

# 大屏数据可视化易用工具的研究与开发<sup>①</sup>



王荟奥, 蔡永香, 杨岸霖, 游小玲, 何宗宜

(长江大学 地球科学学院, 武汉 430100)

通信作者: 蔡永香, E-mail: caiyx2002cn@126.com

**摘要:** 大屏数据可视化是对数据分析结果的表达, 是数据赋能决策的重要环节. 针对大屏数据可视化软件开发周期较长、成本较高等问题, 本文基于 Vue 前端框架及 Echarts 可视化组件, 研制开发了一个大屏数据可视化易用工具 C317DataUI, 通过对可视化组件拖拽式操作进行界面布局, 使用组件的数据连接面板进行数据配置管理, 并提供了部分场景模板, 可以快速实现大屏数据可视化的应用表达, 满足行业用户进行数据可视化表达时低成本、高效率的需求.

**关键词:** Vue; Echarts; 拖拽式; 组件式; 数据可视化

引用格式: 王荟奥, 蔡永香, 杨岸霖, 游小玲, 何宗宜. 大屏数据可视化易用工具的研究与开发. 计算机系统应用, 2022, 31(2): 114-119. <http://www.c-s-a.org.cn/1003-3254/8329.html>

## Research and Development of Easy-to-use Large-screen Data Visualization Tools

WANG Hui-Ao, CAI Yong-Xiang, YANG An-Lin, YOU Xiao-Ling, HE Zong-Yi

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

**Abstract:** Large-screen data visualization is an expression of data analysis results and an important part of data-enabled decision making. Given the long development cycle and high cost of large-screen data visualization software, an easy-to-use large-screen data visualization tool C317DataUI is developed on the basis of the Vue front-end framework and the Echarts visual components. The interface layout of the visual components is conducted by drag-and-drop operation, and data configuration management is carried out via the data connection panel of the components. The tool also provides some scene templates that can quickly deliver the expression of large-screen data visualization and thereby meet the needs of industrial users for data visualization with a low cost and a high efficiency.

**Key words:** Vue; Echarts; drag-and-drop; component type; data visualization

### 1 概述

大屏数据可视化是以大屏为主要展示载体进行数据的可视化呈现. 随着大数据分析技术的流行, 行业应用中进行数据分析及大屏可视化展示以提供决策支持的需求日益迫切. 国内外许多互联网公司都有很多有用的可视化实践, 以 D3.js 为例, 其由 Mike Bostock 和斯坦福可视化组的 Jeff Heer 制作, 而以此为基础开发的可视化工具层出不穷, 如 Data.js、RAWGraphs 等<sup>[1]</sup>.

随着可视化的需求越来越大, 出现了一些商业性

可视化组件库, 如阿里巴巴旗下的 DataV 是专业级大数据可视化工具, 有图形化编辑界面, 支持多种数据源, 能灵活部署及发布等, 但免费版本的功能受限, 商业应用的费用也很高. 也有一些开源性可视化组件库, 如百度旗下的 Echarts, 有着丰富的可视化类型, 能进行深度的交互式数据探索, 支持多维数据、动态数据等, 有着丰富的视觉编码手段, 但必须通过程序编码进行定制开发.

由于系统数据类型、分析需求各不相同, 可视化

<sup>①</sup> 基金项目: 地理信息工程国家重点实验室开放基金 (SKLGE2017-M-4-6)

收稿时间: 2021-04-13; 修改时间: 2021-05-11; 采用时间: 2021-05-27; csa 在线出版时间: 2022-01-17

表达的内容及方式不同,定制开发实现成本较高、运维管理复杂.如果能建成一个数据易用可视化工具,能借助可视化组件,基于数据面板配置管理数据,快速实现数据可视化,就能够大大节省成本,提高工作效率.

本文从方便实用角度出发,着重开发出一个拖拽式大屏数据可视化工具,使用数据连接面板、可视化组件库、场景模板以及发布工具,通过对可视化组件的拖拽式操作,让没有编程基础的人也能快速的将数据可视化,做出高大上的可视化应用.

该工具命名因研究办公室名为C317,该工具主要特点是数据可视化页面,数据的英文是Data,可视化页面用UI表示,所以取名为C317DataUI.

## 2 系统架构与开发技术

### 2.1 系统整体架构

本系统采用B/S结构<sup>[2]</sup>,在该结构的基础上将系统分为3层:数据层、服务层、表现层,如图1所示.

