

基于前后端交互的就医无忧系统的设计与实现

黄晟祺, 许林瑞, 汪玉笏, 曾 媛, 李功权

(长江大学地球科学学院, 湖北 武汉 430100)

✉2696175145@qq.com; 835566891@qq.com; 1139729237@qq.com;
1666203035@qq.com; 195648169@qq.com



摘 要: 传统的Web开发模式中前后端代码高度耦合, 导致系统分工不明确, 开发效率低, 因此前后端分离的开发模式在当前Web应用中显得尤为重要。本文面向智慧就医问题, 引入前后端分离的渐进式框架Vue.js(一套轻量级的构建用户的渐进式框架)、ElementUI组件库(饿了么团队开发的一款Vue框架)、webpack构建技术(一个模块打包器), 后端基于Dubbo分布式架构进行开发, 前后端利用Ajax(创建交互式的网页开发技术)进行数据交互。设计实现了基于GIS技术(一种空间信息分析技术)的就医无忧服务系统, 系统具备空间查询、属性查询、缓冲区分析、周边设施查询等基础功能。实验表明, 前后端分离的开发模式使开发人员分工明确, 提高了系统性能和开发效率。

关键词: 前后端分离; Vue.js; ElementUI; Dubbo

中图分类号: TP311 **文献标识码:** A

Design and Implementation of a Worry-free Medical Service System based on Front-end and Back-end Interaction

HUANG Shengqi, XU Linrui, WANG Yujia, ZENG Ai, LI Gongquan

(School of Geosciences, Yangtze University, Wuhan 430100, China)

✉2696175145@qq.com; 835566891@qq.com; 1139729237@qq.com;
1666203035@qq.com; 195648169@qq.com

Abstract: High coupling of front-end and back-end codes in traditional Web development model leads to unclear system division of labor and low development efficiency. Therefore, development model of front-end and back-end separation is of particular importance in current Web applications. This paper aims to solve the problem of smart medical service by using the progressive framework Vue.js (a lightweight progressive framework for building users), ElementUI component library (a Vue framework developed by the Ele.me team), and webpack construction technology (a module packer). Back-end is developed based on the Dubbo distributed architecture, and Ajax (creating interactive Web development technology) is used for data interaction between front-end and back-end. The worry-free medical service system based on GIS technology (a kind of spatial information analysis technology) is implemented, which has basic functions such as spatial query, attribute query, buffer analysis, and surrounding facility query. Experiments show that the development model of front-end and back-end separation defines a clear division of labor and improves system performance and development efficiency.

Keywords: front-end and back-end separation; Vue.js; ElementUI; Dubbo

1 引言(Introduction)

传统的开发方式将前端代码和后端代码混合在一起, 如ASP、JSP技术等, 这种开发模式存在代码可读性差、开发效率低等问题^[1]。针对以上问题, 产生并发展了前后端分离的开

发架构。前端组件化、工程化, 后端数据化; 前端实现人机交互逻辑, 提供业务数据展示, 后端为前端提供业务数据支撑; 前后端通过RESTful接口进行数据交互^[2]。前后端分离的开发模式分工明确, 提升了系统的性能, 让效率最大化, 同

时使系统的开发也更加专业,降低了开发成本。

本文设计了一款基于前后端分离的就医无忧服务系统,系统采用前后端分离的开发模式,采用了Dubbo、Spring MVC、Spring Security分布式架构、权限技术栈、Echarts的报表技术栈、MySQL的持久化技术栈,以及HTML5、CSS3、jQuery、Node.js、Vue.js(一套轻量级的构建用户的渐进式框架)、ElementUI等技术栈、第三方服务七牛云(存储服务)和百度地图API。

2 相关研究基础(Related research foundation)

