

**毕 业 设 计**

****

**题目：电力电缆温度在线监测系统服务器端程序设计**

|  |  |
| --- | --- |
| **姓 名：** | 温凯凯 |
| **学 号：** | 201505050418 |
| **班 级：** | 物联网151 |
| **所在院系：** | 电子信息与人工智能学院 |

2019年6月13日

电力电缆温度监测系统服务端程序设计

# 摘 要

伴.随.我.国.国.民.经.济.的.迅.猛.发.展.，.城.市.建.设.和.企.业.现.代.化.程.度.提.高.。.作.为.国.民.经.济.命.脉.的.电.力.遇.到.了.前.所.未.有.的.挑.战.。为.了.满.足.人.民.的.供.电.需.求.，城.市.电.网.中.电.缆.所.占.的.比.重.也.越.来.越.大.。.然.而.大.量.电.缆.敷.设.在.距.离.较.长.、走.向.复.杂.的.电.缆.沟.内.。这.些.电.缆.长.期.运.行.在.高.电.压.、大.电.流.环.境.下.，极.易.引.起.温度.上.升.、造.成.温度.异常.，导.致.电.缆.接.头.爆炸.，引.发.火.灾.。本课题研究的是一套电力电缆温度监测系统，该系统可以通过采集端实时采集电缆关键节点温度，经过服务端对数据分析处理并在客户端上进行展示和预警。达到对电缆渠中的电力电缆温度实时监测、提前预警、实时查看等目的，可以有效的减少因为电缆温度过高造成的财产损失。

电力电缆温度监测系统包括三部分：服务端、采集端、客户端。其中服务端设计基于HTTP协议，并使用PHP进行开发。服务端与采集端以及客户端之间统一使用HTTP协议的POST方法传送JSON格式的字符串进行通信。服务端的主要功能包括用户管理（用户注册\登录\修改信息）、设备管理（设备添加\删除\属性设置）、温度信息保存、温度数据分析等功能。客户端包括移动端和PC端，其中PC端由Java开发，利用Java GUI实现了用户管理、设备管理、温度报警、数据查询等界面的展示，并通过调用服务端的接口实现界面的跳转以及数据的展示。

关键词：实时监测，提前预警，实时查看，服务端设计，客户端设计

**Design of Server Program for Power Cable Temperature Detection System**

# **ABSTRACT**

With the rapid development of China's national economy, the degree of urban construction and enterprise modernization has been improved. As the lifeblood of the national economy, electric power is facing unprecedented challenges. In order to meet the people's demand for power supply, the proportion of cable in urban power grid is also growing. However, a large number of cables are laid in long distance and complex trenches. These cables operate in high voltage and high current environment for a long time, which is easy to cause temperature rise and abnormal temperature, resulting in explosion of cable joints and fire. This paper studies a set of temperature monitoring system for power cable. The system can collect the temperature of the key nodes of the cable in real time through the acquisition terminal, analyze and process the data at the service terminal, and display and early warning on the client side. Real-time monitoring, early warning and real-time viewing of the temperature of power cables in cable canals can effectively reduce property losses caused by excessive temperature of cables.

Power cable temperature monitoring system consists of three parts: server, acquisition and client. The server design is based on HTTP protocol and developed with PHP. The POST method of HTTP protocol is used to transmit JSON format strings to communicate between server, collector and client. The main functions of the server include user management (user registration\login\modification information), device management (device addition\deletion \attribute setting), temperature information preservation, temperature data analysis and other functions. Client includes mobile and PC. PC is developed by Java. Java GUI is used to display user management, device management, temperature alarm, data query and other interfaces. The interface jump and data display are realized by calling the interface of server.

**Key words**: Real-time monitoring, early warning, real-time viewing, server design, client design

目 录

[摘要 Ⅰ](#_Toc29740_WPSOffice_Level2)

[ABSTRACT Ⅱ](#_Toc29423_WPSOffice_Level2)

[1绪论](#_Toc29740_WPSOffice_Level1) [1](#_Toc29740_WPSOffice_Level1)

[1.1选题背景](#_Toc18678_WPSOffice_Level1) [1](#_Toc18678_WPSOffice_Level1)

[1.2研究现状](#_Toc20340_WPSOffice_Level1) [1](#_Toc20340_WPSOffice_Level1)

[1.3课题研究的目的及意义](#_Toc29129_WPSOffice_Level1) [2](#_Toc29129_WPSOffice_Level1)

[2电力电缆温度监测系统设计](#_Toc21262_WPSOffice_Level1) [3](#_Toc21262_WPSOffice_Level1)

[2.1 系统总体方案设计](#_Toc22882_WPSOffice_Level1) [3](#_Toc22882_WPSOffice_Level1)

[2.1.1 系统总体框架](#_Toc18678_WPSOffice_Level2) [3](#_Toc18678_WPSOffice_Level2)

[2.1.2 系统各模块功能描述](#_Toc20340_WPSOffice_Level2) [3](#_Toc20340_WPSOffice_Level2)

[2.1.3 系统工作流程分析](#_Toc29129_WPSOffice_Level2) [4](#_Toc29129_WPSOffice_Level2)

[2.2 应用技术选择与使用](#_Toc32553_WPSOffice_Level1) [5](#_Toc32553_WPSOffice_Level1)

[2.2.1 Web服务器选择](#_Toc22882_WPSOffice_Level2) [5](#_Toc22882_WPSOffice_Level2)

[2.2.2 开发语言的选择](#_Toc32553_WPSOffice_Level2) [6](#_Toc32553_WPSOffice_Level2)

[2.2.3 开发框架的选择](#_Toc32249_WPSOffice_Level2) [6](#_Toc32249_WPSOffice_Level2)

[2.2.4 数据库的选择](#_Toc8653_WPSOffice_Level2) [7](#_Toc8653_WPSOffice_Level2)

[2.2.5 服务端与客户端通信方式的选择](#_Toc17271_WPSOffice_Level2) [7](#_Toc17271_WPSOffice_Level2)

[2.3 开发及调试工具的使用与选择](#_Toc32249_WPSOffice_Level1) [8](#_Toc32249_WPSOffice_Level1)

[2.3.1 操作系统CentOS的使用](#_Toc11571_WPSOffice_Level2) [8](#_Toc11571_WPSOffice_Level2)

[2.3.2 PhpStorm和Postman的使用](#_Toc24903_WPSOffice_Level2) [8](#_Toc24903_WPSOffice_Level2)

[2.3.3 版本控制Git的使用](#_Toc6376_WPSOffice_Level2) [8](#_Toc6376_WPSOffice_Level2)

[3 电力电缆温度监测服务端设计](#_Toc16219_WPSOffice_Level2) [9](#_Toc16219_WPSOffice_Level2)

[3.1数据库功能设计](#_Toc8653_WPSOffice_Level3) [9](#_Toc8653_WPSOffice_Level3)

[3.1.1 用户信息表user\_base设计 9](#_Toc22818_WPSOffice_Level2)

[3.1.2 设备信息表device\_info设计 9](#_Toc3703_WPSOffice_Level2)

[3.1.3 用户设备权限表uid\_did\_map设计 1](#_Toc3703_WPSOffice_Level2)0

[3.1.4 设备温度表device\_temperature设计 1](#_Toc3703_WPSOffice_Level2)0

[3.2服务端软件功能设计](#_Toc17271_WPSOffice_Level3) [11](#_Toc17271_WPSOffice_Level3)

[3.2.2 Nginx服务器的安装与调试](#_Toc22818_WPSOffice_Level2) [12](#_Toc22818_WPSOffice_Level2)

[3.2.3 PHP7的安装](#_Toc3703_WPSOffice_Level2) [13](#_Toc3703_WPSOffice_Level2)

[3.2.4项目代码设计](#_Toc9581_WPSOffice_Level2) [14](#_Toc9581_WPSOffice_Level2)

[4 系统运行调试](#_Toc616_WPSOffice_Level2) [30](#_Toc616_WPSOffice_Level2)

[4.1 测试环境概述](#_Toc16885_WPSOffice_Level3) [30](#_Toc16885_WPSOffice_Level3)

[4.2 测试过程](#_Toc24602_WPSOffice_Level3) [31](#_Toc24602_WPSOffice_Level3)

[4.2.2功能测试](#_Toc28292_WPSOffice_Level2) [32](#_Toc28292_WPSOffice_Level2)

[5 总结与展望](#_Toc20937_WPSOffice_Level2) [36](#_Toc20937_WPSOffice_Level2)

[致谢](#_Toc8653_WPSOffice_Level1) [37](#_Toc8653_WPSOffice_Level1)

[参考文献](#_Toc17271_WPSOffice_Level1) [38](#_Toc17271_WPSOffice_Level1)

# 1绪论

## 1.1选题背景

随着我国国民经济的迅猛发展，城市建设和企业现代化程度提高。作为国民经济命脉的电力遇到了前所未有的挑战。为了满足人民的供电需求，城市电网中电缆所占的比重也越来越大。特别是一些经济发展快如北京、上海的一线城市的电网呈现高密度和高负荷的增长。仅仅依靠空中架设的电网已经不能满足日益增长的需求，输送电力的电缆逐渐从天上走到地下。通过在电缆渠里铺设电缆，不仅可以解决电力运输的问题，还可以有效的利用空间[1]。

但是电缆铺设在电缆渠也带来了一些问题：高昂的人力维护成本。众所周知电缆绝缘性的优劣势决定电力电缆正常运行的主要因素，而电缆的接头与导体温度又是影响电缆绝缘性能的关键指标。电缆渠中的电缆长时间工作在高电压，强电流的环境中，接口处温度容易上升，造成电缆接口爆炸，引起火灾，造成不必要的财产损失。所以，如果能对电力电缆接口的温度有效的监控、预警就可以有效地减少电缆渠中电力电缆的维护成本以及降低因为温度引起的财产损失[2]。

电力电缆的温度监测可以分为离线监测和在线监测两种方法。作为传统的离线监测包括示温蜡片法、引线接头测温法、接触式电信测温法、红外测温法等，然而这些方法需要断电后监测人员携带仪器到现场进行监测，实时性差，不能做到节约人力和物力成本要求。

本次毕业设计针对上述问题提出了一套可以实时在线监测电力电缆温度的系统。本系统可以实现对电缆渠中的电力电缆温度实时监测、实时预警、实时查看等功能。电力电缆温度在线监测系统包括三部分，包括采集端、服务端、客户端。采集端负责实时采集现场数据；服务端负责用户管理、保存数据、分析数据并发出预警；移动端负责用户登录、实时展示采集结果；为了满足不同场景的需要，客户端又分为移动端和PC端。客户端主要负责设备管理、用户管理、查看实时温度、实时预警等。

## 1.2研究现状

目前，我国在电力电缆的运行电缆绝缘监测、发热监测、导体额定载流量计算等问题首选的技术就是实时在线监测技术。发热监测又是最能反映出电缆实际运行状况的指标。发热监测之所以能够比较有效的反映电缆的运行状况，是因为电缆在运行过程中并不能将电力完全输送过去，期间会产生电阻损耗。而电阻损耗最直接的表现就是产生热量。也就是说，电阻损耗可以通过电缆的温度表现出来。所以，运行电缆允许的最高温度可以作为电缆的性能指标之一。

