

# Design and Development of Student Dormitory Social Platform WebAPP Based on Vue.js and Javalin Framework

Haoming Huang, Chengjue Liu, Ying Zheng, Luyue Wang, Tai Peng

North China University of Technology, Beijing  
Email: void593767573@gmail.com

Received: Dec. 5<sup>th</sup>, 2019; accepted: Dec. 19<sup>th</sup>, 2019; published: Dec. 26<sup>th</sup>, 2019

---

## Abstract

Based on Vue.js and Javalin framework, this paper implements a dormitory social platform mobile application through WEB APP technology with the architecture of the separated front and rear ends, including student information storage, dormitory announcement, dormitory socialization, etc. The improved communication efficiency between members also reduces the management burden of the accommodation. The platform is realized by WebAPP technology, and it can directly enter the APP through WeChat scanning QR code, and the use cost is extremely low. The implementation method of this paper also provides a new idea for the development of the information platform application on the mobile phone.

## Keywords

Mobile Software, Web APP, Vue.js, Social Platform, Dormitory Management

---

# 基于Vue.js和Javalin框架的学生宿舍社交平台 WebAPP应用程序的设计与开发

黄浩铭, 刘成珏, 郑滢, 王露越, 彭泰

北方工业大学, 北京  
Email: void593767573@gmail.com

收稿日期: 2019年12月5日; 录用日期: 2019年12月19日; 发布日期: 2019年12月26日

---

## 摘要

本文基于Vue.js和Javalin框架, 通过WEB APP技术, 以前后端分离的架构, 实现了一款宿舍社交平台手

文章引用: 黄浩铭, 刘成珏, 郑滢, 王露越, 彭泰. 基于 Vue.js 和 Javalin 框架的学生宿舍社交平台 WebAPP 应用程序的设计与开发[J]. 软件工程与应用, 2019, 8(6): 390-401.

DOI: 10.12677/sea.2019.86048

机应用程序,包括了学生信息储存、宿舍公告、宿舍间社交等功能,提升了宿舍成员间的沟通效率,同时也减轻了宿管的管理负担。本平台通过WebAPP技术实现,可以通过微信扫描二维码直接进入本APP,使用成本极低,本文的实现方法也为手机上信息平台应用的开发提供了新的思路。

## 关键词

手机软件, 前后端分离, 社交软件, 宿舍管理

Copyright © 2019 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

随着互联网+国家战略的推行,国内各行各业都利用互联网技术来优化现有的业务模式,并推行更为便利的信息平台,以提升用户的体验。学生宿舍楼中,各个宿舍房间成员之间往往缺乏有效的交流手段,互相不甚熟识,长久以往,容易积累矛盾,不利于宿舍楼中良性关系的发展,也不便于宿管进行管理。传统的信息平台应用往往都需要下载后才能使用,然而宿舍人员众多,此种信息平台不易落地,因此,我们利用最新的互联网应用开发技术,以前后端分离的架构,制作了一款宿舍社交平台应用。用户只需扫描 URL 二维码即可进入本应用,而无需下载独立的 APP,使用成本极低。因为本平台通过 WebAPP 技术实现,所以用户可以通过微信扫描二维码直接进入本 APP,而无需独立下载安装,使用成本极低。本文的实现方法为手机上信息平台应用的开发提供了新的思路,此种 WebAPP 开发模式既增加了开发效率,又提升了用户的使用体验。本文对 WebAPP 技术的研究为互联网+浪潮下的创新手机应用开发提供了新的思路。

## 2. 平台设计与实现

### 2.1. 平台架构设计

本平台采用前后端分离架构,图 1 为本平台的架构图,前端中视图与视图模型进行响应式双向绑定,开发人员通过模型获取数据,并修改视图模型。模型层通过服务器以拉取数据,前端后端之间通过 RESTful 风格的 HTTP API 接口异步发送 AJAX 请求来完成通讯,后端内通过 Javalin 来进行请求处理,以 Jetty 作为容器,MySQL 作为持久化储存数据库。前后端分别由两个团队完成开发,最后统一进行联调,开发效率较高。本架构实现了高内聚、低耦合,可扩展性和可维护性较佳。

