

浙江大学

本科实验报告

物联网应用网站

开发报告

课程名称: B/S 体系软件设计

姓 名: 黄仁泓

学 院: 竺可桢学院

系: 竺可桢学院

专 业: 混合班

学 号: 3180101969

指导教师: 胡晓军

2020 年 6 月 29 日

浙江大学实验报告

课程名称: B/S 体系软件设计 实验类型: 综合型
实验项目名称: 物联网设备管理平台网站
学生姓名: 黄仁泓 专业: 计算机科学与技术 学号: 3180101969
同组学生姓名: 无 指导老师: 胡晓军
实验地点: 无 实验日期: 2021 年 05 月 05 日

报告共分为四部分, 第一部分开发报告&功能实现具体阐述了预期功能的实现和开发过程, 包括各个需求的具体实现和截图。第二部分为使用方法。第三部分为测试报告。第四部分为开发心得。

第一部分: 开发报告&功能实现

一、设计概述

设计目标和任务

本部分为物联网设备管理平台的设计文档, 初步设计项目所需的实验技术框架、模块设计、类图设计、界面设计等。

本项目为物联网应用类型网站, 具体的说, 本项目实现了设备的登记/修改/删除等功能, 提供上报数据查询统计界面, 并提供地图界面展示设备信息, 以便于设备的管理。能有效处理指定物联网终端模拟器发送的数据, 提供较强的设备管理的功能。同时保证网站的美观性和平台适配性(能够在手机浏览器/微信等应用内置的浏览器友好显示)

二、技术框架

项目整体采取 Browser/Server 架构, 采用前后端分离的开发模式, 前后端数据交互, 前端页面调用后端提供的接口以实现。

2.1 前端技术

本项目暂定使用 Vue.js +Element UI+ Apache Echarts 进行设计

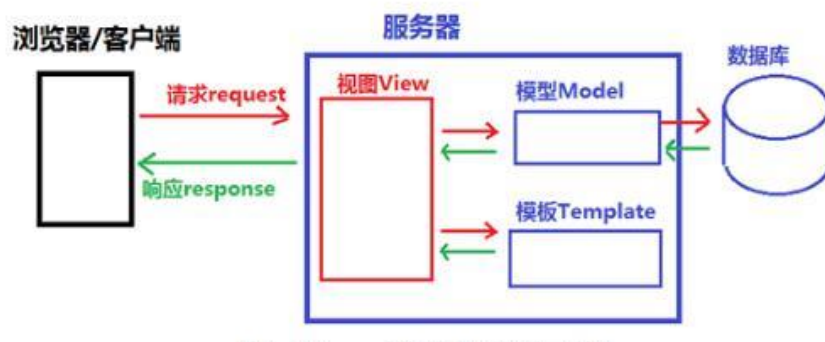
Vue 是一套构建用户界面渐进式框架的 java script 框架, 其高效的开发效率、渐进式的开发理念, 对于开发者而言更加容易上手。因为 vue 采取自底向上的增量开发设计, 其核心库只关注视图层, 非常容易与其他库或已有项目整合。

而 Element UI 作为 Vue 的组件库, 可以提供成熟和美观的 UI 界面。

对于地图绘制以及表格绘制等功能，本项目采用 echarts 绘制柱状图、折线图等，而设备地图绘制则采用百度开放平台的 JavaScript API——Bmap 来绘制。

2.2 后端技术

本项目暂定采用 Django 作为后端框架，是一个开源 Web 应用框架，为了各组件间保持松耦合关系，类似于 MVC 模式，django 采用 MTV 设计模式。或者采用 Spring Boot 作为后端的框架进行设计。



图表 1 MTV 设计模式

而本次项目中，数据库方面采用关系型数据库 MySQL 等数据库即可。本项目采用 Django 作为后端，利用 django 自带数据库创建和迁移，可以迅速完成建表等部署。

三、需求分析及功能实现

1. 搭建一个 mqtt 服务器，能够接收指定的物联网终端模拟器发送的数据

（已实现）

MQTT 是一个基于客户端-服务器的消息发布/订阅传输协议，可以以极少的代码和有限的带宽，为连接远程设备提供实时可靠的消息服务。Mqtt 设备客户端（智能家居、机器人）发送信息，通过 MQTT 协议，目前已经扩展出了数十个 MQTT 服务器端程序，我们可以利用对应的包以接受、处理 mqtt 消息。

我采用 python 语言下的 django 进行开发，使用 paho-mqtt 库来进行 MQTT 的通信进行处理。通过设定对应的回调函数，接受数据。下面是项目接受的 MQTT 端口。将读入进来的数据读入到 MySQL 的数据库中。

```
Host = "localhost" # 服务器地址
port = 1883 # 通讯端口
username = 'username' # 用户名
password = 'password' # 密码
topic = 'testapp' # 订阅主题名
```

