

基于 Spring Boot 的云端数据监控管理与可视化应用系统^①



杜英魁¹, 王 杨¹, 关 屏², 彭 跃³, 赵丽娟³, 刘 枢³

¹(沈阳大学 信息工程学院, 沈阳 110014)

²(沈阳恒源伟业环境检测服务有限公司, 沈阳 110003)

³(辽宁省环境监测实验中心, 沈阳 110161)

通讯作者: 王 杨, E-mail: 853158550@qq.com

摘 要: 为了实现对存储在云端空气质量数据的管理与实时可视化展示, 构建了一个基于 Spring Boot 的云端数据监控与可视化系统. 系统为 B/S 架构, 采用 Spring Boot 框架搭建后端微服务实例, 使系统的配置与监控变得简单. Vue.js 框架实现前端页面开发. 使用 Axios 插件封装的 Ajax 技术来实现数据交互, 实现前后端连接与逻辑交互同时, 减少服务器开销与响应. 本系统通过云端数据库可以查询到 11 种空气成分信息, 包括: PM_{1.0}、PM_{2.5}、PM₁₀、CO、CO₂、NO、NO₂、O₃、SO₂、甲醛、TVOC, 除此之外还监测包括温度、湿度、风速、坐标, 时间等其他相关属性. 实现了数据监控下载、报警管理、百度地图可视化多个功能. 程序将部署在阿里云端, 方便用户远程访问 web 项目. 其中 Spring Boot 框架与 Vue 的结合实现了系统的前后端分离, 使系统具有良好的稳定性、实时性与高效性.

关键词: Spring Boot 框架; Vue; 阿里云; 百度地图; Axios

引用格式: 杜英魁, 王杨, 关屏, 彭跃, 赵丽娟, 刘枢. 基于 Spring Boot 的云端数据监控管理与可视化应用系统. 计算机系统应用, 2020, 29(5): 123-127. <http://www.c-s-a.org.cn/1003-3254/7383.html>

Cloud Data Monitoring Management and Visual Application System Based on Spring Boot

DU Ying-Kui¹, WANG Yang¹, GUAN Ping², PENG Yue³, ZHAO Li-Juan³, LIU Shu³

¹(School of Information Engineering, Shenyang University, Shenyang 110014, China)

²(Shenyang Hengyuan Weiye Environmental Inspection Service Co. Ltd., Shenyang 110003, China)

³(Liaoning Environmental Monitoring Experimental Center, Shenyang 110161, China)

Abstract: In order to realize the management and real-time visual display of the stored air quality data in the cloud, a cloud data monitoring and visualization system based on Spring Boot was established. The system was a B/S architecture, the Spring Boot framework was used to build a back-end micro-service instance to make the configuration and monitoring become simple, the Vue.js framework implemented front-end page development. The Axios plug-in was used to implement data interaction. It reduced the server overhead and response while implementing front-end connection and logical interaction. The system could query 11 kinds of air component information through the database, including: PM_{1.0}, PM_{2.5}, PM₁₀, CO, CO₂, NO, NO₂, O₃, SO₂, formaldehyde, TVOC, in addition to monitoring temperature, humidity, wind speed, coordinates, time and other related attributes. The system realized multiple functions such as data monitoring

① 基金项目: 辽宁省重点研发计划指导计划 (2018104013); 沈阳市双百工程 (1901402); 沈阳市中青年科技创新项目 (RC180338); 辽宁省教育厅创新人才支持计划 (LR2016074); 辽宁省重点研发计划 (2019JH2/10300014)

Foundation item: Guiding Program of Key Research and Development Program of Liaoning Province (2018104013); Double Hundred Project of Shenyang Municipality (1901402); Mid- and Young-Aged Science and Technology Innovative Program of Shenyang Municipality (RC180338); Talent Supporting Program of Education Bureau, Liaoning Province (LR2016074); Key Research and Development Program of Liaoning Province (2019JH2/10300014)

收稿时间: 2019-09-25; 修改时间: 2019-10-29; 采用时间: 2019-11-05; csa 在线出版时间: 2020-05-07

download, alarm management, home web front-end visualization, Baidu map visualization, etc. The program was deployed in the Alibaba Cloud, which was convenient for users to access web projects remotely. The combination of Spring Boot framework and Vue realized separation of front and back ends, which made the system to have sound stability, real-time and high efficiency.