前端负责视图层的渲染与用户交互功能并向后端通过,基于现今在国内较为热门的 MVVM 框架 Vue.js 实现。前端程序利用 JavaScript 完成交互逻辑的编写,通过向后端 API 接口发送异步 AJAX 请求来实现获取数据或变更数据,并利用获取的数据改视图模型(ViewModel),然后利用 Vue.js 提供的双向绑定机制来将视图模型的数据变更作用至视图层(View),更改 HTML DOM 树中的内容,从而实现数据的更新。为减少代码体积并提升兼容性,前端代码在被编写完成后将会利用 Babel 编译器与 WebPack 打包器进行编译打包处理,最终提供将代码打包为一个单一的 js 文件与 css 文件,并内联至 HTML 单页中,形成一个单页应用(Single Page Application)。本平台界面十分简洁,便于用户交互,利用 Vue.js 实现的程序也更有利于后期进行更新与维护。

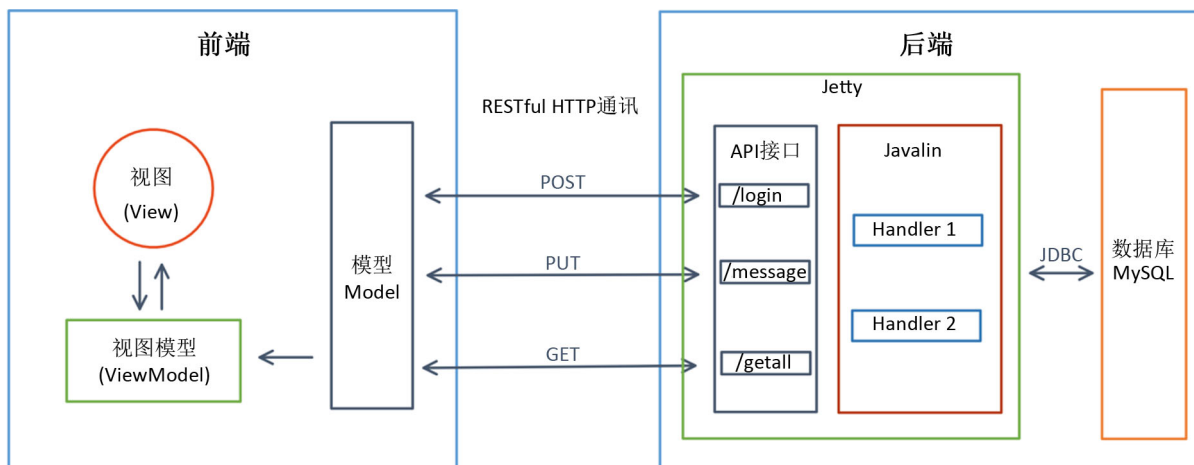


Figure 1. Server architecture diagram  
图 1. 服务器架构图

后端程序利用 Javalin 框架来编写对 HTTP 请求的处理程序，暴露出 API 接口供前端请求调用，然后利用 JDBC 接口来将数据持久化至 MySQL 数据库。Javalin 编写的服务程序将会被部署至 Jetty 容器中，Jetty 实现了对异步非阻塞 IO 的支持，使得本程序在高并发下仍能维持良好的性能。Javalin 提供了方便的请求处理方法(Handler)，极大地提升了开发效率。

## 2.2. 平台主要功能

如图 2 所示，本宿舍平台有四个主要功能：

- 1) 用户登录；
- 2) 宿舍成员信息展示；
- 3) 宿舍留言板；
- 4) 宿舍社交。

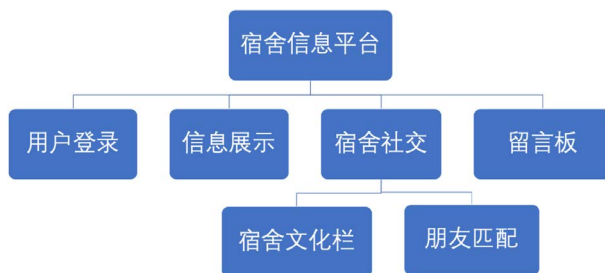


Figure 2. Main module distribution chart  
图 2. 主要模块分布图

平台采用 WebAPP 技术实现，应用以 Web 页面的形式呈现给用户，由浏览器负责交互与数据的异步拉取。所以用户只需扫描二维码即可进入本平台，无需下载独立的应用安装包，即可使用上述的所用功能。