图表 2

下面是读入进来的设备数据：

id	client_id	info	value	alert	lat	lng	timestamp
1	device0003	Device Data 2021/06/26 15:22:38	11	0	30.17281839847565	120.15304485559464	1624692158670
2	device0001	Device Data 2021/06/26 15:22:38	19	0	30.47739017009735	119.99543110132218	1624692158670
3	device0002	Device Data 2021/06/26 15:22:39	51	0	30.49904718399048	120.4702555179596	1624692159664
4	device0005	Device Data 2021/06/26 15:22:40	63	0	30.35447030067444	120.41989493370056	1624692160664
5	device0003	Device Data 2021/06/26 15:22:40	14	0	30.130592536926272	120.18421720266343	1624692160811
6	device0004	Device Data 2021/06/26 15:22:44	58	0	30.14647574424744	120.00382305383683	1624692164664
7	device0001	Device Data 2021/06/26 15:22:46	39	0	30.108012652397157	119.92130843400956	1624692166817
8	device0002	Device Data 2021/06/26 15:22:47	26	0	30.498814964294436	120.21179875135422	1624692167668
9	device0005	Device Data 2021/06/26 15:22:48	53	0	30.423066234588624	120.13406218290329	1624692168678
10	device0003	Device Data 2021/06/26 15:22:48	57	0	30.344313192367554	120.21256972551346	1624692168814
11	device0003	Device Data 2021/06/26 15:22:49	79	0	30.23083419799805	120.39350821971894	1624692169819
12	device0002	Device Data 2021/06/26 15:22:51	28	0	30.449258303642274	120.04772498607636	1624692171675
13	device0004	Device Data 2021/06/26 15:22:52	47	0	30.119951510429384	120.2591132760048	1624692172671
14	device0003	Device Data 2021/06/26 15:22:52	4	0	30.28749103546143	120.35869218111038	1624692172831
15	device0003	Device Data 2021/06/26 15:22:52	11	0	30.38294496536255	120.19372384548188	1624692172831
16	device0001	Device Data 2021/06/26 15:22:53	24	0	30.351881623268127	119.94432518482209	1624692173831
17	device0005	Device Data 2021/06/26 15:22:54	17	0	30.17179446220398	120.26992597579957	1624692174691
18	device0001	Device Data 2021/06/26 15:22:54	50	0	30.278176999092103	120.1294530749321	1624692174835
19	device0002	Device Data 2021/06/26 15:22:56	3	0	30.427501940727236	120.06941579580308	1624692176688

图表 3

2. 实现用户注册、登录功能，用户注册时需要填写必要的信息并验证，如用户名、密码要求在 6 字节以上，email 的格式验证，并保证用户名和 email 在系统中唯一。

(已完成)

通过 MD5 算法或加盐 hash 的形式以实现验证用户密码。用户注册所需要填写的信息，利用 view 中获取信息并将信息（用户名和 email）存储于数据库中。

在读入用户信息的时候，注意匹配信息格式，验证邮箱信息是否正确，验证用户密码长度是否否在 6 字节以上，并且通过维护数据库的方式，保证用户名和 email 在系统保持唯一。

下面是具体实现的格式检查。

```
if len(password1) < 6:
    message = "密码长度必须大于等于6"
    return render(request, 'login/register.html', locals())
if CheckEmail(email) is False:
    message = "邮箱必须符合格式"
    return render(request, 'login/register.html', locals())
if password1 != password2: # 判断两次密码是否相同
    message = "两次输入的密码不同!"
    return render(request, 'login/register.html', locals())
else:
    same_name_user = models.User.objects.filter(name=username)
    if same_name_user: # 用户名唯一
        message = '用户已经存在, 请重新选择用户名!'
        return render(request, 'login/register.html', locals())
```

其次我在 session 中记录用户的登录时场景，可以保存用户登录的状态。
通过下面的登录和注册界面进行登录或注册：

BS PROJECT



图表 4 登录界面

BS PROJECT



图表 5 注册界面

3. 提供设备配置界面，可以创建或修改设备信息，包含必要信息，如设备 ID、设备名称等（已完成）

设计好配置类型，当用户创建/修改设备信息时，通过前端获取修改信息，并针对设备的数据格式对数据库进行修改。这部分可以通过较为成熟的组件库以实现。

在设计中我采用 vue 框架下的 axios 基于 promise 的 HTTP 客户端这一工具，用于将数据从前端发送到后端中。在配置好 vue 和 django 的跨域设置下，在用户填写好表单信息后，可以将表单信息通过 axios 发送到后端，进行创建/修改/删除设备信息数据库等操作。

下面是创建/删除/修改等界面的实际功能图(位于侧边栏的设备配置界面):



设备id	设备名	设备描述
device0002	倚天屠龙剑	可斩恶龙
device0003	电视	使用电子技术传送活动的图像画面和音频信号的设备
device0001	ipad	平板电脑
device0004	智能冰箱	智能控温
device0005	摄像头	安全监控记录

图表 6 创建设备

BS Project-IOT网站 欢迎浏览

创建设备

修改设备

删除设备

查询设备记录

设备ID

设备ID

设备名字

设备名字

设备描述

设备描述

创建设备

取消

设备id	设备名	设备描述
device0002	倚天屠龙剑	可斩恶龙
device0003	电视	使用电子技术传送活动的图像画面和音频信号的设备
device0001	ipad	平板电脑
device0004	智能冰箱	智能控温
device0005	摄像头	安全监控记录

图表 7 修改设备

BS Project-IOT网站 欢迎浏览

创建设备

修改设备

删除设备

查询设备记录

设备ID

删除的设备ID

设备名字

删除的设备名字

删除设备

取消

设备id	设备名	设备描述
device0002	倚天屠龙剑	可斩恶龙
device0003	电视	使用电子技术传送活动的图像画面和音频信号的设备
device0001	ipad	平板电脑
device0004	智能冰箱	智能控温
device0005	摄像头	安全监控记录

图表 8 删除设备

BS Project-IOT网站 欢迎浏览

创建设备

修改设备

删除设备

查询设备记录

设备名

device0001

数值区间

30~50

是否告警

否

查找

取消

设备名	设备信息	数值	是否告警	纬度	经度
device0001	Device Data 2021/06/26 15:22:46	39	0	30.108012652397157	119.92130843400956
device0001	Device Data 2021/06/26 15:22:54	50	0	30.278176999092103	120.1294530749321
device0001	Device Data 2021/06/26 15:23:05	42	0	30.181979417800903	120.17391884326935
device0001	Device Data 2021/06/26 15:23:09	49	0	30.374096846580507	120.33790692090989
device0001	Device Data 2021/06/28 17:18:01	38	0	30.323145866394043	120.15620596408844
device0001	Device Data 2021/06/28 17:18:37	38	0	30.488905143737796	120.16524951457978

