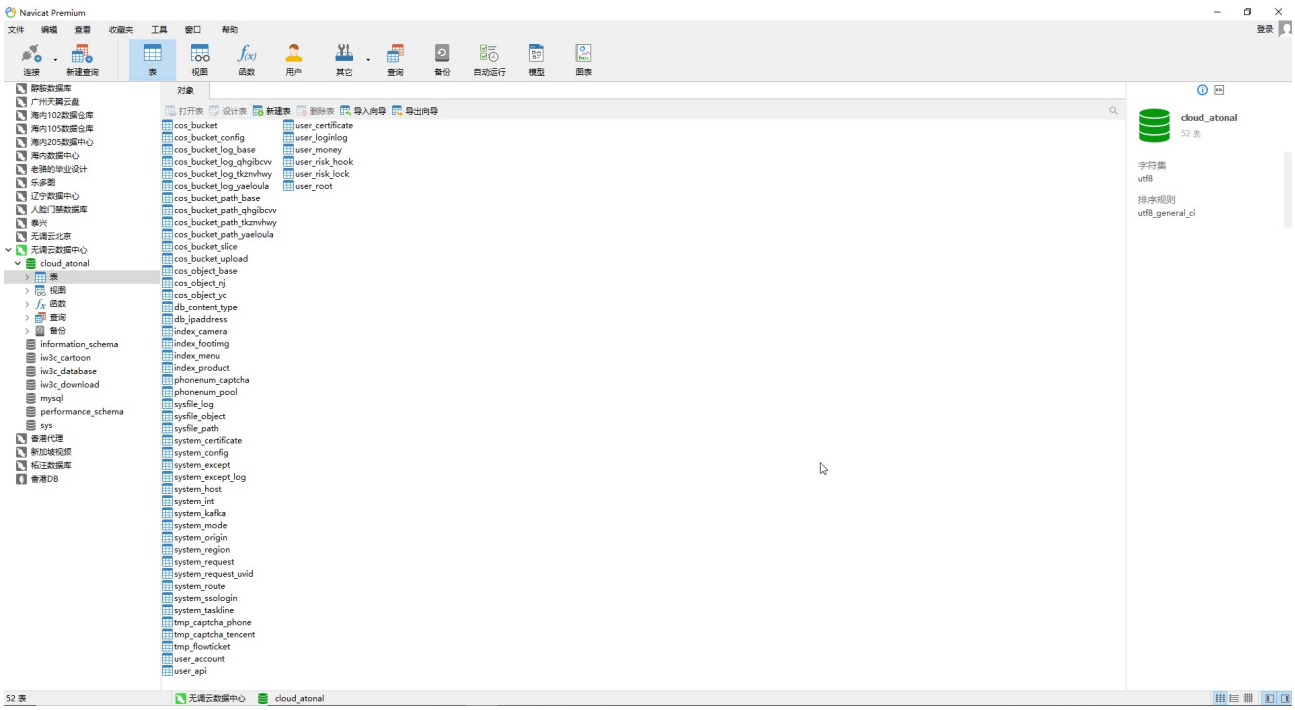

图 2.4 MYSQL 数据库中的表

2.3.3用户账户表

2.3.3.1 用户信息表

存储系统中所有用户的基本信息，比如用户所被分配的角色 Id 、加密存储后的用户密码，用户密码加密参数等，具体信息如表 2.1 所示：

表 2.1 用户信息表

字段名称	数据类型	说明
Id	int	
UserId	int unsigned	用户系统唯一 Id
UserUnionId	varchar(64)	用户联合身份 id
NickName	varchar(32)	用户的昵称
PassWord	varchar(255)	用户的密码
PassToken	varchar(255)	用户的密码加密 Token
IconUrl	varchar(255)	用户头像图片的地址
MailAddress	varchar(64)	用户的邮箱地址
TelPhone	varchar(20)	用户的手机账号
RegTime	varchar(24)	用户注册时间
UserRootAuth	varchar(32)	用户根权限 Id
AccountStatus	varchar(32)	用户账号的状态信息

2.3.3.2 用户 API 信息表

存储系统中对应不同用户的 API 密钥信息，主要信息如 表 2.2 所示：

表 2.2 用户 API 信息表

字段名称	数据类型	说明
Id	int	
ApiUuid	varchar(64)	API 的唯一 Id
UserId	int	所属的用户的 Id
ApiAppToken	varchar(128)	API 密钥对的 TOKEN
ApiAppSerect	varchar(128)	API 密钥对的 SERECT
ApiAccessRole	varchar(255)	API 扮演的 API 角色（预留字段）
ApiStatus	varchar(16)	API 密钥的状态信息

表 2.2[续]

字段名称	数据类型	说明
updateTime	int	API 密钥的修改时间
user_certificate	int	API 密钥最后一次使用的时间

2.3.3.3 用户上传的证书表

存储系统中用户上传的 HTTPS 证书信息，用于用户需要部署 HTTPS 业务的中，主要信息如表 2.3. 所示

表 2.3 用户上传的证书表

字段名称	数据类型	说明
Id	int	
certificateId	varchar(64)	证书的唯一 Id
serialNumber	varchar(255)	证书的编号
certificateSha	varchar(255)	证书的 sha 值(对公钥进行 sha256 摘要)
certificateProduc	varchar(255)	证书品牌名称
outSignDate	int	证书有效时间
Host	varchar(255)	证书所对应的域名的信息
pemFile	text	证书的 pem 文件内容
privateKey	text	证书的 私钥 文件的内容
userId	int	所属用户的 Id
updateTime	varchar(30)	证书添加到系统的时间

2.3.3.4 用户登录日志表

存储系统中用户登录请求的日志，主要信息如 表 2.4 所示

表 2.4 用户登录日志表

字段名称	数据类型	说明
Id	int	
LogId	varchar(255)	日志 id
LogTimeStr	varchar(255)	日志生成的时间，字符串
LogTime	int	日志生成时间戳

表 2.4［续］

字段名称	数据类型	说明
UserId	varchar(255)	用户的唯一 id / 用户名不存在时为用户名
AppAccessId	varchar(255)	
ClientIp	varchar(255)	客户端 ip 地址
UserAgent	text	客户端 UA
ClientId	varchar(255)	设备 id
LoginHost	varchar(255)	客户端的 HOST
LoginAddress	varchar(255)	登录的 IP 位置
Status	varchar(255)	登录状态
Msg	varchar(255)	补充说明信息
RequestId	text	请求 Id

2.3.3.5 用户余额信息表

存储用户的余额信息的表，主要信息如表 2.5 所示：

表 2.5 用户余额信息表

字段名称	数据类型	说明
Id	int	
UserId	int	用户的 Id
Balance	float	用户的余额
Credit	float	信用额度
Freeze	float	冻结额度
Available	float	用户实际可用的额度
RegTime	varchar(32)	开户时间
LastPayId	varchar(255)	最后一笔订单编号
LastPayTime	varchar(32)	最后一笔订单支付时间

2.3.3.6 触发风控系统限制表

记录因触发风控系统限制而被临时禁止登录到系统的信息，主要信息如 表 2.6 所示：

表 2.6 触发风控系统限制表

字段名称	数据类型	说明
Id	int	
RiskId	varchar(255)	风险记录 Id
LogTime	varchar(30)	记录风险 的时间
Ip	varchar(255)	被记录风险的 Ip
UserId	int	被记录风险的 用户的 UserId
ClientId	varchar(255)	被记录风险的 客户端的 Id
RiskTime	varchar(15)	被记录风险的抑制登录时间
Reason	varchar(255)	被记录风险的原因

2.3.3.7 风控系统限制表

记录因触发风控系统限制而被永久禁止登录到系统的信息，主要信息如 表 2.7 所示：

表 2.7 风控系统限制表

字段名称	数据类型	说明
Id	int	
RiskId	varchar(255)	风险记录 Id
LogTime	datetime	记录风险的时间
UserAgent	text	被记录风险的设备的 User-Agent
ClientId	varchar(255)	被记录风险的 客户端的 Id
UserId	varchar(255)	被记录风险的 用户的 UserId
Ip	varchar(255)	被记录风险的 Ip
Origin	varchar(255)	被记录风险的来源域名信息

2.3.3.8 角色信息表

存放用户的角色信息，主要信息如表 2.8 所示：

表 2.8 角色信息表

字段名称	数据类型	说明
Id	Int	
AuthId	varchar(32)	角色的 Id
PermissionName	varchar(255)	角色的名称

表 2.8[续]

字段名称	数据类型	说明
Description	text	角色的描述信息
IsLock	tinyint(1)	该角色是否锁定不可修改

2.3.4系统基础表

2.3.4.1 RSA 双向数字加密公钥信息表

存放系统用于 RSA 双向数字加密使用的公私钥，对应域名，特征值，文件类型等信息，主要信息如 表 2.9 所示：

表 2.9 RSA 双向数字加密公钥信息表

字段名称	数据类型	说明
Id	int	
host	varchar(64)	SSL 证书对应的域名
pubMd5	varchar(32)	SSL 证书的公钥的 MD5
endTime	varchar(32)	证书到期时间
filetype	varchar(16)	证书的文件格式
filevalue	text	证书的内容

2.3.4.2 系统配置表

存放系统的全局配置项目的信息，主要信息如 表 2.10 所示：

表 2.10 全局配置项表

字段名称	数据类型	说明
Id	int	
ConfigKey	varchar(255)	系统配置变量的键名
ConfigValue	text	系统配置变量的值
ConfigPs	varchar(255)	配置项目注释

2.3.4.3 报错日志信息表

存放系统的报错日志的信息，主要信息如表 2.11 所示：

表 2.11 报错日志信息表

字段名称	数据类型	说明
Id	int	
code	varchar(255)	报错信息的代码
msg	varchar(255)	报错信息的英文说明
ps	varchar(255)	报错信息的中文说明

2.3.4.4 报错信息表

存储系统的具体报错信息的表，主要信息如表 2.12 所示：

表 2.12 报错信息表

字段名称	数据类型	说明
Id	int	
ExceptId	varchar(255)	异常信息 Id
RequestId	varchar(255)	请求 Id
ServerId	varchar(32)	服务器的 Id
Except_Type	varchar(255)	异常类型
Except_Session	longtext	发生异常时服务器的 SESSION 数据
Except_Content	longtext	发生异常的日志的内同
ExceptTime	varchar(32)	发生异常的时间 整型
ExceptTimeStr	varchar(32)	发生异常的时间 字符串

2.3.4.5 域名信息表

存储系统中用户自定义域名信息的表，主要信息如表 2.13 所示：

表 2.13 域名信息表

字段名称	数据类型	说明
Id	int	
DomainId	varchar(255)	域名的 Id
DomainHost	varchar(255)	域名的信息
ProductName	varchar(255)	关联的产品名称
ProductWorkId	varchar(255)	关联的产品的业务 Id
ProductWorkCode	varchar(255)	关联的产品的业务 Code

表 2. 13［续］

字段名称	数据类型	说明
panelId	int	关联的面板站点的 Id
RegionId	varchar(255)	关联的 Bucket 的 Id
IsSSL	tinyint(1)	是否开启 SSL 功能
forceSSL	tinyint(1)	是否强制启用 SSL 证书加密
SSLId	varchar(255)	关联的 SSL 证书的 Id
UserId	int	所属的用户信息
UpdateTime	varchar(30)	记录更新的时间

2. 3. 4. 6　接口权限表

存储系统中， 接口权限及详情信息的表， 主要信息如表 2.14 所示：

表 2. 14 接口权限表

字段名称	数据类型	说明
Id	int	
Interface_Id	varchar(32)	接口 Id 信息
Interface_Add	varchar(255)	接口请求地址
Interface_Name	varchar(255)	接口名称信息
Interface_Des	text	接口描述信息
Interface_ParentId	varchar(255)	接口的父亲节 Id
Interface_Child	text	接口的子节点 Id
IsApi	tinyint(1)	是否对外真实暴露的 API 接口
AuthRoot	text	根权限管理控制 Id
InsertTime	varchar(32)	接口权限的添加时间
UpdateTime	varchar(32)	接口权限的修改时间

2. 3. 4. 8 Kafka 消费者群组表

存储系统中，Kafka 消费者所在群组的配置和用户以及密码信息，主要信息如表 2.15 所示：

表 2. 15 Kafka 消费者群组表

字段名称	数据类型	说明
Id	int	
TopicId	varchar(64)	Kafka 消费端的 Id
GroupId	varchar(64)	Kafka 消费端所属的群组的 Id
TopicName	varchar(255)	kafka 消息通道的 Id
TopicArea	varchar(255)	Kafka 消费端所属区域的代码
TopicPartition	int	消息队列的分区的 id
KafkaServer	varchar(255)	kafka 服务器的地址
KafkaAuth	varchar(255)	kafka 服务器加密方式
KafkaUser	varchar(255)	Kafka 用户名
KafkaPass	varchar(255)	kafka 密码

2. 3. 4. 9 API 请求表

存储系统中允许从前端浏览器直接访问 API 接口的域名，主要信息如表 2.16 所示：

表 2. 16 API 请求表

字段名称	数据类型	说明
Id	int	
OriginId	varchar(64)	被允许前端请求的来源域名的 Id
OriginValue	varchar(255)	被允许请求的来源域名的值
Ps	varchar(255)	备注信息
UpdateTime	varchar(32)	更新时间

2. 3. 4. 10 服务器配置信息表

存储系统中不同地域中用来提供服务的服务器的配置信息，主要信息如表 2.17 所示：

表 2. 17 服务器配置信息表

字段名称	数据类型	说明
Id	int	
RegionId	varchar(64)	地域 Id
RegionName	varchar(16)	地域名称(简写)
RegionTitle	varchar(255)	地域名称

表 2.17［续］

字段名称	数据类型	说明
RegionType	varchar(255)	服务类型
DnsType	varchar(255)	DNS 解析记录类型
DnsValue	varchar(255)	DNS 解析记录值
PanelHost	varchar(255)	宝塔面板的地址
PanelPort	varchar(255)	宝塔面板的端口号
PanelApi	varchar(255)	宝塔面板的 API 密钥

2.3.4.11 客户端请求日志表

存储系统中所有来自客户端的请求的日志，主要信息如表 2.18 所示：

表 2.18 客户端请求日志表

字段名称	数据类型	说明
Id	int	
RequestId	varchar(255)	请求的唯一 ID
ExceptId	varchar(255)	异常响应处理的唯一 ID
ServerId	varchar(255)	响应服务器 ID
IP	varchar(255)	请求客户端的的 IP 地址
UA	text	请求客户端的 UA
REFFER	text	请求客户端的 Reffer
ClientId	varchar(255)	请求客户端的 ClientId
RequestUrl	text	请求的 Url
RequestContent	longtext	请求的数据包体
ResponseContent	longtext	响应的数据包体
RequestTime	varchar(255)	请求的时间

2.3.4.12 客户端特征信息表

将请求的用户的 User-Agent 和 Ip 使用 MD5 加密得到请求系统后台的客户端的特征信息 UvId，并将相关信息存储到本表中，主要信息如表 2.19 所示：

表 2. 19 客户端特征信息表

字段名称	数据类型	说明
Id	int	
UvId	varchar(32)	访问请求记录的特征值
Ip	varchar(255)	请求客户端的 Ip 地址
Uv	text	请求客户端的 Uv
Plus	text	备注信息
InsertTime	int	数据新增时间
UpdateTime	int	数据修改时间
UvId	varchar(32)	访问请求记录的特征值

2. 3. 4. 13 用户路由权限表

存储当前系统中的菜单项的名称，所属模块，前端自定义信息，以及拥有访问权限的用户角色信息的表，主要信息如表 2.20 所示：

表 2. 20 用户路由权限表

字段名称	数据类型	说明
Id	int	
RouteId	varchar(32)	路由接口 Id
ModeId	varchar(32)	所属模块 Id
RouteType	varchar(16)	路由接口类型 group , menu , link
RouteAdd	varchar(255)	路由地址
RouteIcon	varchar(255)	路由接口的图标信息
RouteName	varchar(255)	路由名称
ParentId	varchar(32)	路由的父节点 id
ComponentName	varchar(255)	路由对应的前端组件的名称
ComponentPath	varchar(255)	路由对应的前端组件的地址
MenuShow	varchar(32)	控制路由节点在前台菜单中是否显示 show / hidden
Redirect	text	重新定性的地址
Childs	text	路由的直接子路由 Id 列表
AuthRoot	text	路由的角色权限

表 2.20［续］

字段名称	数据类型	说明
CreateTime	varchar(255)	路由接口创建日期
UpdateTime	varchar(255)	路由接口更改日期

2.3.4.14 任务队列信息表

存储系统中推送到异步进程守听的异步消息任务队列的具体参数，执行情况，运行时间等信息的表，主要信息如表 2.21 所示：

表 2.21 任务队列信息表

字段名称	数据类型	说明
Id	int	
TaskId	varchar(64)	任务的 Id
TaskName	varchar(255)	任务的名称
TaskParam	text	任务队列的参数
TaskResult	text	执行的响应结果
TaskProcess	text	过程状态参数
TaskStatus	varchar(255)	任务执行的状态
CallbackUrl	varchar(255)	回调请求的地址
CallbackToken	varchar(255)	回调请求的校验 token
RunGroupId	varchar(64)	运行的任务的设备的 id
UserId	int	发起任务的用户的 Id
InitTime	int	任务初始化时间
StartTime	int	任务启动时间
EndTime	int	任务结束时间
RunTime	int	运行用时
WaitTime	int	任务等待用时

2.3.5 临时缓存表

2.3.5.1 手机验证码数据信息表

存储系统中下发给手机验证码的数据信息，主要信息如表 2.22 所示：

表 2. 22 手机验证码数据信息表

字段名称	数据类型	说明
Id	int	
CaptchaId	varchar(255)	验证码 Id
PhoneNum	varchar(16)	手机号
Code	varchar(6)	发送到手机的验证码
RequestTime	varchar(32)	发送验证码的时间
ExpireTime	varchar(32)	验证码过期时间
Captcha_Status	varchar(8)	验证码的状态新
Send_Status	varchar(128)	验证码信息的发送状态
RequestId	varchar(255)	发送验证码请求的 Id

2. 3. 5. 2 水墙滑动验证信息表

存储系统中调用腾讯防水墙接口对用户进行滑动验证时的信息，主要信息如表 2.23 所示：

表 2. 23 水墙滑动验证信息表

字段名称	数据类型	说明
Id	int	
CaptchaId	varchar(255)	验证请求唯一 Id
CaptchaStatus	varchar(255)	当前验证请求的状态信息
EvilLevel	int	风险等级 0-100
Front_Ticket	varchar(255)	验证成功的票据，当且仅当 ret = 0 时 ticket 有值。
InitTime	datetime	腾讯防水墙初始化的时间
EndTime	datetime	腾讯防水墙终止的时间

2. 3. 5. 3 工作流票据状态信息表

存储系统中调用系统的工作流的票据的状态信息的，主要信息如表 2.24 所示：

表 2. 24 工作流票据状态信息表

字段名称	数据类型	说明
Id	int	
FlowTicketId	varchar(255)	工作流的流票据 Id

表 2.24［续］

字段名称	数据类型	说明
FlowTicketType	varchar(255)	工作流的类型
TicketStatus	varchar(255)	工作流的流程状态 请注意不同的工作流可能会存在不同的状态描述信息，初始状态统一为 init，最终态为 end
TicketRegTime	varchar(32)	工作流的开始时间
TicketEndTime	varchar(32)	工作流的结束时间
TicketParamJson	text	作业票的相关信息 Json
RelationUserId	int	关联用户 Id
CaptchaPhoneId	varchar(255)	短信验证代码 Id
CaptchaMailId	varchar(255)	邮件验证代码 Id
CaptchaPicId	varchar(255)	图形验证代码 Id

2.3.6具体业务表

2.3.6.1 存储桶资源及配置表

存储 cos 存储桶的资源和基础配置的表，主要信息如表 2.25 所示：

表 2.25 存储桶资源及配置表

字段名称	数据类型	说明
Id	int	
BucketId	varchar(64)	文件仓库 Id
BucketCode	varchar(8)	8 位随机字符组成的仓库短码
BucketName	varchar(255)	文件仓库的名称
BucketTitle	varchar(255)	文件仓库的标题
BucketArea	varchar(255)	文件仓库所在的地域，当前默认为 yc-cos
BucketRegionId	varchar(255)	文件仓库所在的地域 Id
PanelSiteId	int	宝塔面板的站点 Id
DnsRecordId	varchar(32)	DNS 解析记录的 Id
BackUpOption	varchar(255)	仓库全局备份配置选项 aliyun , qiniu, aliyun_qiniu
AuthorzationOption	varchar(255)	仓库全局访问控制选择 pw_pr , w_pr , w_r (公有读写，公有读私有写，私有读写)
OutPutOption	varchar(255)	仓区全局访问输出配置项目 atonal aliyun qiniu auto

表 2.25［续］

字段名称	数据类型	说明
UserId	int	用户 Id
CreateTime	varchar(30)	仓库建立时间

2.3.6.2 存储桶配置项目表

存储 cos 系统中,对应每个 bucket 存储桶的具体的详细配置项目的表 ，主要信息如表 2.26 所示：

表 2.26 存储桶配置项目表

字段名称	数据类型	说明
Id	int	
BucketId	varchar(255)	存储桶的资源 Id
ConfigKey	varchar(255)	存储桶的配置关键字段
ConfigType	varchar(255)	配置字段的类型
ConfigValue	longtext	配置项目的数据的类型
UpdateTime	varchar(32)	配置项修改的时间

2.3.6.3 存储桶访问日志表

存储 cos 系统中，对应每个 bucket 存储桶的具体的访问日志的表，主要信息如表 2.27 所示：

表 2.27 存储桶访问日志表

字段名称	数据类型	说明
Id	int	
FileId	varchar(255)	文件 Id
LogId	varchar(64)	日志 Id
LogTime	int	日志记录时间
ParseTime	varchar(16)	文件解析耗时
Status	varchar(255)	文件解析的状态
UvId	varchar(32)	用户特征值 Id

