

基于 B/S 架构的博客系统设计与实现

王悦 黄媛媛

四川大学锦城学院计算机与软件学院 四川 成都 611731

【摘要】本文简述的是基于 Vue 框架设计开发的一个知识分享类博客 App。该 App 基于“易于分享，美好体验”的理念，汲取了当下优质博客 App 的优点，是一款操作简单且页面简洁美观的知识分享类博客 App。本项目的前端部分使用了 Vant 组件，Element 组件和 Vue 框架，后端框架采用了 Express 框架，采用 MySQL 数据库管理系统来存储数据，运行环境是 Node.js，是一款高效便捷的轻量级博客 App。该 App 具有查看文章，发表文章，登录注册，收藏评论点赞等功能。

【关键词】博客 App；Vue；MySQL；Express；Vant

1 引言

随着互联网的发展，人们获取信息的手段从书本变成了电子产品，在获取信息的同时还可以在互联网上表达自己的观点。但这些内容杂乱无章，分布于各个论坛与网站，人们要靠搜索引擎检索出若干信息，再从这些信息中寻找自己的需要的部分，博客的出现很好的解决了这一问题，人们不仅可以在博客上按关键字，类别去查找需要的信息，还可以在博客上发表自己的观点，方便交流也便于以后查看。人们可以分享自己的专业知识，也可以分享生活中的美好瞬间。本文将对主要选用技术，用户需求分析以及功能模块几大部分归纳介绍该 App 的实现流程。

2 主要技术选型介绍

2.1 Vue.js

Vue.js 是一个轻量，可组件化的 MVVM 库，Vue 使用简单直接，可以更好地组织与简化 Web 开发。

2.2 Element 桌面端组件库

Element-UI 是配合 Vue 使用的一套 UI 库，Element 简化了常用组件的封装，提升了后台系统的可用性。

2.3 Node.js

Node.js 是 Javascript 语言的一个运行环境。

3 需求分析

该设计是一个知识分享类博客 App，目的是给网友们提供一个知识分享的平台。该设计参考了当下热门同类型 App 的功能设计，其目标受众是所有年龄群体，重点放在使用电子产品最多的十五到三十五的年轻群体

基于对当下热门交友分享类软件的分析，本项目需要具备以下功能：

用户功能：登录，注册，退出账号，修改个人信息，对不同类型访问者设置不同权限。

文章功能：发表文章，查看文章，文章分类。

用户互动功能：评论文章，点赞文章，收藏文章，关注其他用户。

3 项目设计

3.1 模块设计

本 App 分为消息管理模块，评论管理模块，收藏管理模块，分类管理模块，用户管理模块，内容管理模块六个功能进行模块开发设计，目的是给用户提供一个简

洁，实用，便捷的线上分享类博客 App。用户未登录时可以查看文章内容，但不能进行评论，点赞，关注，发表文章等功能。当未登录用户点击这些功能时 App 会跳转到登陆页面提醒用户登录，模块设计如图 1 所示：

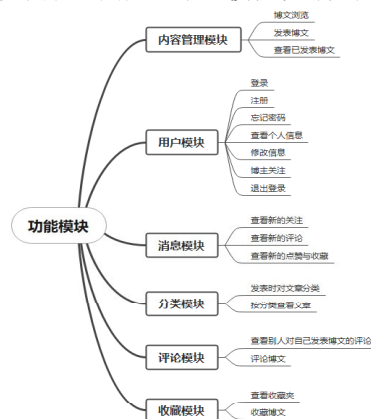


图 1 功能模块设计

3.2 发表文章流程设计

用户登录账号后即可发表文章，在发表文章界面输入标题与文章内容，选择类别，上传图片（可以不上传图片）后点击发表按钮后，前端通过 axios 把数据传入后端，同时跳转到发表成功界面。其中上传图片大小不能超过 2MB，必须是 JPG 格式。