### 2.2.1. 用户登录与鉴权

如图 3 所示，进入应用时会要求输入宿舍编号与密码，来完成对用户身份的鉴权操作，并将与权限对应的 token 发送至前端，服务器端也将 token 储存至内存，以便于下次验证，前端在向需要特定权限才

可以访问的接口发送请求时，需在 header 中附上 token，来实现特定权限操作。Token 是用于区分用户 Session 的唯一标识符，用于判断用户权限。登录与鉴权机制避免了避免其他宿舍成员访问到本宿舍成员的敏感信息。图 4 为登录界面图。

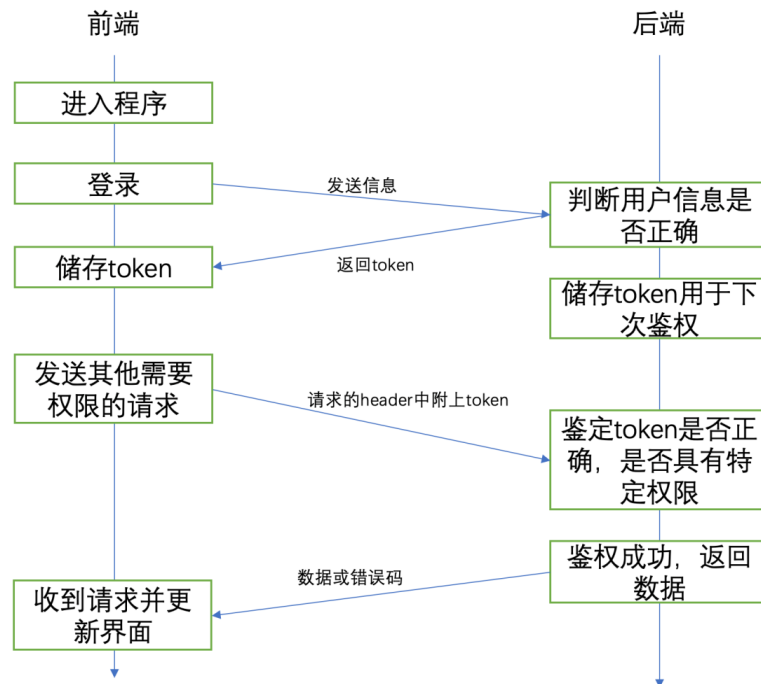


Figure 3. Login process timing diagram

图 3. 登录流程时序图

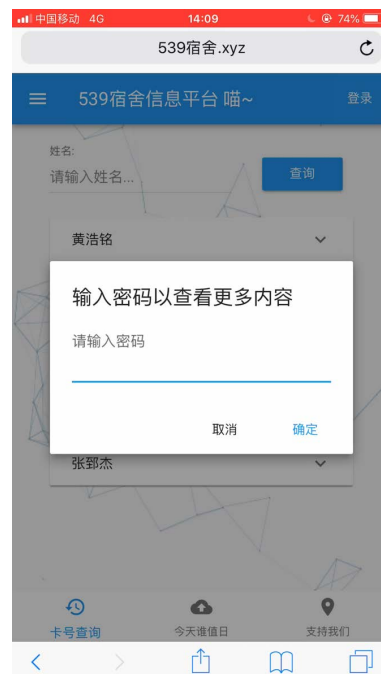


Figure 4. Login UI schematic diagram

图 4. 登录界面图

### 2.2.2. 宿舍成员信息展示

用户登录后可以查看本用户的宿舍信息，数据保存在后端，需登录后方可查看，可以方便宿舍和学生进行信息的查询与管理，如图 5 所示。



Figure 5. Dormitory member information UI diagram

图 5. 宿舍成员信息界面图

### 2.2.3. 留言板

用户可以使用本平台的留言板功能，对其他宿舍匿名或实名留言，其他宿舍的用户将能在本系统中查看留言。此功能在宿舍与宿舍之间构建了沟通的桥梁。图 6 为留言板界面图。



Figure 6. Message board UI diagram

图 6. 留言板界面图

#### 2.2.4. 宿舍间社交

如图 7 界面所示, 用户可以利用本平台进行宿舍间社交, 本宿舍的成员可以修改本宿舍的文化栏, 向其他宿舍的成员展示本宿舍的文化。其他宿舍的成员可以通过朋友匹配, 来获取其他宿舍文化栏的内容, 从而找到心仪的朋友。此功能促进了宿舍之间的交流, 为宿舍与宿舍之间搭建了沟通的桥梁, 使不同宿舍的成员之间的关系更加和睦稳定。