2.3.6.4 存储桶逻辑位置表

存储 cos 系统中,对应每个 bucket 存储桶中的文件的逻辑位置和信息的表，主要信息如表 2.28 所示：

表 2.28 存储桶逻辑位置表

字段名称	数据类型	说明
Id	int	
AuthorzationOption	varchar(255)	仓库全局访问控制选择 pw_pr , w_pr , w_r (公有读写，公有读私有写，私有读写)
FileExt	varchar(32)	文件的后戳类型
FileId	varchar(64)	文件 Id
FileName	varchar(255)	文件名称
FilePath	varchar(255)	所在文件目录
FileSize	varchar(32)	文件大小
IsDir	varchar(255)	是否是文件夹
ObjectId	varchar(255)	对象系统的 Id
UpdateTime	int	更新时间
UploadTime	int	创建时间
UserId	varchar(32)	上传的人的 Id

2.3.6.5 文件分片信息表

存储 cos 系统中，通过分片上传的方式上传到系统中的分片信息的表 ，主要信息如表 2.29 所示：

表 2.29 文件分片信息表

字段名称	数据类型	说明
Id	int	
CreateTime	int	创建时间
SliceId	varchar(128)	分片 Id
SliceMd5	varchar(255)	分片的 MD5 值
SlicePath	varchar(255)	分片的文件路径地址
SortId	int	分片的排序 Id

表 2.29［续］

UploadTime	int	上传时间
UpTaskId	varchar(128)	上传的任务的 Id
UserId	int	用户 Id
SliceId	varchar(128)	分片 Id

2.3.6.6 文件上传任务信息表

存储 cos 系统中，通过分片上传/简单上传两种文件上传方式上传到服务器的任务信息，文件特征值的表，主要信息如表 2.30 所示：

表 2.30 文件上传任务信息表

字段名称	数据类型	说明
Id	int	
UpTaskId	varchar(255)	上传的分片的 Id
FileSize	double	文件的大小
cosArea	varchar(255)	cos 文件区域
cosBucketId	varchar(255)	cos 存储桶的 Id
FileName	varchar(255)	文件的名称
FilePath	varchar(255)	文件的路径地址
UploadType	int	上传方式
FileMd5	varchar(255)	文件的 Md5
sliceNum	int	分片的数量
sliceSize	int	分片的大小
cacheTime	int	文件缓存时间
createTime	int	任务创建时间
taskStatus	varchar(255)	任务的状态

2.3.6.7 文件对象表

存储 cos 系统中，存储到文件系统中的实际文件对象的表，主要信息如表 2.31 所示：

表 2.31 文件对象表

字段名称	数据类型	说明
Id	int	
ObjectId	varchar(64)	文件在 OSS 云端的 Id
ObjectKey	varchar(128)	文件在 OSS 云端的路径地址
FileMd5	varchar(32)	文件 MD5
FileSize	varchar(32)	文件大小，字节计算
FileExt	varchar(32)	文件的后戳
tmpPath	varchar(255)	上传的临时存储的文件地址
AliOssPath	varchar(255)	阿里云对象存储的文件地址
Cloud189Path	varchar(255)	天翼云对象存储的文件地址
QiNiuPath	varchar(255)	七牛云对象存储的文件地址
UploadTime	int	文件的上传时间

2. 3. 6. 8 文件映射表

存储系统内部常用数据中 ，对应文件后戳和 content-type 映射关系的表，主要信息如表 2.32 所示：

表 2. 32 文件映射表

字段名称	数据类型	说明
Id	Int	
FileExt	varchar(255)	文件的后戳类型
ContentType	varchar(255)	文件对应的 content-type 的类型

2. 3. 6. 9 ip 地址运营商信息表

存储系统内部常用数据中， 指定的 ip 地址和该 ip 地址所属的运营商以及位置信息的表，主要信息如表 2.33 所示：

表 2. 33 ip 地址运营商信息表

字段名称	数据类型	说明
Id	int	
Ip	varchar(255)	IP 地址
IpInt	varchar(16)	IP 地址的整数形式
CacheTime	varchar(32)	数据抓取时间

表 2.33[续]

字段名称	数据类型	说明
ApiSource	varchar(255)	数据来源地址
continent	varchar(15)	大洲
country	varchar(50)	国家
zipcode	varchar(15)	邮编地址
timezone	varchar(15)	时区
accuracy	varchar(15)	数据精度
owner	varchar(255)	所属机构
isp	text	所属运营商
source	varchar(255)	采集方式
areacode	varchar(10)	国家编码
adcode	varchar(10)	行政编码
asnumber	int	自治区编码
lat	varchar(20)	维度
lng	varchar(20)	经度
radius	varchar(20)	定位半径
prov	varchar(255)	省份
city	varchar(255)	城市
district	varchar(255)	区县
multiaddress	varchar(255)	高精度定位数据（ipip）
info	varchar(255)	ipv6 响应的位置信息

2.3.6.10 轮播图信息表

存储系统首页做为轮播图片展示的图片信息，和点击链接地址的表，主要信息如表 2.34 所示：

表 2.34 轮播图信息表

字段名称	数据类型	说明
Id	int	
Ip	varchar(255)	IP 地址

表 2.33［续］

IpInt	varchar(16)	IP 地址的整数形式
CacheTime	varchar(32)	数据抓取时间
ApiSource	varchar(255)	数据来源地址
continent	varchar(15)	大洲
country	varchar(50)	国家
zipcode	varchar(15)	邮编地址
timezone	varchar(15)	时区
accuracy	varchar(15)	数据精度
owner	varchar(255)	所属机构

2.3.6.11 页脚信息表

存储系统首页展示的页脚信息的表，主要信息如表 2.35 所示：

表 2.35 页脚信息表

字段名称	数据类型	说明
Id	int	
FootId	varchar(255)	图片的 Id
SortId	int	排序 Id
FootUrl	varchar(255)	图片的地址信息
FootTitle	varchar(255)	图片的标题
FootDes	varchar(255)	图片的 Logo 的描述 信息

2.3.6.12 菜单信息表

存储系统首页展示的菜单信息的表，主要信息如表 2.36 所示：

表 2.36 菜单信息表

字段名称	数据类型	说明
Id	int	
SortId	int	排序的 Id
MenuName	varchar(255)	菜单的名称
MenuLink	varchar(255)	菜单的链接地址

2. 3. 6. 13 产品信息表

存储系统首页展示的产品信息的表，主要信息如表 2.37 所示：

表 2. 37 产品信息表

字段名称	数据类型	说明
Id	int	
SortId	int	排序 Id
ProductName	varchar(255)	产品的名称
ProductDes	varchar(255)	产品的描述
ProductIcon	varchar(255)	产品的图标
ProductPrice	varchar(255)	产品的价格信息
ProductSrc	varchar(255)	点击以后跳转的链接地址

2. 3. 6. 14 手机号信息池信息表

存储系统中手机号自有信息池信息的表，主要信息如表 2.38 所示：

表 2. 38 信息池信息表

字段名称	数据类型	说明
Id	int	
PhoneNum	varchar(16)	手机号
Status	int	0 正常 , 1 空号 , 2 停机 , 3 黑名单 , 4 无数据 ， 5 其它
RecordTime	varchar(32)	记录时间
UpdateTime	varchar(32)	更新时间
PhonePs	varchar(255)	手机号备注信息
AuthUserid	varchar(255)	关联添加或删除本条记录的用户 uid

2. 3. 6. 15 手机号验证日志信息表

存储系统中手机号验证日志信息的表，主要信息如表 2.39 所示：

表 2. 39 手机号验证日志信息表

字段名称	数据类型	说明
Id	int	
CaptchaId	varchar(255)	验证请求 id
PhoneNum	varchar(16)	被验证的手机号

表 2.39[续]

Status	varchar(255)	验证请求结果 -1 格式错误， 0 正常 , 1 空号 , 2 停机 , 3 黑名单 , 4 无数据， 5 其它
Extend	text	验证请求的其它附带返回结果
RequestTime	varchar(32)	请求验证的时间
RequestId	varchar(255)	请求验证手机号的接口的请求 Id

2.3.6.16 外链文件请求日志表

存储系统中存放在 oss 系统的系统外链文件的请求日志的表，主要信息如表 2.40 所示：

表 2.40 外链文件请求日志表

字段名称	数据类型	说明
Id	int	
LogId	varchar(64)	日志 Id
FileId	varchar(255)	被访问的文件的 Id
UvId	varchar(32)	请求的客户端的特征 Id
ParseTime	varchar(16)	文件解析用时
LogTime	int	日志记录的时间
Status	varchar(255)	本次请求的状态

2.3.6.17 OSS 文件对象表

以文件对象的形式存储的用户上传到系统文件外链模块的文件和存储在阿里云 OSS 系统中的真实文件地址数据映射关系表，主要信息如表 2.41 所示：

表 2.41 oss 文件对象表

字段名称	数据类型	说明
Id	int	
ObjectId	varchar(64)	文件在 OSS 云端的 Id
ObjectKey	varchar(128)	文件在 OSS 云端的路径地址
FileMd5	varchar(32)	文件 MD5
FileSize	varchar(32)	文件大小，字节计算
FileExt	varchar(32)	文件的后戳

表 2.41[续]

字段名称	数据类型	说明
UpdateTime	int	文件的修改时间
UploadTime	int	文件的上传时间
UserId	varchar(32)	文件上传人的 id

2.3.6.18 OSS 虚拟逻辑地址表

存储系统中存放在 oss 系统的系统中的文件的虚拟逻辑地址的表，主要信息如表 2.42 所示：

表 2.42 oss 虚拟逻辑地址表

字段名称	数据类型	说明
Id	int	
FileId	varchar(64)	文件 Id
FilePath	varchar(255)	所在文件目录
FileName	varchar(255)	文件名称
FileSize	varchar(32)	文件大小
FileExt	varchar(32)	文件的后戳类型
IsDir	varchar(255)	是否是文件夹
ObjectId	varchar(255)	对象系统的 Id
UploadTime	int	创建时间
UpdateTime	int	修改时间
UserId	varchar(32)	上传的人的 Id

2.4 API 请求格式设计

2.4.1 请求、响应格式标准

前端或第三方应用、服务器请求后端 API 接口时 一律使用 x-www-form-urlencoded 格式进行编码。服务端响应时，一律响应 JSON 请求 。需要注意的时，所有设备请求后端接口时，需要按照表 2.43 下面的标准，发起请求，下文中除特殊声明，均不再书写请求头部内容，下文中的请求内容默认为 content 格式化前的数组。

表 2. 43 请求响应头表

字段名	字段用途
Client-Device-Id	设备准入 ID
Client-Device-Token	设备准入 TOKEN， 注意 此处传入的值 为服务器响应的 Client_Device_Token
Client-Encrypt-Rsa	使用客户端公钥进行 RSA 加密后得到值 使用的 RSA 加密证书的公钥的 md5 值
encrypt	true , false
content	JSON 格式化后的数组 如果 encrypt 为 true ,需要 RSA 公钥加密
device_type	pc-client 、app-client 、server

服务端响应请求时， 同样按照上述标准进行响应，响应内容格式如表 2.4.2 所示。

注：在实际业务中，有时服务器响应给客户端的消息过长，在 RSA 加密时消耗了大量的时间。因此，现规定（特别说明除外），服务器响应的数据，如 content_size 的值 大于 64K （64 * 1024）时，将不会对响应的数据进行加密。 正常情况下，当您请求后端接口时，后端将向您返回加密后的数据。但因系统内部故障，或传入参数不规范导致系统在运行过程中触发异常时，系统将返回异常错误代码。为方便开发者调试解觉问题，规定，系统返回异常错误代码时，数据一律不加密。特别的，规定 5000000 为系统内部未知异常，此时回传的数据包体中将包含 ExceptId 字段，您可记录下 ExceptId , 提供给开发人员，以方便复现排除您遇到的问题

表 2. 43 请求响应体表

字段名	字段用途
status	默认值 success
encrypt	true / false
request_time	发起请求的时间
spend_time	后端执行时处理花费的时间
request_id	本次请求的唯一 id 用于 DEBUG 调试或反馈问题

表 2. 43[续]

request_id	本次请求的唯一 id 用于 DEBUG 调试或反馈问题
session_expire	请求设备的准入许可密钥过期时间
content_size	响应内容的字符串长度，注意可能存在一定误差
content	本次请求的响应内容，如果 encrypt 的值为 true 会使用 RSA 公钥将数组 JSON 序列化后加密返回，否则直接返回数组

图 2.5、图 2.6 展示的是系统在运行过程中，前端程序向后端传输的 JSON 数据包体。

本课题使用的 RSA 双向加密技术，使得别有用心的攻击者很难直接构造参数进行攻击。但本技术存在一个安全缺陷，即攻击者可以对数据包进行监听并进行请求重放，为避免这一问题，您可在请求的 content 中加上下面两个字段，以防止 Replay Attack。

表 2. 44 请求响应头表

字段名	类型	用途
request_replay	int	当前请求时的时间戳 （10 位）
request_replay_abs	int	请求容差，默认值为 3,单位 s

图 2. 5 前端传给后端的 JSON 数据包

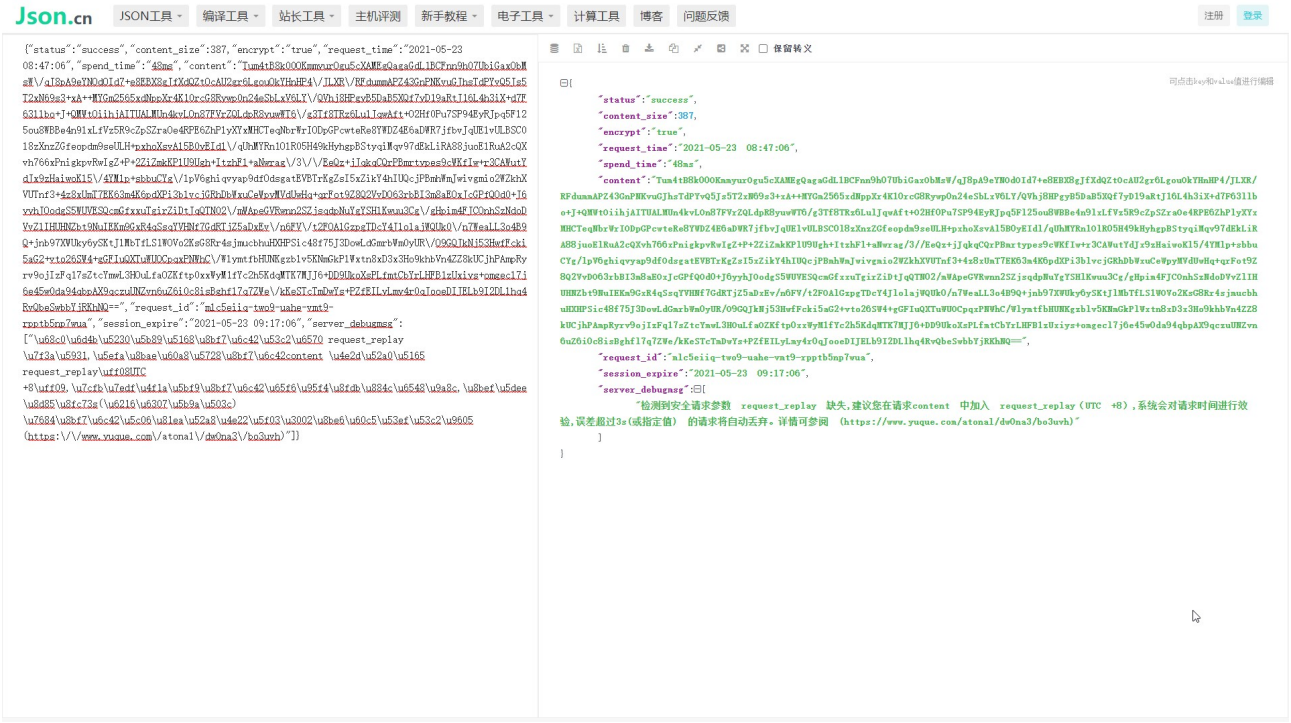

图 2.6 后端响应给前端的 JSON 数据包

2.4.2 设备跨域访问设计

为了解决域名跨域、不同设备请求的问题，无调云计算平台采用业界通用的 header 头部认证法。如表 2.45 所示，任何期望与后端 API 系统进行通讯的设备（无论是前端 HTML5 页面，小程序，App, 或服务端程序）均需要访问设备注册接口，注册设备访问许可。设备访问许可密钥由（ Client_Device_ID，Client_Device_TOKEN） 组成，由服务器统一授权颁发。每个设备的授权访问许可，有效期为 30 分钟（1800s），在设备访问许可的生命周期内，访问后端任意接口，会自动续费生命周期。

表 2.45 跨域访问设计表

接口地址		请求方法
/Client/Session/Reg		GET / POST
请求头部		请求内容
Client-Device-Id	设备准入 ID	空
Client-Device-Token	设备准入 TOKEN ，注意此处传入的值为服务器响应的 Client_Device_Token	

2.5安全设计

2.5.1 RSA 双向非对称加密

本系统采用 RSA 非对称加密技术，服务器端存留着多个不同版本的 RSA 双向非对称加密的私钥，客户端则持有 RSA 加密的公钥。当客户端向服务器进行数据交互的时候，会使用公钥对数据进行 RSA 的加密，加密后的数据只有持有私钥的服务端才能对数据进行解密。服务端响应客户端数据时，使用 RSA 的私钥对数据进行加密，客户端收到数据必须使用公钥对数据进行解密。服务端会定期对 RSA 双向密钥进行更换，过期的密钥将被弃用。借助 RSA 双向非对称的密钥的保护，可确保数据在传输过程中不被第三方监听，不被恶意篡改。

2.5.2 请求时间戳校验，防止重放攻击

在 RSA 双向数据加密的情况下，恶意攻击者无法读取当前设备的请求数据包，但仍可通过重发请求来实现对业务系统的攻击。(即攻击者发送一个目标主机已经接收过的包，来达到欺骗系统的目的。主要用于身份认证过程，破坏认证的正确性[10])因此系统设计并采用了一套时间戳抗重放攻击方案，客户端在请求服务器后台接口的时候，可以在请求的包体中携带 request_replay 和 request_replay_abs 这两个参数，规定请求的过期时间，系统会校验服务器时间戳 与 request_replay 传入时间戳相减得到的绝对值是否在 request_replay_abs 规定允许的范围内，如果超出这一范围会自动放弃对本次请求的处理，并告知用户本次请求存在被恶意重放的风险。使用本方案，客户端不需要存储任何“抗重放因子”，不占用存储空间，且获取“抗重放因子”的方法简单[10]。

2.5.3 基于登录日志的安全风控系统

本系统会记录用户每次登录时的 IP 地址、User-Agent、ClientId 和登录状态信息，并存入系统的用户登录日志数据库中。当用户登录系统时，系统会根据日志数据，对当前用户的登录状态进行分析，判定用户当前的登录请求是否存在风险，如存在风险，则要求用户进行短信或邮件验证登录。系统会自动对连续异常登录的用户采取惩罚措施，禁止指定的用户账户、IP、User-Agent、ClientId 在一段或永久登录到系统，防止通过密码穷举或撞库破解系统的密码。

2.6 开发调试设计

云计算平台因其业务的特殊性，通常会有很多的用户自行开发相关的第三方应用接入系统。在用户接入系统的过程中，通常会出现传入字段有误，缺少字段等问题。为了方便用户自查错误，反馈问题，方便开发人员能够及时的发现问题，定位错误，本系统设计了一套快速开发调试系统。

2.6.1 用户请求日志

如图 2.7 所示，针对来自前端浏览器中的每一个独立的请求，系统会在后台为其分配一个唯一的，独立的 Id，用以标识当前请求信息。通常情况下，系统会在返回的 JSON 包体中，通过 request_id 字段响应当前请求的 Id,其格式为一组标准 UUID。同时，系统也会在响应的 Header 头部中通过 Client-Request-Id 字段响应当前的请求 Id.用户可将请求 Id 提供给开发人员，开发人员通过查阅数据库或者使用前端调试工具，可以快速的查阅本次请求时用户传入系统的参数，和系统响应给用户的 JSON 数据，如图 2.8 所示 ，开发人员可对当前请求的时间，SESSION 有效时间，请求的接口地址，请求的 IP 地址，请求的 Reffer 地址，请求的数据包体的内容，响应的数据包体的内容进行查阅和分析。