图表 9 查询设备

4. 提供设备上报数据的查询统计界面

(已完成)

用户提出请求 request 查询设备上报数据的查询统计信息，使用 axios 的 post 包发送到后端，并在后端的数据库中进行统计，并将统计结果返回至前端的视图中。而前端可以采用成熟、美观的可视化组件 Element UI + Echarts 即可实现。

目前统计信息可支持查询特定设备的所有数据记录（可支持条件筛选），可以获取特定设备的 value 数值并绘制相应的折线图，可以查询特定设备平均经纬度等功能。

5. 提供地图界面展示设备信息，区分正常和告警信息，并可以展示历史轨迹

(已完成)

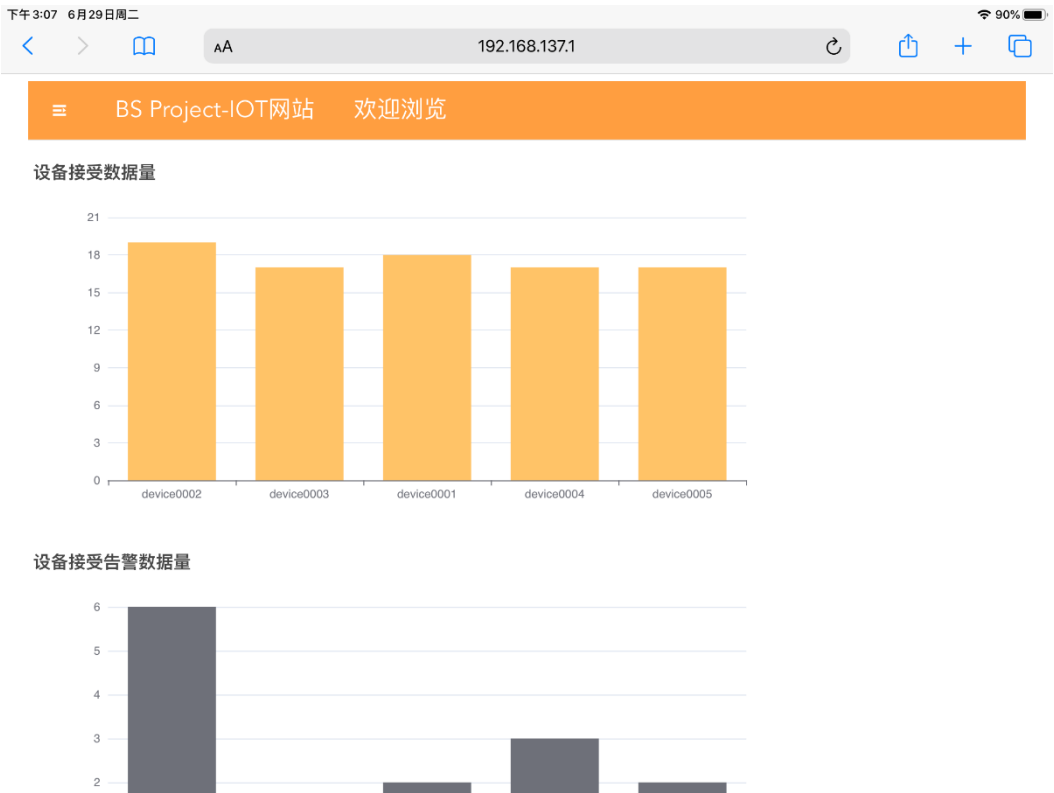
在前端中加入对应的地图组件（百度地图等），并将后端数据库返回的设备信息（地理位置）传给前端的百度地图 Bmap 组件进行可视化，实时展示设备的地理位置，并可以区分正常和告警状态，也可以展示设备的历史的轨迹。

同时地图支持漫游和缩放，可以便于用户精准定位到设备位置信息。

或桌面端进行判断，若某种组件并不支持某浏览器，则更换对应的组件库即可。在项目中，主要使用的组件库 echarts 和 element-UI 均对移动端有很好的支持，在移动端(ipad)上，显示的界面如下：



图表 11 移动端登录界面



图表 12 首页统计

下午3:096月29日周二192.168.137.190%

BS Project-IOT网站 欢迎浏览

创建设备

修改设备

删除设备

查询设备记录

设备名

device0002

数值区间

20~80

是否告警

输入是否告警,格式例如: 是 或 否

查找

取消

设备名	设备信息	数值	是否告警	纬度
device0002	Device Data 2021/06/26 15:22:39	51	0	30.49904718399048
device0002	Device Data 2021/06/26 15:22:47	26	0	30.498814964294436
device0002	Device Data 2021/06/26 15:22:51	28	0	30.449258303642274
device0002	Device Data 2021/06/26 15:22:56	3	0	30.427501940727236
device0002	Device Data 2021/06/26 15:23:00	38	0	30.293308711051942
device0002	Device Data 2021/06/26 15:23:09	85	1	30.11632106304169
device0002	Device Data 2021/06/28 17:17:52	41	0	30.397896695137025