Key words: Spring Boot framework; Vue; Alibaba cloud; Baidu map; Axios

近年来,随着国家经济结构的转型,推动经济高质量发展,对于环境问题越来越重视。而空气污染依旧是环境保护中的一个顽疾,空气质量的优劣直接关系到人民的日常生活,对人们的健康有着直接的影响,能否获取空气质量相关最新的信息,对人们来说越来越重要^[1]。许多国内外学者进行了相关系统的开发与研究分析。史姣姣等建立了基于物联网的 PM_{2.5} 监测系统,实现了硬件设计与针对 PM_{2.5} 在线监测 Web 平台,实现了手机实时查看与统计分析^[2]。程家盼实现了基于 GIS 技术的空气质量数据可视化,采用先进的 AcrGIS 平台对空气质量监测数据进行可视化研究^[3]。周剑等设计并实现了基于多种交互方式的分布式空气质量监测系统,构建了云端数据交互、三维可视化交互等模块^[4]。然而这些空气数据监测或可视化系统要么数据过于单一,要么对于数据未能实时可视化。

同时,各种软件架构更新迭代造成了巨大的资源浪费,为了降低软件产品的耦合性,各种微服务框架应运而生^[5]。因此采用了目前国内外最流行的微服务框架 Spring Boot 来进行本系统的搭建。同时系统使用 Vue 来实现前端页面开发,Vue 支持双向数据绑定,可以使用 Axios 实现复杂的逻辑交互,实现了系统对于数据实时监控管理与实时可视化的需求^[6]。

1 云端空气质量数据

本系统主要针对传输到云端的空气数据的进行监控管理与可视化开发而搭建。数据由空气监测无线传感器设备进行在线采集,采用自定义协议长连接实现传输,云端再接收、读取、解析。将解析完的最新数据缓存到 Redis 数据库,每天固定时间再存储到 MySQL 数据库中。如发生失误导致信息未能及时上传,恢复通信后也会补传数据。Redis 数据库与 MySQL 数据库结合作为数据存储的载体,不仅保证数据的不丢失、不冗余,同时也是本系统业务逻辑的承载体,是系统的核心组成部分^[7]。系统数据库解析存储的空气数据有:细

颗粒物 (PM_{2.5})、可吸入颗粒物 (PM₁₀)、二氧化硫 (SO₂)、二氧化氮 (NO₂)、一氧化碳 (CO)、臭氧 (O₃)、温度、湿度、甲醛等。空气质量数据库表设计如表 1。

表 1 空气质量数据库表设计

名	类型	注释
DEVICE_CODE	varchar	设备标识码
COLLECT_TIME	datetime	采集时间
SIM	varchar	SIM 卡号
PM1_0	int	PM _{1.0}
PM2_5	int	PM _{2.5}
PM10	int	PM ₁₀
FORMALDEHYDE	float	甲醛 0.00-9.99
TEMPERATURE	int	温度
HUMIDITY	int	湿度
CO	int	一氧化碳
CO2	int	二氧化碳
NO	int	一氧化氮
NO2	int	二氧化氮
O3	int	臭氧留
SO2	int	二氧化硫
TVOC	int	有机气态物质
WIND_SPEED	int	风速
WIND_DIRECTION	int	风向
LONGITUDE	float	经度
LATITUDE	float	纬度
ELECTRICITY	int	太阳能电源电量

2 系统关键技术

2.1 Spring Boot 微服务

Spring Boot 微服务实际上是大量开源库的集合,自配 Tomcat, Jetty 容器,无需部署 WAR 包只需在 pom.xml 中写好相关依赖,减少了系统开发的繁琐^[8]。Spring Boot 框架下有着完善的 MVC 设计模型支持,对于项目的配置工作采用自动化配置的方式,使得在 Spring Boot 环境下进行 MVC 模式的开发变得尤为简单^[9]。Spring Boot 还提供 REST 获取进程运作期参数来进行监控,本系统主要为 GET、POST、DELETE 等分别用于获取资源、新建资源、删除资源等。本系统需要对 MySQL 数据库的增删查等操作,所以系统使用

MyBatis 编写 SQL 语句来实现, 采用 JDBC 与数据库建立连接^[10].

2.2 前端 Vue 框架

前端 Vue 框架是一款渐进式 MVVM 框架, 使用 node.js 技术, 安卓 vue-cli 脚手架构建工具, 由 vue-router 配置路由, 使用 Axios 插件封装的 Ajax 进行数据交互, 实现数据持久化, 保证良好的用户交互体验^[11].