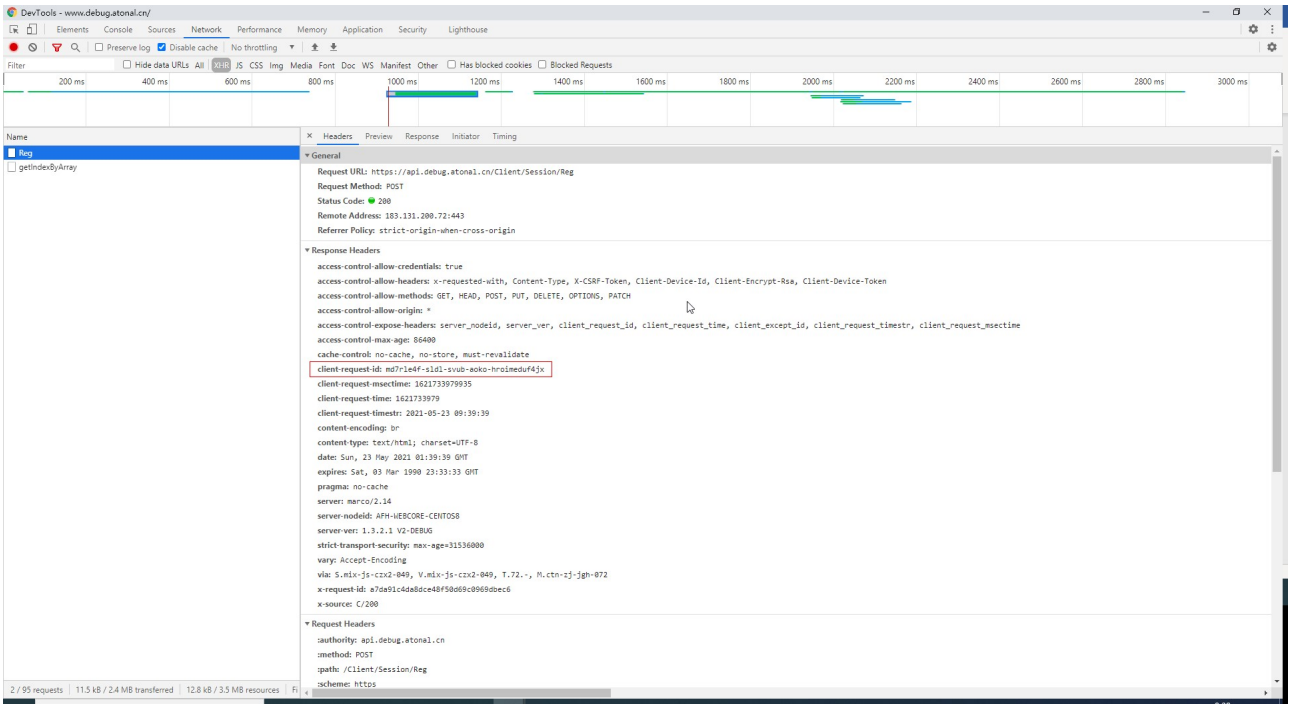

图 2.7 Chrome 调试工具中显示的响应 Header

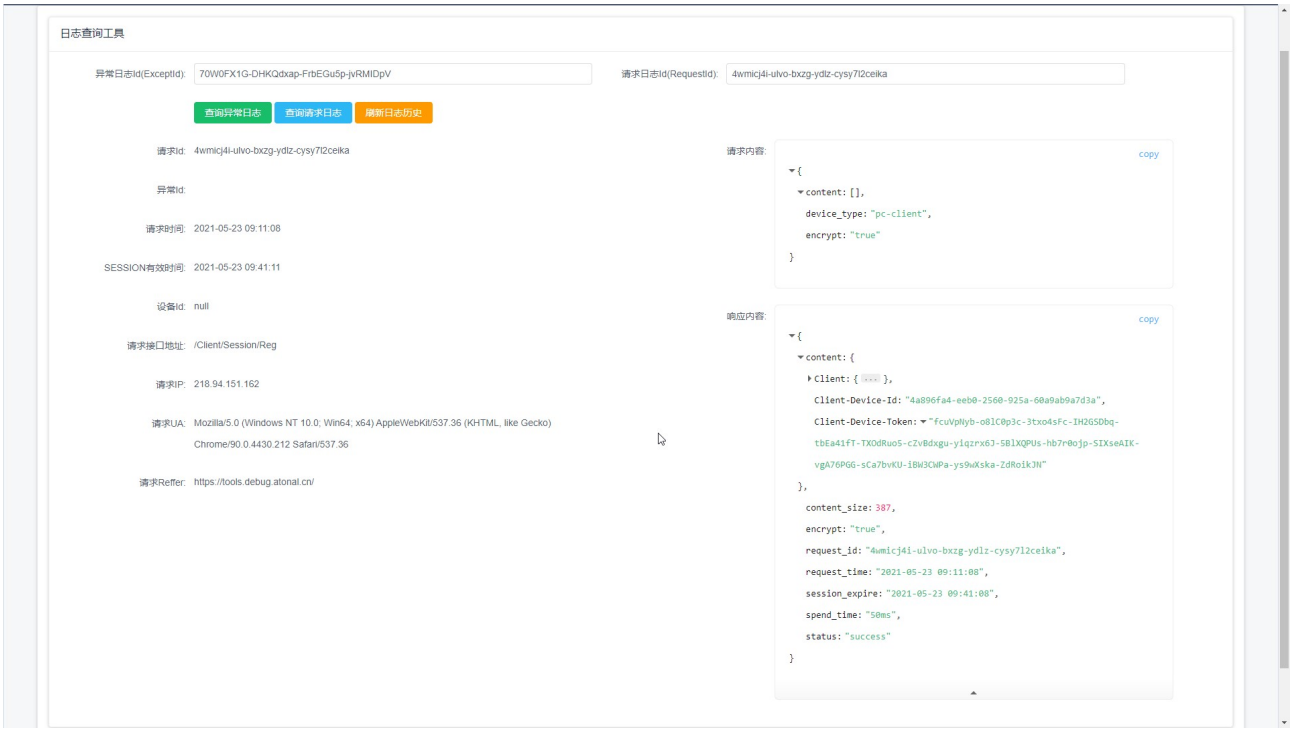

图 2.8 通过前端开发调试器查询请求日志

2.6.2异常错误日志

当系统在运行过程中，触发异常时，系统会自动捕获异常，并将其抛入异常处理模块中。根据遇到的异常情况的不同，本系统中的异常主要分为两类，一类是系统自定义异常（参阅 2.6.3 小节），一类是未定义的未知异常。和 2.6.1 小节中提到的用户请求日志类似，系统会对用户的每一条异常请求分配一个异常 Id,并在响应给客户端的 JSON 数据包中使用 ExceptId 字段进行标识，并在响应的 Header 头部使用 Client-Except-Id 字段进行标识。用户可以自信阅读响应的错误信息判断大致的错误，若出现无法定位的错误，可将错误 Id 提交给开发人员，开发人员通过查阅数据库或者使用前端调试工具可快速查阅本次错误的具体异常信息。如图 2.9 所示，和 2.6.1 小节中所述类似，开发者在获得本次请求的基本信息的同时还可阅读 ThinkPHP 打印出来的异常日志信息。本图中，前端应用请求了一个不存在的 API 接口地址，据此后端响应了 500 未知错误。

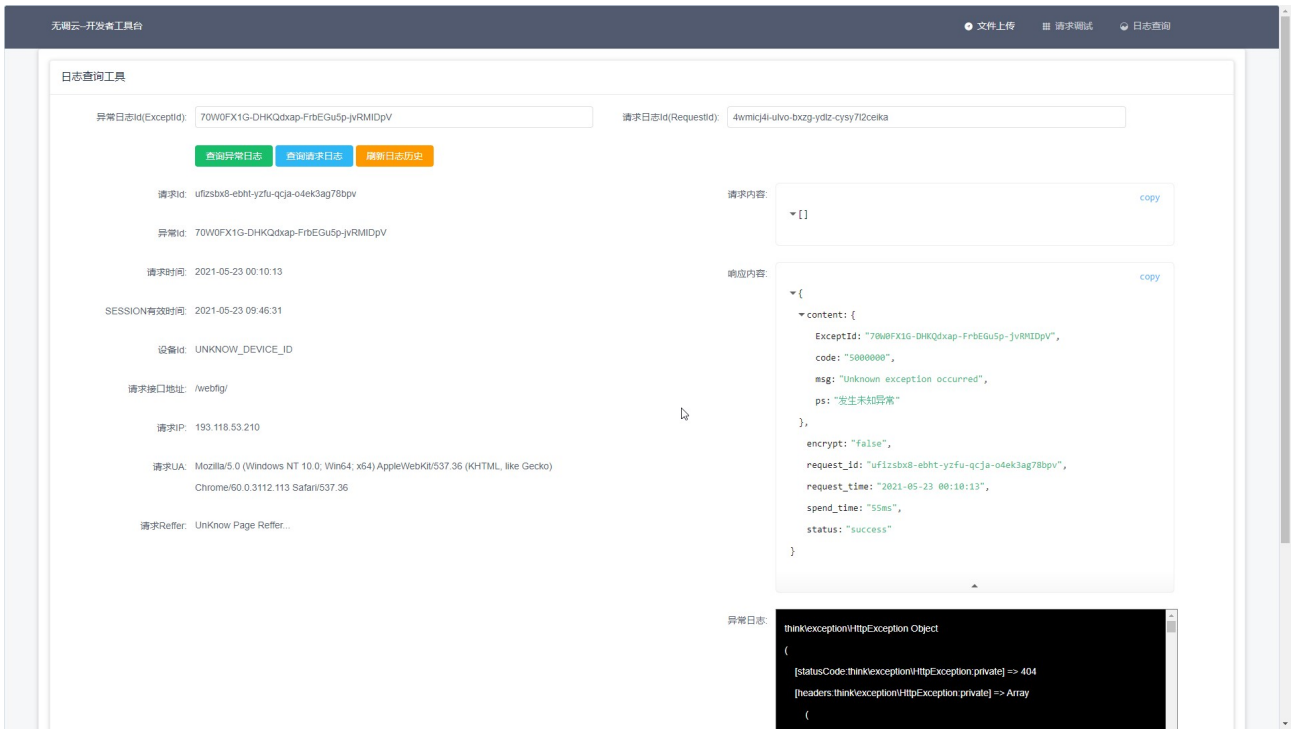

图 2.9 通过前端开发调试器查询异常日志

2.6.3 异常代码定义

为帮助用户在请求系统的 API 接口时能够快速定位问题，本系统设计了一套自定义的异常代码消息组。通常系统响应的异常代码由 8 位数字组成。根据所触发的异常的不同，如表 2.46 所示，异常代码可大致分为如下这几类。系统将会在用户触发相关异常时，抛出相关异常的错误代码和对应相关异常的英文描述信息。

表 2.46 异常代码分类

异常代码范围	异常信息类型
4030XXX	没有权限请求系统相关底层接口或拒绝访问
4031XXX	没有权限请求相关业务接口或拒绝访问
5000XXX	系统内部错误或参数有误或未知异常
5001XXX	请求工作流相关接口时触发了错误或异常
5002XXX	请求验证码相关接口时触发了错误或异常
50030XX-50031XX	请求相关权限配置接口，异常配置接口，路由配置接口，域名配置接口时发生错误或异常

表 2.46［续］

50032XX	请求文件操作时出现错误或者异常
50033XX	请求存储桶资源是时出现错误或者异常
50034XX	请求域名绑定接口或者 SSL 证书接口时发生错误或者异常
50035XX	请求文件上传任务时发生错误或者异常

2.6.4开发提示消息

为避免用户不规范传入参数导致系统在运行过程中可能会出现的各种异常或数据风险，我们特别设计了 DEBUG_MSG 功能。当用户传入的参数被该模块捕获时，系统会在响应到前端的 JSON 包体中使用 debug_msg 字段来注明当前用户的本次请求可能存在的问题或导致的风险。当后端代码 API 逻辑或者接口发生变动时，亦可以通过本功能向用户推送接口更新消息。如图 2.10 所示，在使用 Chrome 打开自主研发的前端调试工具，使用 F12 打开 Chrome 浏览器的开发者模式，在 Console 面板下已经可以显示当前系统响应给用户的提示消息（建议用户使用 request_replay 字段来避免重放请求攻击）。

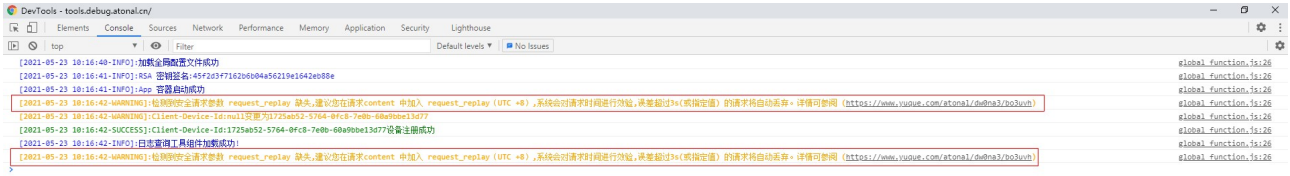

图 2.10 Chrome 调试工具中显示的开发提示信息

第三章 系统实现与测试

3.1 系统首页

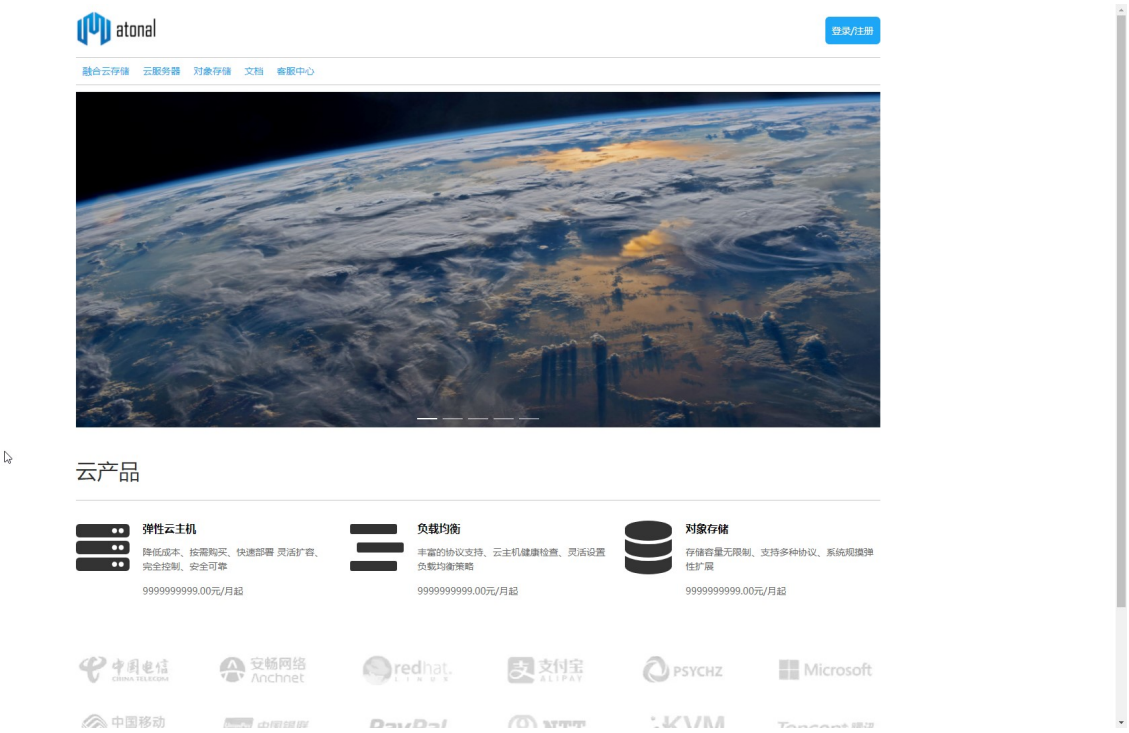

图 3.1 系统前台首页

图 3.1 是用户访问系统网站后看到的前台首页页面。用户可以在网站首页中快速了解系统中的一些热门产品，以及产品的基础定价，点击产品菜单和轮播图可以打开系统中的不同模块的详情。用户点击系统右上角的登录/注册按钮，可以访问系统后台的用户中心，对用户在系统中的业务进行管理。

3.2 用户登录、注册

在各种主流的 WEB 系统中，用户的登录注册一直是各种安全风险的切入点和发生点。本系统前后端分离的特性使得跨域登录的问题更加突出。针对这一问题，本系统专门设计了一套复杂的用户登录和注册安全审计逻辑（如图 3.2，图 3.3 所示）可以使得跨域登录变的更加的安全，结合多因子用户身份认证方法，结合票据加密传输[11]。

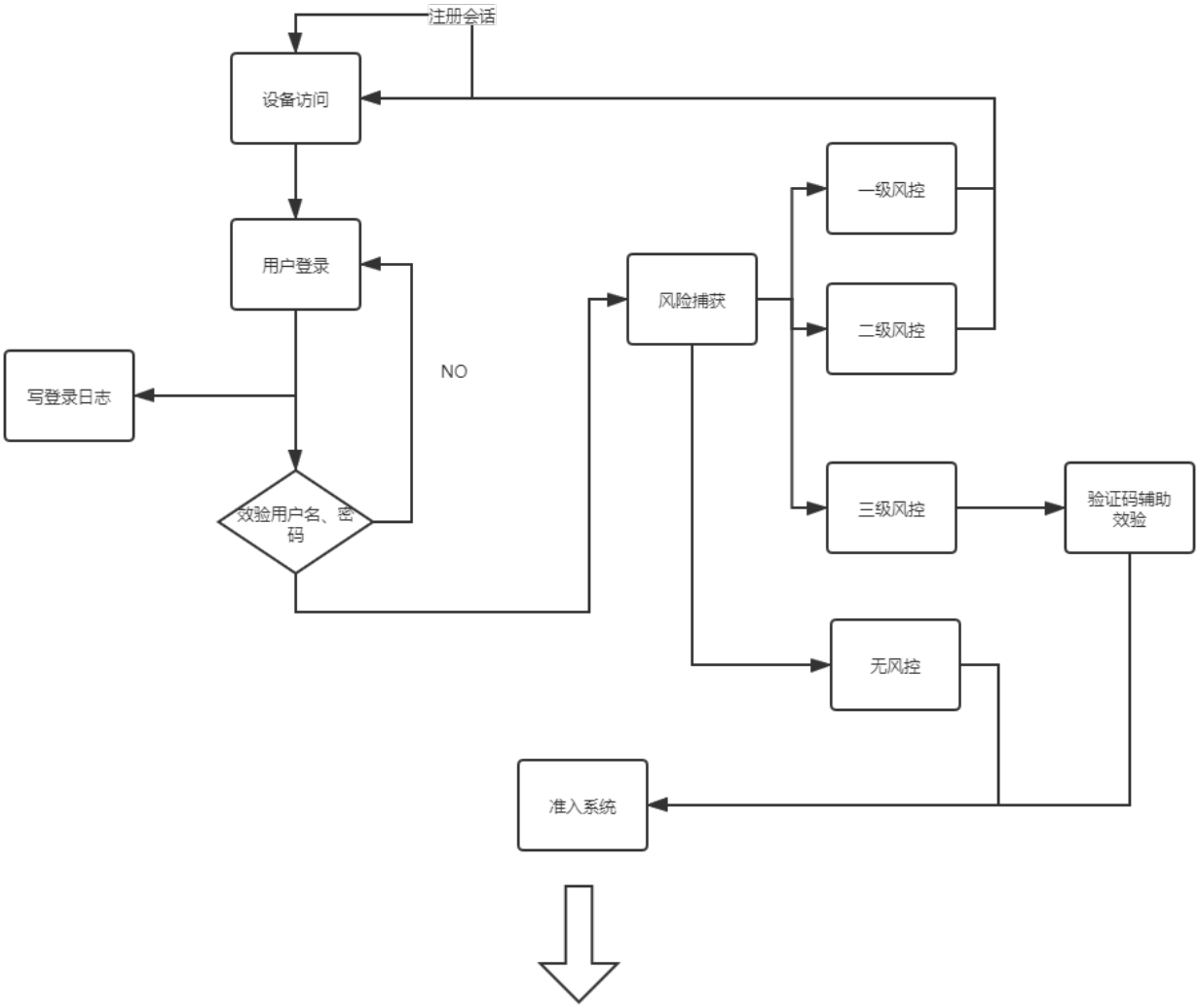
注册会话
设备访问
用户登录
写登录日志
效验用户名、密码
NO
风险捕获
一级风控
二级风控
三级风控
验证码辅助效验
无风控
准入系统

图 3.2 用户登录逻辑图

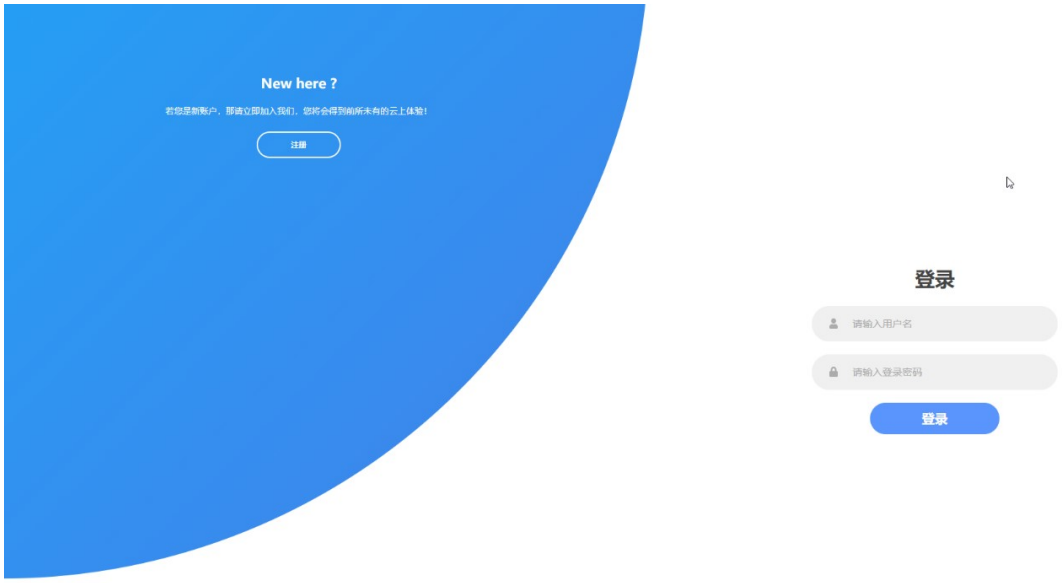

图 3.3 用户登录界面

具体的实现逻辑如下：当用户访问登录页面时，前端浏览器首先和后端程序通讯注册当前用户的访问的临时会话通道。当用户在图 3.3 所示的登录界面输入完密码并点击登录按钮时，前端应用会将用户号的密码先进行第一轮 MD5 加密和用户账户的数据包发送到后端接口时，系统首先查询日志数据库，尝试捕获一级和二级风控。如果成功捕获一级或二级风控，系统会直接拒绝用户的登录请求，并响应前端浏览器触发风控登录限制。否则系统会效验当前用户的用户名和密码信息，并将效验结果写入到登录日志中。如果用户的密码效验正确，系统会尝试进行三级风控捕获，当成功捕获到三级风险时，系统会响应前端浏览器进行手机验证码二步认证，并向用户手机发送验证码消息来确认用户的身份信息，若没有捕获三级风控，则系统将告知前端浏览器当前用户登录成功，并在当前的 SESSION 空间内写入用户基本信息数据。