图表 13 查询记录

下午3:126月29日周二192.168.137.189%

BS Project-IOT网站 欢迎浏览

设备名

device0002

数值区间

输入Value数值区间,格式例如: 11~50

是否告警

输入是否告警,格式例如: 是 或 否

查找

取消

地图操作

缩小

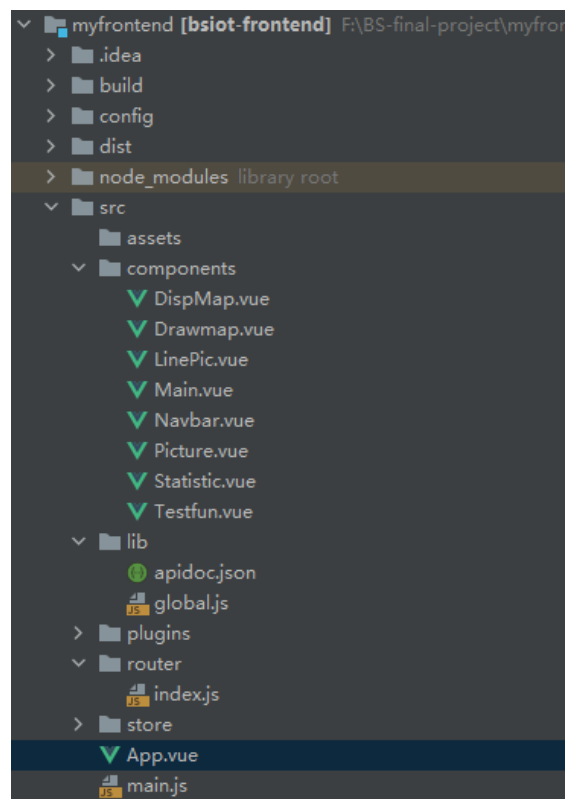
放大

图表 14 移动端地图显示

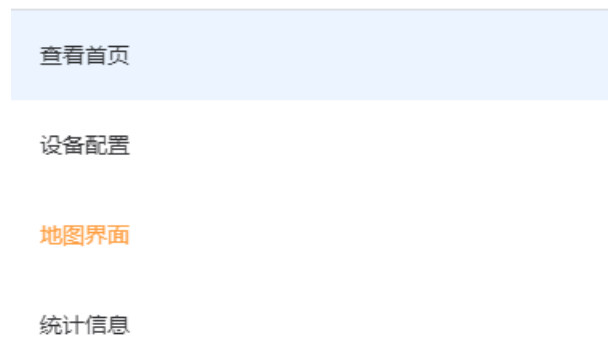
四、项目结构

前端（frontend）

前端部分采用 vue 的脚手架结构，具体文件布局见下：



其中在路由设置方面，下面四个模块分别对应着



查看首页：<http://localhost:8081/picture> 对应 Picture.vues

设备配置：<http://localhost:8081/test> 对应 Testfun.vues

地图界面：<http://localhost:8081/Drawmap> 对应 Drawmap.vues

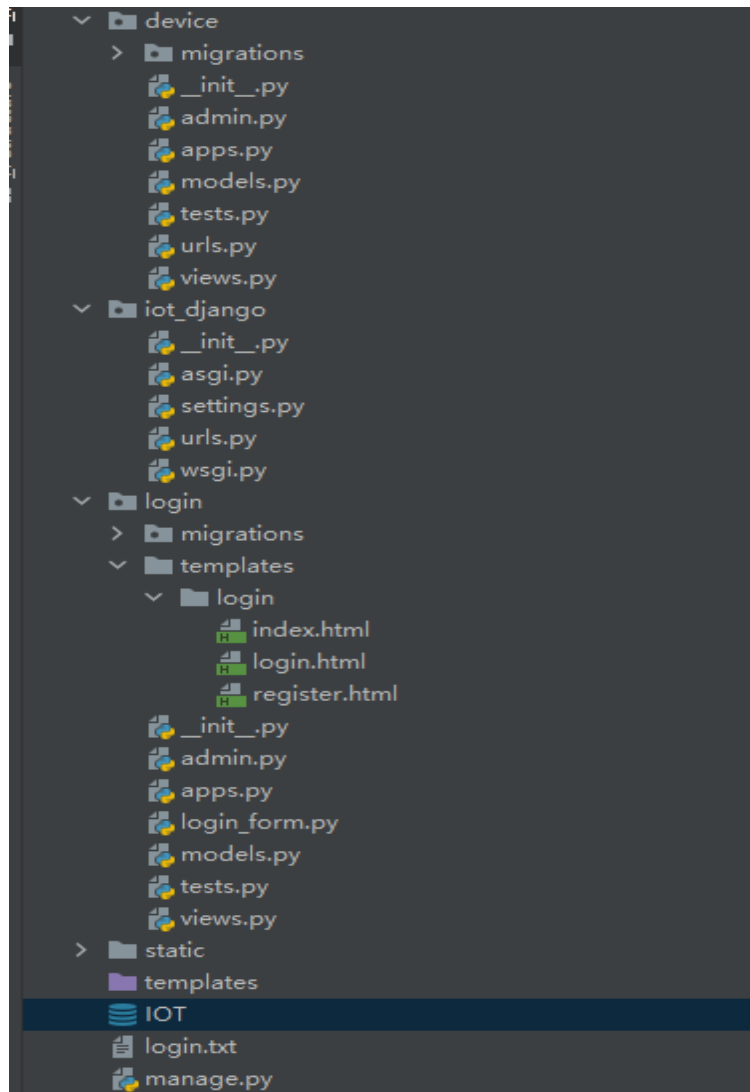
统计界面：<http://localhost:8081/line> 对应 LinPic.vues

