

系统概述

物联网（IoT，Internet of things）即“万物相连的互联网”，是互联网基础上的延伸和扩展的网络，将各种信息传感设备与网络结合起来而形成的一个巨大网络，实现在任何时间、任何地点，人、机、物的互联互通。本系统要求实现一个物联网应用的网站，在后端能够接收物联网终端发送的数据；在前端实现用户注册、登录，配置设备以及展示各种信息的功能。

总体设计

系统功能

- 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。
- 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，用户登录后可以进行以下操作。
 - 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；
 - 4.提供设备上报数据的查询统计界面；
 - 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；
 - 6.首页提供统计信息（设备总量、在线总量、接收的数据量等），以图表方式展示（柱状体、折线图等等）；
- 7.样式适配手机端，能够在手机浏览器/微信等应用内置的浏览器中友好显示。

系统性能

- 系统应具有较高的稳定性、鲁棒性
- 系统需要保证较快的网站响应速度
- 系统能对各种异常状况进行检测并报错
- 系统维护/重启时间应尽可能短

输入输出要求

输入

物联网数据通过终端模拟器发送。

用户要求数据在网页上通过交互按钮和输入框输入，包括：

- 注册/登录时输入用户名和密码
- 配置设备时输入设备ID，设备名称
- 选择两个查询界面
 - 上报数据的统计界面
 - 地图界面

输出

设备配置数据输出到物联网终端模拟器；

用户名，密码，设备历史数据等信息输出到数据库。

运行环境

服务器

硬件组件	信息
CPU	2.0G，四核
内存	8.0G
硬盘	512G
网卡	浙江大学校园网

软件组件	信息
操作系统	Windows 10
数据库	MySQL 8.0
服务器	mosquitto/python

客户端

硬件组件	信息
支持的设备类型	PC网页端以及手机主流浏览器网页端
网卡	浙江大学校园网

其他软件组件	信息
操作系统	Windows10
浏览器	Firefox、Chrome等主流浏览器

系统结构

功能结构

根据系统的交互对象不同，可以将系统分为三个模块，分别对应和用户的交流，和数据库的交流，以及和物联网终端的交流。

模块1

完成前端界面，注意需要对接手机浏览器界面；完成后端与前端数据对接的通道；设计交互友好的方式展示信息。

模块2

完成数据库的建表配置文件，完成和数据库的交流，在这部分考虑可能的数据库注入攻击，进行简单的防御；同时对密码进行加密处理，在数据库中不储存明文。

模块3

完成和物联网终端的交流。

技术简介

前端技术

HTML, CSS, JavaScript是目前很成熟的前端页面构建方案。发展至今, 这些语言已经能够简单的实现文本, 语音, 视频的传输, 同时构建现代化的网页设计。相关的示例和代码资料在网络上容易找到, 可行性强。HTML (超文本标记语言) 是一种标识性的语言, 由一系列标签和属性构成。CSS (层叠样式表) 是一种表现HTML文件样式的语, 能够对页面布局进行精确控制。JavaScript是一种脚本语言, 是浏览器的一部分, 能够给HTML 增加动态效果。

react是目前流行的前端架构之一, 能够让开发者更加方便的构建现代化页面。其发展至今, 有着很多能够方便使用的库, 满足本系统要求的可视化内容。教程丰富, 社区活跃, 范例极多, 开发人员易于掌握, 具有实施可行性。

本项目计划使用react+ant design+echarts进行前端的构建。

前后端接口

分为操作接口和资源接口, 数据格式为json; 使用python+flask+react项目结构。

Flask 是一个使用 Python 编写的轻量级 Web 应用程序框架, 通过flask可以轻松在python后端和react前端之间搭建起桥梁。通过flask可以在python接收POST信息和发送回复, 通过axios可以在react组件中发送POST信息和接收回复, 从而实现前后端的交流。因为这里没有实现后端主动发消息, 前端被动接收的模式, 因此考虑用一个按钮控制前端轮询, 即定时refresh并获取最新的信息。

数据库

MySQL 是最流行的关系型数据库管理系统, 在 WEB 应用方面 MySQL 是最好的 RDBMS(Relational Database Management System: 关系数据库管理系统)应用软件之一。

与其他的大型数据库例如 Oracle、DB2、SQL Server等相比, MySQL 自有它的不足之处, 但对于一般的个人使用者和中小型企业来说, 比如对于我们需要搭建的系统来说, MySQL 提供的功能已经绰绰有余; 而且由于 MySQL是开放源码软件, 因此可以大大降低总体拥有成本。MySQL使用者多, 和python语言接口成熟, 对于本系统来说可行性较强。