2.3 百度地图 API

目前国内外各大互联网公司均推出了各自的地理信息服务平台, 如高德地图、百度地图、谷歌地图等^[12]. 考虑本系统开发环境、功能需求等因素, 采用百度地图作为数据可视化开发的重要平台. 百度地图 API 是百度公司免费开放的一系列地图服务接口, 开发者在获得接口使用权限后即可在自己的系统中使用地图服务^[13]. 本系统可视化开发需要百度地图 API 提供的热力图、覆盖物类等功能来完成.

2.4 云服务器

在阿里云官网上申请账号, 就能拥有自己的云主机, 并可以进行属于自己的相关部署, 例如安全规则等, 然后就可以对外服务. 通过 MobaXterm 工具连接阿里云服务器, 安装 JDK、Tomcat 等相关配置, 搭建好的 Web 项目, 将打包好的 .war 文件上传到 Tomcat 的 webapps 文件夹下, 在浏览器中连接公网 IP, 即可访问已部署好的 Web 项目^[14].

3 系统总体设计

本系统的前端实现数据监控下载、报警管理、百度地图可视化的页面. 后端主要包括 API 接口、数据库操作模块、采集到的云端数据. 用户可以直接通过 node.js 发出各种指令, 如: 查询数据、删除数据等. 后端根据请求内容对存储在数据库中的云端空气质量数据进行读写操作, 并将请求结果返回页面, 然后页面进行更新. 通过 Axios 封装的 Ajax 技术作为中间层实现前后端请求响应与数据交互, 使用 JavaScript 向后端发出请求并处理响应, 通过 XML 与 XMLHttpRequest 对象可以在不重载页面的情况下实现前后端数据交互^[15]. 中间层的存在, 实现了前后端分离. 系统总体框架如图 1.

4 系统总体实现

4.1 数据交互实现

本系统中数据交互模块是非常重要的部分,

Axios 插件中封装的 Ajax 技术使用前端技术来实现数据交互 (请求-获取-处理的过程), 在实现前后端分离后, 后端更专注于业务逻辑和数据处理^[16]. Spring Boot 框架完全支持 REST 开发规范, 通过“@RestController”注解的使用即可实现对控制器的声明和响应, 将系统的数据格式自动转化为 JSON 格式^[17]. 数据真正的交互在于这个 JSON 数据的请求和返回. JSON 的本质是 JS 对象以键值对呈现的字符串. 而“@RequestMapping (value="/device", method={RequestMethod.DELETE, RequestMethod.GET, RequestMethod.POST, RequestMethod.PUT}), produces={"application/json;charset=UTF-8"})”注释是用来限制整个类中的请求方法和定义类的 URL. 表 2 为本系统定义的部分 API 接口信息列表.

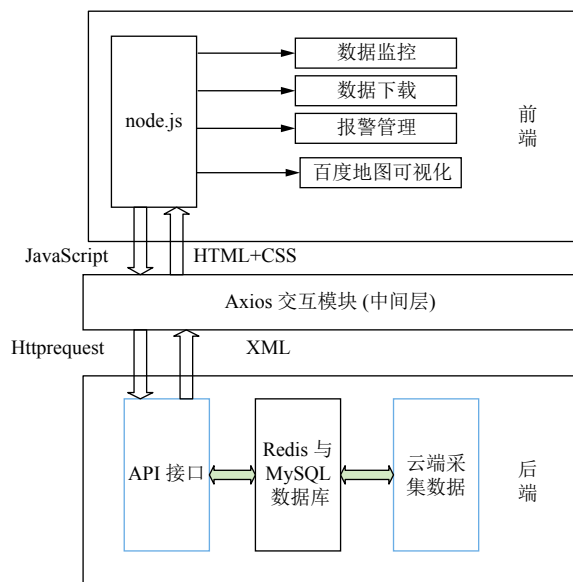


图 1 系统总体框架图

Ajax 以轮询的方式实现使用异步调用, 传递数据方式主要有两种: GET 是从后端获取要渲染的数据, 而 POST 则是向后端传递要添加或者处理过的数据. 前端拿到传递过来的 JSON 数据, 因为后端有一个对象转化为 JSON 字符串的过程, 所以 JSON 数据传递到前端后, 也有一个数据解析/解读的过程^[18]. 应用解析好的数据, 页面进行内容填充或者局部动态更新. 反过来前端向后端传递改变的数据, 也会打包成 JSON 字符串形式, 后端再进行响应与数据解析, 再完成对数据库的读写操作.