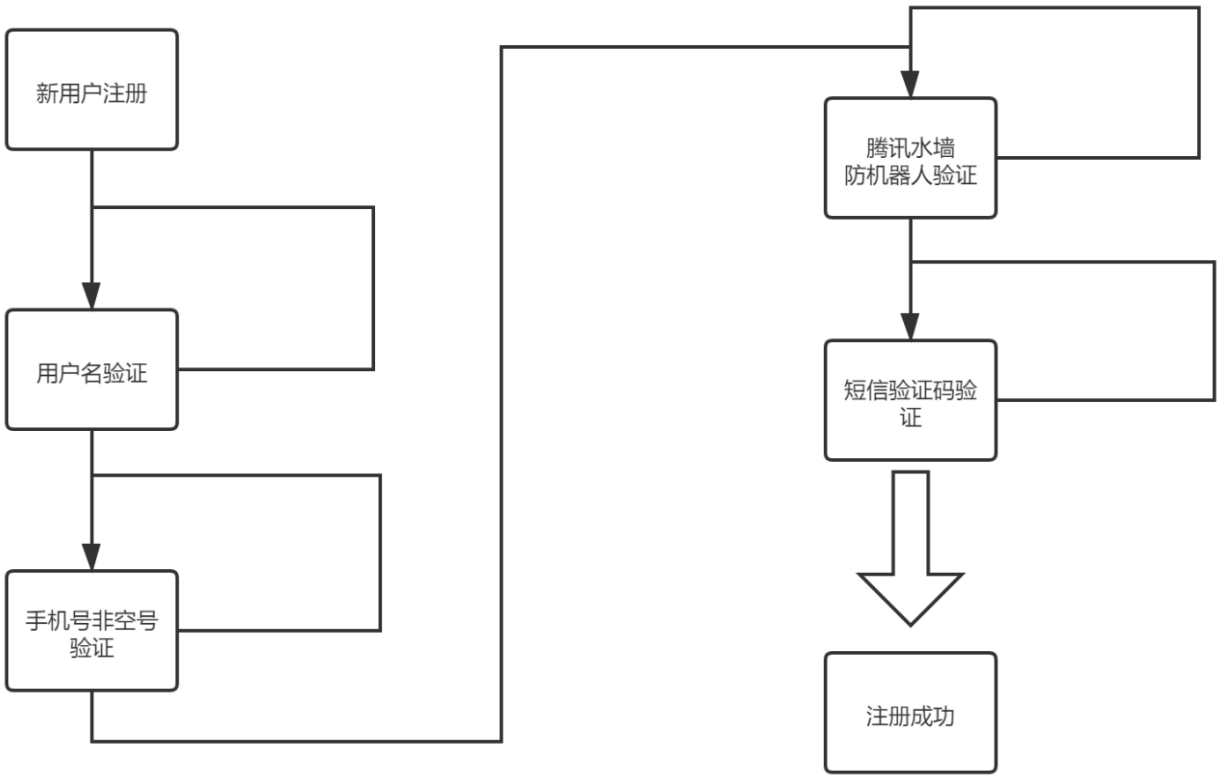

图 3. 4 用户注册逻辑图

系统内置的流控制器将严格限制用户注册时的验证步骤。图 3.4 展示的是系统后台对注册用户密码的加密流程。用户必须按照图 3.4 所示的过程进行逐步效验，任何一个步骤均不可跳过或跨越，否则系统将响应流票据出错。如图 3.5 所示用户在注册新用户时，系统首先会查证指定的用户的用户名是否已经在系统中绑定注册并且符合规则规定，然后将要求用户输入手机号。系统将依据云端大数据和自建数据库自动判断尝试注册的手机号码是否为空号，欠费号，羊毛号，并自动阻断相关号码的注册流程。当系统后台认为当前用户的手机号不存在安全风险时，后台接口将要求前端浏览器调起腾讯防机器人水墙模块对恶意注册的机器人行为进行验证。用户需要滑动图 3.6 所示的滑块验证码完成人机验证，此时系统将对申请用户申请注册的手机发送短信验证码，验证当前手机号是否归属当前用户所有，并且能够正常的接收到来自系统的短信验证码。用户输入正确的验证码后，系统会在数据库中插入当前注册用户的基本信息，并生成供用户密码加密的随机 TOKEN。

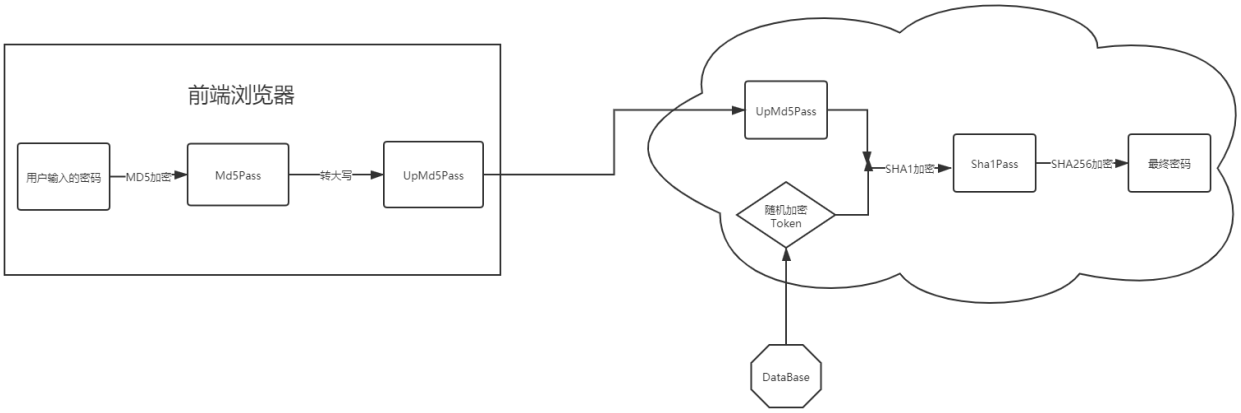

图 3.4 用户密码加密流程

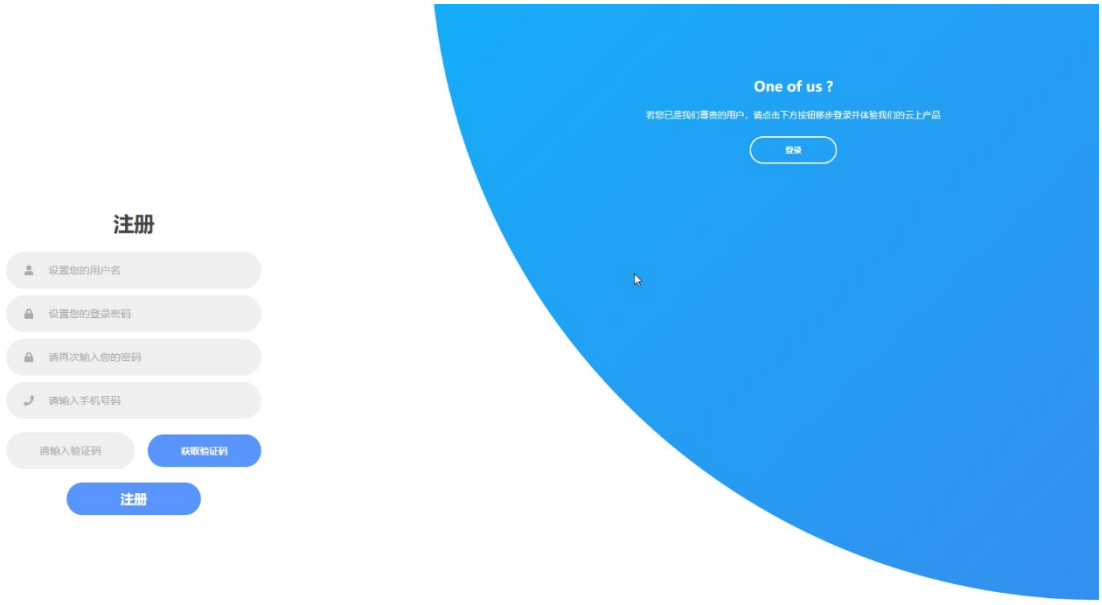

图 3.5 用户注册界面

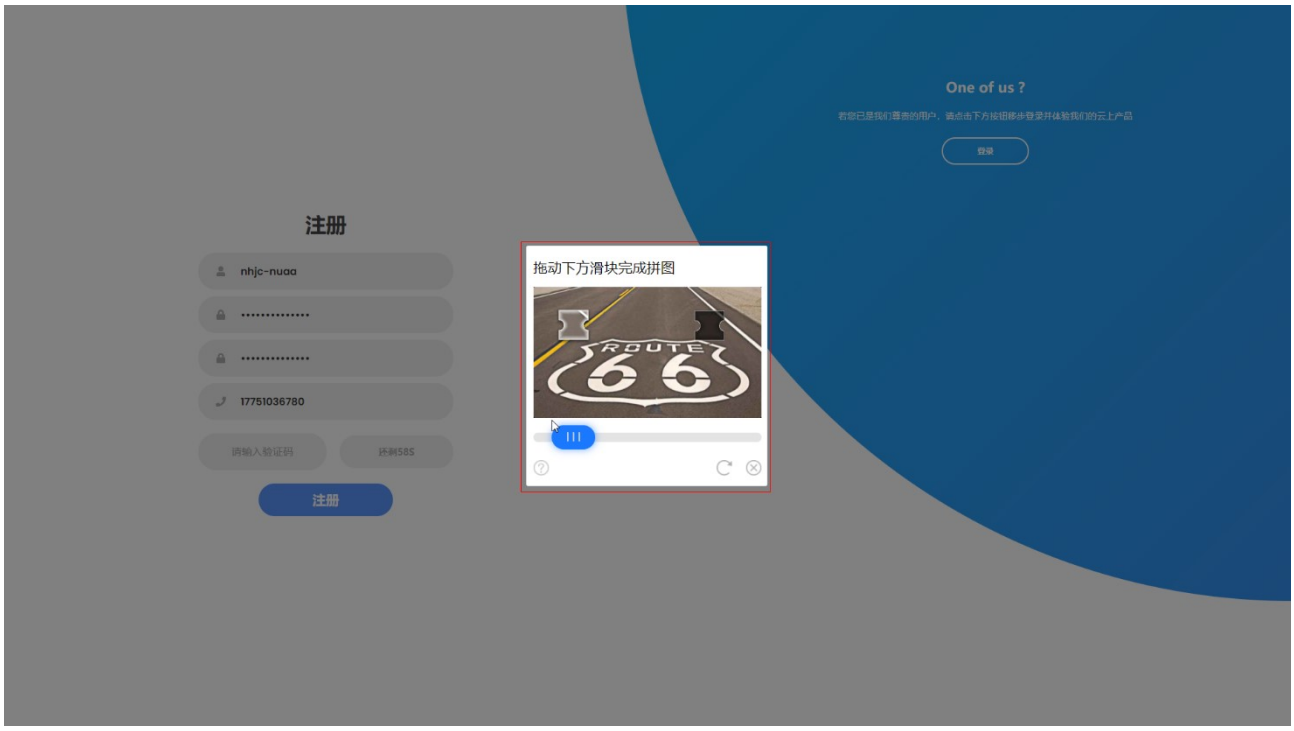

图 3.6 用户注册界面的腾讯防水墙验证

下面是系统对不同等级的风控事件的评估标准以及可能会采取的行动。

1.最高等级风险（永久封禁 ）

系统会针对当前请求的 User-Agent,ClientId,Ip 进行精确匹配，直接禁止被匹配到的设备登录到系统。注意基于 IP 地址的封禁对 Api 请求同样生效，且不会自动删除。

规则：

（1）同一 IP 地址，在一小时内累计统计到 300 次错误登陆 ，将永久禁用此 IP

（2）同一 ClientId，在一小时内累计统计到 2 次错误登陆,将永久禁用此 ClientId

2. 中度分享控制（一定时间内的访问控制）

系统会针对当前请求的 User-Agent,ClientId,Ip 进行精确匹配，并对匹配结果采取一段时间内的访问控制。

规则：

（1）同一个 IP 地址，在 5 分钟内被累计统计到 100 次错误登陆，将禁止此 Ip 地址登录系统 15 分钟

（2）同一个 UserId，在 5 分钟内被累计统计到 15 次错误登陆，将禁止此 UserId 登录

系统 15 分钟

(3) 同一个 ClientId，在 5 分钟内,被累计统计到 15 此错误登陆，将禁止此 ClientId 登录到系统 15 分钟

（4）同一个 ClientId,UserId,在 5 分钟内，累计统计到 5 此登陆错误,将禁止此 UserId 登录到系统 15 分钟

3.低风险（需要二步认证）：

系统会针对当前请求的信息 进行处理

规则：

（1)登录地与最近 10 次登陆地所在的省、市、区位置存在不同

（2)2 小时内 IP,ClientId , UserId 有过被中度风险封禁的记录

（3)连续两次输入错误的账号密码

3. 3 用户前台

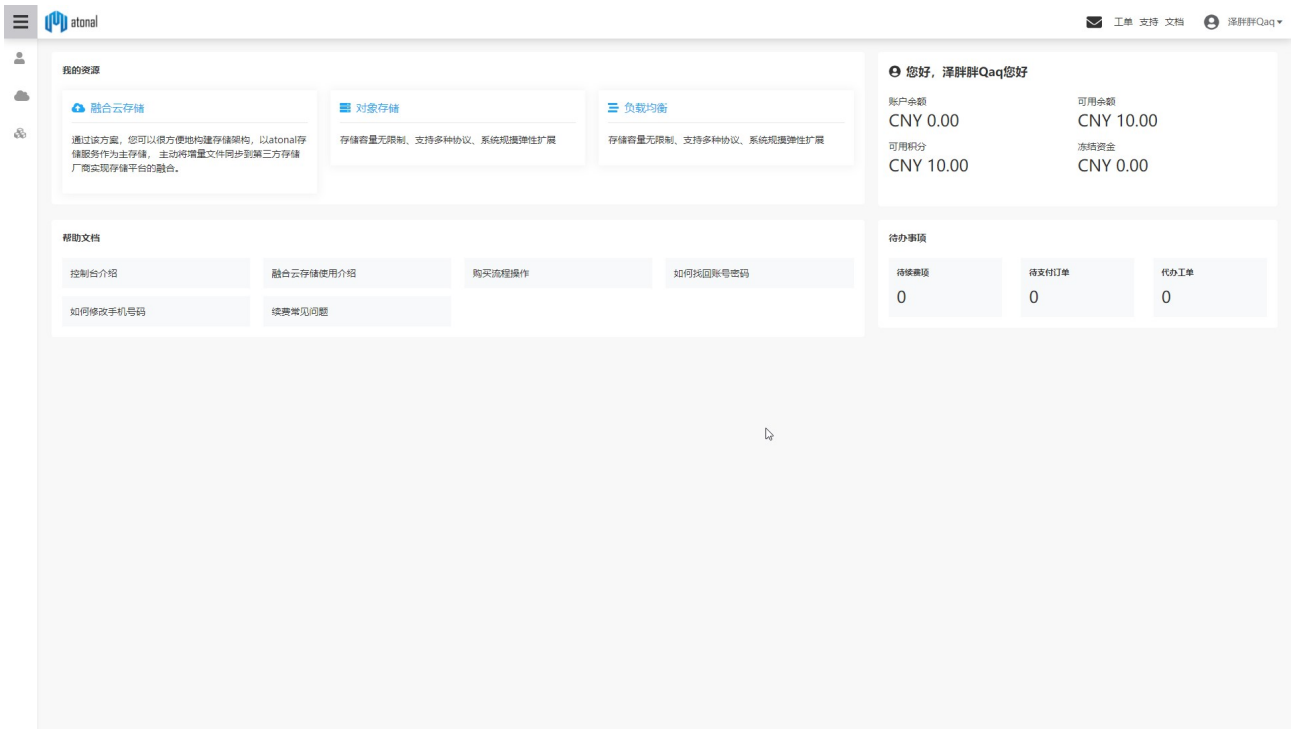

图 3. 7 用户前台首页

当用户登录到系统后，会看到如图 3.7 所示的用户前台首页。用户可以在本页面中查看当前账户的余额积分信息，可点击我的资源选修卡，快速访问用户在系统中也有的业务资

源，并对其进行配置。用户可点击帮助文档选项卡阅读本系统中一些常见问题的解决说明和方案。可在待办项选项卡中查看当前用户需要处理的工单、需要续费的项目和需要支付的订单信息。