Figure 7. Dormitory social UI diagram

图 7. 宿舍间社交界面图

## 4. 实现细节与代码结构

### 4.1. 前端

前端程序通过 JavaScript 编写, 程序加载后, 创建 Vue 示例并挂载至 DOM 树, 进入主页。通过 vue-router 导航至各个组件, 形成单页 WebAPP 应用。使用 vuex 来进行状态管理, 储存 token 等全局状态。

### 4.2. 后端

后端程序通过 Java 编写, 程序启动后, 利用 Javalin 框架设定 URL 至 Handler 的映射, 当接受到请求后, Jetty 会自动完成请求解析并调用 Javalin 中的方法, Javalin 将请求分发至对应的 Handler 中, Handler 中完成业务逻辑的实现, 最后向 Jetty 返回响应, Jetty 将响应发送回前端完成本次请求。后端代码结构如图 8 所示。

数据库设计如图 9 所示, user 表储存用户信息, announcement 表储存宿舍文化公告栏, message\_board 表储存宿舍留言, announcement 表和 message\_board 表通过外键列链接到 user 表。

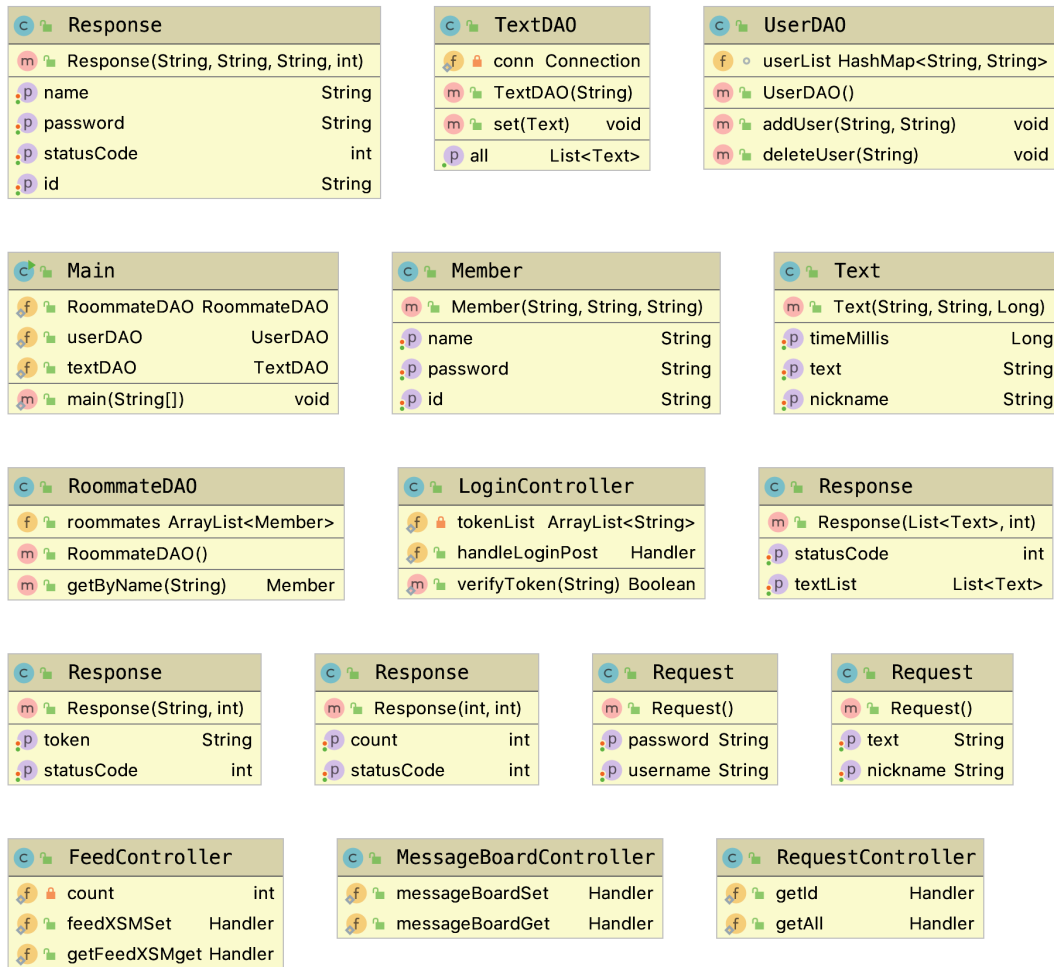


Figure 8. Backend UML diagram  
图 8. 后端程序 UML 图

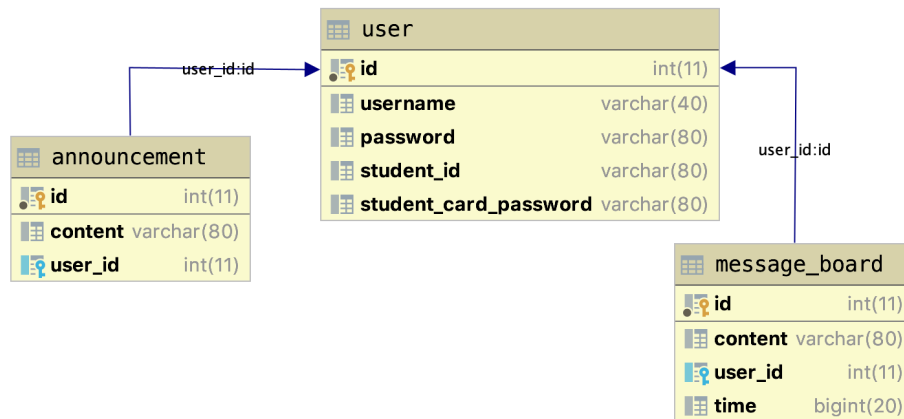


Figure 9. Database ER diagram  
图 9. 数据库 ER 图

### 4.3. 前后端通讯 API 接口设计

前后端利用 HTTP 链接实现通讯，API 接口设计符合 RESTful 风格，具体接口定义如图 10 所示



接口定义

接口名	URL	请求类型	参数名	参数位置	参数类型	参数描述	返回值名	返回值类型	返回值描述
登录	/login	POST	username password	body body	string string	登录者的用户名 登录者的密码	token	string	token, 用于登录鉴权