2.1 前后端分离技术

2.1.1 定义

前后端分离的核心思想是前端HTML页面通过Ajax调用后端的RESTful API接口,并通过JSON数据进行交互^[3]。它的开发模式是前端技术人员只专注于前端的工作,负责页面设计、框架搭建以及把后端返回来的数据渲染到页面上,不用考虑和理解后端开发的复杂代码,而后端技术人员负责提供接口给前端技术人员。前后端人员分工明确,大大提高了工作效率。前后端分离的最终目标是前端和服务端是完全独立的项目,前端项目分为浏览器和控制层,服务端项目分为服务层、DAO层等。项目开发时前后端完全分开,展示和交互是前端的作用,业务服务是后端的作用,最后生产发布时分开部署,即可完成前后端最大程度解耦^[4]。

2.1.2 前后端分离的优势

(1)前后端分开部署服务器,系统性能提升

前后端的服务器分开部署,减轻了服务器的压力,提升了页面的流畅度以及用户的体验度,最大程度地解决了随着系统应用范围的增大、用户访问量的增加,服务器访问过慢或服务器崩溃的问题。前后端分离极大地完成了前后端解耦,提升了系统性能。

(2)打造高质量人才

全栈开发者的精力有限,大部分的全栈开发者学习广而不专,要掌握数据库、服务器开发、前端开发等多种技术。相比而言,前后端分离是前后端开发人员在不同的研究领域里各自学习,对于前端开发人员,把精力放在HTML5、CSS3、jQuery、bootstrap、Vue.js、webpack、Node.js、ElementUI,以及设计模式上;对于后端开发人员,重点学习Java、Spring+Spring MVC原理及源码等,使得在有效的时间,让开发人员不断学习,积累经验,更加精通自己的专业知识,作出更好的产品,提高开发效率。因此,前后端分离的模式更有利于打造高质量的前后端人才,使开发更加专业。

(3)分工明确,让效率最大化

前后端分离可以避免在业务逻辑复杂的系统里前后端代

码混合在一起,当出现bug的时候可以快速定位是前端开发人员还是后端开发人员的问题,再由对应的人负责解决,比如客户端的问题则不需要后台的工作人员参与调试,由前端开发人员定位解决即可。当前端需要修改一个模块时,后端只需在原基础上再提供一个接口即可让项目的扩展变得更加方便。前后端并行开发,双方互不干扰,分工明确,让效率达到最大化,也有助于抢占市场。

(4)增强用户的体验感

前后端分离模式的出现,实现了高内聚低耦合,节约了开发时间,让前端能够采用更多的时间追求页面的美观程度,后端也能利用更多的时间开发出可用性高、性能好的系统。同时,在后端服务器暂停时,前端服务器可正常访问,只是没有后端提供数据,增强了用户的体验感。应用JavaScript代码对于保证应用程序快速响应请求,而不必消耗大量CPU时间和精力至关重要。同样,客户端代码表现不佳可能导致用户将应用程序视为无响应^[5]。所以,前后端分离既能让前端人员利用更多的时间提高性能,也能更好地增强用户的体验感。

(5)降低开发成本

后端开发人员开发出一套后端接口后,可以支持PC端、APP端、微信端、WebApp端等多种前端应用程序,并且可以共用一个数据库,极大地降低了开发成本。

2.2 GIS技术

前端组件化设计思想对于WebGIS有较好的借鉴意义。由于WebGIS的主要表现显示为在线地图,为了增强加载速度和用户体验感,应避免多次刷新地图页面,因此包含地图的单页面WebGIS成为发展趋势^[6]。GIS技术还包括空间大数据技术。空间大数据可视化是将计算机可视化技术、二维GIS可视化技术、三维GIS可视化技术等相结合,实现对多源、异构、海量、动态数据的可视化表达^[7]。其分析结果以热力图、格网图、散点图、密度图、OD图等方式表达大数据空间分析对象的聚合程度、变化趋势和关联关系等,直观清晰地展现数据内含的价值^[8]。