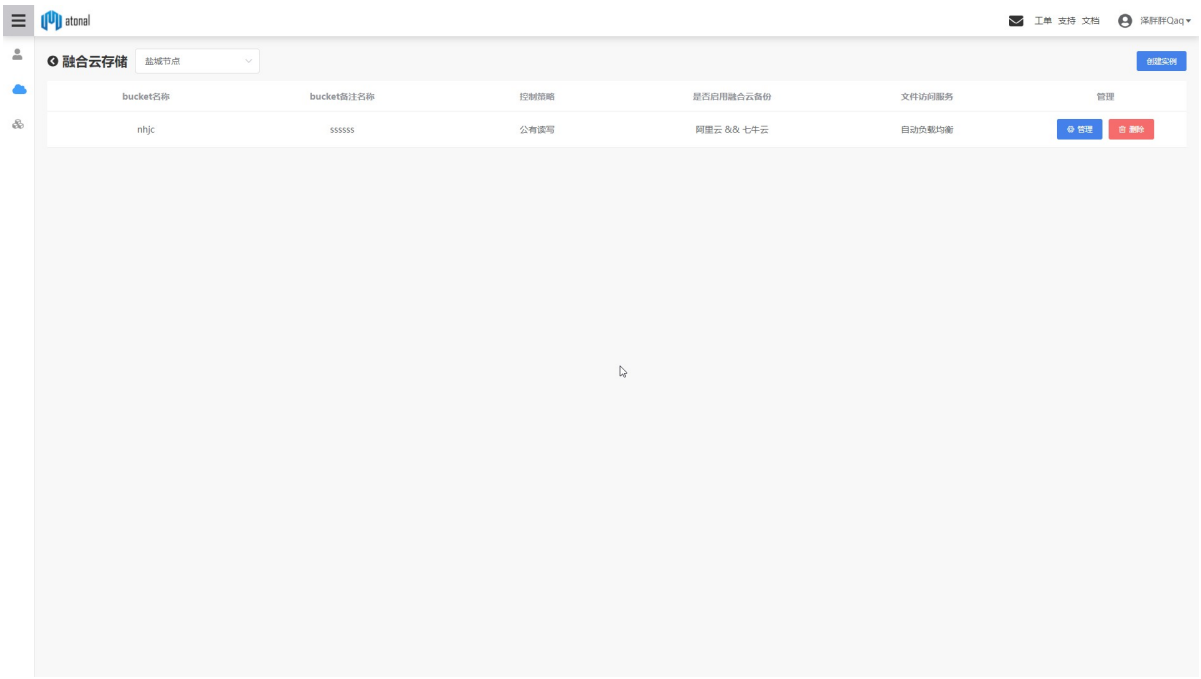

图 3.8 融合云存储业务首页

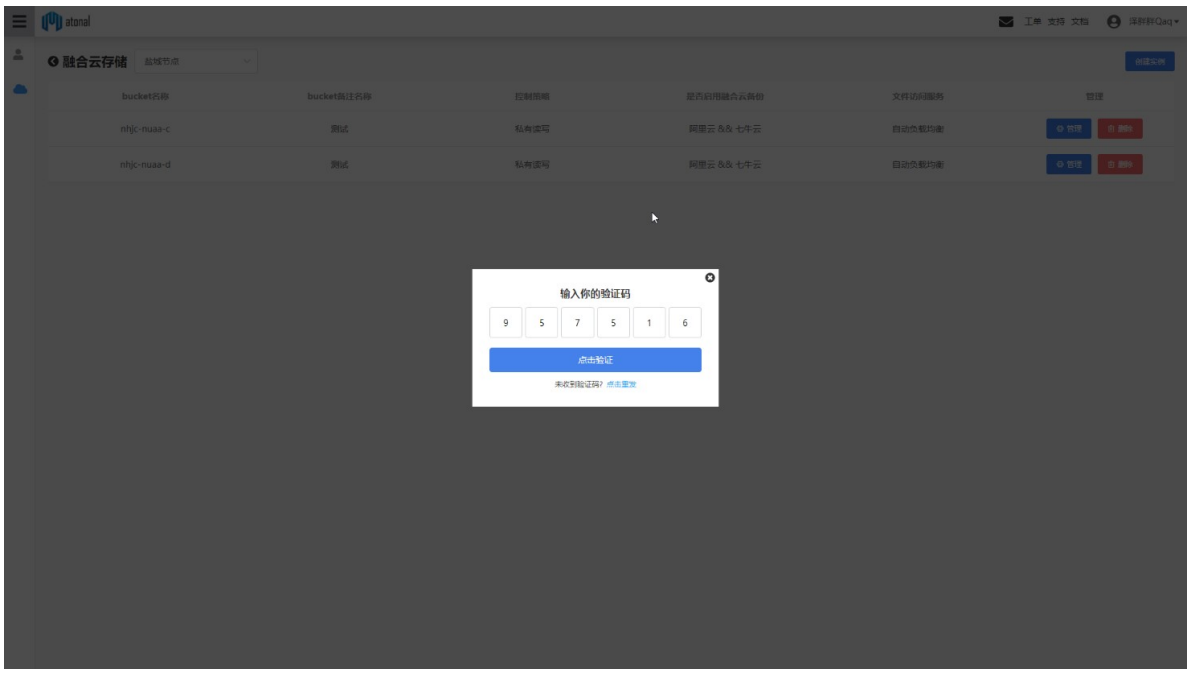

图 3.9 融合云存储删除存储桶二步短信验证

图 3.8 主要展示了本课题的核心业务，融合云存储系统的控制管理页面。用户可选择不同的资源存储分区，对不同资源去下的存储桶资源进行管理。

图 3.9 展示的是当用户进行删除存储这种高危操作时，需要验证手机验证码进行二步认证，避免用户误操作和恶意操作，极大程度上的保护了用户的数据的安全性。

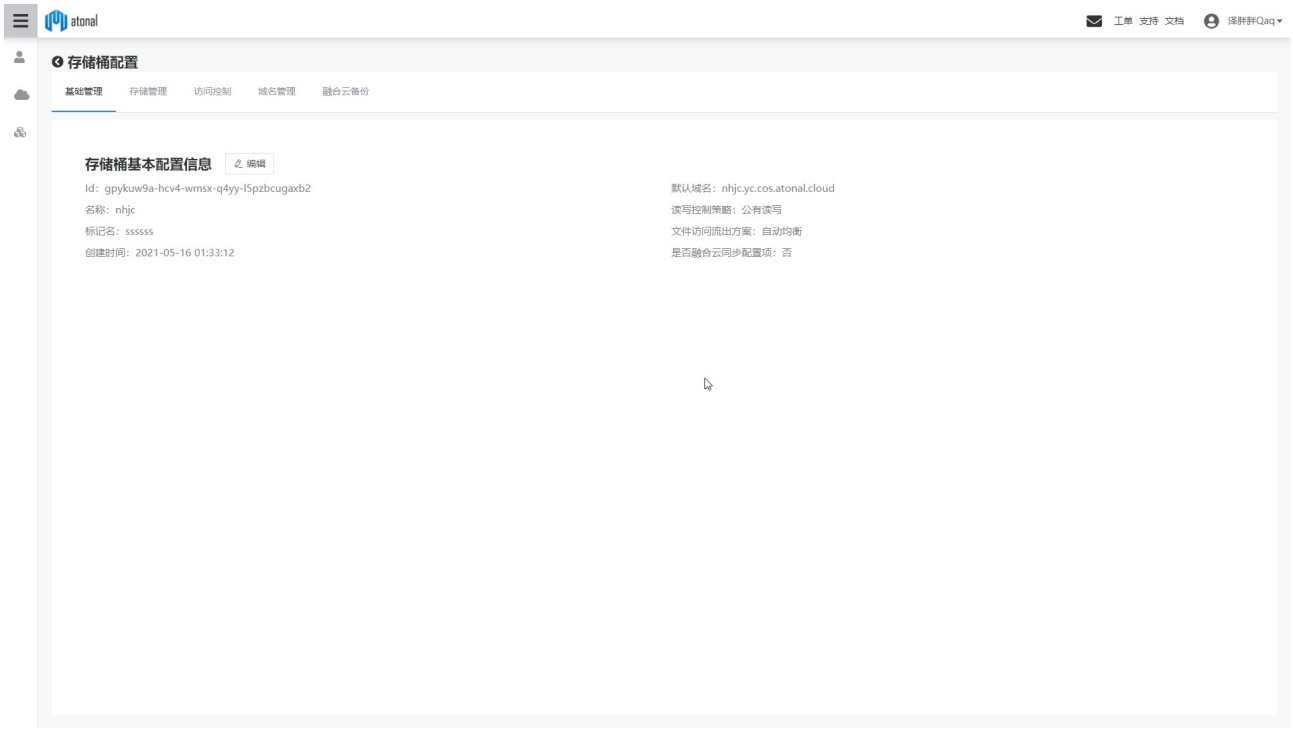

图 3.10 存储桶基本信息页面

用户可以点击存储桶旁边的管理按钮，进入图 3.10 所示的存储桶基本信息的管理页面，查看当前存储的桶的基础情况信息，点击上方菜单栏中的存储管理按钮，可以对用户存储桶下的文件资源进行管理。系统会自动按照文件结构展示用户文件信息。用户可在本界面中对存储的文件进行复制、移动、删除、重命名等操作。用户可以通过本界面将本地的文件上传到系统，系统会根据用户存储桶配置的融合云存储策略自动同步备份到阿里云或七牛云存储中，用户可自由针对每一个文件设置被访问时的读出策略，选择合适的数据流出源，并可针对每一个文件配置单独的访问权限。

如图 3.11 所示，用户可在前端浏览器中直接上传文件到存储桶中。针对每一文件，系统支持配置成自动均衡、阿里云流出、七牛云流出这三种不同的数据流出方式。当用户选择使用自动均衡时，系统会根据文件的大小、文件的类型、系统服务器的网络、存储负载情况

动态地决策直接从系统向用户流出数据，还是通过重定向引导用户访问存储在阿里云或七牛云上的冗余备份资源。

针对每一个文件，系统支持配置成公有读写、公有读私有写、私有读写这三种不同的文件访问权限。当用户不配置时，系统默认继承存储桶的访问控制权限。如用户配置成私有读写权限，那么该文件的删除和下载访问均需要通过 API 密钥认证当前用户的访问权限；如果用户配置成公有读私有写，那么文件可以被任意知道文件地址的人读取，但文件的删除和修改需要通过 API 鉴权才能实现；如果用户配置为公有读写，则任何人都可以在不经过 API 密钥认证的情况下对文件进行读写操作。系统支持文件秒传功能，即对用应当前服务器上已经存在的文件，用户不需要重复进行上传。采用的原理是，当时用户在上传文件到系统时，需要在创建服务器上传任务的第一个包体里面携带需要上传的文件的 MD5 信息，系统通过检索文件特征值 MD5 来判断系统中是否已经存在系相同的文件，如果存在对应特征值和需要上传的文件的 MD5 的值相同的文件，那么系统会自动在用户目录下创建逻辑文件，并告知用户秒传已经成功。如果服务器不存在用户需要上传的文件的 MD5，根据文件的大小不同，用户可以选择通过文件简单上传到服务器（即将文件封装在一个单独的 HTTP 表单请求中一次性上传到服务器上)），或者采用文件分片上传的方式将文件切片上传到服务器（即通过对文件进行切割，分别简单上传多个比较小的文件，图 3.12 展示为简单上传和分片上传的区别），服务器校验文件分片的 MD5 信息后对文件进行合并。

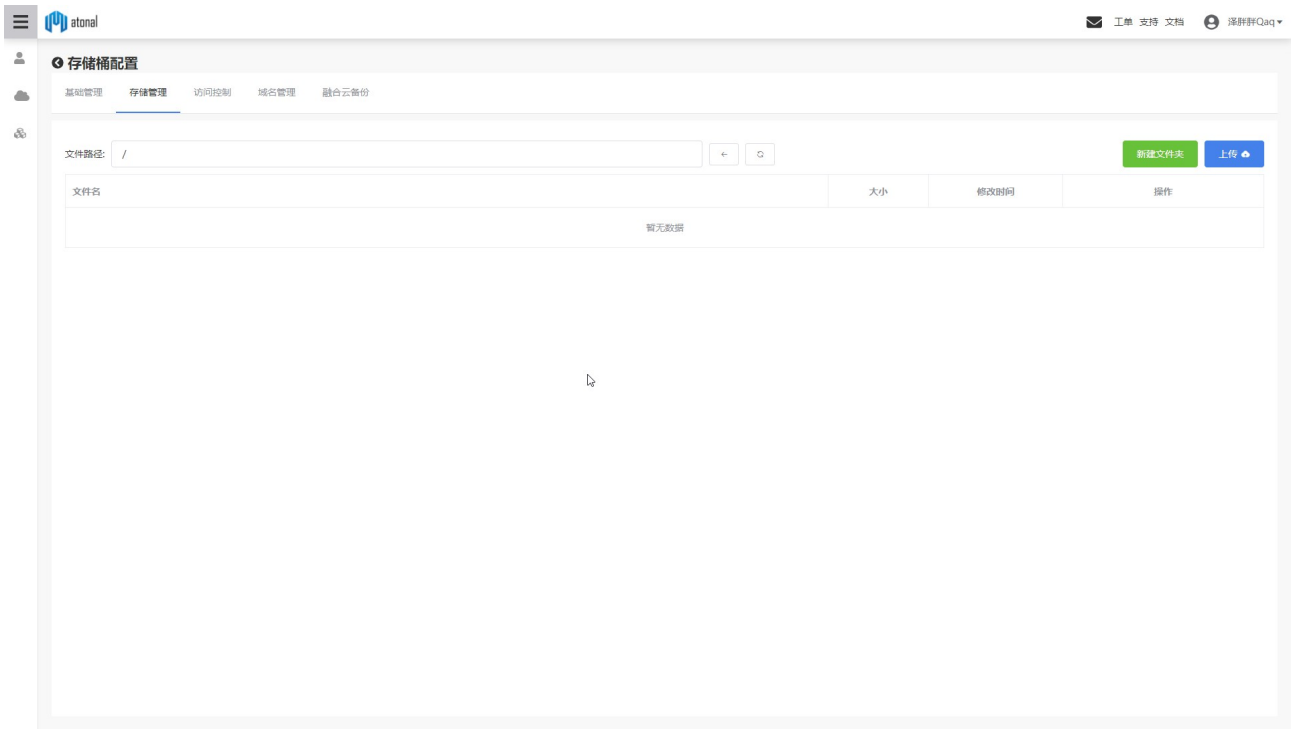

图 3.11 存储桶文件管理页面

图 3.12 简单上传和分片上传

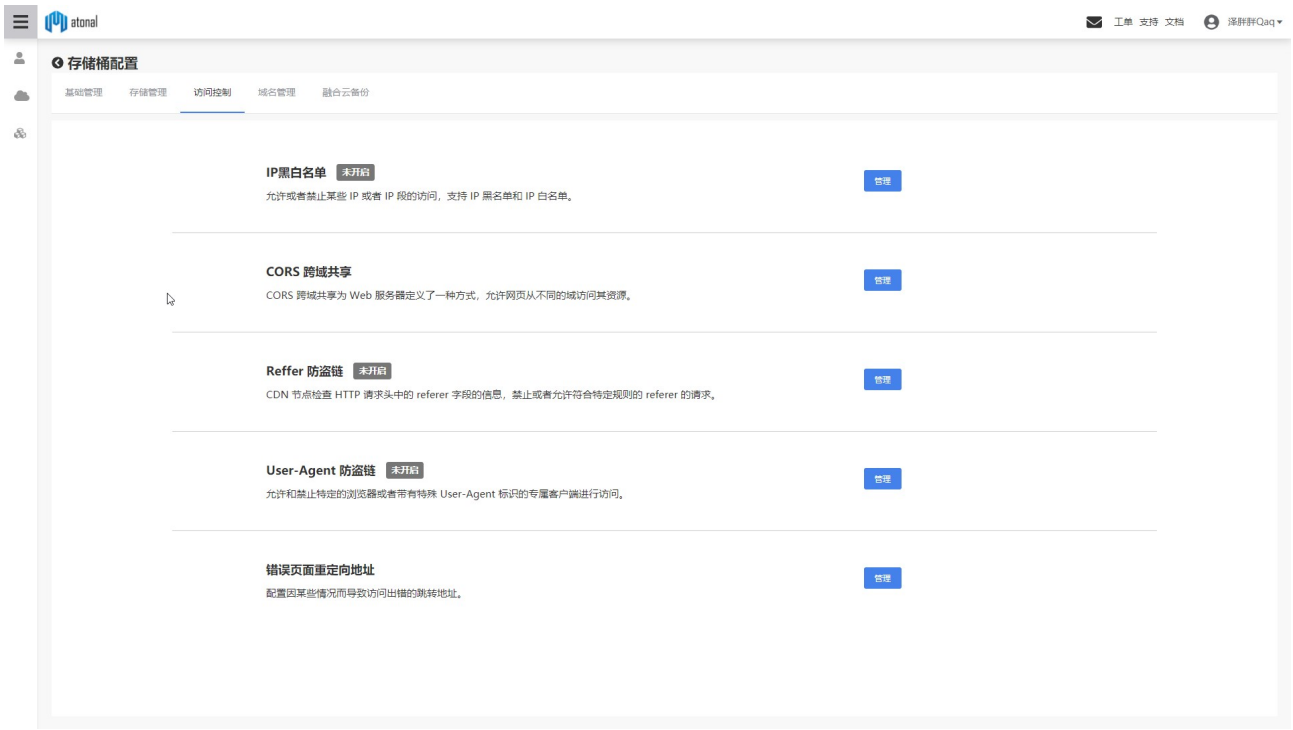

图 3. 13 存储桶访问权限配置页面

用户可以点击访问控制按钮 配置存储桶中的文件的访问控制功能。系统会自动根据来访客户端的 IP 地址、Reffer、User-Agent 这些特征信息，按照用户指定的访问规则进行控制，自动阻止或放行指定的请求，防止用户在存储桶的中的资源被恶意用户恶意访问而导致高额流量费用的损失。通过 Reffer 防盗链功能，用户还可以确保存储桶中存储的文件资源只被指定的前端网站访问，避免有其它站长恶意盗取链接的情况。

CROS 选项则支持用户在前端界面中直接对存储桶发起请求，用户可以自行配置暴露的请求头部、允许的请求头部、请求域名地址和缓存时间等信息，从而实现前端界面跨域对存储桶中的资源进行读写操作。

如图 3.14 所示用户可以点击域名管理按钮，访问存储通的域名管理界面。用户可以给存储桶配置自定义的域名，并给域名部署相关的 SSL 证书，在配置 SSL 证书的情况下，系统支持配置存储桶仅允许使用 HTTPS 资源进行访问。系统会为用户创建的存储桶分配通配符域名，用户可以直接使用系统默认分配的通配符域名来访问用户账户下的资源。为了方便用户进行调试，用户可以访问图 3.15 所示的地址来测试存储桶的解析配置信息。