后端技术

计划使用python语言构建后端, python连接MySQL使用MySQLdb库, 连接react前端的技术在前后端接口部分进行了阐述, 连接MQTT使用paho.mqtt.client库。连接MQTT服务器的代码参考博客: http://blog.csdn.net/qg_31927785/article/details/88859959

MQTT协议

MQTT (Message Queuing Telemetry Transport, 消息队列遥测传输协议), 是一种基于发布/订阅 (publish/subscribe) 模式的"轻量级"通讯协议, 该协议构建于TCP/IP协议上, 由IBM在1999年发布。MQTT最大优点在于, 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。作为一种低开销、低带宽占用的即时通讯协议, 使其在物联网、小型设备、移动应用等方面有较广泛的应用。

方法:

- (1) Connect。等待与服务器建立连接。
- (2) Disconnect。等待MQTT客户端完成所做的工作, 并与服务器断开TCP/IP会话。
- (3) Subscribe。等待完成订阅。

(4) UnSubscribe。等待服务器取消客户端的一个或多个topics订阅。

(5) Publish。MQTT客户端发送消息请求，发送完成后返回应用程序线程。

数据包结构：

一个MQTT数据包由：固定头（Fixed header）、可变头（Variable header）、消息体（payload）三部分构成。MQTT数据包结构如下：

(1) 固定头（Fixed header）。存在于所有MQTT数据包中，表示数据包类型及数据包的分组类标识。

Byte 1中bits 3-0为标识位，7-4为数据包类型，Byte 2为“剩余长度”。

(2) 可变头（Variable header）。存在于部分MQTT数据包中，数据包类型决定了可变头是否存在及其具体内容。

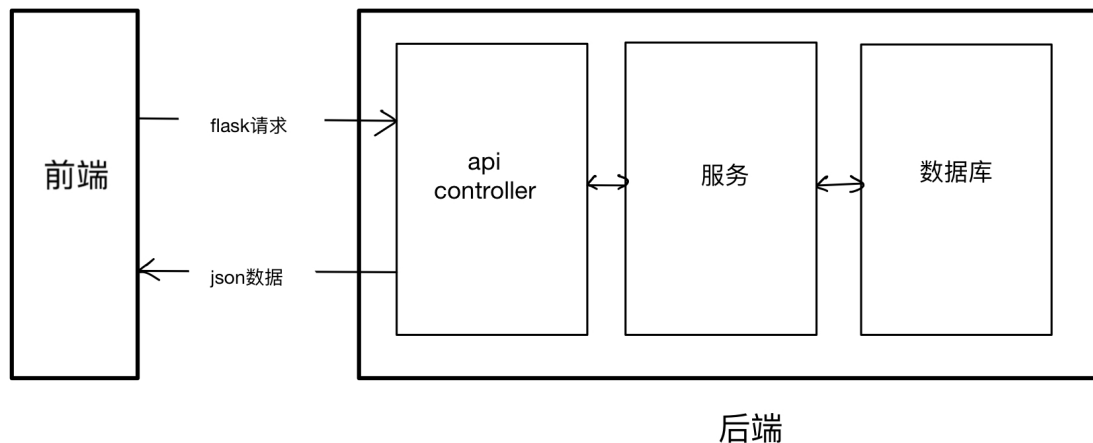
可变头的内容因数据包类型而不同，较常的应用是作为包的标识：很多类型数据包中都包括一个2字节的数据包标识字段，这些类型的包有：PUBLISH (QoS > 0)、PUBACK、PUBREC、PUBREL、PUBCOMP、SUBSCRIBE、SUBACK、UNSUBSCRIBE、UNSUBACK。

(3) 消息体（Payload）。存在于部分MQTT数据包中，表示客户端收到的具体内容。

Payload消息体位MQTT数据包的第三部分，包含CONNECT、SUBSCRIBE、SUBACK、UNSUBSCRIBE四种类型的消息：

- CONNECT，消息体内容主要是：客户端的ClientID、订阅的Topic、Message以及用户名和密码。
- SUBSCRIBE，消息体内容是一系列的要订阅的主题以及QoS。
- SUBACK，消息体内容是服务器对于SUBSCRIBE所申请的主题及QoS进行确认和回复。
- UNSUBSCRIBE，消息体内容是要订阅的主题。