2.3 系统技术的介绍

2.3.1 Vue.js

Vue.js是一套轻量级的构建用户的渐进式框架。Vue聚焦于图层,是让开发人员利用简单方便的API实现对页面数据的绑定和组合的视图组件,它的核心是一个响应的数据绑定系统。Vue是在View层(即DOM层)与Model层(即js逻辑层)之间通过ViewModel绑定DOM Listeners与Data Bindings,以实现双向数据绑定的功能。Vue.js的优点是渐进式,开发者可以以script脚本或单独创造等不同方式的创造方法逐步引用该

组件库，灵活性极强，而其他大型框架则需自上而下全面引用^[9]。

2.3.2 ElementUI

ElementUI是饿了么团队开发的一款非常流行的Vue框架，该框架封装了各种各样的基于Vue 2.0的组件，用于网页的快速布局，方便快速开发功能强大、风格统一的页面。选择基于Vue.js的组件库，可以提高系统的兼容性。目前开源组件库支持Vue 3.0版本，但Vue 3.0版本并不普遍适用，且部分组件不兼容，因此Vue 2.0版本仍是当下最流行、种类最多的组件库。掌握Vue的基础知识，以及对Vue脚手架有一定的了解是学习ElementUI的基础。ElementUI提供布局容器、按钮、多选框、输入框、表单、图标等组件的详细代码，仅需要复制文本即可使用。它简单易学，省去了很多烦琐的Vue代码，让页面设计变得美观、快捷和简单。

3 基于前后端和GIS技术的就医无忧服务系统设计与实现(Design and implementation of a worry-free medical service system based on front-end and back-end and GIS technology)

3.1 系统的架构设计

开发者在设计系统的时候，需综合系统功能性和非功能性两方面的要求确定开发规范，这样不仅有利于开发者进行模块化开发，方便多人协同开发，保证开发进度，还方便开发人员编写利于阅读、可维护的代码，避免过多人员参与，造成代码的冲突，可以整体提升开发的效率^[10]。对Web应用来说，客户端和服务端独立的架构能最大程度地完成前后端解耦。根据功能分类，服务器包括前端服务器和后端服务器，前端服务器包括HTML、CSS、JavaScript、图片等静态资源；后端服务器主要作用于业务逻辑，结合Spring应用一些成熟的框架，可以扩展功能和均衡负载^[11]。系统架构图如图1所示。

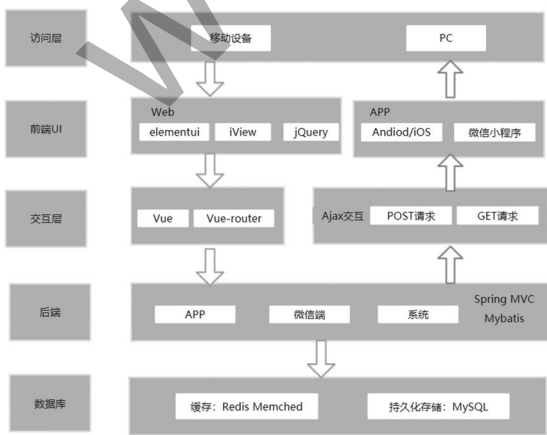


图1 系统架构图

Fig.1 System architecture diagram

3.2 前端实现

虽然学习Vue框架和其他相关js组件需要一定的时间和精力，但对开发功能复杂的Web前端系统来说，整体使用Vue框架更节约时间和精力，因此可以应用Vue框架以提高开发效率和效果^[12]。

3.2.1 统计分析模块

统计分析模块主要分为人流热度分析、高峰期人流分析，可对医院人流信息进行收集分析并进行图形化反馈。

(1)人流动态分析

对于人流动态分析，地图以点的大小、颜色表示该区域内人流量的多少，输入查询区域、查询日期和查询时间段可显示人口流动热力图，动态展示人流量的变化。通过这种方式可以直观地看出人流变化规律，页面下方采取柱状图、雷达图等进行可视化分析，展示某一区域内人流量最多的十大医院，帮助用户选择合适医院。数据可视化的作用是直观清晰地向用户传达数据所表达的信息，它是数据分析或数据科学中的一个步骤^[13]。