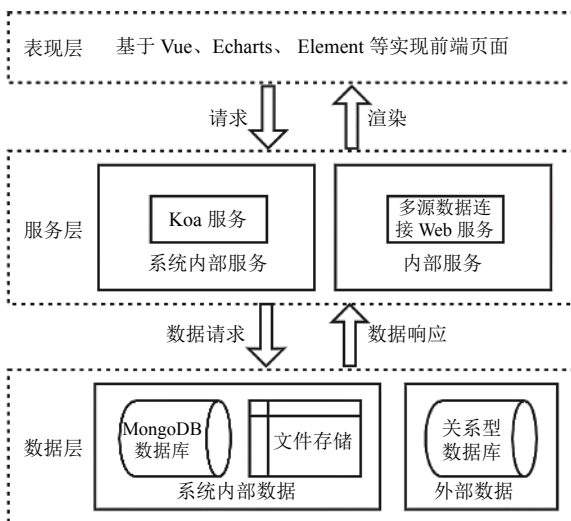


图1 系统3层架构图

#### (1) 数据层

数据层主要的功能是给服务层中的业务逻辑提供各种类型的数据支持,它的主要任务是数据存储与管理,并将数据传递给上层的业务逻辑层进行处理.

#### (2) 服务层

服务层是处于应用程序数据层和表现层之间.服务层隐藏了业务逻辑层的细节,在内部组织业务微服务,采用封装和接口的形式,提供粗粒度、面向表现层的服务.系统所有的交互都是从表现层进入.

#### (3) 表现层

表现层位于3层架构的最上层,它是直接与客户端交互的一层.它负责接收客户端的请求,解析客户端的请求,获取数据后,再返回给客户端,进而完成了一层数据请求<sup>[3]</sup>.

### 2.2 主要开发技术

考虑到要建成可视化工具C317DataUI的先进性与科学性,我们采用了成熟的前后台框架与技术来开发.前端采用Vue.js框架、Element组件库;后端采用Koa框架实现系统服务, Spring Boot框架实现多源数据连接Web服务,采用JSON格式进行数据传输,提高数据传输效率.

Vue.js是目前流行的创建界面的渐进式前端框架.它完全能够驱动使用Vue生态系统支持的单个文件组件和库开发的复杂单页应用程序<sup>[4]</sup>,而Vuex是一个专为Vue.js应用程序开发的状态管理模式.它采用集中式存储管理应用的所有组件的状态,并以相应的规则保证状态以一种可预测的方式发生变化<sup>[5]</sup>.Element是一套开源的组件库,旨在快速搭建页面,并提供了配套的设计资源,能充分满足定制化的需求.

Koa是一个新的后端框架,它没有捆绑任何中间件,而是提供了一套优雅的方法,让开发者能快速而愉快地编写服务端应用程序<sup>[6]</sup>.Spring Boot不需要进行繁琐的定义配置,而是使用一种自动配置方式,目前已成为快速应用程序开发的领导者.

## 3 关键问题及解决方案

设计构建可视化工具C317DataUI的目的是实现数据快捷可视化,减少用户开发工作量,按照简单易用的原则进行软件开发.主要实现的功能需求有:可视化组件调用、配置、布局;页面编辑与发布,模板的发布、使用、编辑;多源数据的上传、管理与连接,以及用户信息的管理等功能.其中涉及到的关键问题主要包括以下两个方面.

#### (1) 多源数据配置问题

为了让C317DataUI简单易用,设计采用傻瓜式的配置数据方式,不需编写代码,就能灵活处理不同数据源,支持MySQL、SQL Server等常见关系型数据库,让需要表达的数据能灵活与可视化组件连接.并且在交互过程中,能对图表中的数据进行联动修改,而如何连接、统一处理不同来源的数据是解决问题的关键.

我们的策略是将所有数据连接并存储到 MongoDB 数据库, 然后转换成一种公共的数据格式来可视化表达, 这样就只需要对公共格式数据的可视化表达编写代码实现即可, 可以提高代码的复用性. 此外, 对公共数据表设置监听, 当数据表被修改后, 用代码实现源数据的联动更新.

(2) 可视化组件配置问题

同样, 为了让 C317DataUI 简单易用, 我们设计对可视化组件采用拖拽式布局, 自动生成页面代码. 由于每个可视化组件都有许多的属性, 如颜色、位置、大小、数据绑定、字体等, 且组件不同, 属性的个数和类别也有所不同, 如何在尽可能降低代码量的情况下, 实现属性快速配置并自动生成页面代码是解决问题的关键.