电力电缆埋在地下的电缆渠中，周围土壤的湿度，回填土的密度都会影响到电缆的散热性能。另外，还有电缆的金属保护套引起的环流损耗、涡流损耗，也会影响电缆的导电性能。

目前，针对电缆温度的监测主要使用的方法是在线监测。本系统也是一套在线实时监测系统。采集端采集现场电缆温度，上传到服务端；服务端分析处理温度数据；移动端展示温度数据和预警信息；PC端进行用户管理、设备管理以及温度查看。

## 1.3课题研究的目的及意义

电力电缆温度在线监测系统包括三个部分：采集端、服务端、客户端。采集端基于无线传感网络技术（Wireless Sensor Network）利用DS18B20温度传感器以及GPRS（General Packet Radio Service）对电力电缆接口处温度进行实时采集和发送，可以有效的减少人力采集温度的成本并且实现温度的实时采集及传送。客户端为了满足不同场景的需要，又分为了移动端和PC端。PC端用于对用户、设备进行管理，同时可以查看实时温度数据、预警信息等。移动端利用便携的优点，采用Android技术对采集到的温度进行实时报表展示，以及故障预警。服务端对采集端采集到的温度进行存储，当PC端和移动端请求服务端接口时，服务端对温度数据分析计算，发出预警，并向移动端发送实时数据。

本课题研究的电力电缆温度在线监测系统可以对电缆温度进行实时监测，当电缆温度达到某个阈值或者通过计算预测电缆温度出现异常将进行提前预警，提前处理。这样可以有效的避免因为电缆温度过载导致的电力事故，减少停电故障引起的财产损失。并且，可以通过采集的大量的温度数据对电力事故多发地区的电缆温度实时进行分析预测，在温度异常之前发出预警。另外，该系统还能对事故多发地的电缆布局提出改良方案。该研究可以提高经济效益，减少事故带来的损失以及电缆渠电力电缆的人力维护成本；因为可以做到提前预警温度异常，延长了电缆的使用寿命，从而节约成本。

# 2电力电缆温度监测系统设计

## 2.1 系统总体方案设计

### 2.1.1 系统总体框架

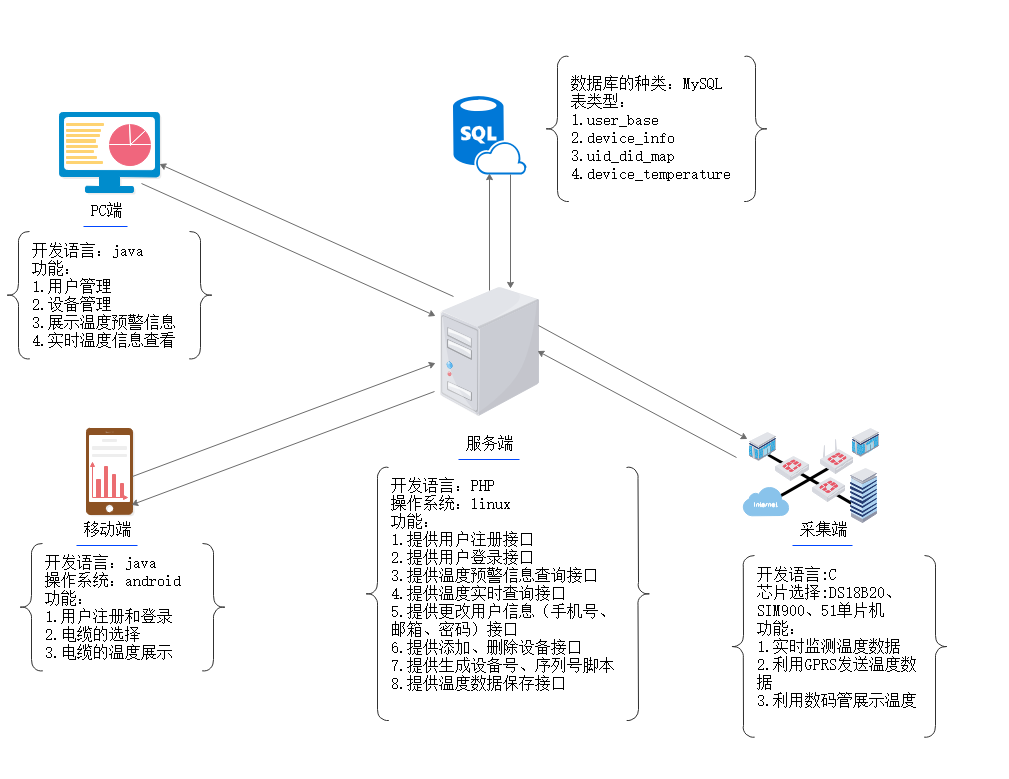
电力电缆温度监测系统框图如下图2-1所示，该系统主体分为四部分：采集端、服务端、移动端、PC端。采集端负责采集现场温度数据并定时上传到服务端；服务端负责用户管理、数据处理以及响应采集端和移动端的请求；移动端负责用户的注册登录、设备权限管理以及温度信息和预警信息的展示；PC端负责用户信息管理、设备管理、温度预警信息查看、实时温度查看等功能。

图2-1 系统总体框图

### 2.1.2 系统各模块功能描述

采集端利用WSN技术，通过DS18B20温度传感器在现场采集电缆温度数据，之后通过SIM900利用GPRS网络向服务器发送HTTP的POST请求，调用服务端温度保存接口将温度数据上传到服务端数据库。

服务端通过提供接口响应移动端、采集端、PC端的请求。其中对移动端和PC端提供了以下接口。

1. 用户注册接口
2. 用户登陆接口
3. 修改邮箱/电话号/密码接口
4. 获取用户拥有设备权限接口
5. 添加/删除用户可以查看设备权限接口
6. 查看设备温度信息接口

对采集端提供了保存温度接口。服务端通过对以上接口进行详细的设计，为电力电缆温度监测系统提供了完备的用户管理、设备管理、温度保存、温度查看等服务。

移动端通过Android技术，设计了一个电力电缆温度监测系统的App。用户可以通过App随时随地查看电缆的温度信息，并获得电缆的实时温度预警信息，提前做出处理，最大程度上降低损失。

### 2.1.3 系统工作流程分析

如图2-2所示，电力电缆温度监测系统由三部分组成：采集端、服务端、客户端。贯穿电力电缆温度监测系统的数据流就是电缆的温度数据，该数据首先由采集端在现场通过DS18B20温度传感器进行采集，并利用SIM900通过GPRS网络发送HTTP的POST请求到服务端。服务端收到采集端传送的温度数据，对数据进行格式处理，并存储到云端数据库中。服务端从数据库拿到数据，进行分析处理。在客户端（移动端\PC端）请求温度预警信息查看接口以及温度信息查看接口的时候，服务端将对温度数据处理结果推送到客户端。客户端接收到服务端响应的温度数据之后，在前端进行展示。至此完成了电力电缆温度监测系统的整个主干流程。

另外，当用户进行注册、登录以及设备增删时，客户端将请求服务端对应的接口。服务端做出响应，并操作数据库完成用户管理以及设备管理。系统设计时考虑到每一台设备都需要一个唯一的设备号以及设备添加时的验证码，所以服务端提供了一个可以生成制定数量设备序列号和设备验证码的脚本。当出厂新设备时将调用该脚本为设备生成一个唯一的序列号以及验证码。

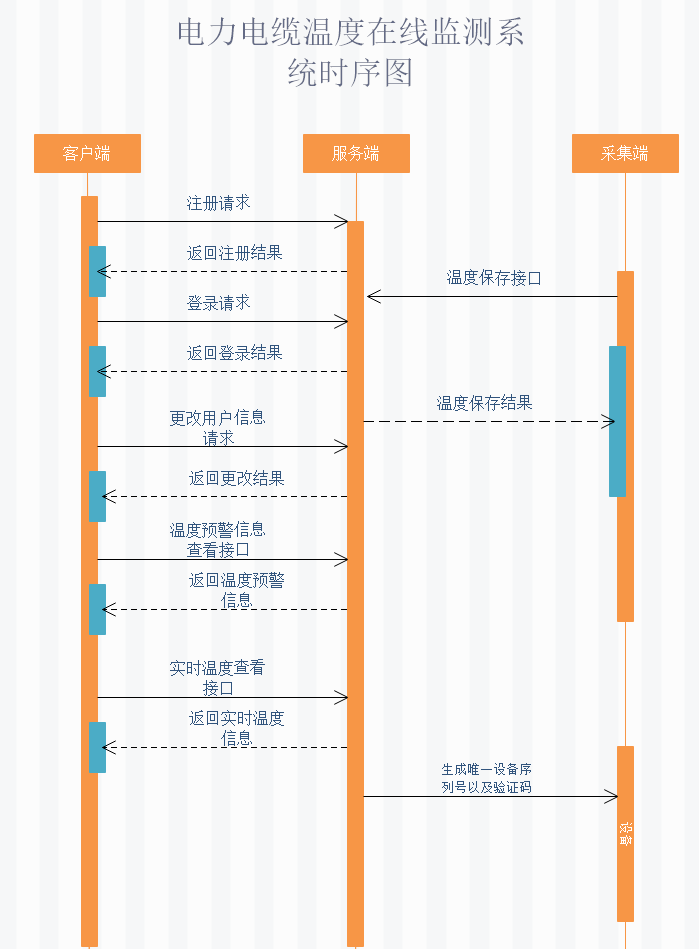


图2-2 服务端时序图

## 2.2 应用技术选择与使用

### 2.2.1 Web服务器选择

现在各大互联网公司主流的web服务器有Nginx和Apache两种，本次毕业设计电气电缆温度监测系统服务端设计将要搭载到阿里云的云服务器上，考虑到资源开销问题，需要一个轻量级的Web服务器，并且考虑到多个采集端为了保证温度数据的实时性，将密集的访问服务器的接口，所以需要一个能够承受较高并发量的Web服务器。综上所述，本次课题最终选择了Nginx服务器。Nginx (engine x) 是一个开源的且高性能的[HTTP](https://baike.baidu.com/item/HTTP" \t "https://baike.baidu.com/item/nginx/_blank)和[反向代理](https://baike.baidu.com/item/%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86/7793488" \t "https://baike.baidu.com/item/nginx/_blank)web服务器。Nginx服务器有以下优势：

1. 轻量级，同样提供Web 服务，Nginx比Apache 占用更少的内存及资源 ，Nginx的官方测试的结果中，能够支持五万个并发连接，而在实际的运作中可以支持二万到四万个连接，具有较高的抗并发能力。
2. 在高并发下Nginx 能保持低资源，低消耗，高性能。
3. 高度模块化的设计，编写模块相对简单 ，社区活跃，各种高性能模块出品迅速。

### 2.2.2 开发语言的选择

Web服务端开发语言主要有Java和PHP等，本次毕业设计电力电缆温度监测系统搭载在阿里云的云服务器上，而阿里云对PHP有着良好的支持。并且Nginx和PHP-FPM的组合，是一种稳定、高效的PHP运行方式，效率要比传统的Apache和mod\_php高出不少。除此之外，PHP还具有以下优势：

1. 良好的安全性

PHP是开源软件，所有PHP的源代码每个人都可以看得到，代码在许多工程师手中进行了检测，同时它与Nginx编绎在一起的方式也可以让它具有灵活的安全设定，PHP具有了公认的安全性能。开源造就了强大，稳定，成熟的系统。