获取指定学生信息	/student/id	GET	id token	path header	string string	指定学生的id token, 用于登录鉴权	name id	string int	学生姓名 学生ID
获取全部成员信息	/student	GET	token	header	string	token, 用于登录鉴权	students	array	所有成员信息列表, 每个元素结构与/student相同
给宿舍留言	/messageboard	PUT	nickname token text	body header body	string string string	留言者昵称, 可空 token, 用于登录鉴权 留言文本			
获取宿舍留言板内容	/messageboard	GET					message	string	宿舍留言板内容
修改宿舍文化栏	/announcement	PATCH	content token	body header	string string	新宿舍文化栏内容 token, 用于登录鉴权			
获取宿舍文化栏	/announcement	GET	id	query	string	宿舍编号	message	string	宿舍文化栏内容

Figure 10. API interface definition

图 10. API 接口定义

## 5. 关键技术与算法

### 5.1. 前后端分离

近些年, 国内互联网产业飞速发展, Web 产品更新迭代的频率极快, 且业务逻辑也愈加复杂, 对性能的要求也越来越高, 这对软件工程开发领域也带来了新的挑战。

传统的 Web 开发模式下, 前端通过编写模板(如 JSP)与 jQuery 代码, 来实现动态界面的功能, 前端和后端共同完成模板代码的编写, 此种开发模式下, 前端不等地需要学习一些后端模板的编写技巧, 后端也需要修改一些页面上的元素, 分工不够明确, 代码耦合性极高。

由于前后端基本没有耦合, 修改前端代码不会影响后端代码, 如果前端需要添加模块, 只需在后端增加相应的接口即可, 这使得应用的扩展变得容易。除此之外, 只要前端代码遵循后端接口的调用规则, 一套后端代码可以支持多种前端应用程序, 极大的降低了开发成本[1]。