我们采用分类处理的策略, 即按属性性质对属性进行分类, 分为通用属性和个性化属性, 并分开编码, 开发出不同属性的配置组件. 可视化组件属性配置时, 通过调用相应的配置组件, 从其接口获得配置参数来完成, 从而减少编码的工作量.

4 系统总体设计与实现

4.1 功能模块设计

考虑到用户可视化的需求及软件的简单易用, 按照系统功能类别我们将系统划分为我的作品、创意模板、数据管理和用户管理 4 个大的功能模块, 其中每个大功能模块由几个小的功能模块组成, 如图 2 所示.

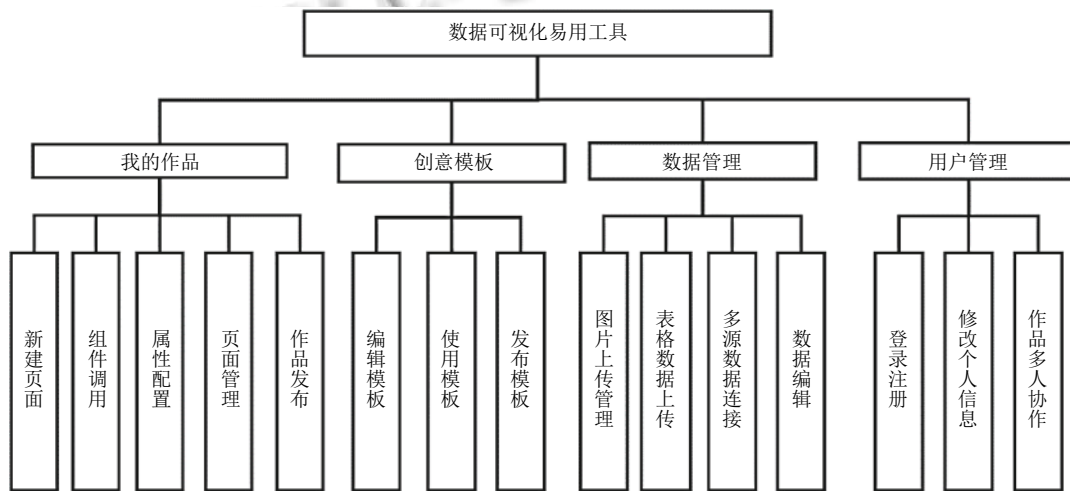


图 2 系统功能模块图

4.2 数据库设计

C317DataUI 选取了当下流行的 MongoDB 数据库来进行数据的存取与管理.

MongoDB 是一个基于分布式文件存储的数据库, 是非关系数据库当中功能最丰富, 最像关系数据库的数据库<sup>[7]</sup>. 考虑到 C317DataUI 为 Web 端工程, 对数据存取的速度要求较高, 且由于数据源多样, 有不同的数据库数据, 有 Excel 表格数据等, 因此对数据库的可扩展性要求高, MongoDB 数据库以其巨大的优势能够胜任本项目的各项需求.

(1) 具体结构设计

本数据库按照项目实际需求定义了 5 个集合分别是用户集合 (user)、图片集合 (images)、页面集合 (pages)、

数据集合 (data)、数据库连接信息集合 (jdbc).

其中用户集合 (user) 存储用户注册的相关信息. 当用户登录时, 将按用户输入的用户名和密码与用户集合中的数据进行对比验证, 实现用户登录等功能. 图片集合 (images) 主要存储用户上传的图片数据. 页面集合 (pages) 是最重要且最复杂的集合, 存储页面中所有信息, 例如在编辑器中添加的组件相关信息: 大小、位置、图表标题等, 一条记录代表一个页面. 数据集合 (data) 主要存储上传的数据信息, 这些数据与可视化组件进行数据绑定后就能进行可视化显示. 数据库连接信息集合 (jdbc) 负责记录多源数据库信息的存储, 如数据库类型、名称、IP 地址、端口号、用户名、密码等相关信息.