1. 跨平台特性  
    PHP几乎支持所有的操作系统平台及数据库系统，并广为流行。
2. 执行速度快, 效率高

PHP是一种强大的CGI脚本语言，语法混合了C、Java、Perl和PHP式的新语法，执行网页速度比CGI、Perl和ASP更快，而且内嵌Zend加速引擎，性能稳定快速，占用系统资源少，代码执行速度快。PHP消耗相当少的系统资源。

1. 应用范围广

目前在互联网有很多网站的开发都是通过PHP语言来完成的，例如，搜狐、网易和百度等，在这些知名网站的创作开发中都应用到了PHP语言。

### 2.2.3 开发框架的选择

本次服务端设计采用Yii2.0框架，Yii Framework是一个基于组件、用于开发大型 Web 应用的高性能 PHP 框架。 Yii框架具有以下优点。

1. 易于安装 安装处理使用Composer。
2. 利用现代技术 Yii是一种纯粹的面向对象框架，并且利用PHP的一些更高级的功能,包括延迟静态绑定,SPL类和接口,和匿名函数。
3. 高度可扩展性

非常容易根据需求来进行定制. 实际上框架的每一个组件都是可以扩展的。

1. 简化的安全方案
2. 缩短开发时间

Yii框架基于MVC（Model-View-Controller）架构, 具体分层和目录如下：

model层：本次设计所有对数据库的直接操作都默认放到该层实现。

controller层：本次设计的对数据的输入输出控制（接口）放在该层。

service层：本次设计对输入数据进行处理的逻辑都在service层实现。

command: 本次设计的超出温度阀值发送邮件脚本在该目录实现。

Enum: 本次设计所有的枚举常量都在该目录下保存。

Webroot: 后台管理系统的web资源都保存在该文件下。

config: 本次设计的所有配置文件都放在该目录下。

components: 本次设计的基类以及工具类放在改目录下

### 2.2.4 数据库的选择

互联网公司主要使用关系型的数据库有MySQL和Oracle两种。MySQL更像是Oracle的轻量版，但是MySQL也具有自己的优势。MySQL是开源的数据库，而Oracle需要用户支付额外的费用。MySQL更加轻量，对硬件的要求更低。另外对PHP有很好的支持。并且MySQL也可以很好的处理千万级数据量的大型数据库。

本次毕业设计电力电缆温度监测系统最大的并发量来自于采集端存储温度数据。在系统设计之初，考虑到电缆的温度并不会产生突变，是一个缓慢变化的过程。所以本系统设定采集端每隔一分钟向服务端发送一条温度数据，这样一个采集端一天的数据量是一千条左右。当使用后期单表的数据量过大的时候，可以对数据量大的表进行分表操作，这样可以解决MySQL存储大量数据的问题。所以使用MySQL能够完全满足本课题的需求。除此之外，云服务器的配置有限，使用MySQL作为数据库的话能够更好的发挥云服务器的性能。

综上所述，本次毕业设计选择了MySQL作为了存储数据的数据库。

### 2.2.5 服务端与客户端通信方式的选择

Web应用中服务端与客户端涉及到一些敏感信息如用户名和密码时常采用HTTP的POST请求来请求服务端，因为POST的请求参数不会显示在URL中并且也不会保存到浏览历史和web服务器的日志中。所以POST请求是一种更加安全的请求方式。因此本系统涉及到的客户端请求服务端接口时都是通过HTTP的POST请求来进行访问的。

另外，本课题规定了服务端和客户端之间通信的参数都采用JSON串的格式进行传输。JSON串是一种轻量的数据交换格式。它由键值对的形式组成，易于阅读和解析。在java和PHP中都有对应的JSON处理函数。所以在客户端请求服务端时，将把参数转化成JSON串的形式添加到POST方法的报文主体中。服务端收到请求之后将对JSON串解析，拿到对应的参数并进行处理。

## 2.3 开发及调试工具的使用与选择

### 2.3.1 操作系统CentOS的使用

CentOS（Community Enterprise Operating System）是Linux发行版之一，作为开源的Linux操作系统，CentOS常被用作Web服务端的操作系统。CentOS具有超高的稳定性，可以保证系统长时间的稳定运行。CentOS还支持Nginx以及PHP和大部分web开发工具。因此本次毕业设计的服务端操作系统选择了阿里云的CentOS 7.2作为服务器的操作系统。

### 2.3.2 PhpStorm和Postman的使用

PhpStorm是 [JetBrains](https://baike.baidu.com/item/JetBrains" \t "https://baike.baidu.com/item/_blank) 公司开发的一款商业的 PHP 集成开发工具，旨在提高开发人员的工作效率，能够深刻理解开发者的代码，提供智能代码补全，快速导航以及即时错误检查。使用PhpStorm在编写PHP项目的时候可以很方便快捷地进行函数查找、代码补全、函数快速修正。使用PhpStorm开发能够有效的提高开发效率。

在web接口开发中，常常需要对接口的输入输出数据流进行测试调试。Postman可以提供几乎所有类型的HTTP请求，并查看响应报文。所以Postman是一个web开发者必不可少的测试调试工具。因此本课题选择了Postman作为服务端接口开发时的功能测试工具。

### 2.3.3 版本控制Git的使用

Git是一个开源的分布式版本控制系统，可以有效、高速地处理从很小到非常大的项目版本管理。在本地进行项目团队开发的时候，可能由于大家的进度不同会产生很多不同的版本，这个时候就需要Git来进行版本控制，Git版本控制可以保证项目开发有序的进行。此外，由于代码是在本地进行开发，为了保证服务端代码和本地保持一致，可以使用linux rz、sz等工具把代码上传到服务器。但是项目比较大的话这样的传输速度就很慢，严重的拖慢了开发进度。这时可以在github建立一个远程Git仓库，当然也可以在自己的服务器上搭建远程的Git仓库。把本地的项目上传到github对应的仓库之中，之后再服务端使用git pull命令将代码进行同步。因为Git是分布式版本管理工具，每次上传代码都传送的是做出更改的文件而不是整个项目，所以使用Git进行代码同步的速度要比直接上传整个项目要快的多。

使用Git之后服务端代码开发流程就是首先在本地开发，开发完成之后使用git add命令将变更的代码加入暂存区，再使用git commit命令将更改添加到本地仓库，最后使用git push命令将代码推送到远程仓库。到这里已经将代码同步到了github的远程仓库上。之后再服务端使用git pull将服务端代码同步。就可以使用Postman等接口测试工具进行测试。如果发现代码出现bug并且不好修复可以使用git reset进行版本回退。

# 3 电力电缆温度监测服务端设计

## 3.1 数据库功能设计

电力电缆温度监测系统共涉及到四个表的操作，分别是user\_base, uid\_did\_map, device\_info, device\_temperature。

### 3.1.1 用户信息表user\_base设计

如图3-1所示，user\_base表设计用来存储用户的基本信息，包括用户名(user\_name)、手机号(mobile)、邮箱(email)以及密码(password)。此外还有三个辅助字段：用户唯一标识(uid)、盐值(salt)和更新时间(update\_time)。在注册登录以及更改账号/邮箱/密码的接口都涉及到了对改表的操作。用户唯一标识uid是在注册时生成的八位用户唯一标识，在用户的后续操作中都根据uid作为唯一标识。Salt是保存用户密码的时候添加的盐值，该盐值将和用户密码拼接在一起再进行哈希处理，加盐哈希作用是防止恶意的彩虹表破解，以保证数据库中用户密码的安全。而保存盐值的作用是在用户登录以及更改密码的时候拿出盐值与密码进行哈希，与数据库中的密码进行对比，判断登录是否成功。

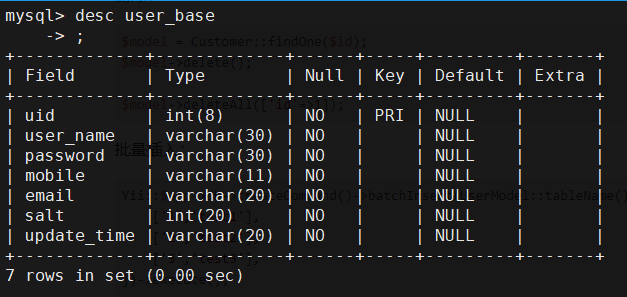


图3-1 user\_base表

### 3.1.2 设备信息表device\_info设计

如图3-2所示，device\_info表是设备信息表，该表是电力电缆温度采集端的信息表。包括了设备唯一标识(did)、验证码(verify\_code)、设备状态(status)、设备阈值温度（threshold）以及更新时间(update\_time)。did是设备出厂的时候生成的八位唯一标识，这个号码将伴随设备的整个生命周期，作为设备的唯一标识。verify\_code也是设备在出厂的时候生成的八位密码。移动端在添加用户对设备的权限的时候以及采集端在保存温度数据的时候，都会验证这个验证码。这样做的目的是为了保证权限以及温度数据的安全性。status表示设备的运行状态，该值默认为0，0表示设备不在线，1表示在线。在移动端添加查看设备权限的时候，如果设备不在线服务端将拒绝请求；采集端在第一次上传温度数据的时候，服务端会将该表中的status置1。

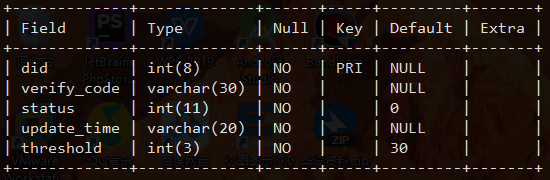


图3-2 device\_info表

### 3.1.3 用户设备权限表uid\_did\_map设计

如图3-3 uid\_did\_map表所示，uid\_did\_map表是用户唯一标识（uid）和设备唯一标识（did）的映射表。该表有两个字段uid和did，这两个字段共同作为该表的主键。当移动端添加用户查看设备的权限的时候会在该表添加一条uid到did的映射。当uid和did作为共同主键在表中存在的时候，服务端会拒绝移动端的添加请求。移动端对用户权限删除只需删除表中对应记录即可。另外，移动端想获取用户能够查看的所有设备时，只需要提供用户的唯一标识（uid）在该表进行查找即可找到用户拥有的设备权限。

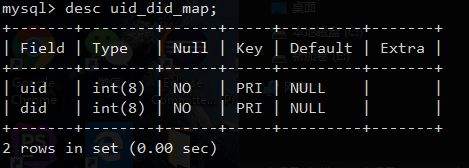


图3-3 uid\_did\_map表

### 3.1.4 设备温度表device\_temperature设计

如图 3-4 device\_temperature所示，device\_temperature表包含三个字段：设备唯一标识（did）、温度（temperature）、更新时间（update\_time）。采集端请求服务端保存数据的时候，根据did在该表写入数据，服务端会生成时间update\_time一起存入表中。update\_time存储的是温度存储时的时间戳，将在客户端请求温度预警信息接口以实时温度查看接口时作为限定条件使用。

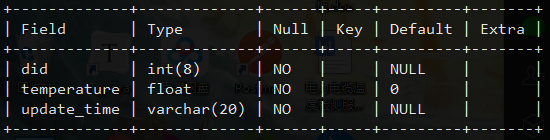


图3-4 device\_temperature表

## 3.2服务端软件功能设计

### 3.2.1 Yii框架安装与调试

1. 首先下载composer的源代码:

php -r "copy('https://install.phpcomposer.com/installer', 'composer-setup.php');"