表2 部分 API 接口信息表

Controller 接口	URL	功能
DeviceController	/device/getAllLatestDevice	查询 Device 表最新上报数据
DeviceController	/device/deleteOneDevice/{id}	删除在 Device 表此 id 的全部数据
DeviceController	/device/insertOneDevice	在 Device 表增加一组数据
DeviceController	/device/queryDeviceByUserAndTime	在 Device 表根据用户名和时间查询设备信息
ReportController	/report/getAllReportByDeviceCode	根据设备号查询全部数据
DayController	/day/queryOneDay	查询最新 15 条日均值
HourController	/hour/queryOneHour	查询最新 15 条时均值

4.2 前端页面实现

出于对业务需求的分析,系统需要实现多个功能来完成对数据的实时监控管理与可视化开发.主要实现的功能为:数据监控下载、报警管理以及百度地图可视化.在浏览器中连接相应的公网 IP 地址,用户就可以看到系统功能页面的实现,从而进行使用.

(1) 数据监控实现

系统需要对云端的数据实现监控管理,输入想要查看的设备号,前端通过 GET 请求获取到全部云端空气质量数据,在浏览器的页面通过表格的样式进行渲染,表格随的数据更新不断刷新.使得用户更加清晰观察数据更新状态.实现效果如图 2 所示.

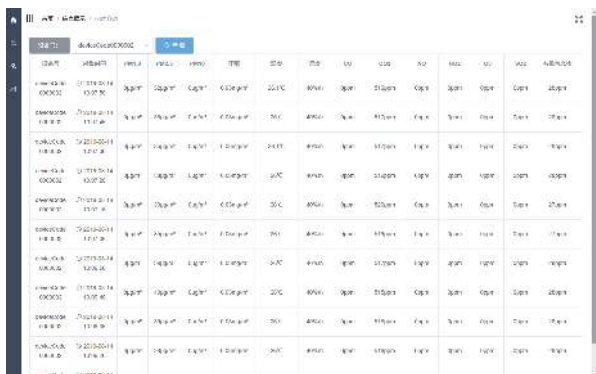


图 2 数据监控图

(2) 数据下载实现

系统拥有对云端的数据进行备份下载的功能,输入要下载数据信息(即设备号、时间段),查看点击确定后便可选择下载,页面实现效果如图 3 所示.

(3) 报警管理实现

系统需要对上传云端数据状态进行监控,通过查询设备最新数据上传的时间与上一次上传的之间间隔确定上传数据的状态,如果超过 15 分钟依然未有新的上传数据,该设备 ID 将被标红.用户可以对报警的设备进行管理,可以删除或许修改信息.页面每隔 3 分钟刷新一次,实现方法为: window.setInterval(code, 180 000). 实现效果如图 4 所示.

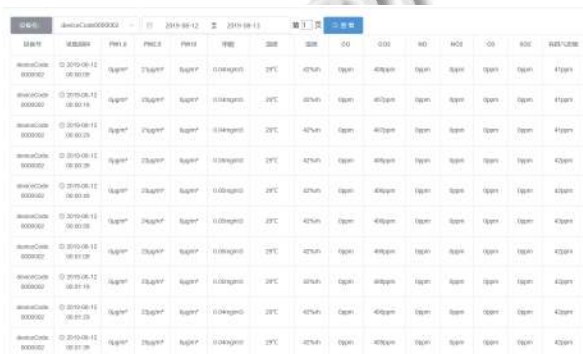


图 3 数据下载图



图 4 报警管理图

(4) 百度地图可视化实现

百度地图可视化是本系统主要的可视化展示方式,百度地图 API 提供了向地图添加自定义标注覆盖物的功能^[19].开发通过移除旧标注的“removeOverlay 方法”和添加新标注的“addOverlay 方法”来实现设备与数据值报警百度地图可视化.用标注颜色来反映设备上传数据的状态或监测的数据超标情况,页面也是每隔 3 分钟刷新一次.百度地图报警可视化页面如图 5 所示.

同时百度地图提供了热力图这一非常流行的地图可视化功能,其主要就是将要素密集程度用特殊而高

亮的方式来展示, 叠加在网络地图上的不同颜色来描述该地区中空气污染分布情况^[20], 本系统中紫色表示空气污染程度最深、橙色次之、绿色表示程度最轻, 每隔3分钟刷新一次页面. 其中PM_{2.5}热力图可视化页面实现效果如图6所示.