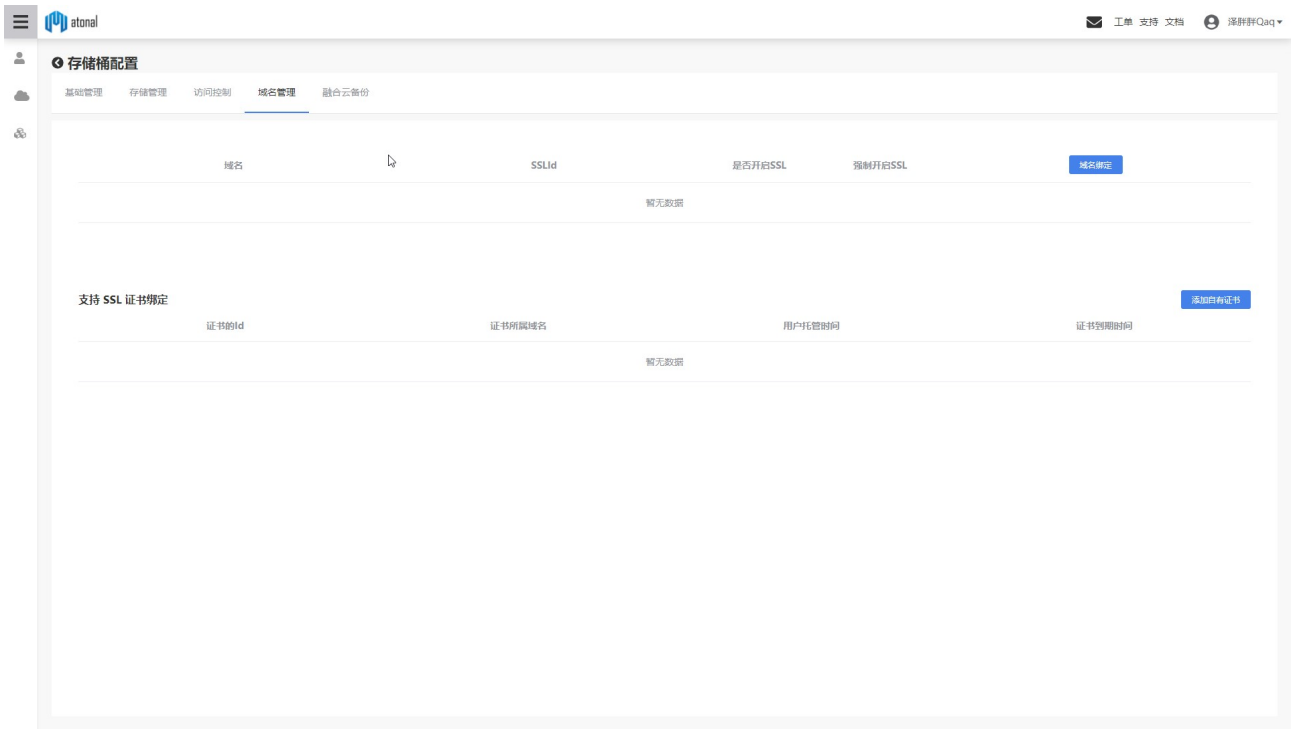

图 3. 14 存储桶访问域名管理页面

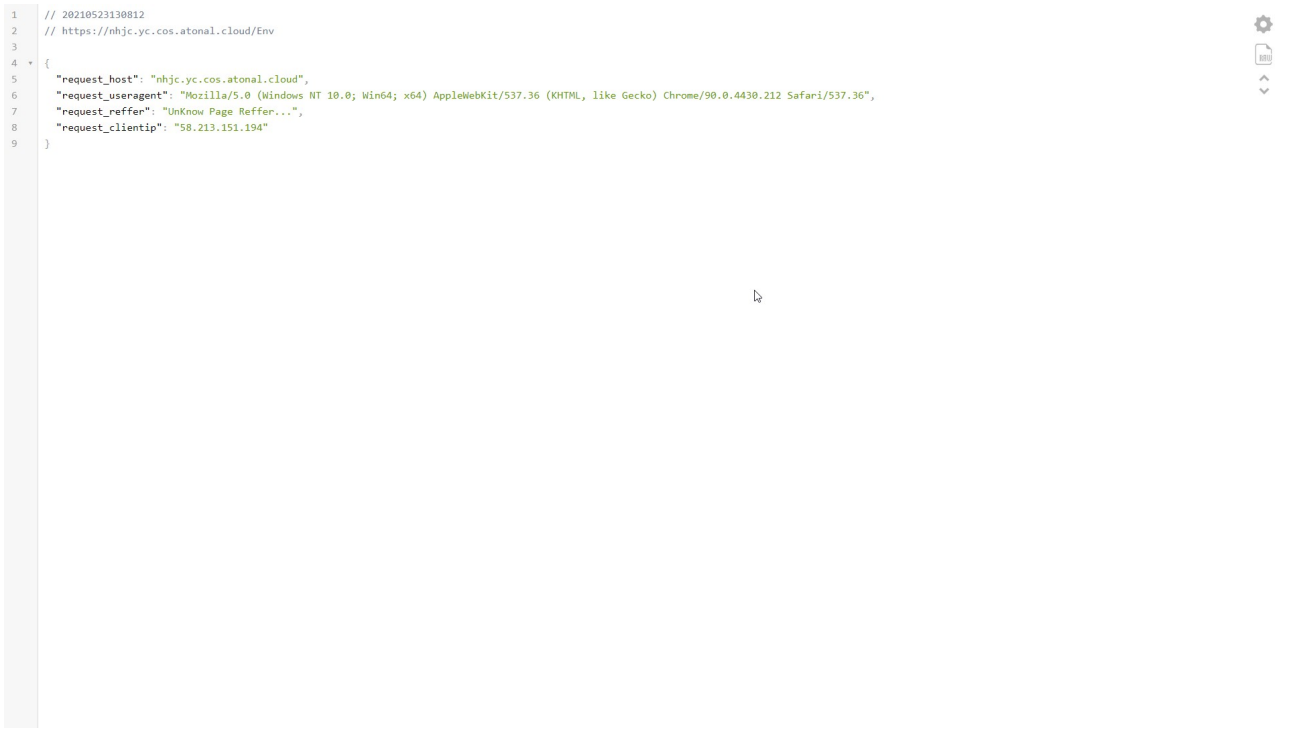

图 3. 15 存储桶环境变量显示界面

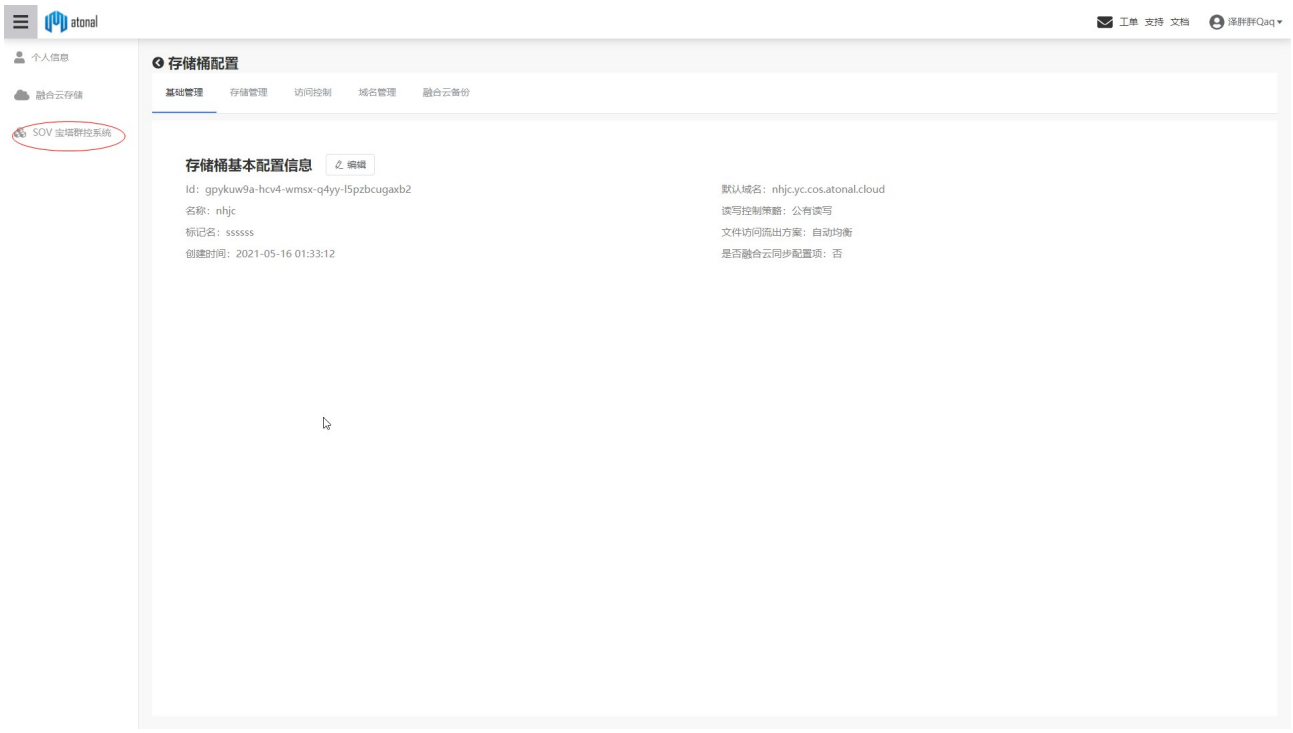

图 3. 16 SOV 服务器菜单栏

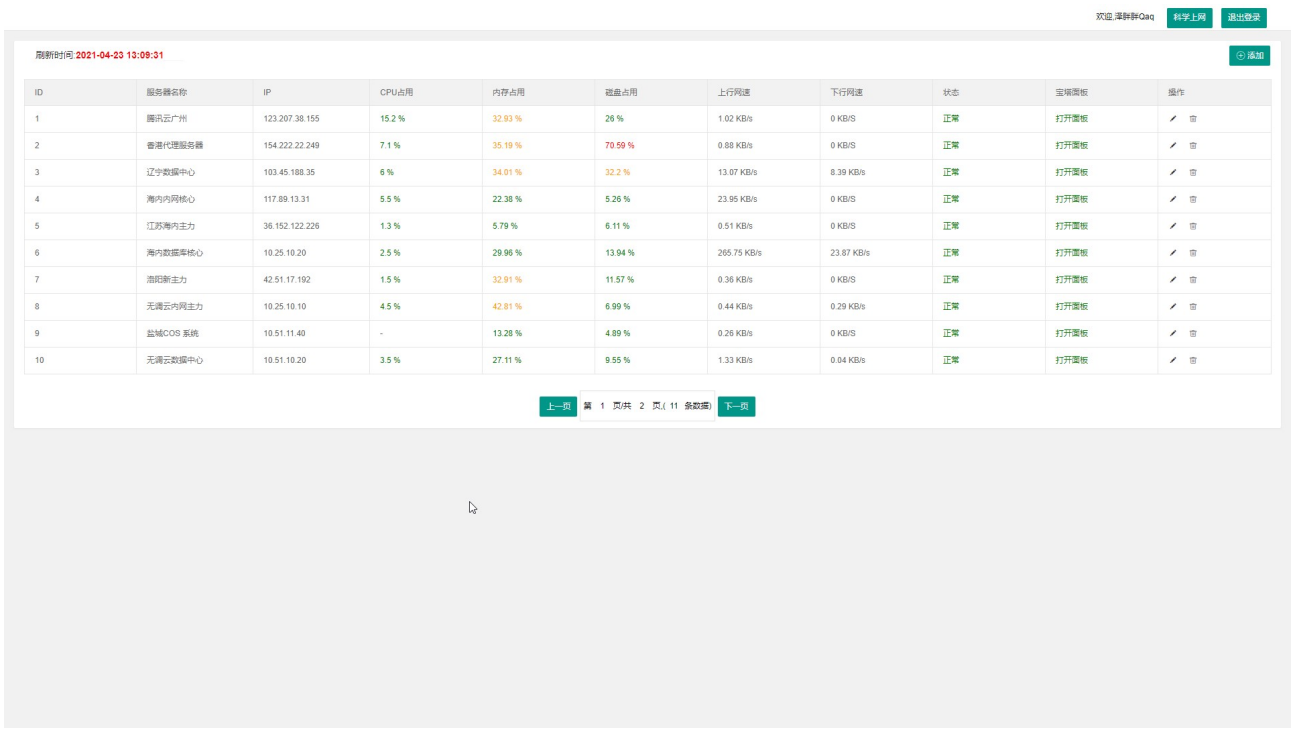

图 3. 17 SOV 服务器管理系统

系统支持使用 SSO 免登录授权。用户可以点击图 3.16 所示的后台界面菜单栏中的 SOV

服务器集群管理功能按钮，一键免登录访问系统内部互相信任的其他应用模块。具体流程是，用户在前端页面中点击相应的按钮，向系统后台发起一个指定应用的 SSO 免密登录请求，系统后台根据当前用户输入的 UserId 和 SSO 应用的私钥计算本次登录的 Token 签名，然后将用户的 UserId、Token 签名和应用的 AppId 封装到 URL 中。通过前端浏览器引导用户跳转，第三方应用拿到用户的 UserId 和 Token 签名后对请求进行校验，校验通过后执行后端的会话登录逻辑，并重定向用户到系统内部页面（如图 3.17 所示）。

3.4 业务中台

业务中台是系统管理人员的后台，管理员可以在业务中台中对系统进行各项配置。

3.4.1 权限及角色管理

图 3.18 展示的是系统的用户角色管理页面。管理员可以在本页面中根据用户所需要访问的功能，分工不同，个性化的创建不同的用户角色，通过设置不用用于所扮演的不同的用户角色来约束用户所具备的访问权限。图 3.19 展示的是系统的模块权限管理页面，系统的模块管理功能和用户角色发生关联，管理员可制定不同模块的角色访问规则。当且仅当用户所在的角色拥有访问该模块的访问权限的时候，用户才能够在右上角的用户个人信息的下拉列表中找到对应的模块，并点击响应的模块对其进行访问。

图 3.20 和图 3.21 展示的是系统的接口权限管理页面，管理员可以针对系统中所有需要进行鉴权的接口进行权限配置。管理员可以在该页面中按照接口所属的树形结构逻辑，添加不通的接口（需要和后端程序配合），管理员可以点击不同的接口查看指定接口的描述信息，并在右侧的角色权限框中对该接口允许被访问的用户的角色进行配置。当修改的接口节点为某些节点的父节点的时候，用户可以勾选权限管理框中的递归权限控制按钮，系统会自动对该接口下的所有子接口添加子接口不具备的父节点的管理权限。