1. 如图3-5, 执行php composer-setup.php

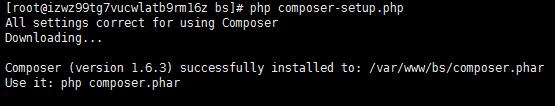


图3-5 安装composer

1. 添加到全局环境，执行composer可以看到图3-6所示，则表示Composer安装完成。



图3-6 composer安装完成

1. 安装Yii框架，执行如下命令。

composer create-project yiisoft/yii2-app-basic /var/www/bs

见到图3-7所示，表示Yii框架代码安装已经完成，接下来就可以在Yii框架的基础上进行接口的开发。

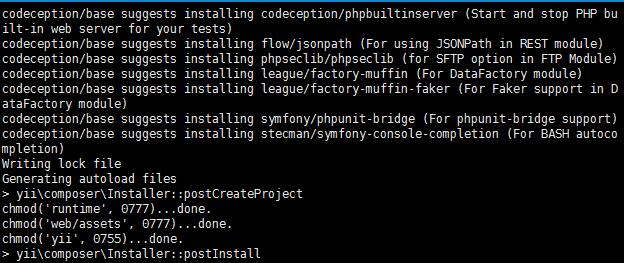


图3-7 安装Yii框架代码

### 3.2.2 Nginx服务器的安装与调试

(1) 首先安装必要的库（Nginx 中gzip模块需要 zlib 库，rewrite模块需要 pcre 库，ssl 功能需要openssl库）。

1. 指定pcre的安装目录 cd /usr/local/，执行wget <ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-8.36.tar.gz>获取对应的源码，解压源码包tar -zxvf pcre-8.36.tar.gz，执行./configure文件生成makefile。就绪之后，依次执行make,make install，安装pcre库完毕。
2. 指定zlib的安装目录 cd /usr/local/,执行wget http://zlib.net/zlib-1.2.8.tar.gz
3. 获得对应源码，解压源码包tar -zxvf zlib-1.2.8.tar.gz，执行./configure文件生成makefile。就绪之后，依次执行make,make install，安装zlib库完毕。
4. 指定ssl的安装目录 cd /usr/local/，执行wget http://www.openssl.org/source/openssl-1.0.1j.tar.gz 获取对应的源码，解压源码包tar -zxvf openssl-1.0.1j.tar.gz，执行./configure文 件生成makefile。就绪之后，依次执行make,make install，安装ssl库完毕。
5. 安装Nginx服务器
6. 指定Nginx的安装目录 cd /usr/local/
7. 执行wget http://nginx.org/download/nginx-1.8.0.tar.gz，获得源码包。
8. 对源码进行解压，tar -zxvf nginx-1.8.0.tar.gz
9. 执行./configure --prefix=/usr/local/nginx
10. 依次执行make, make install

服务端的Nginx web服务器安装完毕，接下来执行/usr/local/sbin/nginx 启动nginx服务器。Nginx安装完成之后要进行相应的配置，Nginx的配置文件在/usr/local/conf/nginx.conf里，如图3-8所示，需要修改部分配置参数。root需要指向Yii框架的安装目录中的入口文件web下。错误日志也重定向到指定目录。

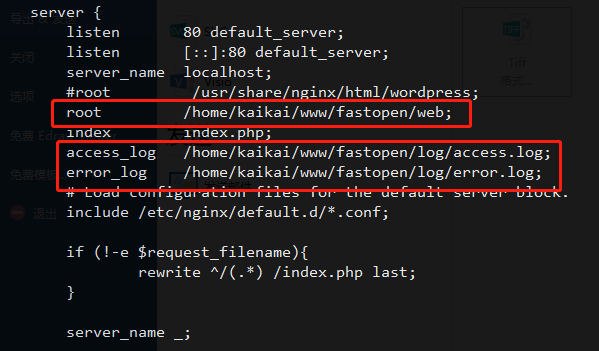


图3-8 nginx配置

另外，需要配置 Nginx的fast\_cgi功能。配置此功能的目的是为了让Nginx更好地支持php，避免每次接收到新的连接都进行fork开辟子进程，FASTCGI管理器将提前开辟好cgi,当有新的连接来时直接进行分配，这样可以有效地提高服务器的效率。添加配置文件如图3-9所示。

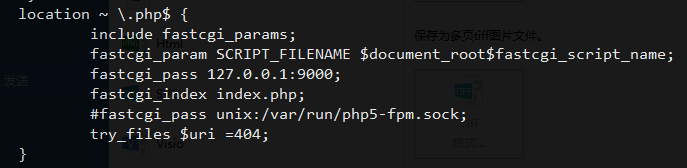


图3-9 fast\_cgi配置

### 3.2.3 PHP7的安装

1. 首先下载PHP7的安装包。

wget http://am1.php.net/distributions/php-7.2.0.tar.bz2

1. 安装必要的依赖工具。

yum -y install libjpeg libjpeg-devel libpng libpng-devel freetype freetype-devel libxml2 libxml2-devel zlib zlib-devel curl curl-devel openssl openssl-devel

1. 准备编译工作。

./configure --prefix=/usr/local/php7

1. 依次执行 make, make install。
2. 至此，php7安装完成。接下来启动php-fpm。

/usr/local/php7/sbin/php-fpm

Php-fpm将作为一个守护进程在后台运行，如图3-10，当Nginx收到新的请求之后fastcgi将通过fastcgi\_pass将用户的请求发送到127.0.0.1:9000进行解析，php-fpm则监听该端口，当收到请求之后启用worker进程进行解析，处理完成之后，将处理结果返回给Nginx, Nginx再返回给用户。

图3-10 请求路由图

### 3.2.4 项目代码设计

1. 项目结构设计

电力电缆温度监测服务端设计遵从MVC（Model-View-Controller）架构，主要目录结构主要有：

1. config

config目录存放的是项目的配置文件，如db.php中存放的是数据库配置，web.php中存放一些框架的配置文件。在有新的请求时，Yii框架会首先加载config配置文件中的配置。

1. domain

domain目录存放的是项目中的服务类，这个目录下的方法全部为接口层面提供服务，把主要的业务逻辑处理都放到该目录下，尽量避免在controller层有太多的业务逻辑代码，这样做的目的是提高代码的可读性。

1. components

components目录下存放的是项目的公共组件。比如公共基类、日志服务类、工具类等。每个新的接口类都会继承公共基类，公共基类中会重写beforeAction以及afterAction方法，主要用于在业务代码之前处理参数以及结束之后处理响应。日志服务主要用来在代码运行时打印必要的信息，用于追查问题。服务端的日志级别分为三种级别：info、error、warning。分别用于打印信息，错误以及警告。工具类中有一些常用的工具方法，比如判断是否是合法的密码，是否是合法的手机号等。

1. controller

controller目录存放的是请求接口方法，controller中的方法以action开头，如actionLogin()等，Yii框架根据URL中的地址信息路由到具体的接口方法。每个controller类都会继承一个基类，基类会重写beforeAction以及afterAction方法，在基类中拿到参数，在controller中调用domain中方法处理业务逻辑。所以controller层中主要做的事情就是接收请求参数，返回响应。

1. enum

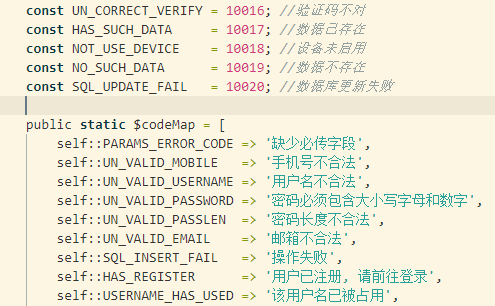
如图3-11所示，enum目录下存放项目中的常量，例如错误码及错误原因等信息。

图3-11 服务端常量

1. models

models层负责连接数据库，提供对数据库的增删改查等操作。一般models层的方法都在domain的服务类中进行调用。在config目录中已经指定了数据库，所以在调用models层方法时只需要提供表名以及数据还有限定条件即可。如果插入失败，会抛出异常，所以在controller中需要捕获异常。

1. 请求及响应格式规定

JSON串是一个由键值对组成的字符串，由于拥有简洁和清晰的结构层次便于编写和解析，被中大型互联网公司广泛作为服务端与客户端之间的参数通信格式。电力电缆温度监测系统服务端设计的所有接口设计都使用HTTP的POST方法进行请求，如图3-12所示，要求客户端在请求接口时POST方法的主体中传送请求参数构成的JSON串。服务端处理之后，同样将处理结果按照如图3-13所示的JSON串的格式传送给客户端，其中的外层的code表示HTTP的状态码，status表示服务端的处理状态，uid表示用户的唯一标识，bizData里是业务参数，如果为空表示成功响应，否则，code表示错误码，reason表示错误原因。



图3-12 客户端请求参数示例 图3-13 服务端响应示例

1. 用户注册接口设计

如图3-14所示，用户注册接口提供给客户端（移动端\PC端），客户端在收集到用户的注册信息之后首先会在本地做一些非空校验，之后上传到服务端。服务端对用户信息做必要得重复性校验和规则校验，首先会判断用户名、手机号是否已经被占用。所有校验通过会生成一个唯一的用户表示uid，并对用户的密码进行加盐哈希处理，之后在user\_base表保存用户信息。如果校验未通过，则返回具体的错误码和失败原因，客户端会提示用户的错误原因，如名字包含特殊字符，密码太短等。

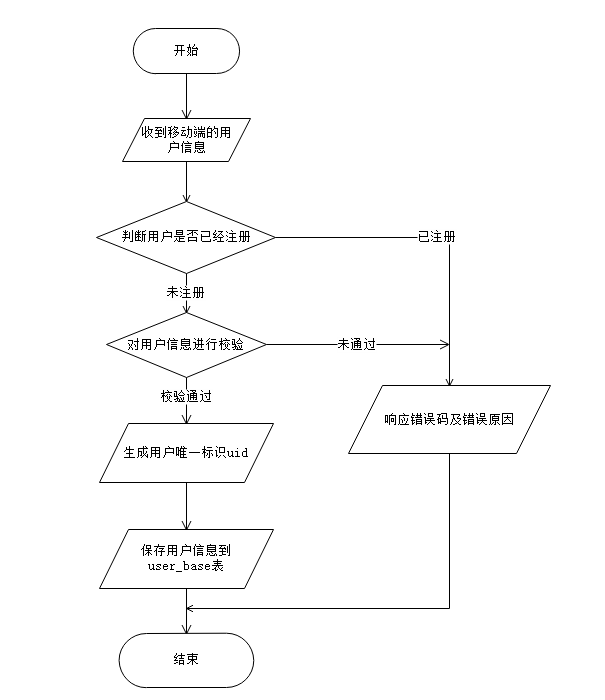


图3-14 用户注册接口流程图

用户注册的时候移动端会给服务端上传用户的用户名、手机号、邮箱和密码。利用正则表达式对部分不合法的手机号字段进行了限制；对用户名进行了全汉字的限制；邮箱必须包含@并且大于7位的限制；密码必须为大小写字母加数字且大于六位的限制。部分关键代码如下：

1. 校验手机号的是否是合法字段，对部分字段的手机号进行了过滤：

$ret=preg\_match('/^(13[0-9]|14[5679]|15[012356789]|16[56]|17[0-9]|18[0-9]|19[89])[0-9]{8}$/', $mobile);