### 4.3 主要功能实现

#### (1) 多源数据配置

多源数据配置主要分为数据连接与存储、数据转换与绑定、数据监听与更新 3 个部分,如图 3 所示。

##### 1) 数据连接与存储

多源可视化数据主要分为 Excel 数据和不同种类的数据库数据两大类。其中 Excel 数据,通过 Excel 插件 (x-data-spreadsheet) 实现本地 Excel 文件的连接与

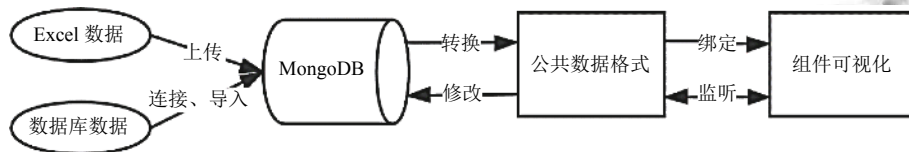


图3 多源数据配置

首先采用 forName 方法将 Driver 类的实例注册到 DriverManager 类中来加载 JDBC 驱动程序,然后创建数据库的连接获得 Connection 对象,使用 DriverManager 的 getConnection 方法传入数据库的路径、用户名和密码;再通过 executeQuery 方法来查询数据库,返回结果集 ResultSet,最后遍历结果集得到数据库数据存入 MongoDB 数据库中。

##### 2) 数据转换与绑定

从 MongoDB 数据库中读取需要可视化表达的数据,利用格式转换函数将其转化为一种公共数据格式,如格式 1 所示。将其与可视化组件的数据源绑定,即可实现数据的可视化显示。

格式 1. 公共数据格式

```
columns:
[“日期”“访问用户”“下单用户”“下单率”],
rows:
[{"日期": "1/1", 访问用户: 1 393, 下单用户: 1 093, 下单率: 0.32 },
{"日期": "1/2", 访问用户: 3 530, 下单用户: 3 230, 下单率: 0.26 }]
```

##### 3) 数据监听与更新

对公共格式数据设置数据监听,当数据发生改变时,与公共格式数据绑定的可视化图表也会发生相应的改变。需要同步更新数据源时,代码修改 MongoDB 数据库中对应的数据库。此时,实现数据监听采用的是 Vue.js 中的 watch 函数,起监听数据变化的作用。

#### (2) 可视化组件配置

可视化组件配置主要包括组件属性分类、分类后属性配置、页面发布与生成 3 个部分。

读取,然后调用数据存储接口存储在 MongoDB 数据库中,供可视化组件使用。

数据库数据使用 JDBC 进行连接,并用查询语句遍历数据库,将连接信息与数据集存储在 MongoDB 数据库中,目前已经支持的关系型数据库有 MySQL、Oracle、PostgreSQL、SQL Server 等。针对不同的数据库,需要单独创建一个以 JDBC 连接数据库的程序,下面以连接 MySQL 数据库为例来进行说明。

##### 1) 组件属性分类

Echarts 的可视化组件是通过 JSON 格式的数据项实现属性配置,不同的组件其配置项大同小异,我们把相同的属性配置分为同一类来实现,如折线图、柱状图、K 线图等,这一类图表都同样具有 X、Y 轴,因此把 X、Y 轴属性配置分为一类;所有图表都有图形样式属性“itemStyle”,包括图形颜色、图形类型、描边颜色、描边宽度、描边类型、阴影颜色、透明度等,将其分为一类;每一类进行组件式开发实现,分别称之为 XY 配置组件、图形样式配置组件等,并提供传输参数的接口。在对具有 X、Y 轴属性这类图表组件进行配置时,通过连接 XY 配置组件,获取组件的接口数据来实现属性的配置;对图形样式修改时亦如此,连接图形样式配置组件,获取组件的接口数据来实现样式配置。

##### 2) 分类后属性配置

Vue 官方文档中,组件式开发中 props 变量用来实现可视化图表组件与配置组件之间参数的传递。可视化图表组件属性配置时采用 JSON 数据格式,使用 props 变量在对应配置组件接口处接收数据,实现参数的传递。当配置组件接口数据被修改,接收数据的可视化组件属性配置相应被修改,可视化图表也会发生相应的变化,最终所有配置数据存入 MongoDB 数据库中。

如图 4 所示,可视化组件 1 有 A、B 类属性需要配置,可视化组件 2 有 B、C 类属性需要配置,属性被分成了配置组件 A、配置组件 B 和配置组件 C 三类,通过组件式开发实现。可视化组件 1 配置属性时,通过 props 分别连接配置组件 A 和配置组件 B,获取接口数

据实现属性配置;可视化组件 2 配置属性时,通过 props 分别连接配置组件 B 和配置组件 C,获取接口数据实现属性配置,这样大大提高了代码的复用性,不用每一个可视化组件配置时都单独实现一次。