随着浏览器技术的进步与更新, Web 的开发模式也在逐渐更新、迭代、优化。ECMAScript2015 和 ECMAScript2016 标准中引入了许多新的语法功能, 使前端页面中能独立实现交互成为了可能。客户端设备性能的进步, 也为纯前端页面交互提供了物质基础。因此, “大前端”的概念应声而出, 即“纯前端实现页面及交互逻辑, 独立开发, 独立部署”, 这使得前端开发和后端开发彻底分离, 不再耦合在一起, 前端开发变得更为工程化与标准化。前后端分离不意味着双端没有任何逻辑, 而是双方以 HTTP 链接作为媒介, 双方调用 API 接口进行通讯, 来达成数据的获取与更新。这种通讯方式, 使得前端后端可以独立开发与独立测试, 最后再进行整体联调。

我们开发的信息平台, 前端采用 Vue.js 进行开发, 在前端中独立实现了社交和信息展示功能的界面与交互, 前端通过向后端发送 AJAX 异步请求以拉取数据, 并更新视图模型, 然后利用 Vue.js 的双向绑定技术将数据更新进视图, 通过前端模板技术将信息呈现给用户。前后端通过 HTTP 异步连接进行通讯, 以实现用户数据的更新。如: 用户社交模块中, 用户发布留言板功能, 前端将会向后端发送 PUT 请求, body 中附带数据实体, 后端接受请求后向数据化持久化存储数据。前端下次通过 GET 请求获取留言板信息时, 后端从数据库中读取相应的数据返回前端, 此期间, 页面不会刷新。比传统的服务器端模板技术(如 JSP)开发更为便捷, 用户体验也更为出色。

本宿舍信息平台通过运用前后端分离的模式进行开发, 大幅提升了整体系统的可维护性、可扩展性, 也达到了“降低耦合度, 提高内聚度”的目的, 开发效率大幅提升。



## 5.2. SPA 单页 Web 应用

单页 Web 应用(SPA)是一种特殊的 Web 应用。它将所有的活动局限于一个 Web 页面中, 仅在该 Web 页面初始化时加载相应的 HTML、Java Script 和 CSS [2]。一旦页面加载完成了, SPA 不会再进行页面的整体刷新, 提升了用户体验, 避免了用户在页面整体刷新时的等待。同时, 借助 Webpack 等打包技术, 通过摇树优化, 也能减轻应用的整体代码体积, 让应用加载变得更快。

本信息平台应用了单页 Web 应用技术, 避免了用户交互后页面的刷新, 同时我们借助 Webpack + Babel 的打包与编译技术, 将编写好的平台应用所用的依赖引入, 并合并为一个 JavaScript 文件, 然后将依赖项中不可达的代码通过摇树优化以缩减代码体积, 最后通过压缩进一步缩减代码体积, 从而使我们的应用首屏加载时间在 4G 网络环境下达到了 2.43 s。除此之外, 由于本信息平台为 Web 应用, 因此直接微信扫码主页 URL 所构成的二维码即可进入本应用, 十分便捷。

## 5.3. MVVM 模式与双向绑定

传统 DOM 处理需要编写繁杂的选择器, 逐级操作还原服务器需要的 JSON 数据格式, 不但操作繁琐且易出现致命错误[3]。而通过 Vue.js 这类前端框架采用 MVVM 模式, 实现了视图层(在 Web 开发中, 对应的是 HTML 的 DOM 树)与视图模型(ViewModel)的响应式双向绑定, 开发者只需从模型中获取数据(一般为向后端请求数据)之后将数据更新至视图模型中, Vue.js 会自动检测变更并将数据渲染至视图层中, 避免前端页面开发中 DOM 选择器繁杂的操作, 简化 Web 前端开发流程和降低开发难度, 提升前端开发效率, 降低开发成本和周期。