这些均通过 router 文件下的 index.js 进行管理，而 global.js 提供全局的 js 函数支持。其中路由文件如下：

```
1. routes: [  
2.   {  
3.     path: "/test",
```

```
4.     name: "Testfun",
5.     component: Testfun
6. },
7. {
8.     path: "/statistics",
9.     name: "Statistic",
10.    component: Statistic
11. },
12. {
13.     path: "/picture",
14.     name: "Picture",
15.     component: Picture
16. },
17. {
18.     path: "/drawmap",
19.     name: "Drawmap",
20.     component: Drawmap
21. },
22. {
23.     path: "/map",
24.     name: "DispMap",
25.     component: DispMap
26. },
27. {
28.     path: "/line",
29.     name: "LinePic",
30.     component: LinePic
31. }
32. ]
```

后端 (backend)



后端部分采用 django 框架，主项目名为 iot_django，下属的 app 包括登录(login)、设备管理(device)两部分。

其中 login 部分负责处理登录信息(url 管理如下):

/login: 登录界面

/index/ 登录后中间跳转界面

/register/ 注册界面

/logout 提供退出接口。

而 device 部分负责接受前端传送过来的数据并进行处理,提供相应的 API (url 管理如下):

/device/show_all 获得所有设备的统计信息

/device/find_device 获得所有符合条件的 mqtt 发送的设备记录

/device/create_device 创建设备

/device/update_device 更新设备

/device/delete_device 删除设备

/device/get_device 获得绘制柱状图的所需数据

/device/get_gjdevice 获得绘制柱状图的所需数据

/device/get_device_value 获得绘制折线图所需数据

/device/get_chart_value 获得绘制地图数据

具体接口实现可详见对应 app 下的 views 下的各个接口实现。

五、数据字典

本部分列举初步实现设计，部分细节可能最后会改动。

5.1 user 表

```
class User(models.Model):
    """用户表"""
    name = models.CharField(max_length=128, unique=True)
    password = models.CharField(max_length=256)
    email = models.EmailField(unique=True)
    c_time = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.name

    class Meta:
        ordering = ['c_time']
        verbose_name = '用户'
        verbose_name_plural = '用户'
```

此部分包含了用户登录所需全部信息（用户名、邮箱等），其中 user_id 唯一识别用户身份。我采用 hash 加盐加密以保证用户账户的安全。

5.2 device 表

Django 数据库申明如下：

字段名	类型	含义	特殊备注
clientID	string	设备ID	
info	string	上报信息	
value	int	设备数据	[0,100)
alert	int	正常(0)/告警(1)	超过80报警
lng	double	位置精度	
lat	double	位置纬度	
timestamp	long	上报时间(ms)	

Device 记录设备发送的上报信息，记录着设备的状态(alert)，上传信息，地理位置信息等，用于设备统计信息绘制。

5.3 device 表

字段名	类型	含义	特殊备注
device_id	char(32)	设备ID	primary key
device_name	char(32)	设备名称	
type	int	设备类型	
description	char(128)	设备描述	
update_time	DateTime	最后更新时间	
last_lng	double	经度	
last_lat	double	纬度	
is_online	bool	是否在线	

Device 表唯一记录着每个 device 的 id，为便于统计，将设备类型分类，并且记录设备最后的更新时间和经纬度等，便于统计信息的绘制。

第二部分：使用手册

这部分可详见 使用方法.md

第三部分：测试报告

注册测试:

注册过程中，一定要保证格式的正确性，否则会提示相关错误。

密码必须大于 6:

密码长度必须大于等于6

注册界面

用户名:

密码:

确认密码:

邮箱地址:

验证码: 

[注册](#)

已有账号? [返回登录](#)

密码两次不一致:

两次输入的密码不同!

注册界面

用户名:

密码:

确认密码:

邮箱地址:

验证码: 

[注册](#)

已有账号? [返回登录](#)

邮箱必须唯一：

该邮箱地址已被注册，请使用别的邮箱！

注册界面

用户名：

hrh

密码：

确认密码：

邮箱地址：

1037564852@qq.com

验证码：



注册

已有账号? [返回登录](#)

若注册成功，即返回登录页面。

登录测试：

输入密码进行登录，若密码不正确会返回错误信息

密码不正确!

欢迎登录

用户名:

密码:

验证码:


登录

没有账号? [点击注册](#)

输入密码成功则进入个人主页，并跳转至 IOT 管理中心。

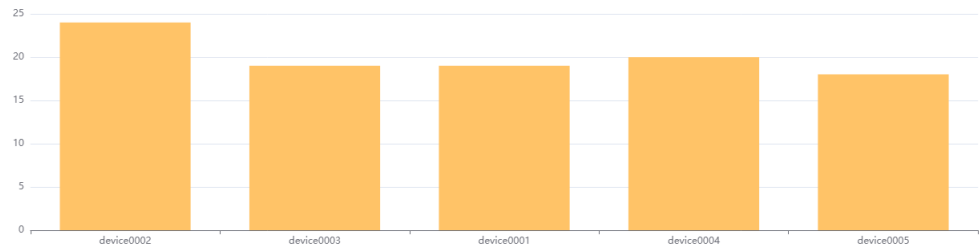
查看首页测试：



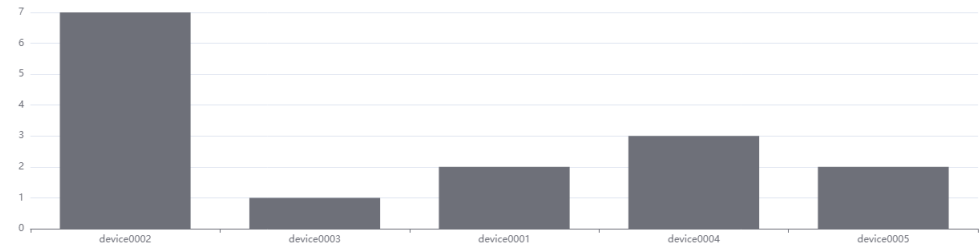
进入首页，能看到用户拥有设备的统计信息，移动到相应位置会显示数据量，若此时正在接受 MQTT 服务器发送的任务，则会动态的进行增长。(下面是运行 MQTT 服务器后)

```
5 import java.util.Vector;
6
7 public class IOTClient {
8     public static void main(String[] args) {
9         int devices = 1;
10        String mqttServer = "tcp://localhost:1883";
11        String topic = "testapp";
12        String clientPrefix = "device";
13    }
```