(2)高峰期人流分析

对于高峰期人流分析，可根据某一医院，输入查询时间段，展示在一天之内该时间段内医院各门诊科室的人流量热力分析图(地图以颜色的深浅代表该时间段内人流量的多少，颜色由绿到红表示客流量由少到多)，动态展示人流热度情况。用户可直观清晰地看出人流变化规律，避免高峰期出行，选择人流量较少的时间段排队检查。

3.2.2 查询分析模块

查询分析模块主要包括空间查询、属性查询、缓冲区分析、周边设施查询和综合查询。

(1)空间查询

输入起点位置和终点医院可进行路线查询，如图2所示。



图2 空间查询

Fig.2 Spatial query

(2)属性查询

输入科室和医院等级, 查询满足条件的所有医院。

(3)缓冲区分析

输入缓冲区半径, 选择查询中心, 查询半径内的所有医院。当鼠标悬浮在其中任意一个红色标记点时可以看到当前医院名称、地址、电话等, 点击右边详情可跳转到该医院更加详细的页面展示, 如图3所示。



图3 缓冲区分析

Fig.3 Buffer analysis

(4)周边设施查询

以当前医院为中心, 查询一定范围内的所有饭店、学校、商场、酒店, 如图4所示。



图4 周边设施查询

Fig.4 Surrounding facilities query

(5)综合查询

输入起始位置和终点医院查询路径, 设置缓冲区, 查询以医院为中心的周边设施如饭店、学校、商场、酒店等, 如图5所示。



图5 综合查询

Fig.5 Integrated query

3.2.3 用户管理模块

用户管理模块包括用户登录注册, 修改账号信息, 如密码、手机号、昵称等。

3.3 后端实现

后端的设计主要有用户管理模块, 包括对用户信息进行删除和修改操作、输入验证码进行验证、用户登录、用户注册; 查询分析模块包括实现对地图的浏览、放大缩小、添加数据、在数据库里进行查询等; 统计分析模块包括人流模拟器的设计, 模拟医院一天内不同时间段的人流数据, 传输到前端以柱状图、饼图、折线图、雷达图等方式进行展现; 数据库的设计包括医院信息表、用户表、周边信息表等共七张表。

系统实现采用Dubbo分布式架构框架, 并具体使用Spring+Spring MVC+Mybatis这三个框架进行开发。在数据的存储上, 采用MySQL、Redis两种不同的方式来解决不同的应用场景。系统中所有的工程都是基于Maven来搭建的, 对项目进行统一规范的管理。为了提高系统的性能, 系统应用七牛云云存储服务、定时组件Quartz、静态化页面FreeMarker协助开发。

3.4 前后端分离的实现

前后端利用Ajax进行交互, 后端提供多个RESTful API接口, 利用JSON格式进行数据交互。经典的Ajax技术实现了网页的局部数据刷新, 而Axios二次封装了Ajax, 它支持创建XMLHttpRequest、PromiseApi, 发出HTTP请求、拦截请求和响应、转换请求和相应数据、转换JSON数据等, Axios插件极好地封装了Ajax, 写法简单清晰, 方便查询错误, 也不容易出错^[14]。

4 结论(Conclusion)

本文以就医无忧服务系统为例介绍了前后端分离的基本结构, 前端负责页面设计、框架搭建等, 后端负责提供接口, 利用Ajax使用JSON数据的格式进行交互, 通过实践证明前后端分离在开发过程中的高效性。同时, 模块化的开发方式也让分工更明确, 也使项目有了更高的复用性和可扩展性, 提高了编程质量, 减少了不必要的开发时间^[15]。本文介绍的一些技术和工具可帮助开发人员更好地学习前后端交互技术, 并解决与实践相关的问题。传统的开发模式都不再适用, 当前的前后端分离模式也在不断地更新, 以期减少开发成本, 提高开发效率, 这种模式将会有更加蓬勃的明天。

(下转第49页)