在我们的信息平台应用中, 我们将数据更新至图模型中, Vue.js 会自动检测变更并将数据渲染至界面中, 提升了开发效率。

## 5.4. RESTful HTTP 接口

传统的 Web 应用程序的设计主要是基于动作(action)的设计思想, 并且有多种框架支持这种思想的实现。2000 年 Roy Thomas Fielding 在他的博士论文中提出了 REST (REpresentational State Transfer, 表述性状态转移)术语, 它作为分布式超媒体系统设计的一种架构风格, 将资源和资源的表示分开, 为构建高性能、简单性、可移植性的 Web 程序提供了一个架构风格上的准则。在 REST 中, 整个 Web 被看作一组资源的集合, 资源是一个基于 URI (Universal Resource Identifier)的实体, 包括实体对象和业务逻辑, 对资源进行的操作由客户端指定的 URI 和 HTTP 协议动词的组合来实施的。REST 这种基于资源的设计改变了传统的基于动作的设计思想[4]。

我们的信息平台应用了基于 REST 的接口方便开发人员调用, 接口定义如图 10 所示, 减轻了前端后端沟通与调试的成本。

## 5.5. 非阻塞 I/O

本平台采用 Jetty 作为容器, Jetty 是一个基于异步 I/O 模型实现的 Web 容器。异步 I/O 模型属于非阻塞 I/O 模型, 此类模型在进行 I/O 操作时不会导致请求进程的阻塞, 使单一线程能够处理来自多个客户端的请求, 避免了多线程上下文切换时的性能开销, 从而在高并发下获得更高的性能。经测试, 基于非阻塞 I/O 模型的 Jetty 框架在测试场景下取得了更优的性能, 如图 11 所示。

## 6. 与传统开发技术的对比

本文对宿舍信息平台应用的实现, 如上文所示, 通过前后端分离分离、单页 Web 应用、非阻塞 IO 等技术, 相比于传统的 Web 开发技术(如 JSP, Servlet 等), 提升了开发效率与用户体验。对比如图 12 所示。

框架名称	框架I/O模型类型	每秒请求数
Netty	非阻塞I/O	1329
Jetty	非阻塞I/O	1461
Tomcat	阻塞I/O	1027

**Figure 11.** Comparison of non-blocking I/O model and blocking I/O model

**图 11.** 非阻塞 I/O 模型与阻塞 I/O 模型的对比

	传统Web信息平台应用	原生手机端信息平台应用	基于本文所实现的信息平台应用
跨平台	√		√
开发效率	低	中	高
耦合性	中	低	低
可扩展性	低	中	高
开发成本	中	高	低
部署成本	中	中	中
推广成本	中	高	低
性能	低	高	中

**Figure 12.** Comparison chart

**图 12.** 对比图

## 7. 部署与测试

### 7.1. 代码部署环境

代码部署在阿里云服务器上，硬件配置如图 13 所示，软件版本如图 14 所示。

配置类型	型号
CPU	Intel ® Xeon ® Platinum 8269CY
内存	ECC服务器内存 1G
硬盘	SSD 40G
网络带宽	1Mbps

**Figure 13.** Hardware configuration

**图 13.** 硬件配置

软件/框架名称	版本
Java Runtime	build 1.8.0_211-b12
Nginx	1.14.2
MySQL	8.0.15 for linux-glibc2.12 on x86_64
Javalin	2.1.1
Vue.js	2.5.17

**Figure 14.** Software version

**图 14.** 软件版本

前后端代码共同部署到同一个服务器上，通过 Nginx 来实现对静态资源请求和动态资源请求的分别

转发，当请求的是动态资源(一般为异步请求，如获取用户信息)时，Nginx 将会将该请求通过反向代理自动转发至监听本地端口的后端服务程序中；当请求的是静态资源时，Nginx 将会自动寻找对应路径的静态资源并直接返回给前端，如图 15 所示。