1. 校验用户名的校验，用户名必须是汉字：

$ret = preg\_match("/^[\x{4e00}-\x{9fa5}]+$/u", $userName);

1. 对邮箱的校验，包含@且长度大于7位：

$ret = strstr($email, '@');  
if ($ret === false || strlen($email) < BsEnum::EMAIL\_LESS\_LEN) { return false; }

1. 对密码的校验，必须包含大小写字母和数字，最少六位：

$result = preg\_match( '/(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?!.\*\s)/', $password);

另外，在注册的时候会生成用户的唯一标识uid。这里服务端的uid生成方法选择了将用户名和密码拼接到一起之后进行MD5（消息摘要算法），然后将16进制转化成十进制取0到8位作为用户的唯一标识。关键代码如下所示：

$unique\_no = substr(base\_convert(md5($strInfo), 16, 10), 0, 8);

保存用户密码的时候，为了防止数据库泄漏导致用户密码的不安全的问题。服务端在保存用户密码的时候对密码进行了加盐哈希处理。这里的哈希算法选取sha1算法。加盐哈希可以有效的防止暴力破解和彩虹表破解，保证用户密码的存储安全。

加盐哈希共分为两步：

1. 首先生成生成盐值，uniqid()函数是PHP基于以微秒计的当前时间生成的一个唯一的ID。

base64\_encode(uniqid("BS"))；

1. 将密码和盐值拼接到一起进行Sha1（安全哈希算法）哈希处理。

sha1($password . $salt);

1. 用户登录接口设计

如图3-15所示，移动端收到用户的账户信息和密码之后请求服务端，服务端首先根据账号信息判断是用户名还是密码。然后根据账户信息到user\_base表里拿到盐值和加密之后的密码，将盐值和密码进行加盐哈希处理得到新的加密密码。最后将新的加密密码和数据库里加密的密码进行比较，判断用户是否登录成功。

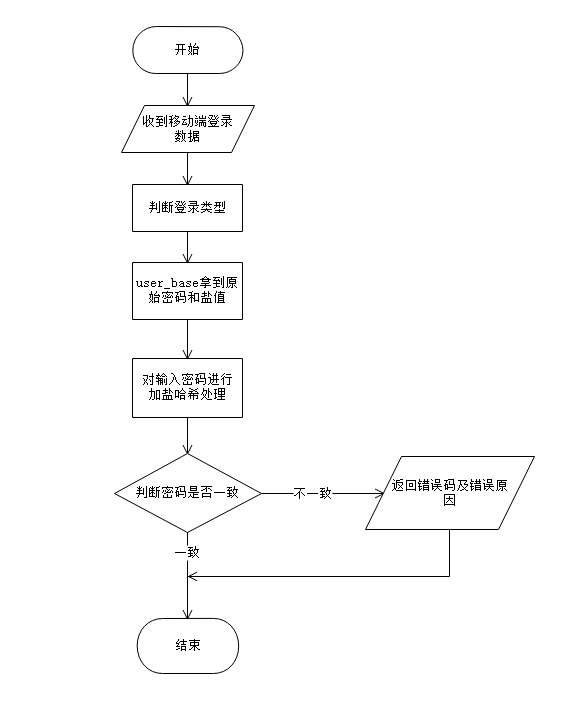


图3-15 用户登录接口流程图

判断登录类型时，如果登录的用户名是纯数字，那么认为是用户名进行登录。其它类型都认为是用户名登录。密码加密时调用的是和用户注册时相同的加密方法，只是登录时的盐值是从数据库中直接拿到的。

1. 修改邮箱/电话号/密码接口设计

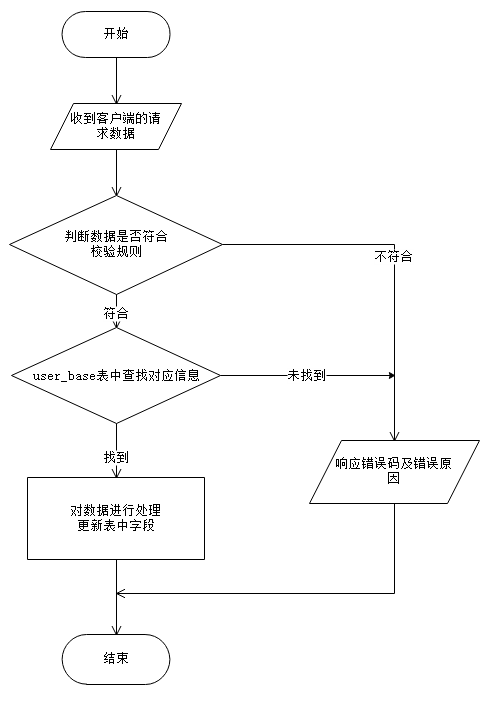
如图3-16所示，当服务端收到修改邮箱、电话号、密码的请求时，先对要修改的内容做对应规则校验。如邮箱必须包含@且大于7位，手机号和密码必须满足正则表达式等。当校验通过之后，根据用户的uid在user\_base表里查看用户是否存在。如果用户不存在，则不允许进行修改操作，向客户端返回错误码及错误原因。如果用户存在，对原始数据进行处理，更改密码时同样需要拿出盐值与新的密码进行加盐哈希处理，之后再更新user\_base表。

图3-16 修改信息接口流程图

1. 获取用户拥有设备权限接口设计

电力电缆温度采集系统设计时建立了uid\_did\_map表，该表保存了用户唯一标识uid和设备唯一标识did的关系映射。uid和did共同作为该表的主键。所以在获取用户设备权限的时候只需要根据用户的uid在该表查询did即可获得用户能够查看的设备权限。

1. 添加/删除用户可以查看设备权限接口设计

如图3-17所示，添加用户设备权限的时候，客户端需要向服务端传送三个字段，用户唯一标识uid，设备唯一标识did，设备验证码verifyCode。设备的验证码是设备出厂时，服务端通过脚本生成的唯一验证码。当用户输入的验证码和device\_info表中的验证码一致时并且当前设备处于在线的状态，用户才能添加设备。

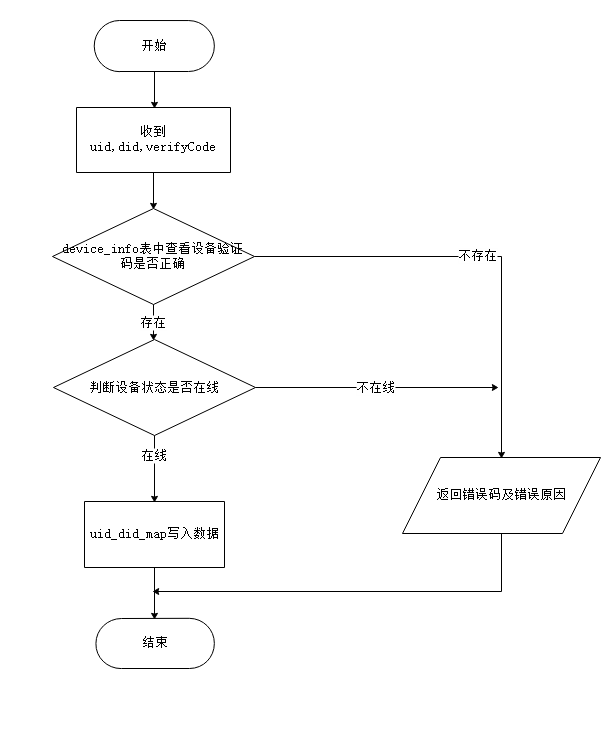


图 3-17 添加设备流程图

1. 查看温度预警信息接口设计

温度预警信息接口用于帮助用户查询温度超过阈值的设备号，系统设计之初，在表device\_info表中threshold字段表示该设备对应的阈值温度。该值默认设置为30摄氏度。服务端单独提供了接口设置该温度。如图3-18所示，当客户端请求查看温度预警信息接口时，服务端会在uid\_did\_map表中找到用户拥有权限的设备号，之后将在device\_info表中查找对应设备的阈值温度，然后根据该温度在device\_temperature表中查找最近一周用户拥有权限的设备并且温度超过阈值的设备号。

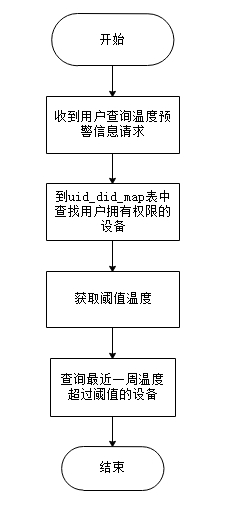


图3-18 查询设备温度流程图

1. 查看实时温度接口设计

客户端通过请求查看实时温度接口获得温度数据，根据温度数据在前端展现一张温度变化的折线图。客户端在请求该接口时必须需要传送三个字段，用户唯一标识uid，设备唯一标识did，以及查询的时间update\_time。还有一个非必传字段interval。首先根据uid在uid\_did\_map表中查找用户拥有权限的设备，判断用户是否拥有查看所请求设备的权限。如果拥有权限，将根据interval字段传来的时间范围在device\_temperature表中查找，如果interval为空，默认设置为一小时（3600秒）。同时当客户端调用该接口时，服务端将对之前的温度进行计算，得到下一时刻的预测温度，连同查询的历史温度共同传送给客户端，工作人员可以根据该温度提前做出预警。

为了准确预测下一时刻的温度，将使用加权平均的算法，根据之前一百条温度数据预测出下一时刻的温度。

1. 温度保存接口设计

温度保存接口提供给采集端调用。采集端每隔一小时调用一次该接口将采集到的温度保存到数据库中。如图3-19每次采集端请求到达服务端，首先查找device\_info表中该设备是否存在，如果不存在返回对应错误码。反之，判断采集端串送的验证码是否和表中验证码是否一致，如果不一致，返回对应错误。如果验证码正确，判断设备的状态是否为0，如果状态码为0，将其置1，并将当前系统时间以及设备did，当前温度保存到device\_temperature表中。

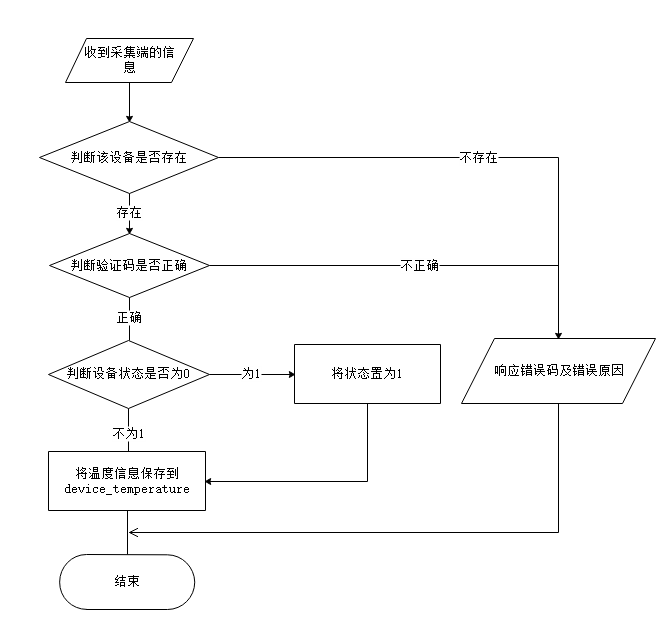


图3-19 采集端保存温度流程图

温度保存到device\_temperature表中关键代码如下：

public function saveBaseInfo($tableName, $data)