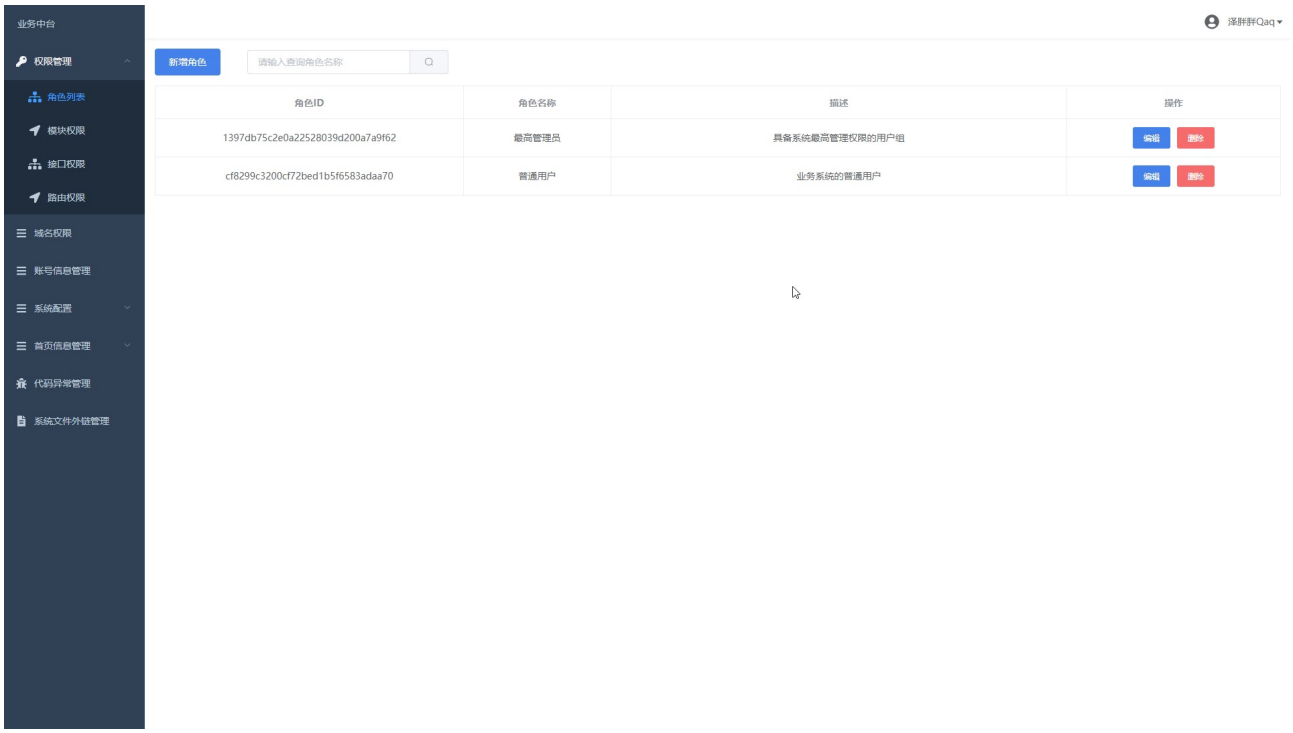

图 3.18 系统用户角色管理页面

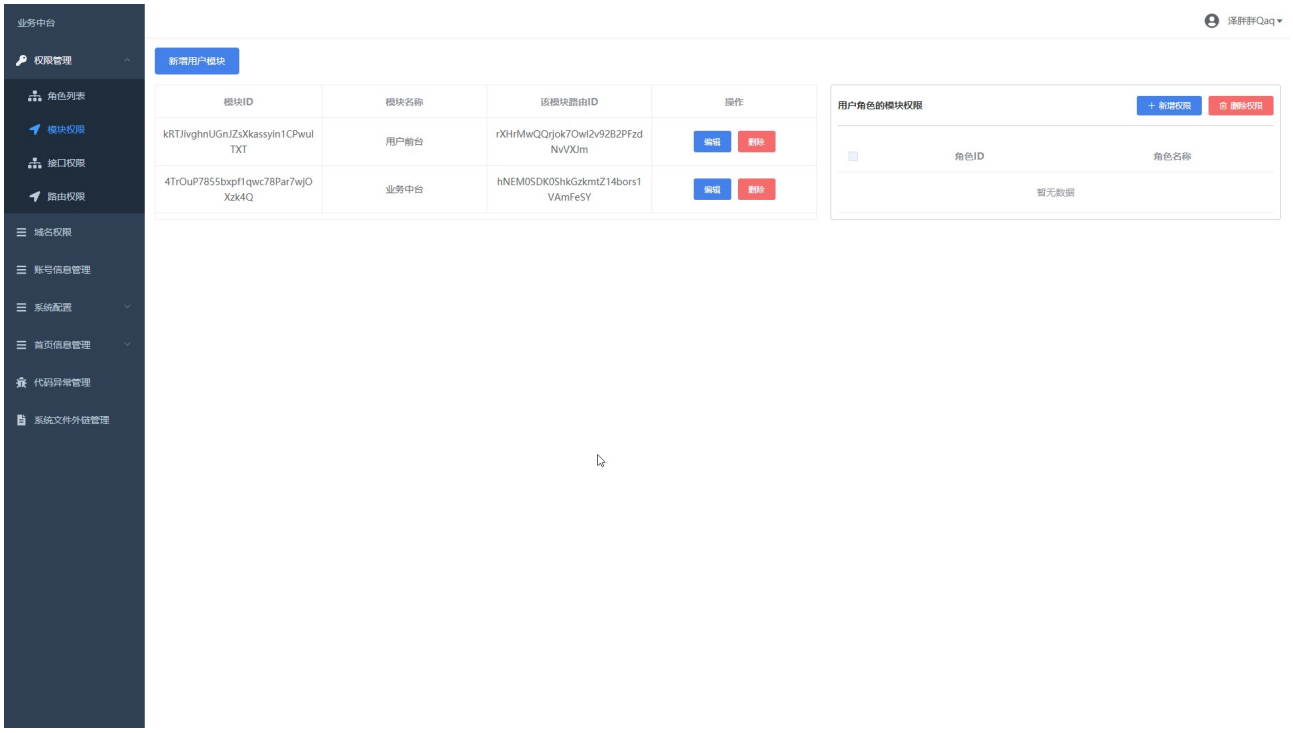

图 3.19 系统的模块权限管理页面

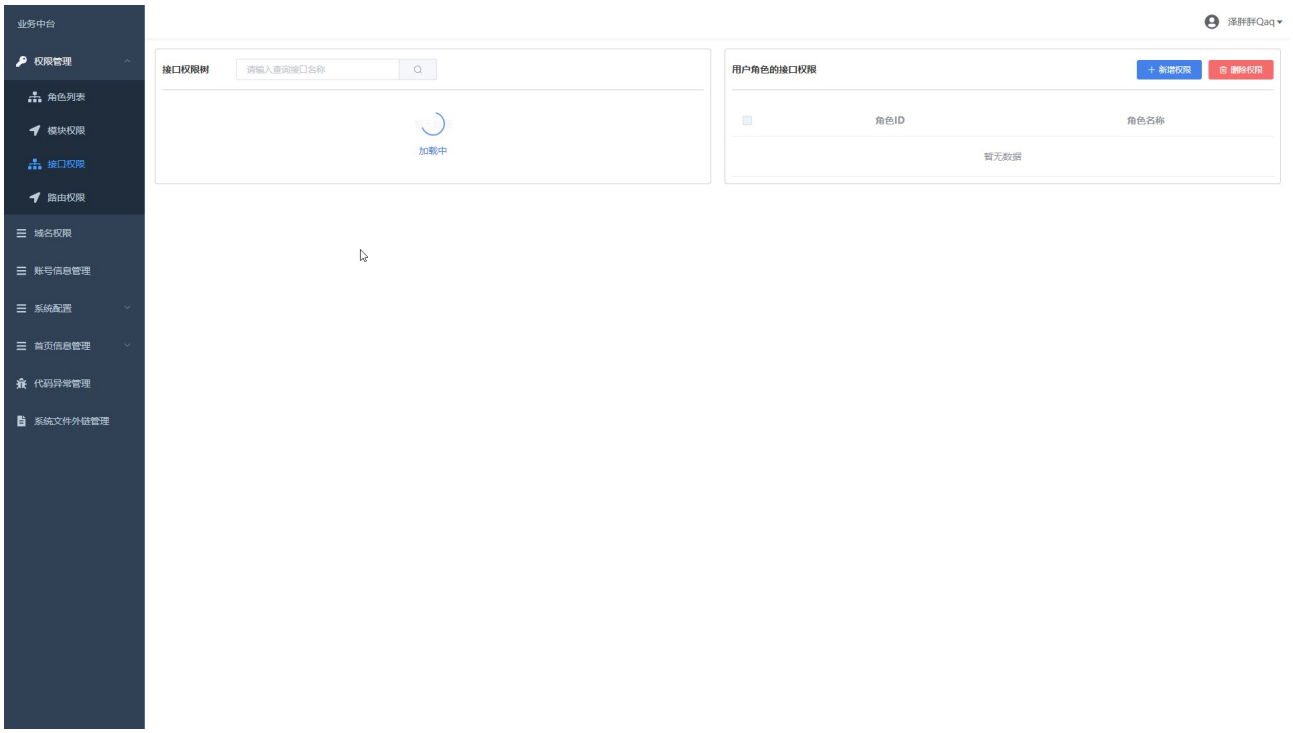

图 3.20 系统的接口权限管理页面 1

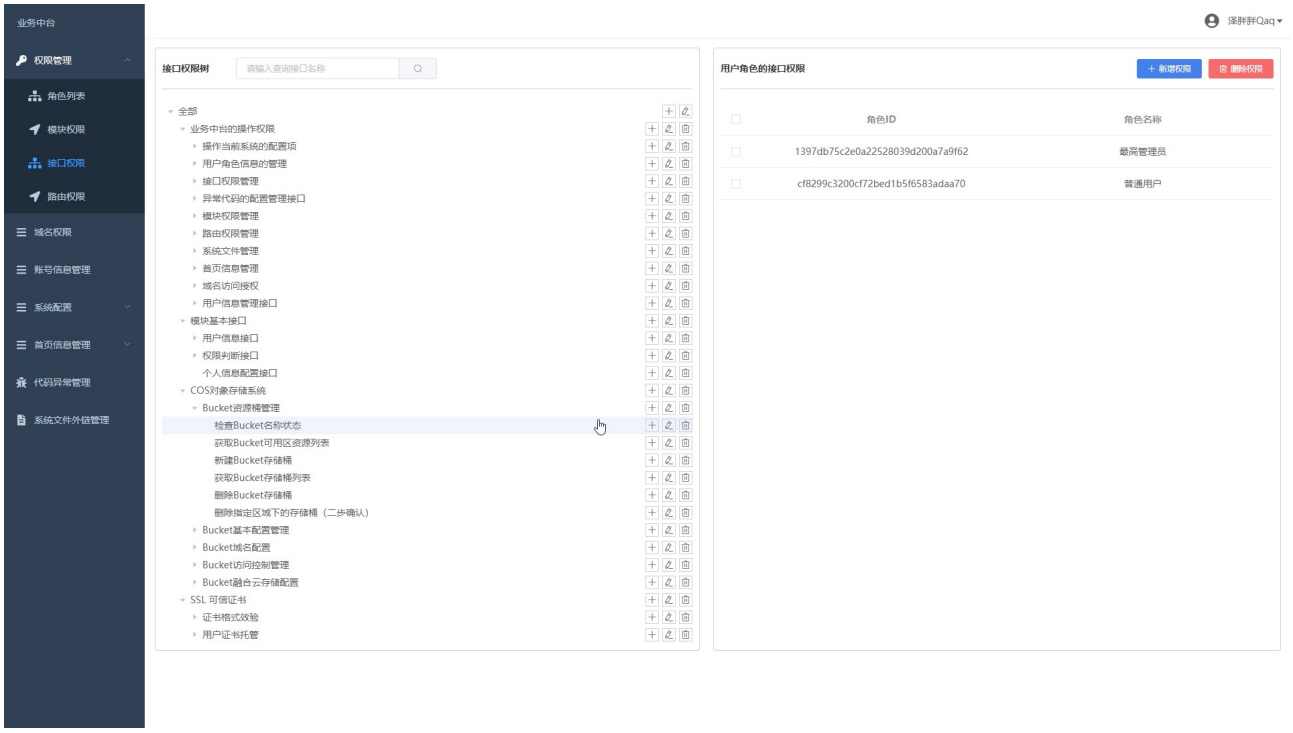

图 3.21 系统的接口权限管理页面 2

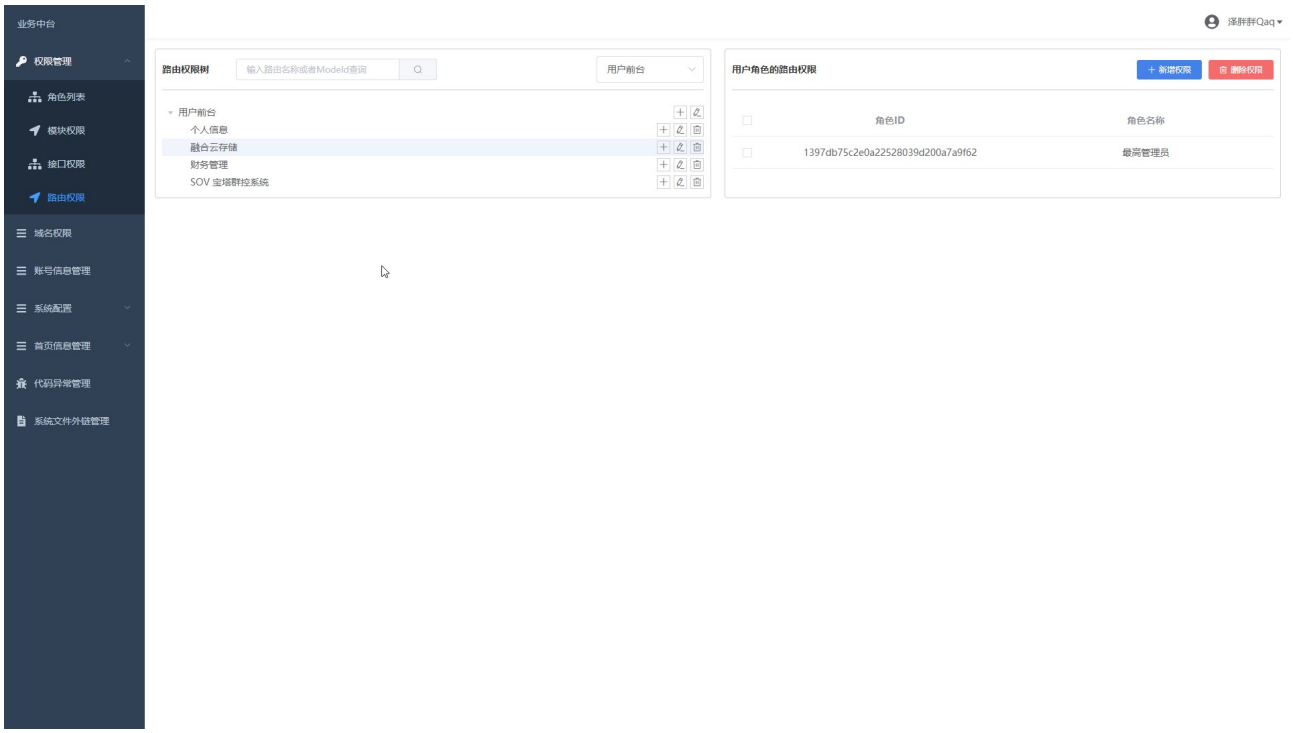

图 3. 22 系统的菜单路由权限管理页面

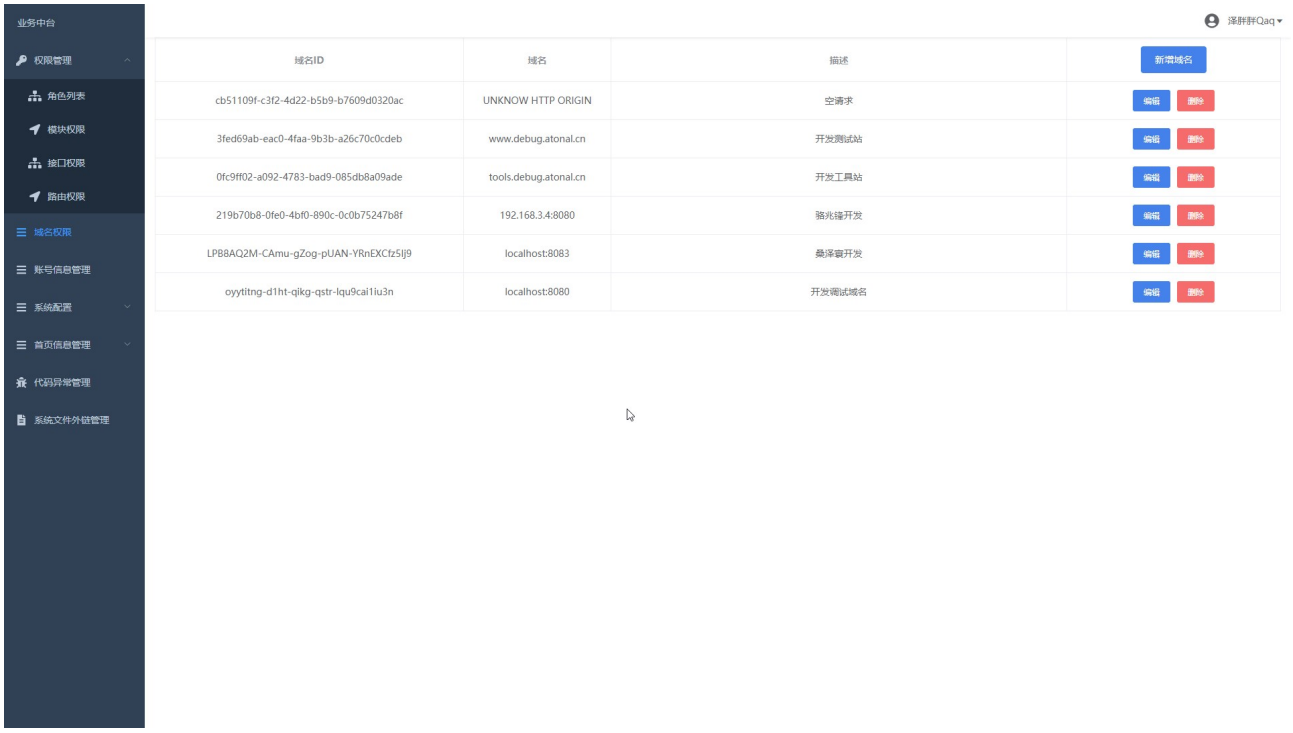

图 3. 23 系统前端访问域名管理页面

图 3.22 展示的是系统的路由权限管理页面，系统的路由权限和模块相互关联。管理员可以通过切换不同的系统模块，对当前系统中的路由权限进行管理配置。管理员可以拖动路由条目对路由菜单的排序进行修改，也可以点击不同的路由项目对访问权限进行修改。对于每一条菜单路由，管理员还可以对路由项目的类型进行配置、系统支持菜单项、菜单组、外链这三种不同的配置项，管理员可以配置路由菜单是否在系统中展示，即系统中的部分页面，用户虽然有访问该页面的权限，但其不会在菜单中展示出来，可用于对不同的页面具体的访问的权限进行管理。

图 3.23 展示的是系统的前端访问域名控制界面。系统的相关 API 接口是直接暴露的，任何人都可以跨域对系统的 API 接口进行调用。因此设计了跨域 API 调用时的域名控制。当用户在前端浏览器中对系统中的 API 接口进行访问的时候，系统会自动对当前访问 API 接口的域名进行判断，系统只允许存在于列表中的域名访问 API 接口，否则用户只能通过后台服务器调用系统中的接口。

图 3. 24 系统用户账户信息管理页面

图 3.24 展示的是用户账户的信息管理界面，系统的管理员可直接在当前页面中常看系

统中的所有用户的基本信息，如姓名，用户名，注册日期，邮箱地址等，管理员可以根据具体的业务需求给不同的用户指派不通的角色身份。用户登录到系统后，会自动扮演指定角色的账户，并拥有相关角色的访问权限。当用户忘记了系统的登录密码时，可以通过人工的方式和管理员取得联系，管理员在核实用户的身份后，可以直接在本页面中选定具体用户账户并对其修改密码。

3.4.2 系统配置项及错误代码管理

图 3.25 和 图 3.26 可以对 系统中的环境变量和 全局变量进行查看和修改。系统的环境变量是系统在运行时自动携带的变量，管理员不可以在界面中直接修改，但可以查看相关变量，帮助开发人员快速定位发现问题。全局变量则是系统的在运行过程可以动态修改的一些变量，通常是一些第三方应用的 API 密钥等等。

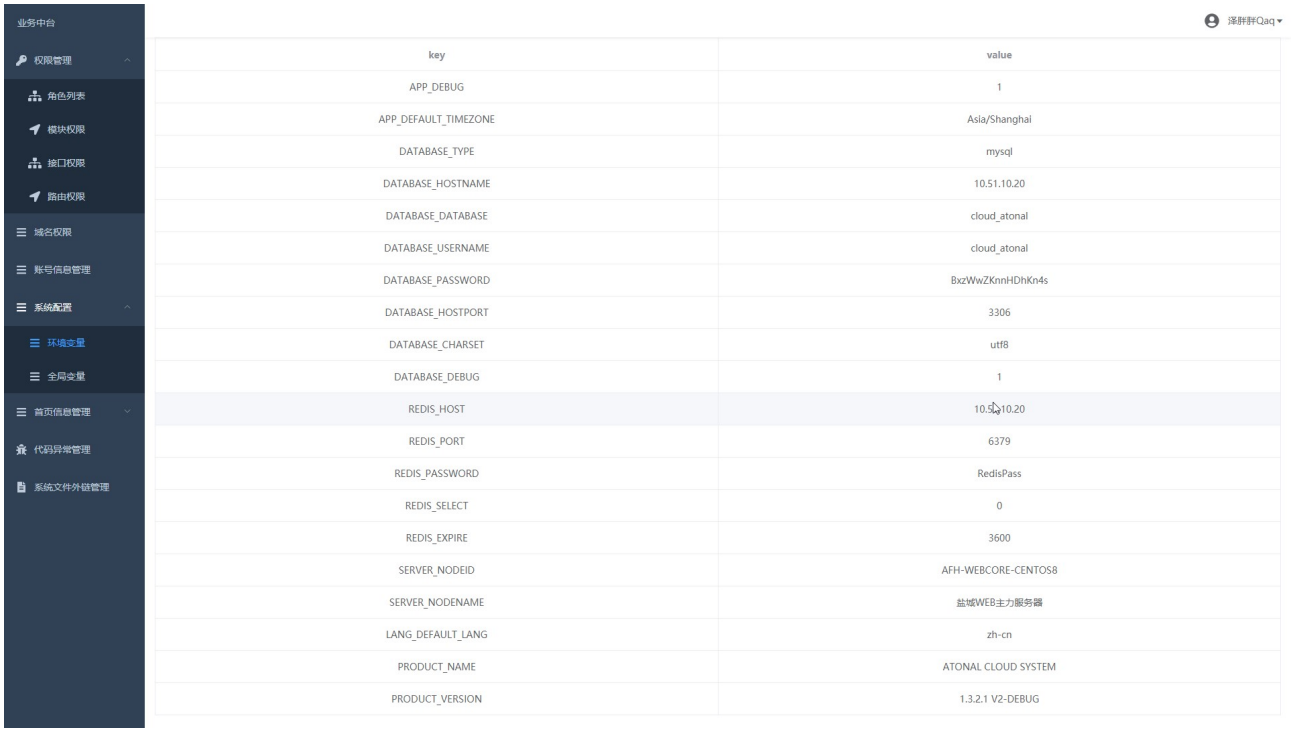

图 3.25 系统的内置环境变量管理页面

图 3.26 系统可修改的全局变量管理页面

3.4.3系统主页及文件外链管理

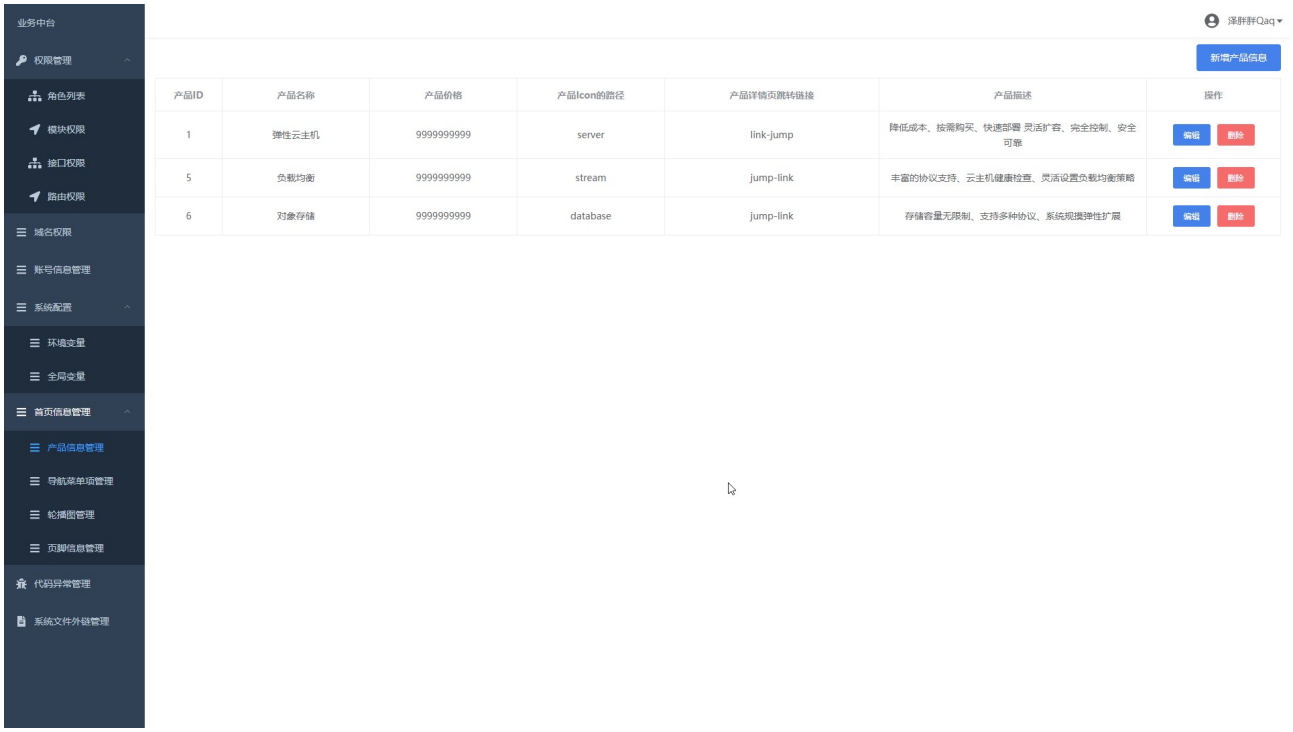

图 3.27 首页产品信息管理页面

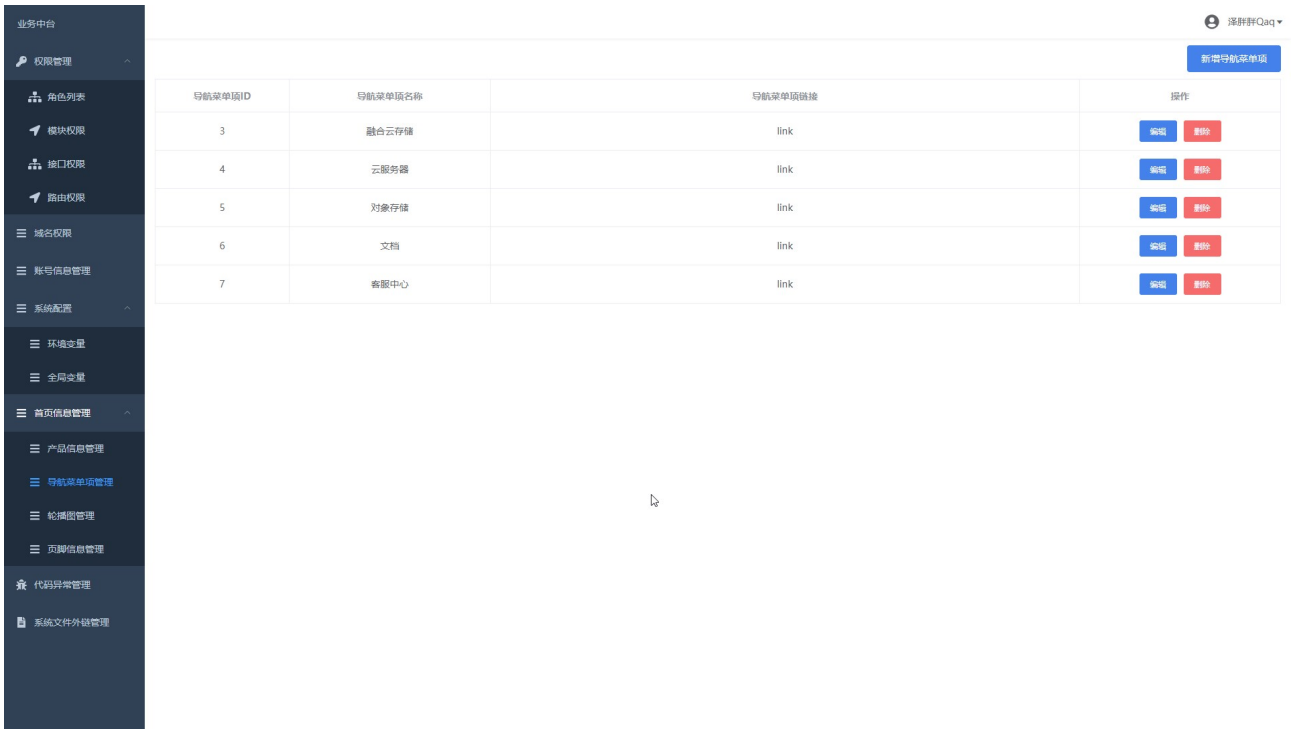

图 3.28 首页导航菜单管理设置页面

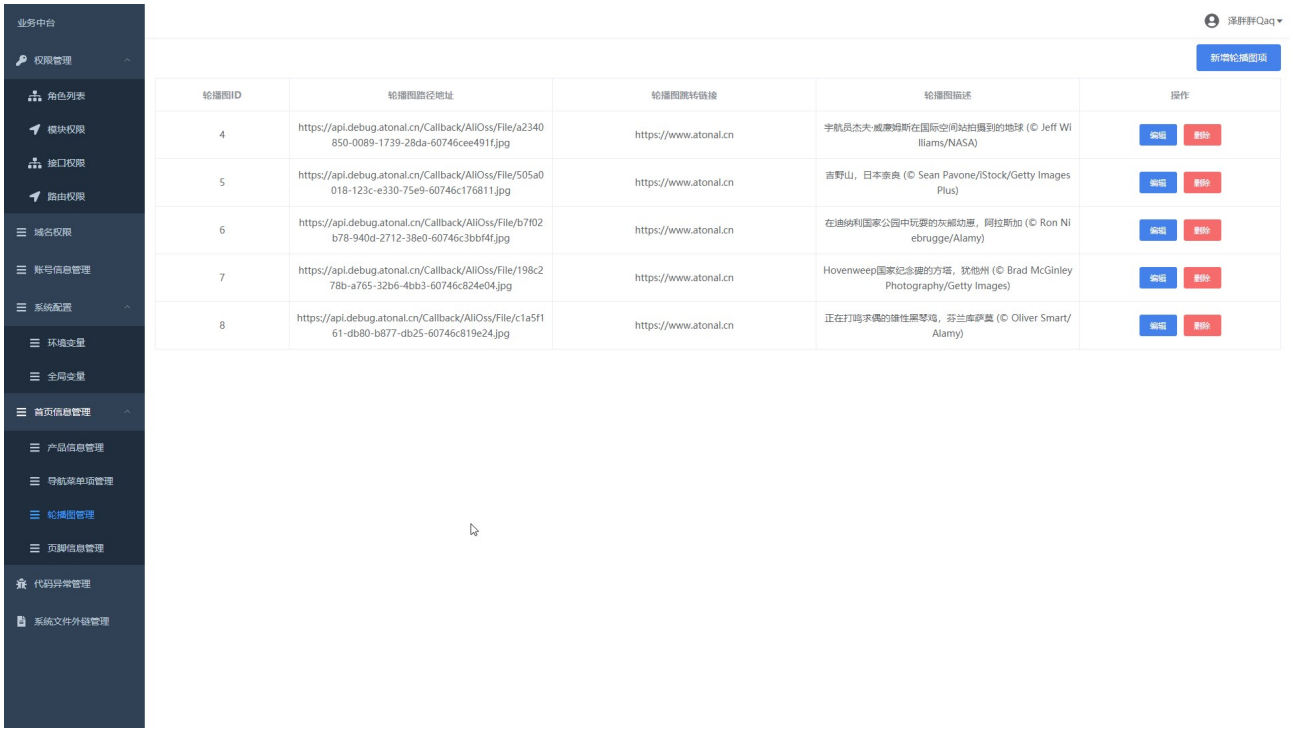

图 3.29 首页轮播图信息管理页面

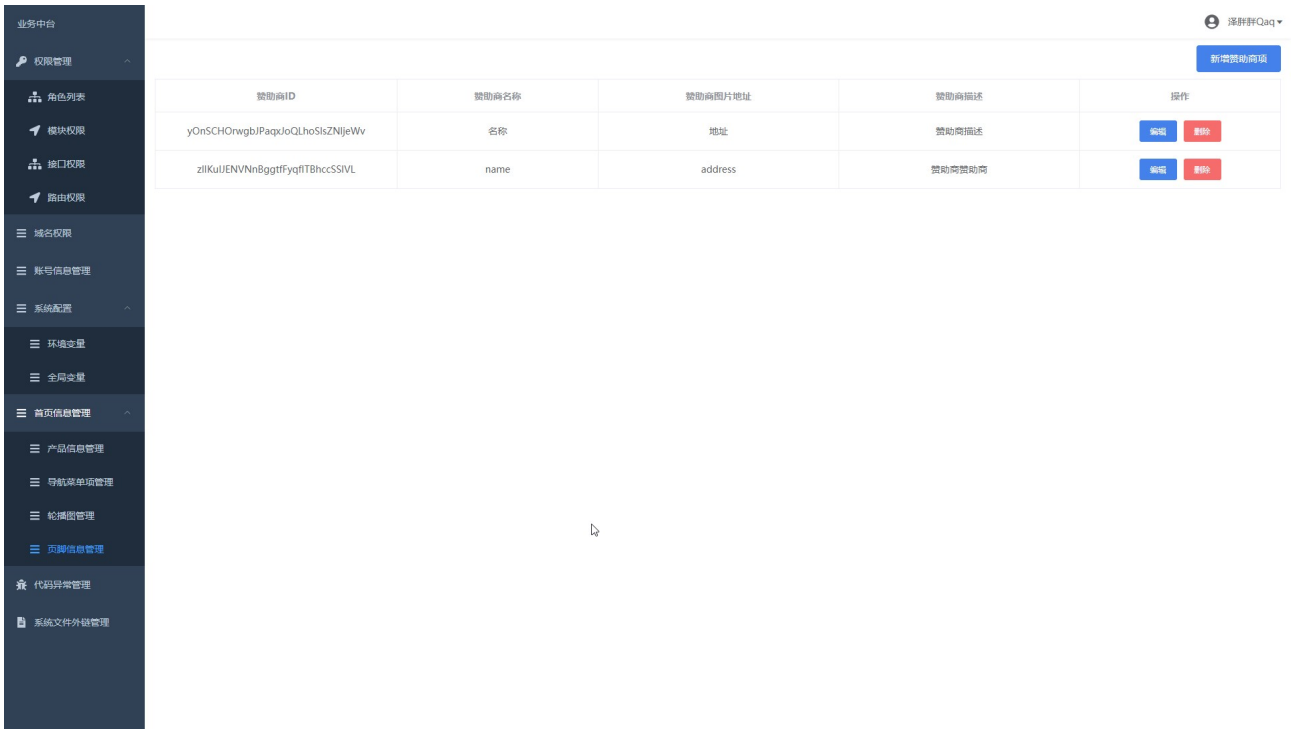

图 3. 30 首页页脚信息管理页面

图 3.27、图 3.28、图 3.29 和图 3.30 是系统首页显示的信息的管理控制页面。管理员可以在这些页面中对这些信息进行修改。修改后的信息会直接显示在系统的首页，用户可以直接看到这些信息。管理员可以借助这个页面对自己的系统进行快速的推广，让用户能够实时了解到当前系统中的热门产品和特价活动。