图5 地图报警可视化



图6 PM_{2.5}热力图可视化

5 结论与展望

本文虽然初步搭建起了对云端数据实时监控管理与可视化应用系统, 但本系统对采集的空气数据可视化处理还不够完善, 可以尝试使用R语言或Python等强大的数据处理分析语言进行更多的可视化分析. 该系统是一项网络化、高效化、现代化的数据管理与可视化展示平台, 在其构建并成功运行的过程中, 认识学习了很多相关技术知识, 为后续开发工作打下基础.

参考文献

- 1 英特尔中国. 智慧城市给你更洁净的空气. 中国公共安全, 2018, (6): 63-65.
- 2 史姣姣, 张泽胜. 基于物联网技术的PM_{2.5}监测系统. 通信

技术, 2018, 51(5): 1142-1147. [doi: 10.3969/j.issn.1002-0802.2018.05.026]

- 3 程家盼. 基于GIS技术的空气质量数据可视化. 科技与创新, 2018, (18): 118-119.
- 4 周剑, 魏广涛, 张胜东, 等. 基于多种交互方式的分布式空气质量监测系统设计与实现. 电子测量与仪器学报, 2018, 32(3): 119-126.
- 5 张雷, 王悦. 基于SpringBoot微服务架构下的MVC模型研究. 安徽电子信息职业技术学院学报, 2018, 17(4): 1-9. [doi: 10.3969/j.issn.1671-802X.2018.04.001]
- 6 焦鹏琢. 基于SpringBoot和Vue框架的电子招投标系统的设计与实现[硕士学位论文]. 南京: 南京大学, 2018.
- 7 范天恩. 基于云服务的物联网嵌入式开发平台研究与设计[硕士学位论文]. 上海: 上海师范大学, 2018.
- 8 祝翔, 董启文, 郁可人. 基于WebSocket的PK答题的设计与实现. 华东师范大学学报(自然科学版), 2018, (2): 89-100.
- 9 裴宏祥, 于晓虹. 基于微服务架构的系统设计与开发. 中国科技信息, 2019, (10): 91-92. [doi: 10.3969/j.issn.1001-8972.2019.10.036]
- 10 刘义忠, 张伟. 基于SSM框架的后台管理系统设计与实现. 软件导刊, 2019, 18(2): 68-71.
- 11 何军, 陈倩怡. Vue+Springboot+Mybatis开发消费管理系统. 电脑编程技巧与维护, 2019, (2): 87-88, 102. [doi: 10.3969/j.issn.1006-4052.2019.02.025]
- 12 张萌, 洪亮. 基于百度地图的车辆租赁管理系统设计与实现. 地理空间信息, 2018, 16(5): 16-18, 22. [doi: 10.3969/j.issn.1672-4623.2018.05.005]
- 13 钱建国, 李智程, 吴财, 等. 基于百度地图API的移动端旅游信息管理系统. 测绘与空间地理信息, 2019, 42(5): 25-28. [doi: 10.3969/j.issn.1672-5867.2019.05.008]
- 14 薛家祥, 叶兴, 吴坚. 基于阿里云平台的光伏发电智能监控系统. 自动化与仪表, 2019, 34(1): 15-19, 23.
- 15 徐光权, 汪龙, 杨晓洪. 基于Ajax技术与MVC架构的水文信息综合管理系统的设计与实现. 中国战略新兴产业, 2018, (32): 80-81, 83.
- 16 吕鹏. 阿里云上基于规则的数据质量管理系统的设计与实现[硕士学位论文]. 济南: 山东大学, 2017.
- 17 王孝宇. 基于微服务架构的敏捷开发平台安全机制的设计与实现[硕士学位论文]. 西安: 西安电子科技大学, 2018.
- 18 黄晓青. 空气质量实时数据可视化系统的设计与实现[硕士学位论文]. 武汉: 华中科技大学, 2019.
- 19 王建勋, 高东辉, 毕超群, 等. 基于百度地图API的历史地震统计软件研究. 电脑与信息技术, 2018, 26(5): 34-35. [doi: 10.3969/j.issn.1005-1228.2018.05.010]
- 20 王胜开, 徐志洁, 张健钦, 等. 逆向热力图的绘制方法. 地球信息科学学报, 2018, 20(4): 515-522. [doi: 10.12082/dqxxkx.2018.170379]