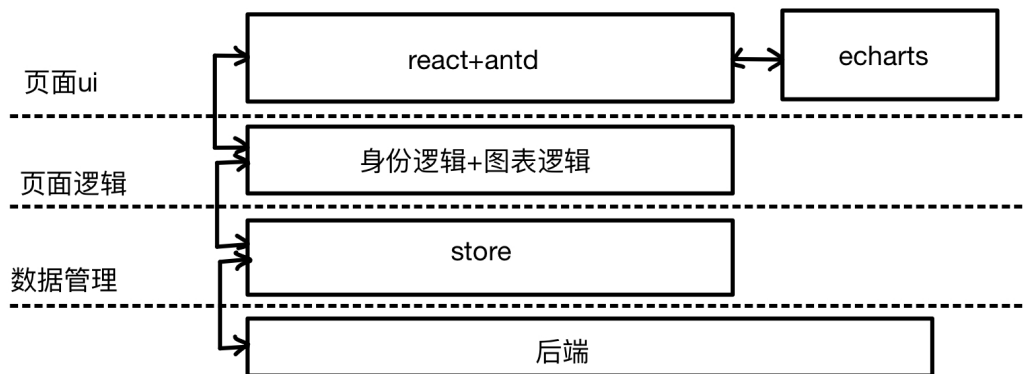
后端架构



希望采用API controller封装和前端交互的部分，使得service部分能够更好地完成和数据库交互的任务

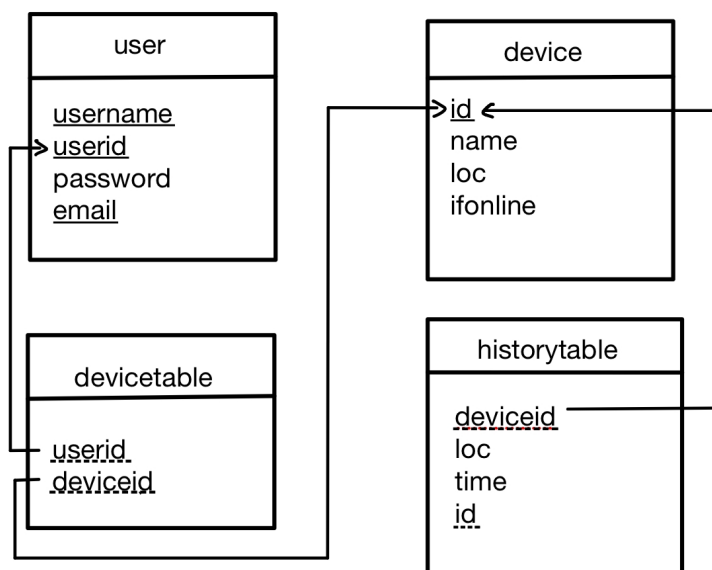
前端架构

采用react+antd构建前端，考虑到统计图的需要，采用echarts画图；由于数据较为轻量，不搭建store，采用自己写的小型store集中数据。



数据库

关系图



数据字典

user

字段名	字段类型	字段备注
username	varchar(20)	用户名
userid	int	用户id
password	varchar(20)	密码（加密后）
email	varchar(30)	邮件地址

device

字段名	字段类型	字段备注
id	int	设备id
name	varchar(20)	设备名
loc	varchar(30)	设备当前位置
ifonline	bool	设备是否在线

devicetable

字段名	字段类型	字段备注
userid	int	包含这个设备的用户id
deviceid	int	被包含的数据id

historytable

字段名	字段类型	字段备注
deviceid	int	设备id
loc	varchar(30)	设备历史位置
time	varchar(20)	该位置时间
id	int	该位置是被记录的第几个位置
value	int	此时的value值
ifoffline	bool	此时是否不在线

系统维护

出错信息

登录

错误类型	处理方法
用户名不存在	重新输入/通过邮箱找回用户名
邮箱名不存在	重新输入
用户名重复	重新输入/通过邮箱找回用户名
邮箱名重复	重新输入
密码错误	重新输入/通过邮箱找回密码

注册

错误类型	处理方法
用户名不符合规范	重新输入
密码不符合规范	重新输入
用户名重复	重新输入
邮箱名重复	重新输入/通过邮箱找回账号

查询

错误类型	处理方法
查询权限不足	返回错误信息
查询信息有误	重新输入

系统崩溃

系统崩溃时重新加载系统，数据库自动备份信息，通过数据库重新构建页面。