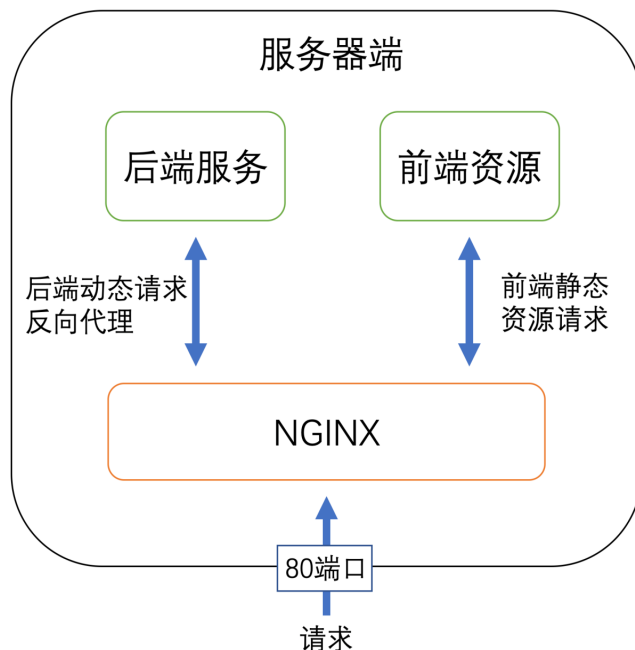


Figure 15. Nginx 转发  
图 15. Nginx 转发

### 7.2. 前端静态资源文件的部署

前端代码通过 ESLint 来实现语法和类型检查，然后利用 Babel 来将其编译为 ES5 兼容的代码以提升对老旧浏览器的兼容性，最后用 Webpack 来进行打包，通过摇树优化和标识符替换降低代码体积，如图 16 所示。



Figure 16. Frontend toolchain  
图 16. 前端工具链

打包好的代码通过 Sftp 上传至服务器，放入 Nginx 指定的 wwwroot 文件夹。

### 7.3. 后端服务的部署

后端代码通过 Maven 来完成编译与打包，最终生成独立的 jar 包，部署至服务器。

### 7.4. 性能测试

通过压力测试，得出图 17 中的数据：

测试期间，前端与后端的所有功能及表现的都符合设计预期，前端界面交互流畅，后端响应迅速。

性能指标	值
事务数量	3827 hits
可用率	100%
花费时间	19.55 secs
传输数据量	36.49 MB
平均响应时间	2.3 secs
最长响应时间	4.3 secs
最短响应时间	0.77 secs
TPS	195.75 trans/sec

Figure 17. Performance

图 17. 性能指标

## 8. 总结

本文基于 Vue.js 前端框架和 Javalin 后端框架, 实现了一套高效的宿舍社交平台, 为宿舍间交友, 宿舍文化交流, 宿舍管理提供了便利。我们已将本平台在北方工业大学馨斋 5 层中进行了小规模推广, 获得了同学、教师和宿管的一致好评。通过本平台, 宿舍的学生们都找到了自己的好朋友, 也降低了宿舍之间的矛盾, 使宿舍楼中形成了和谐的氛围。

由于本平台采用 WebAPP 技术开发, 在浏览器通过 JavaScript 中完成交互和界面的运行, 因此宿舍用户在使用本应用时, 无需下载程序, 只需微信扫描二维码即可进入本 APP, 使用成本极低, 有益于本平台的推广。

本平台采用的前后端分离的开发模式, 改变了传统 Web 的开发模式, 让前端和后端能够分别地被开发与部署, 降低了系统整体的耦合性, 提升了模块的内聚度。前端实现了用户社交和信息展示等功能的交互, 后端实现对数据的持久化储存, 前后端通过 HTTP 接口进行通讯。这种开发架构具有一定的泛用性, 可以被推广至各类 Web 应用的开发工作中。

平台的开发过程中, 还使用了 SPA 单页应用技术与 AJAX 异步加载技术, 使用户在宿舍信息平台的界面交互时, 不会刷新整个页面以加载数据, 提升了宿舍用户的体验。同时, 我们采用了非阻塞异步 IO 技术, 提升了并发性能。通过运用上述技术, 平台的界面的流畅性和效率得以提升。本文对上述技术的研究, 也为 Web 应用开发与优化提供了新的思路。

## 基金项目

北京市大学生科学研究与创业行动计划资助项目。

## 参考文献

- [1] 孟祥双. 前后端分离式 WEB 应用开发研究[J]. 电子元器件与信息技术, 2019, 3(6): 40-43.
- [2] 喻好, 陈锋, 程仁贵. 对单页 Web 应用开发探究及应用[J]. 福建电脑, 2017, 33(2): 136-137.
- [3] 麦冬, 陈涛, 梁宗湾. 轻量级响应式框架 Vue.js 应用分析[J]. 信息与电脑(理论版), 2017(7): 58-59.
- [4] 潘冰. 面向资源的 RESTful Web 应用研究[J]. 微计算机应用, 2010, 31(7): 38-43.