图 3.31 展示的是当前系统中异常错误信息代码的配置页面。管理员可以在这个页面中配置不同的错误代码，及其对应的中英文信息。管理员只需在本页面中配置对应的错误代码的中文报错信息，系统会自动调用腾讯云的文本翻译的 API 接口对中文错误信息进行翻译。开发人员可以在这个页面中对这些代码进行配置，这样只需要在系统代码中抛出指定的 Code 即可，无需重复的输入报错信息。在规范化输出异常信息，减少代码中的无效字符发同时，减少了开发过程中的重复劳动。

图 3.32 展示的系统内部存储的文件外链控制界面。管理员可以把系统中常用的文件存储到这里，从而生成外链，在系统中的其它地方进行引用。管理员可以在此页面中创建不同的文件夹，并对文件夹中的文件的进行复制，移动，删除等操作。

图 3.31 系统异常代码配置管理页面

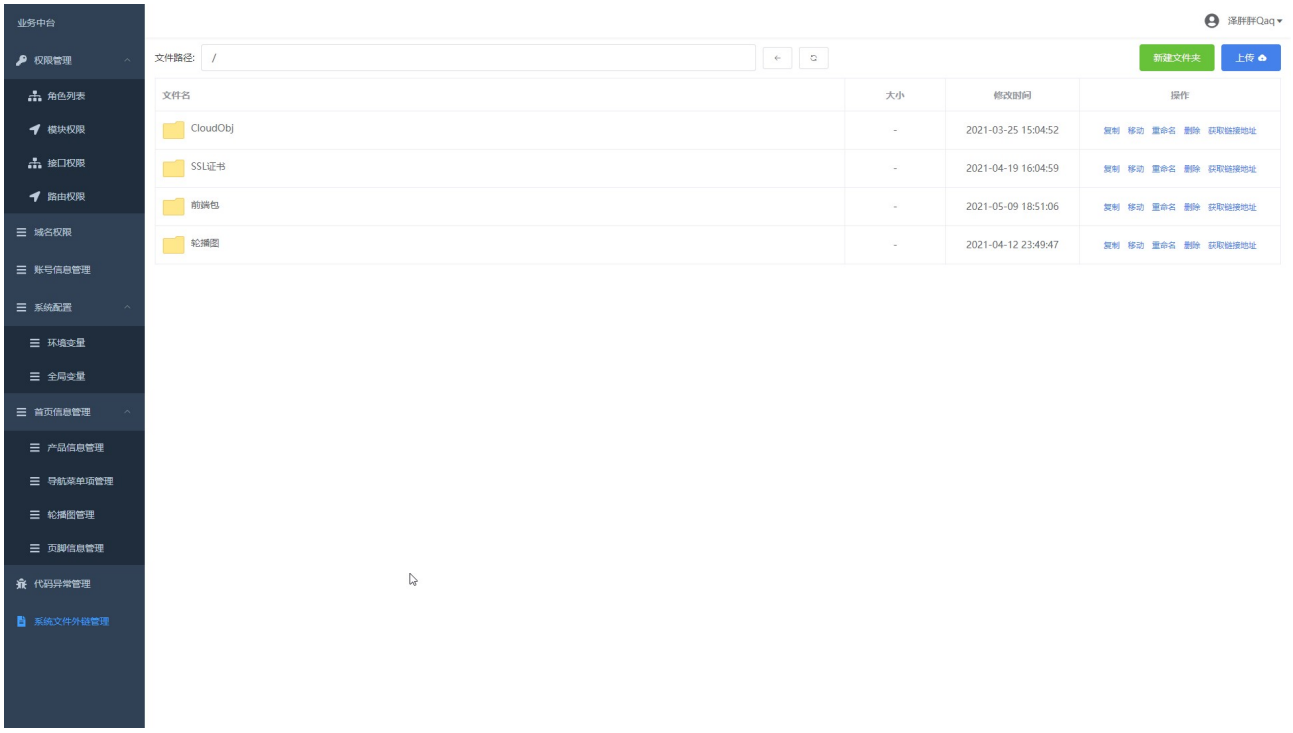

图 3.32 系统文件外链管理控制界面

第四章 总结和展望

4.1 项目总结

本项目最初是来自全年夏天我的一次大胆尝试，即能否通过 Python 脚本去自动操作 VCenter 集群上的服务器内。于是我报名了 2020 年的 C4 网络大赛，并将云计算系统作为自己的选题。去年年末大创选题时，我一下就选中了这个课题。很快半年过去了，在这半年中，我主要利用实习工作的闲暇时间和晚上周末的空闲时间对本课题进行研究和开发尝试。.

我很欣慰，在这半年中，我不断充实了自己，学习到了非常多的新技术，攻克了很多以前自己觉得做不到，不可能的难题。去年暑假结束，我刚离开学校实习时，我还仅仅只是稍稍了解一点 Redis 原理和使用方法，现如今，我不仅能够熟练的使用 Redis 数据库，还掌握了 InFluxDB , Kafka 消息队列，Jenkins 自动化集成，Hadoop 大数据框架，Vue 前端框架等多种在学校里从未涉猎过的新技术，新思想，并在我的毕业设计中得到了不同程度的整合和应用。

本项目在开发过程中，我和位于广州的前端同学互相合作。虽相隔千里，但万水千山阻挡不了我们一起构思产品，思考逻辑，解觉 BUG，优化性能的热情和信心。我深刻的认识到，个人的力量是有限的，在新技术时代，要想做好的产品，就必须要拥有一个互相信任，互相理解，互相配合的团队。

就目前而言，对于团队合作方面，我还有很多的遗憾和不足。首先是自己不够耐心和细心。通常一个接口的开发完成后，没有进行仔细的检查就直接撰写接口文档，并提供给前端同学，往往会出现非常多的意想不到的错误。其次自己在和别人沟通时有很多的缺陷，从某种程度上来说，我心中非常清晰我想构建的产品是怎么样的一种模型，但往往我的描述会让别人产生误解，而我自己在这种往复的沟通中显得非常的不耐烦。由于后端程序和主要架构均为我一人独自设计开发，所以我在开发时，养成了非常不好的坏习惯。没有按照最小化更新量的原则去进行 GIT 版本迭代，而是一拖再拖，等开发出了一个大的模块或很多功能时，才想起来去 GIT 迭代一下。通常情况下，后端程序提供给前端或其他应用的接口参数应该遵

循一定的规范，如小驼峰法，大驼峰法，匈牙利命名法等，但我在开发接口时则随心所欲，时而大写，时而小写。导致内部变量格式非常混乱，不利于项目的后期开发和多人协作。

4.2 未来展望

系统在设计初期，其实就已经为一个完备的云计算生态系统提供了丰富的预留接口和底层组件支持。未来，在以分布式的对象存储系统为核心的基础上，我还将进一步对本系统进行横向的业务拓展。通过整合互利网上分散的 VPS 服务器资源，构建自己独立聚合的代理销售控制系统。通过对 VCenter 系统的进一步开发和了解，整合现有资源，重新开发云虚拟化业务。除此以外，我还将进一步拓宽系统的外部拓展能力，对现阶段已有的 SSO 登录功能的业务逻辑进行完善，支持 OAuth 协议的登录功能。结合自己以前曾借做过的一些零碎的功能，如（应用更新 API，SSL 证书一键签发，Aria2C 离线下载）等，逐渐完善自己的云生态体系。这正如当我所说的，也许对绝大部分人来言，毕业设计只是一份作品，但从我心底，我一直都是以一个产品态度去努力开发努力维护。

参 考 文 献

[1] 穆志纯. 浅谈分布式存储系统架构设计[J]. 电子世界, 2020(22):146-147.
[2] 廖彬, 于炯, 孙华, 年梅. 基于存储结构重配置的分布式存储系统节能算法[J]. 计算机研究与发展, 2013, 50(01):3-18.
[3] 康欣欣. 基于分布式对象存储的媒体资源管理系统建设研究[J]. 信息与电脑(理论版), 2020, 32(17):80-82.
[4] PHP Official. PHP: Preface – Manual. https://www.php.net/manual/en/preface.php, 2020-12-08/2021-03-21
[5] 练振兴. My SQL 读写分离的技术原理[J]. 福建电脑, 2019, 35(08):49-51.
[6] 亓雪冬, 韩立峰. 基于 MVC 的 Web 框架设计与应用[J]. 微型电脑应用, 2021, 37(03):4-6.
[7] 郑朝杰. 医疗挂号系统服务端的设计与实现[D]. 北京邮电大学, 2018.
[8] 陈玺, 马修军, 吕欣. Hadoop 生态体系安全框架综述[J]. 信息安全研究, 2016, 2(08):684-698.
[9] 王昌. 动态网站安全漏洞检测系统的设计与实现[D]. 北京邮电大学, 2020.
[10] 王育红, 夏安祥, 林国庆, 向真. 抗重放攻击方案在工程中的应用[J]. 网络安全技术与应用, 2021(04):8-10.
[11] 吴君楠, 黎铁军, 袁远, 隋荣恒. 一种安全高效的跨域单点登录系统设计与实现[J]. 软件工程, 2018, 21(09):44-47.

致　谢

虽然本次毕业设计仅是个人毕设，但我深知，本次毕业设计的完美完成绝非我一个人的成果，而是一群人的努力，是合作，是共享，是借鉴，是反思，让我学习，让我成长。

牛顿说:“我之所以比别人看得远一些，是因为我站在巨人的肩膀上。” 饮水思源，向自 1946 年第一台电子计算机诞生之日起，所有为电子信息事业默默努力的 IT 开发者致谢，感谢你们创造了这个缤彩纷呈的互联网世界，感谢你们为推动电子信息技术的发展做出的努力和贡献。感谢你们为今天的我们，奠定了如此基石的基础。

“春蚕到死丝方尽，蜡炬成灰泪始干。”向一直默默奉献的老师们致敬，感谢你们教会我知识，教会我方法。“授人以鱼不如授人以渔”，正式有各位老师的引领，我才能意识到自己的缺点和不足，能够在老师的指导下查阅各种资料，不断拓充自己，挑战自己，获得更大的提升。

向我的指导老师王姝懿老师致谢，感谢您长达几个月的毕设开发过程中细致，入微的指导和引领。我是一个比较懒惰的人，幸亏有老师的监督和陪伴，使我最终坚持了下去，最终完成了本次毕业设计。感谢老师一次又一次的帮我审查论文格式，提出修改意见，并督促我准备毕设答辩的各项事宜。

向我的部门领导吴冕之先生致谢。感谢您在我最困难的时候收留了我，感谢您所给我的一个宽松自由的工作环境，使得我在工作之余可以努力学习，大胆实践。海内公司的项目结构和业务逻辑，对本次毕业设计的开发启到了非常大的学习借鉴作用，感谢您的信任，我定当不负期望，愿效犬马之劳。

向堡塔面板的技术总监黄文良先生致谢。您开发的堡塔面板在本次毕业设计中发挥了至关重要的作用。您在开发堡塔面板时所书写的 Python 代码是我在自学 Python 编程的过程中的启蒙导师。您使用 Python 对服务器进行的自动化操作，至今仍然是我学习，体会的对象。感谢您开发出了如此优秀的产品，堡塔面板使得我第一次对 Linux 操作系统有了更加深刻的理解，可以说您是我在自动化运维道路上的奠基人，

向阿里云，七牛云，又拍云，腾讯云，雨云等云计算 IDC 平台的开发者们致谢。本次毕业设计在开发过程中，部分参考并借鉴了各平台的部分系统的业务逻辑，系统界面和操作说明。感谢各位前辈们辛苦付出，我定将牢记开源社区精神，自由，创新，团结，互助，友爱，积极，进取。

特别感谢我毕业设计的前端工程师骆兆锋同学。“桃花潭水深千尺，不及汪伦赠我情。”这份友谊，这段岁月，我将深刻铭记。我不会忘记，那一个个寂静的夜晚，我们畅聊产品的业务逻辑。我不会忘记，那一次次的视频会议，我们共同发现问题，商量方案，解觉BUG。我不会忘记，我们一次此的互相鼓励，给我所长，取我所缺。豪情壮志在心，踌躇满志在云，希望我们的友谊可以像这篇论文一样，在涛涛岁月中，直到永远。

附 录

1.《无调云计算平台 API 接口文档 V4.0》

图 1：API 接口文档

2. 后端代码结构

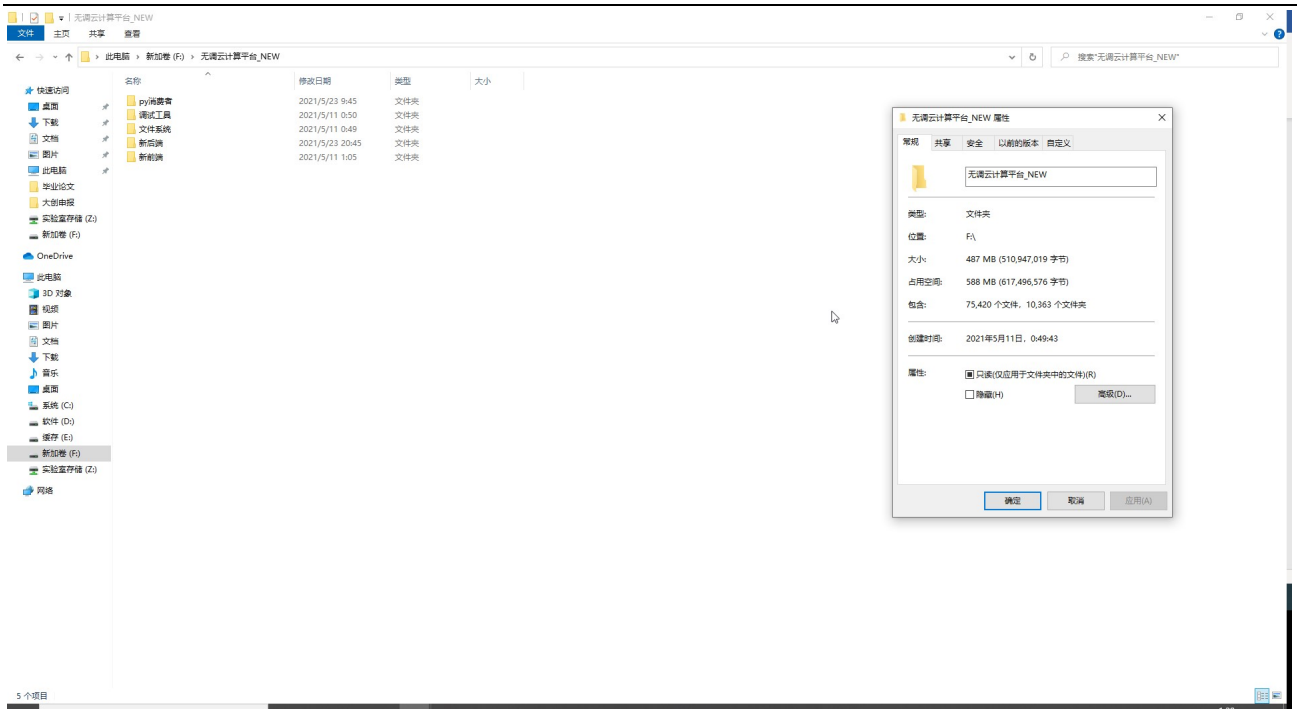

图 2：项目文件夹结构

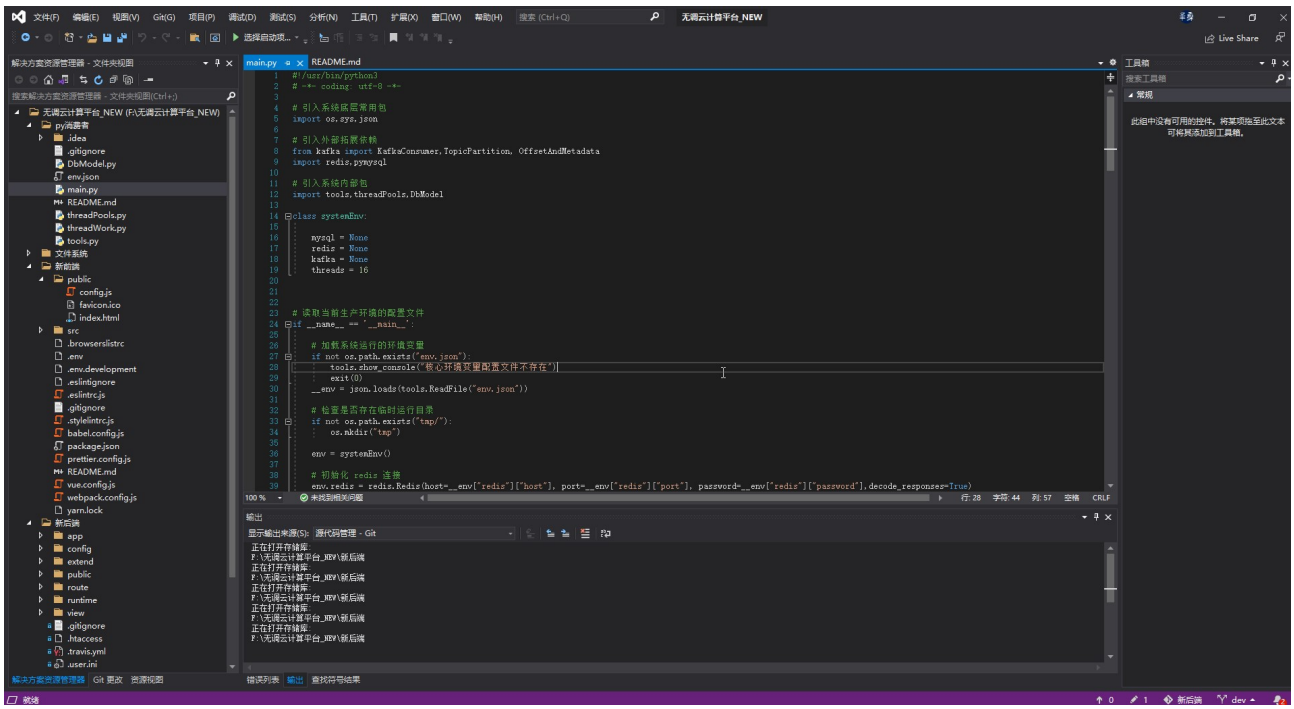

图 3：项目代码结构

3. 代码行统计

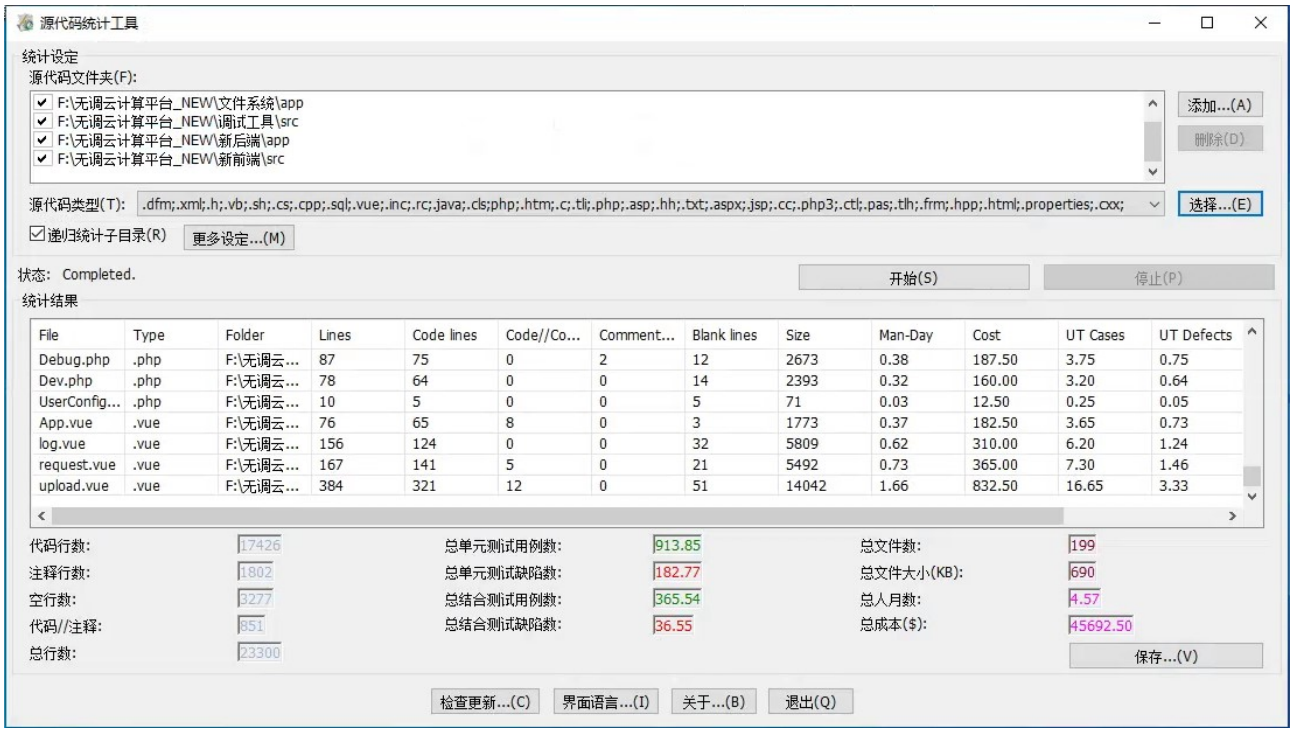

图 4：项目代码统计