{

if (empty($data) || empty($tableName)) {

throw new Exception('参数为空');

}

Yii::$app->db->createCommand()->insert($tableName, $data)->execute();

return true;

}

1. 生成设备号及密码脚本设计

设备在出厂时会自带一个唯一标识码did以及验证码verify\_code，这两个固定参数由服务端的脚本生成。服务端调用该脚本时可以指定生成的设备数量，默认情况下生成十条。

如图3-20所示为生成did的函数，首先由unqid生成一个唯一码拼接到精确到微秒的时间上，再根据这个字符串生成一个md5码。将md5码由十六进制转成十进制取前八位即为设备的唯一标识。由于生成的唯一码以及系统时间是不会重复的，所以生成的设备唯一标识也不会产生重复。并且由于生成了md5，所以设备唯一标识也不具备规律性。

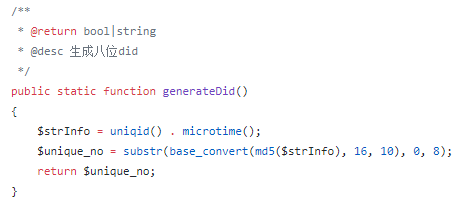


图3-20 生成设备唯一标识码

生成设备验证码的函数如图3-21所示，首先利用一系列随机函数和当前时间以及A-Z字母生成一个随机码，接下来利用MD5函数以及字母数字生成一个随机的验证码。该算法可以保证生成大量验证码的同时拥有很小得重复率。

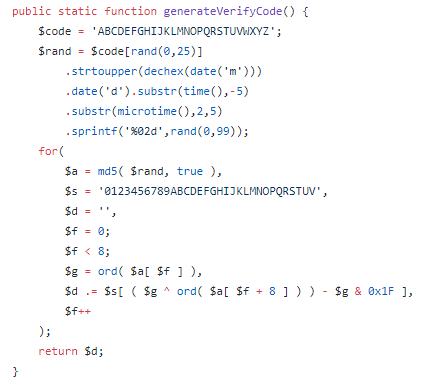


图3-21 生成设备验证码

1. PC端程序设计

考虑到本科阶段专业进行过JAVA的学习，并且JAVA开发桌面程序拥有很高的效率。本次电力电缆温度在线监测系统的PC端程序设计采用JAVA GUI进行开发。

PC端程序设计代码目录结构分为五层，分别为：

1. common

common里包含Util工具类，该类下有一些常用的函数，如sendPostData()函数，用来向服务端发送HTTP的POST数据；json\_encode()函数，用来生成JSON串；json\_decode()函数，用来解析JOSN串等函数。

1. view

view包含GUI的界面布局信息，该目录下有注册登录界面、主页、设备管理、温度查询、温度预警、个人中心等界面布局，界面中所展示的组件都在该目录下定义。

如图3-22所示，用户登录界面需要用户填写用户名以及密码。在该界面可以跳转到注册界面。

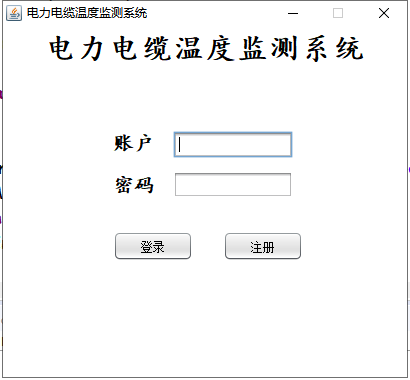


图3-22 用户登录界面

如图3-23所示，用户注册界面用户需要填写用户名、密码、邮箱以及手机号等信息，在客户端对以上信息做了非空校验。当用户点击确认注册之后，将请求服务端的注册接口。服务端将对用户名做汉字限制；对密码做大小写字母及数字校验；对邮箱做包含@并且长度大于7位的校验；对手机号做过滤。

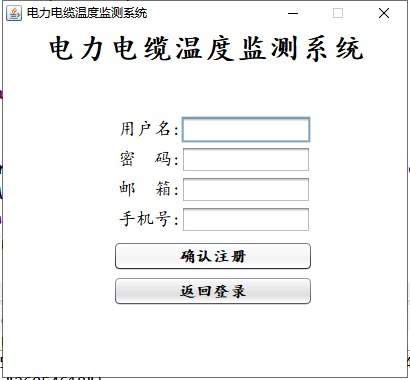


图3-22 用户注册界面

如图3-23所示，为电力电缆温度在线监测PC端主页。该界面包括设备管理、温度查询、温度预警、个人中心等按钮。当用户点击其中一项按钮时，将跳转到对应的界面。

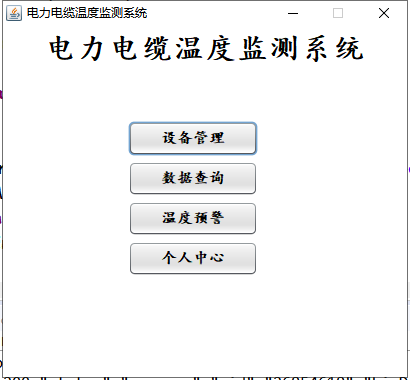


图3-23 系统主页

如图3-24所示，当用户点击设备管理时，将跳转到该界面，展示用户所拥有的设备，用户可以在该界面添加、删除设备、设置设备的阈值温度。添加或者删除时都会调用服务端的添加设备接口或者删除设备接口，点击阈值按钮时将调用设置设备阈值温度的接口。



图3-24 设备管理界面

如图3-25所示，当用户点击温度报警按钮时，将跳转至温度报警信息界面，该界面将展示用户拥有权限的所有超出阈值温度的设备。进入该界面时会启动定时任务，每隔十秒将刷新一次界面，动态展示报警信息。

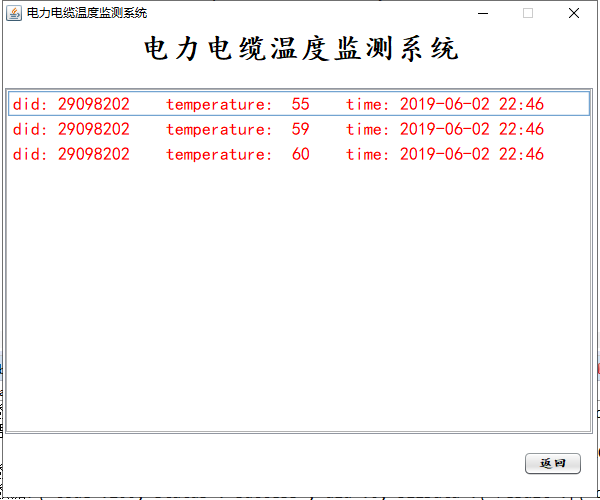


图3-25 温度报警界面

如图3-26所示，用户点击温度查询接口后会跳转到设备选择界面，该界面会调用获取用户设备权限接口获得用户拥有的设备展示到下拉框中。



图3-26设备选择界面

设备选定之后，将会跳转至如图3-27所示温度信息界面，该界面将展示选择设备最近一小时的温度信息折线图以及预测温度值。

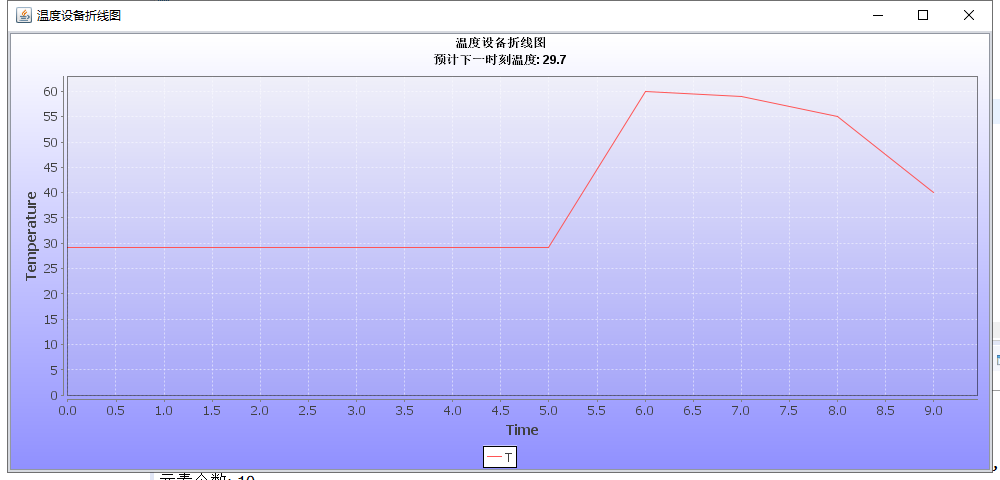


图3-27 温度信息界面

如图3-28所示，用户在点击个人中心之后将跳转到该界面。个人中心界面将展示电力电缆温度监测系统的软件信息，同时用户在该界面下可以修改个人信息，如用户名、账号、密码等。修改个人信息时同样会对修改的信息进行和注册时同样的规则校验。

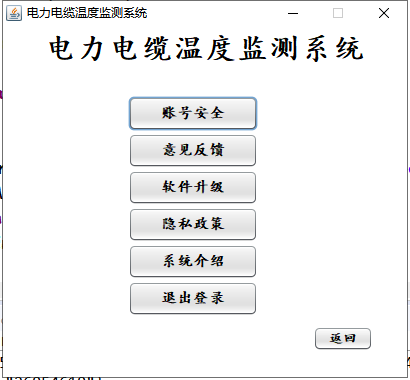


图3-28 个人中心界面

1. domain

domain目录下的类负责监听view类中定义的组件如按钮、文本框等。将监听到的数据进行非空校验处理并与服务端进行通信，接收到服务端的响应之后，处理服务端的响应。本次毕设所有的代码编写都遵循面向对象的原则，因此为了实现service类对view目录中的组件进行控制，如图3-29所示，需要让每一个service类继承view中的类。

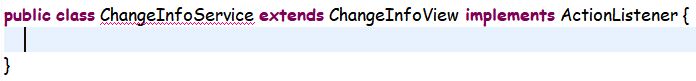


图3-29 类的继承

同时，考虑到系统界面的组件太多，如果对每个组件都重写事件监听，代码讲过于冗余，为此让service中的类继承ActionListener，并重写ActionListener类中的方法，这样只需要两步就可以实现对一个组件的监听。如图3-30所示首先为组件添加事件监听。接下来重写ActionListener类中的actionPerformed方法。

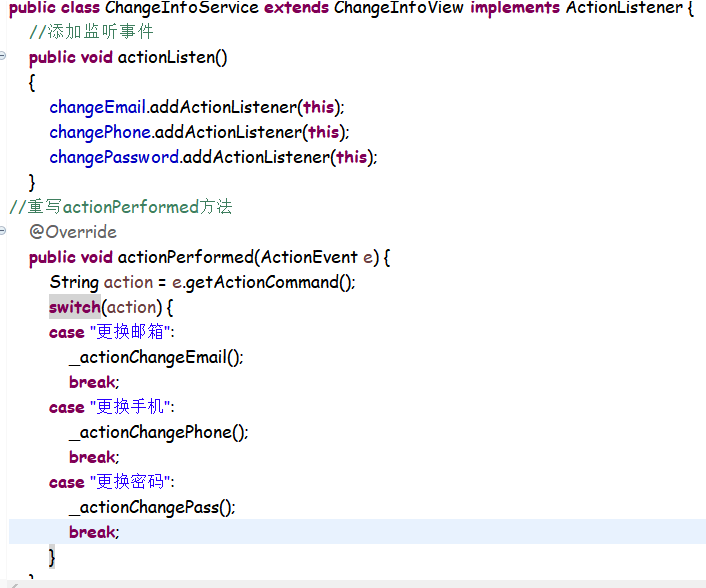


图3-30 事件监听的实现

1. controller

controller层中的类中的方法将界面展示以及事件监听收敛到一个函数中，当主函数或者界面需要跳转的时候将调用controller中的方法。在domain层之外单独设置controller层的目的是使代码收敛，方便修改。