设备接受数据量



设备接受告警数据量



设备配置测试：

删除设备，输入对应的设备 ID 即可进行删除

创建设备

修改设备

删除设备

查询设备记录

设备ID

device0001

设备名字

ipad

删除设备

取消

设备id	设备名	设备描述
device0002	倚天屠龙剑	可斩恶龙
device0003	电视	使用电子技术传送活动的图像画面和音频信号的设备
device0001	ipad	平板电脑
device0004	智能冰箱	智能控温
device0005	摄像头	安全监控记录

删除后

创建设备

修改设备

删除设备

查询设备记录

设备ID

device0001

设备名字

ipad

删除设备

取消

设备id	设备名	设备描述
device0002	倚天屠龙剑	可斩恶龙
device0003	电视	使用电子技术传送活动的图像画面和音频信号的设备
device0004	智能冰箱	智能控温
device0005	摄像头	安全监控记录

由上图可见，device0001 被删除了。

创建设备测试：

输入设备 ID，设备名字，设备描述即可完成创建

创建设备

修改设备

删除设备

查询设备记录

设备ID

device0001

设备名字

ipad

设备描述

平板电脑

创建设备

取消

设备id	设备名	设备描述
device0002	倚天屠龙剑	可斩恶龙
device0003	电视	使用电子技术传送活动的图像画面和音频信号的设备
device0004	智能冰箱	智能控温
device0005	摄像头	安全监控记录

创建后

创建设备

修改设备

删除设备

查询设备记录

设备ID

device0001

设备名字

ipad

设备描述

平板电脑

创建设备

取消

设备id	设备名	设备描述
device0002	倚天屠龙剑	可斩恶龙
device0003	电视	使用电子技术传送活动的图像画面和音频信号的设备
device0004	智能冰箱	智能控温
device0005	摄像头	安全监控记录
device0001	ipad	平板电脑

修改设备测试：

输入修改的设备 ID，设备名字，设备描述即可完成修改(项目 ID 必填)

设备ID

device0001

设备名字

ipad2018

设备描述

设备描述

修改设备

取消

设备id	设备名	设备描述
device0002	倚天屠龙剑	可斩恶龙
device0003	电视	使用电子技术传送活动的图像画面和音频信号的设备
device0004	智能冰箱	智能控温
device0005	摄像头	安全监控记录
device0001	ipad	平板电脑

修改后

设备ID

device0001

设备名字

ipad2018

设备描述

设备描述

修改设备

取消

设备id	设备名	设备描述
device0002	倚天屠龙剑	可斩恶龙
device0003	电视	使用电子技术传送活动的图像画面和音频信号的设备
device0004	智能冰箱	智能控温
device0005	摄像头	安全监控记录
device0001	ipad2018	平板电脑