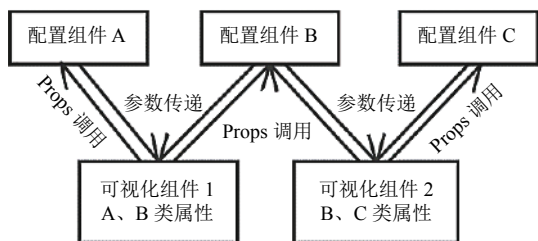


图 4 属性配置原理图

### 3) 页面生成与发布

用户按 UI 设计将布局好的可视化组件配置后,根据 id 从 MongoDB 数据库中读取页面的 JSON 数据,数据包含可视化图表组件的名称、位置、样式、数据等;利用 Vue 的动态组件特性,即<component :is="name"></component>, name 为图表组件的名称,根据

name 的值显示对应组件到页面上<sup>[8]</sup>,实现页面的生成。

使用 Vue Router 实现新页面的发布,它是 Vue 官方提供的路由管理器,适合于构建单页 Web 应用,开发人员需要做的是将组件映射到路由,然后告诉路由渲染页面的规则<sup>[9]</sup>.该工具中发布页面路径格式一般为“http://localhost/editor?id”,其中 id 为编辑页面的唯一标识,从而实现不同页面的发布。

### 4.4 系统成果展示

可视化工具 C317DataUI 开发完成后,这里对部分页面进行成果展示。

进入系统后,系统默认打开系统首页,点击新建页面跳转到编辑页面。

#### (1) 组件调用与属性配置

进入系统编辑页面后,左边是组件列表,点击所要调用的组件,组件就会出现在中间画布上;在画布上的组件可以通过鼠标点击拖拽组件改变其位置和大小;右侧为组件属性编辑,可以对组件进行样式编辑、数据绑定,如图 5 所示。

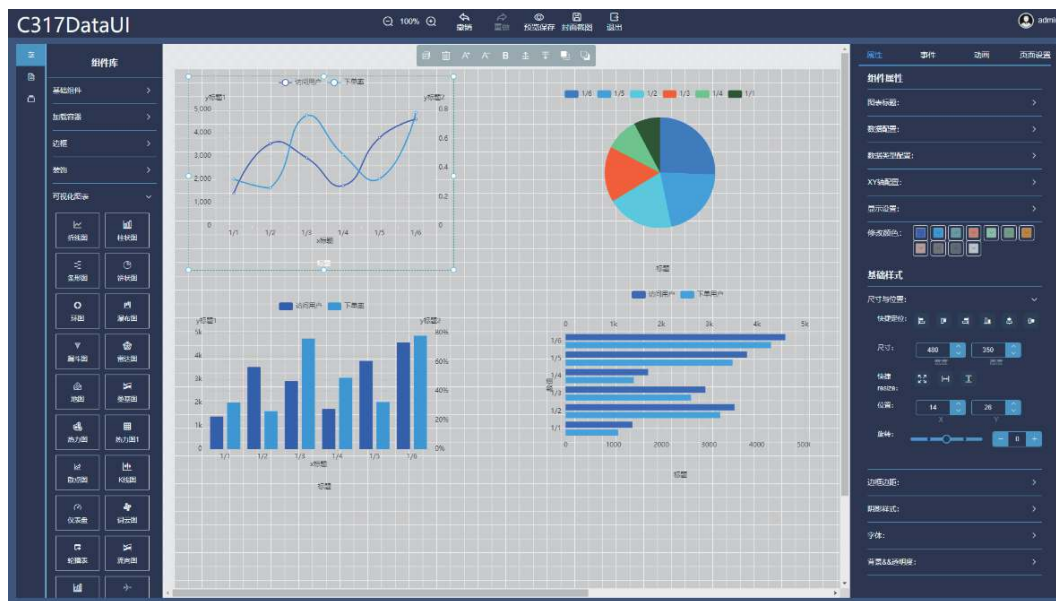


图 5 组件调用与编辑界面

#### (2) 多源数据连接页面

数据分为文件数据和数据库数据,其中文件数据包括图片数据和表格数据.表格数据上传时点击“导入 excel”按钮,图片数据上传时点击“点击上传”按钮打开本地文件夹进行上传;而数据库数据通过连接数据库进行导入,点击“连接数据库”按钮进行数据库的

配置和连接,如图 6 所示。

#### (3) 创意模板页面

C317DataUI 提供了一些创意模板,方便用户参考、使用.用户可以预览和使用模板.将鼠标移到作品上方会出现预览按钮,点击预览就可以查看当前选中的模板.点击使用模板按钮就可以使用该模板并进入