1. enum

如图3-31所示，enum目录下放的是一些常量，如接口地址的请求地址，项目中的一些常量等。



图3-31 常量值

# 4 系统运行调试

## 4.1 测试环境概述

服务端接口完成编写之后，需要观察接口是否能够正常工作。服务端接口能够正常工作的指标就是按照规定输入是否能够得到对应的输出。为此，需要观察服务端的日志是否正常以及数据库数据是否正常操作。由于客户端以及采集端由另外两位同学开发所以需要利用工具对接口进行请求。本次毕业设计服务端选择的测试工具是Postman。Postman可以提供几乎所有的HTTP请求，而电力电缆温度监测系统服务端所有的接口都采用HTTP的POST方法。如图4-1所示，只需要输入请求地址以及请求参数，然后观察服务端的响应。

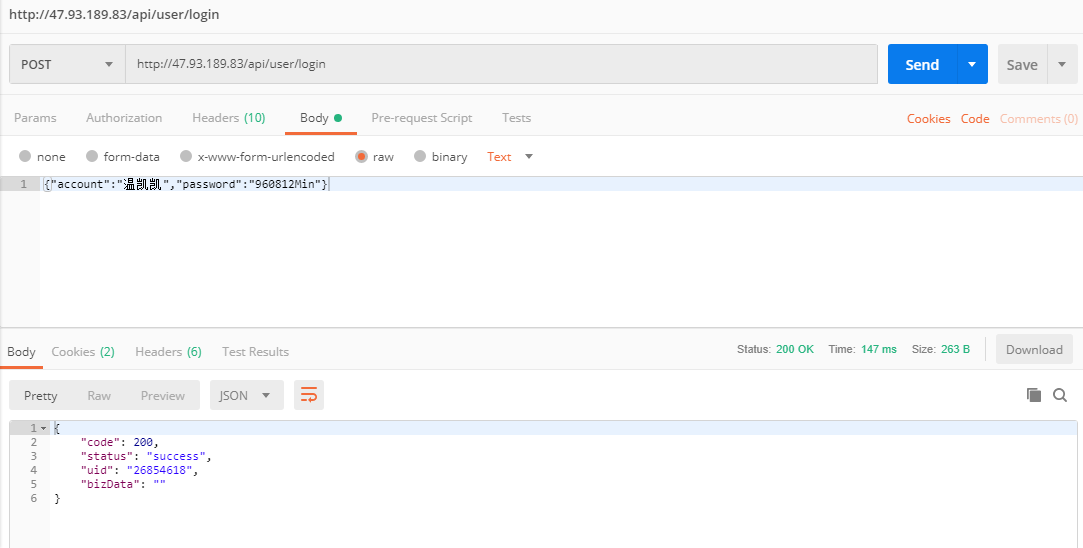


图4-1 Postman界面

服务端代码在编写的时候，在关键参数附近如接收到的参数、处理结果等位置利用Flogger类添加了日志信息。所以在发出请求的时候，同时利用Linux的tail -f \*命令动态观察如图4-2所示服务端的请求日志可以发现大部分问题的原因，服务端日志会在项目运行时，在runtime目录里自动生成log文件，需要定期清理日志文件。

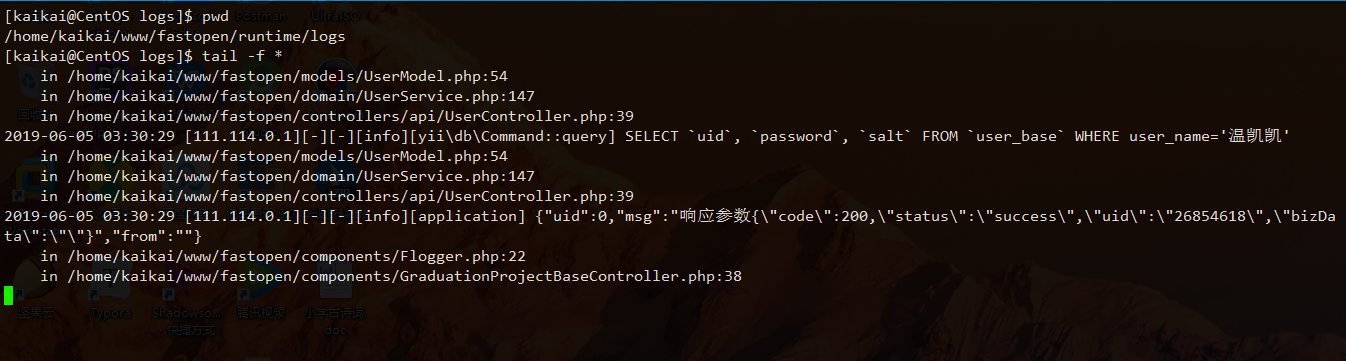


图4-2 服务端日志信息

在开发过程中有时候可能会遇到请求发送成功，但是服务端并没有打印请求日志，这时就需要考虑请求是否到达服务端，所以还需要观察图4-3所示Nginx的请求日志。

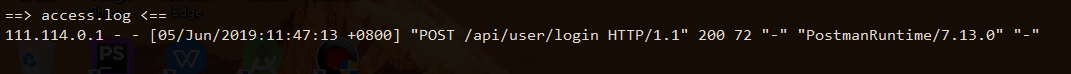


图4-3 nginx日志信息

## 4.2 测试过程

# 4.2.1通信测试

如图4-4所示，通过Postman请求服务端的注册接口，返回响应状态200，bizData为空，表示登录成功。

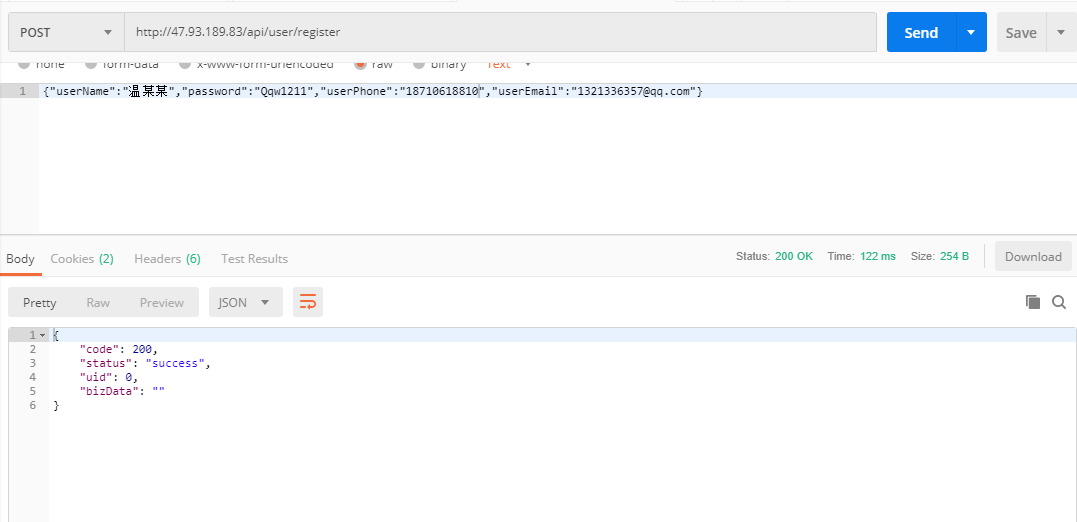


图4-4 Postman注册请求

观察Nginx日志如图4-5所示，成功收到注册请求。

注册Ng

# 图4-5 Nginx注册日志

观察数据库user\_base表，如图4-6所示，数据存储正常，密码加盐哈希正常，uid生成正常。服务端通信测试完毕，通信正常。

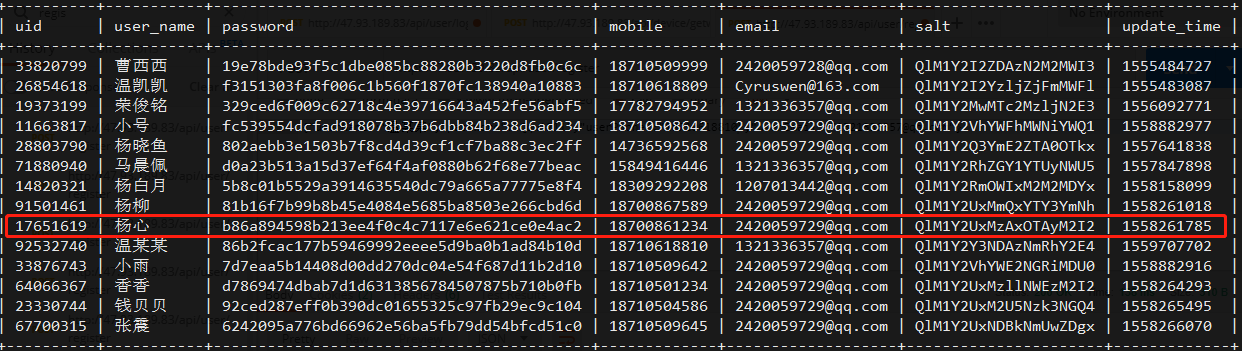


图4-6 user\_base表存储用户信息

# 4.2.2功能测试

1. 用户登录接口测试

用户登录时需要观察服务端日志信息中记录取值加盐哈希是否正确以及输入错误的密码服务端能否做出正确响应。如图4-9服务端日志信息所示，服务端可以正常处理取值加盐哈希操作。