查询记录测试

此部分可以输入各种限制，如设备名，数值区间，告警信息

设备名

device0001

数值区间

20-90

是否告警

输入是否告警,格式例如: 是 或 否

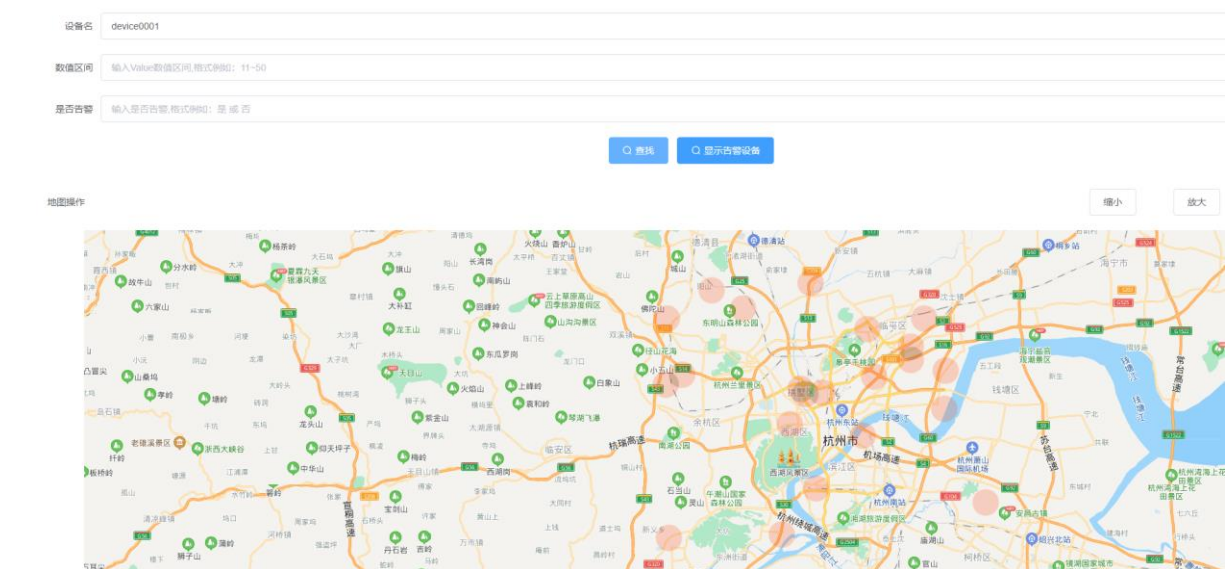
查找

取消

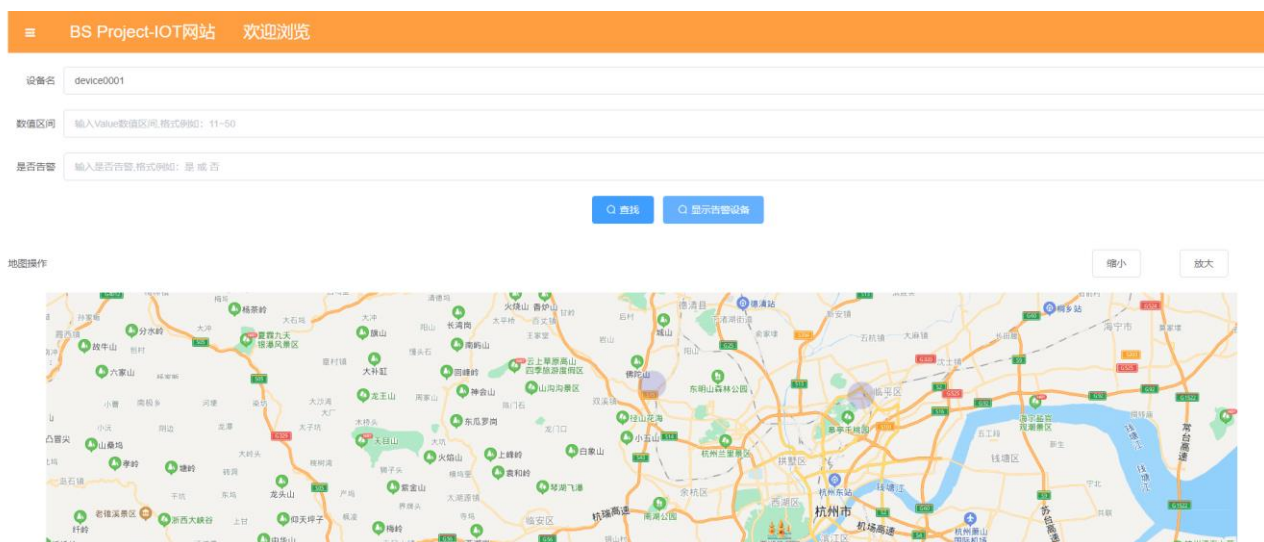
设备名	设备信息	数值	是否告警	纬度	经度
device0001	Device Data 2021/06/26 15:22:46	39	0	30.108012652397157	119.92130843400956
device0001	Device Data 2021/06/26 15:22:53	24	0	30.351881623268127	119.94432518482209
device0001	Device Data 2021/06/26 15:22:54	50	0	30.278176999092103	120.1294530749321
device0001	Device Data 2021/06/26 15:23:01	71	0	30.26106517314911	120.25457111597062
device0001	Device Data 2021/06/26 15:23:05	42	0	30.181979417800903	120.17391884326935
device0001	Device Data 2021/06/26 15:23:09	49	0	30.374096846580507	120.33790692090989
device0001	Device Data 2021/06/28 17:17:52	67	0	30.35795207023621	120.39753990173341
device0001	Device Data 2021/06/28 17:17:57	87	1	30.41978578567505	120.26328010559082
device0001	Device Data 2021/06/28 17:18:01	38	0	30.323145866394043	120.15620596408844
device0001	Device Data 2021/06/28 17:18:09	67	0	30.30093791484833	120.39523023366928

地图界面：

地图界面可以可以输入各种限制，如设备名，数值区间，告警信息，并绘制地图信息



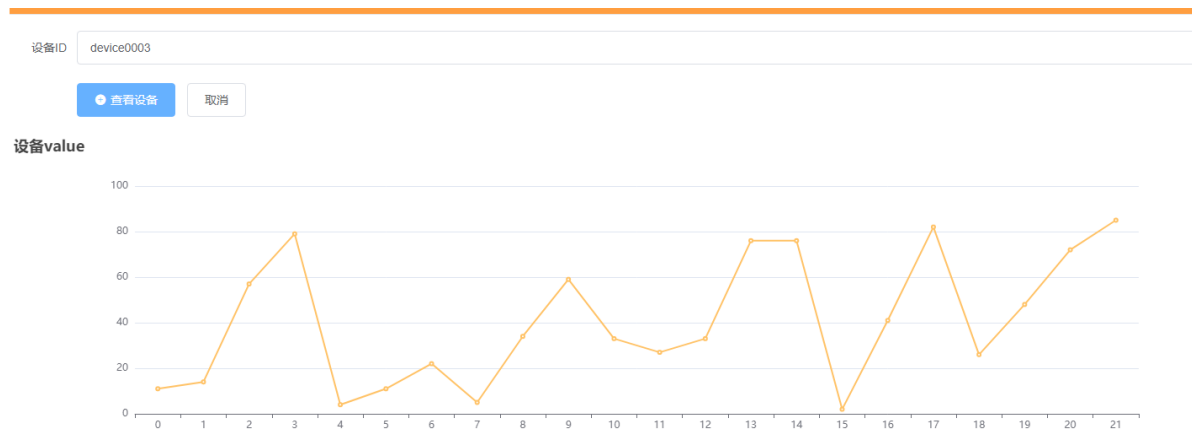
同时可以通过填写是否告警控制显示的告警和正常信息。点击显示告警设备可以标出告警设备位置。