作品编辑页面。

## 5 结论与展望

本文基于 Vue 框架、Echarts 可视化库实现了数据可视化易用工具 C317DataUI, 旨在降低成本、提高效率快速实现数据可视化的应用表达。利用工具, 能通过简单拖拽操作实现可视化页面布局, 使用配置面板

快速配置可视化组件; 无需编写代码, 采用傻瓜式的数据配置方式连接多源数据实现数据快速绑定。该工具适用于有大数据可视化需求的各个领域, 如大宗商品行业交易数据可视化、设备运维管理数据可视化、人员档案管理数据可视化等, 也可以应用在大型企业的报表中, 公共疫情分布展示中等, 以及其他与我们的生活息息相关各种场景数据可视化表达中。

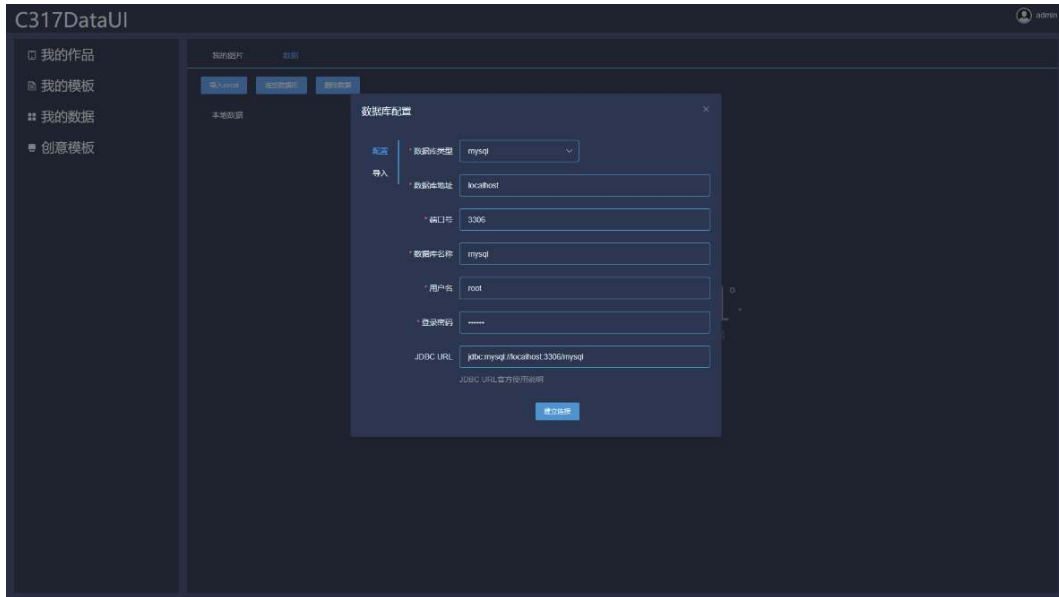


图6 多源数据连接界面

## 参考文献

- 1 陈小燕, 干丽萍, 郭文平. 大数据可视化工具比较及应用. 计算机教育, 2018, (6): 97-102. [doi: 10.3969/j.issn.1672-5913.2018.06.026]
- 2 程炜, 王睿. 基于 B/S 结构的远程医疗系统设计与应用分析. 现代仪器与医疗, 2019, 25(6): 80-83.
- 3 郭玉江. 基于 HTML5 的跨平台移动内容管理系统的设计与实现 [硕士学位论文]. 杭州: 杭州电子科技大学, 2015.
- 4 杨志远. 中国畜禽寄生虫查询系统及开放接口的构建 [硕士学位论文]. 上海: 上海师范大学, 2017.
- 5 林潇. 移动 Web 端网站无障碍人工检测系统的设计与实现 [硕士学位论文]. 杭州: 浙江大学, 2018.
- 6 牛仁腾. 基于 Vue.js 的表单可视化构建系统的设计与实现 [硕士学位论文]. 武汉: 华中科技大学, 2019.
- 7 宋瑜辉. 基于 MongoDB 存储和分析辅助决策系统中的海量日志. 科技创新与应用, 2019, (33): 5-8.
- 8 吴德林. 容器 PaaS 平台系统 WEB 前端的设计与实现 [硕士学位论文]. 北京: 北京交通大学, 2019.
- 9 邓雯婷. 基于 Vue.js 构建单页面 GIS 应用的方法研究. 科技创新与应用, 2018, (14): 5-7, 10.