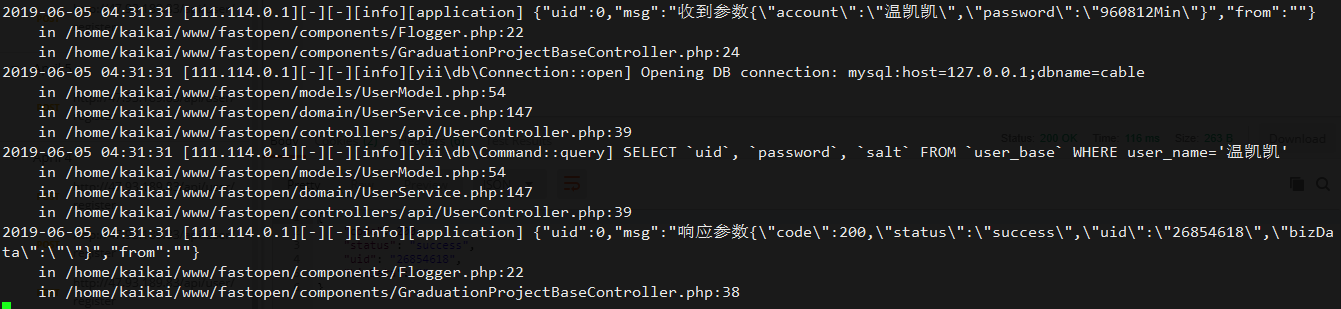


图4-9 服务端日志信息

如图4-10所示，服务端能够识别错误密码，并在返回参数bizData中标明了错误码及错误原因。用户登录接口测试正常。

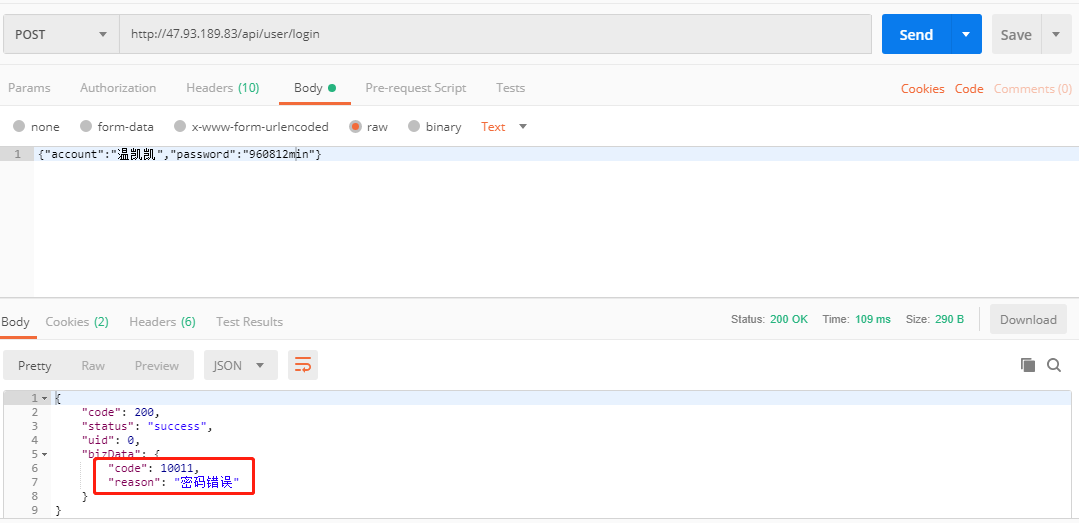


图4-10 postman用户登录接口测试

1. 用户注册接口测试

用户登录时服务端校验了用户名必须是汉字并且大于两个字；用户手机号过滤特殊字段；密码必须包含大小写字母且大于六位；邮箱大于七位并且包含@。针对上述校验对请求参数做出改变，如用户姓名包含特殊字符，密码纯数字等。如图4-11所示，可以看到服务端成功返回了错误原因。注册接口测试正常。

# 

图4-11 postman用户注册接口测试

1. 获取用户拥有设备权限接口测试

如图4-12所示，到uid\_did\_map表中查询uid为26854618拥有的设备权限。

可以看到该用户共拥有九个设备的权限。

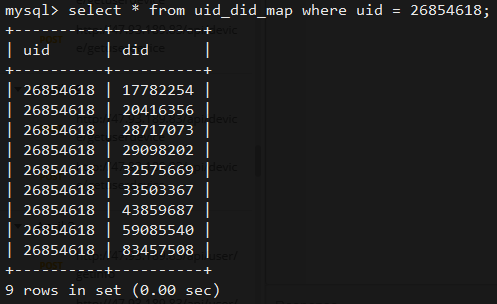


图4-12 用户拥有的设备权限

接来下利用postman请求获取用户设备权限接口，得到如图4-13所示，查询到的用户设备权限与在uid\_did\_map表中查询到的数据相同。所以获取用户拥有设备权限接口功能正常。

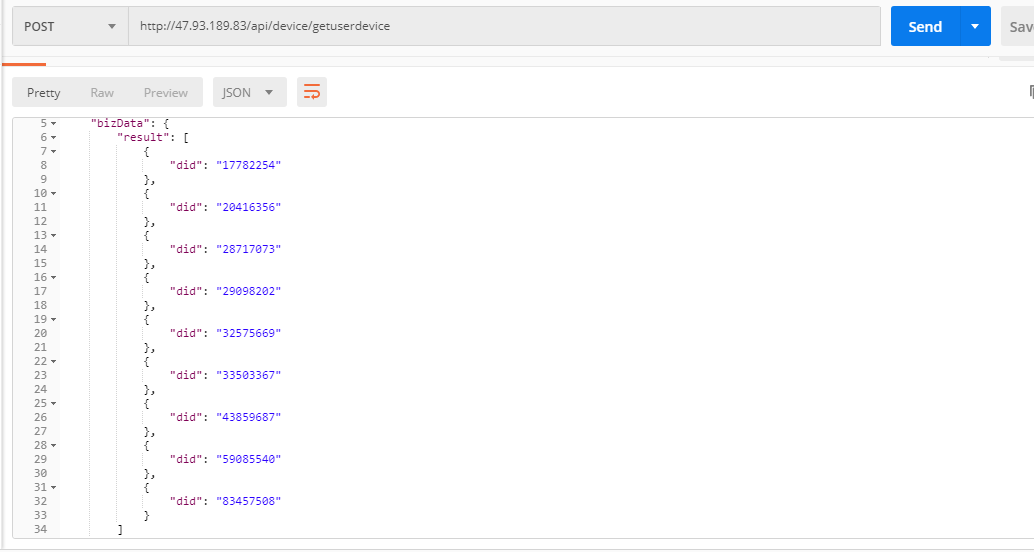


图4-13 获取用户设备权限

1. 查询温度接口测试

如图4-14所示，用postman请求查询温度接口时，只需要提供用户的uid以及设备的did即可查询最近一小时的温度数据，但是由于device\_temperature没有最近一小时的数据，因此多传一个字段interval，该字段是扩展字段，用来查询指定时间的温度。设置interval为14\*24\*3600，表示最近两周的温度数据。



图4-14 温度查询接口响应结果

可以看到forecastTemperature温度预测字段有值，并且返回的温度数据格式正常，所以温度查询接口功能正常。

1. 温度报警接口测试

温度报警接口需要提供用户的uid，服务端会返回最近一周温度超过阈值的设备信息。如图4-15使用postman测试，可以看到服务端成功返回了最近一周该用户所有的超过阈值温度的设备信息，温度报警接口工作正常。

图4-15 温度报警接口响应参数

# 5 总结与展望

目前电力电缆温度在线监测服务端可以完成以下主要功能：

1. 用户管理。电力电缆温度在线监测系统服务端可以完整地完成用户的注册、用户登录以及用户的个人信息的修改，如账号、密码的修改。
2. 设备管理。用户可以通过电力电缆温度在线监测系统服务端完成添加或删除设备权限、设置设备温度阈值、生成新的设备号以及验证码。
3. 温度报警与管理。电力电缆温度在线监测系统服务端可以完成在线温度保存、温度查询、温度预测、温度报警等功能。
4. PC端管理界面。用户可以通过PC端管理界面可视化地使用电力电缆温度在线监测系统服务端的功能。

电力电缆温度在线监测系统服务端的基本功能已经完成，服务端可以连接采集端以及客户端（手机app\PC端），对电缆的温度数据进行报警以及对温度进行预测。服务端采用的是Nginx以及Yii框架的组合，机器条件允许的情况下最大可支持三万到四万的并发量，基本可以满足本系统以后功能延伸的需求。

后期本系统主要可以扩展的功能有两部分。

1. 首先可以开发网页版电力电缆温度在线监测系统。服务端采用的Yii框架支持网页（HTML+CSS）开发，电力电缆温度在线监测系统后续可以在原有基础上开发第三个客户端：网页系统。
2. 温度预测功能可以进行更深层次的开发。目前电力电缆温度在线监测系统服务端对温度的预测是基于对历史温度数据进行加权处理得到，这种做法可以得到一个大概的预测温度，但是不够准确。在服务端代码开发中，笔者已经将预测模块单独分离出来。后期对预测功能进行扩展的时候可以在原有基础上进行二次开发。也可以单独开发更复杂的第三方预测服务，在电力电缆温度在线监测系统的服务端调用。

# 致 谢

四年的大学学习生活在即将划上一个句号，而于我的人生来说却仅仅只是一个逗号，我将面对新的征程的开始。本次毕业设计电力电缆温度在线监测系统服务端设计采用了多种互联网企业所用的技术，如MySQL、Nginx、php、Yii框架、Linux等。从开始学习基础的技术逐步到独立开发出整套系统，我经历了很多难以解决的问题。不过在小组组员以及老师的帮助下，将技术难题一一解决。在此，我要感谢我的毕设老师。也许我不是您最出色的学生，但您却是我所最尊敬的老师。您是如此的治学严谨，学识渊博，视野广阔，思想深刻，您用心为我营造一种良好的学术氛围，让我的论文更加的严谨。同时，我还要感谢一下一起完成毕业论文小组的同学们，如果没有你们的支持和倾心的协助，我是无法解决这些困难和疑惑，最终能够让本文顺利完成。至此论文付梓之际，我的心情无法保持平静，从开始选择课题到论文的顺利答辩，有无数可敬的师长、朋友给了我很多的帮助，在这里请您接受我诚挚的谢意! 最后，再次对那些在论文完成过程中，关心、帮助我的同学和朋友们表示衷心地感谢！

# 参 考 文 献

1. 陈浩超.电力电缆温度在线监测系统设计及应用[D].华南理工大学,2017.
2. 王龙军,李华志,朱雪梅.JSON在Android移动图书馆开发中的应用[J].电脑编程技巧与维护,2019(04):74-75+89.
3. 杨婷婷.PHP开发中MVC模式的应用研究[J].淮南职业技术学院学报,2019,19(02):84-85.
4. 史桂红.基于Android及JSON的天气预报APP设计与实现[J].苏州市职业大学学报,2019,30(01):27-32.
5. 李琨,贾立伟,石晓明.基于Yii2框架的《正常人体功能》课程教育云平台的设计与实现[J].电脑知识与技术,2019,15(08):51-53.
6. 薛煜阳.基于Yii框架的教师工作量管理系统设计与实现[J].电脑编程技巧与维护,2018(05):90-91+130.
7. 苏翔宇,朱爱群.CentOS7下基于Nginx的反向代理及负载均衡研究与实现[J].现代计算机(专业版),2018(10):61-64.
8. 文晖.面向连接的Socket技术在物联网程序设计中的应用[J].兰州石化职业技术学院学报,2017,17(04):12-14.
9. 田明山.Centos7.0新特性分析[J].电脑知识与技术,2017,13(35):122-123.
10. 苏宁,李邦峂,孟庆铭.电力电缆温度智能在线监测系统[J].科技创新与应用,2017(25):98-99.
11. 翟浩.电力电缆故障在线监测及预警系统的研究[D].东南大学,2016.
12. 牧春.MVC设计模式在Yii框架中的应用与研究[J].电子技术与软件工程,2015(12):64-65.
13. 贺佳佳.基于Yii的微店服务系统的设计与实现[D].华中科技大学,2015.
14. 赵晋萍.基于LAMP框架的通用数据测试平台的设计与实现[D].北京交通大学,2015.
15. 胡敏. Web系统下提高MySQL数据库安全性的研究与实现[D].北京邮电大学,2015.
16. 钟蕾,孟庭羽.基于Java GUI实用编程技术的研究[J].信息技术,2014(06):201-202+206.
17. 李华忠.电缆接头温度监测技术的研究[D].哈尔滨理工大学,2014.