可以发现告警的机器比正常的机器的点少了很多。

统计信息：

输入设备号，即可查找该设备的 value 统计信息。

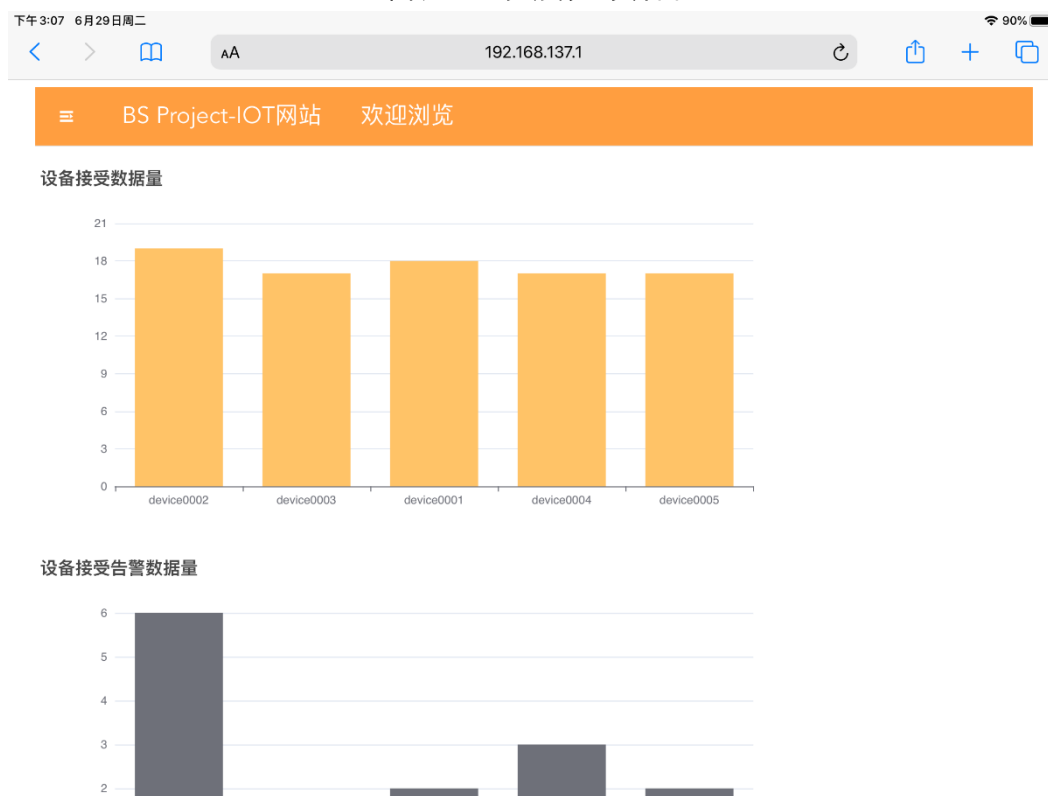


移动端的支持

目前部分组件库一般能对移动端有较好的支持。我们可以在前端 javascript 中对各种移动端或桌面端进行判断，若某种组件并不支持某浏览器，则更换对应的组件库即可。在项目中，主要使用的组件库 echarts 和 element-UI 均对移动端有很好的支持，在移动端(ipad)上，显示的界面如下：



图表 15 移动端登录界面



图表 16 首页统计

下午3:096月29日周二192.168.137.190%

BS Project-IOT网站 欢迎浏览

创建设备

修改设备

删除设备

查询设备记录

设备名

device0002

数值区间

20~80

是否告警

输入是否告警,格式例如: 是 或 否

查找

取消

设备名	设备信息	数值	是否告警	纬度
device0002	Device Data 2021/06/26 15:22:39	51	0	30.49904718399048
device0002	Device Data 2021/06/26 15:22:47	26	0	30.498814964294436
device0002	Device Data 2021/06/26 15:22:51	28	0	30.449258303642274
device0002	Device Data 2021/06/26 15:22:56	3	0	30.427501940727236
device0002	Device Data 2021/06/26 15:23:00	38	0	30.293308711051942
device0002	Device Data 2021/06/26 15:23:09	85	1	30.11632106304169
device0002	Device Data 2021/06/28 17:17:52	41	0	30.397896695137025

图表 17 查询记录

下午3:126月29日周二192.168.137.189%

BS Project-IOT网站 欢迎浏览

设备名

device0002

数值区间

输入Value数值区间,格式例如: 11~50

是否告警

输入是否告警,格式例如: 是 或 否

查找

取消

地图操作

缩小

放大

图表 18 移动端地图显示

第四部分：开发心得

作为一个开发小白，从零开始自己搭建其整个 web 应用是十分吃力的但又令人兴奋的。为了搭建满足各种业务要求的网站，我从零开始学习了 javascript、html、vue 框架、springboot 框架、django 框架以及各种组件等等，在过程中遇到了很多很多的问题，比如框架版本与教程不一致、出现各种各样的 bug，但最后都经过一系列的努力最终解决，还是非常有成就感的。

在项目中，我一开始打算在一起进行前后端开发，然而经过一段时间的学习发现，使用 vue 作为前端框架，并用其他后端技术，中间再使用跨域连接，即可令前后端分离，开发效率大大提高。当然，在前后段分离过程中，遇到了较多的问题，比如跨域的文件配置等等，进行了大量的测试。

一旦前后端分离后，前后端可以按照自己设想的形式运作，即前端发送 get 或 post 请求，而后端处理前端发送的信息并将处理后的信息传送给前端，前端调用相应的组件实现相关的任务。之后碰到的问题都方便调试，开发效率大大提高。

总得来说，这次项目乃至整个 B/S 课程的学习中，我切实地了解到了 B/S 开发中的细节，也对整个 B/S 体系软件开发的流程、工具、开发模式有了更加系统性的了解，培养了我开发小型 web 开发的能力，受益匪浅。因时间有限，最后有些想法都未实现，也有点遗憾吧。