3.3 数据库设计

根据本项目的需求设计了六张表：

(1) 用户表：用户 ID，用户名，用户的登录密码，用户的头像，发表博文的 ID，性别，年龄。

(2) 文章表：博文 ID，用户名，博文标题，博文图片，点赞数量，博文内容，博文类型，用户 ID，博文发表日期。

(3) 点赞表：点赞用户名，点赞用户名，被点赞用户名，点赞日期，点赞用户头像，点赞博文 ID，点赞博文标题。

(4) 收藏表：收藏博文 ID，用户 ID，用户名，博文标题，收藏时间，博文内容。

(5) 关注表：用户名，用户 ID，被关注用户 ID，被关注用户名，用户头像，关注日期，被关注用户名，被关注用户头像。

(6) 评论表：博文 ID，用户名，用户的头像，用户评论的时间，用户评论的内容，被评论用户的 ID，用户

ID, 文章内容。

3.4 后端设计

后端负责把数据存入数据库, 当需要访问或修改数据时, 访问数据库中数据或修改。后端通过 `createConnection()` 连接数据库, 来操作 database 层面, 返回一个 `Connection` 对象。前端可通过访问数据库中的用户表来查询用户是否已经注册。

本项目中使用的路由跳转有 `$router.replace()` 和 `$router.push()`, 两者的区别是 `push()` 可在本地添加一条历史记录, `replace()` 不会添加, 直接跳转。获取数据是通过 `axios` 方式, 统一使用 `post` 请求。页面获取数据后存在状态管理库中的 `store` 中, 页面刷新或触发某些事件时, 通过预先设定的函数把更改后的数据放入 `store` 中, 以保证页面的动态更新。

4 项目实施

4.1 用户管理模块

该模块实现了用户的登录与注册功能, 没有注册过的用户可以在注册页面注册, 注册成功后再登录, 有账号的用户直接在用户页面输入账号与密码登录。

当用户点击 App 中的消息界面, 发表文章界面, 个人中心界面时, 界面都会提醒用户登录, 未登录用户只有浏览博文的权限。

用户登录时需要输入账号与密码, 用户点击登录后, 服务器会在状态管理库中去查找相应的用户 ID 是否存在, 若存在, 匹配密码, 若密码不匹配, 或者用户不存在, 都会提示用户输入正确的用户名与密码。

未注册用户可以点击注册链接进行注册。

当用户输入账号后, 服务器会在状态处理器中查找该账号是否已经存在, 若存在, 显示提示。上传头像图片的格式只允许 JPG 格式, 且不能超过 2MB。注册完成后会跳转到注册成功页面, 并显示当前用户的用户 ID。

用户可在个人中心修改个人信息, 修改个人信息界面的性别选择中使用了 `Element` 组件中的 `<el-radio>` (单选框)。

4.2 分类模块

该模块把文章分成几个类别, 便于用户查找。

用户在上传博文时, 需要选择分类, 一篇博文只能有一个分类, 博文分为“体育”, “知识”, “娱乐”, 在数据库中保存为 `message_type` 字段。

用户可在首页, 通过不同分类来浏览自己感兴趣的文章。

分类同时也增加了不同文章之间的联系性, 当用户阅读完一篇文章后, 系统默认会优先推送相同分类的文章。

用户在发表文章时, 必须选择分类, 文章按分类存入数据 `options`, 通过 `v-for` 把全部分类渲染到 `<el-option>` 选项列表中。文章内容可包含图片, 图片类型只限于 JPG 类型, 且不能超过 2MB。

4.3 内容管理模块

该模块用于处理文章内容, 用户发表的文章会被存储在数据库中, 用户浏览博文时, 前端会向后端数据库请求文章标题, 内容, 分类等渲染到前端界面中实现的。

用户在首页搜索的内容会被保存到后端, 当再次点

击时, 会显示搜索历史, 每一次搜索都会判断是否与状态库中保存的数据相同, 如果这个搜索内容不存在, 就存入历史记录, 如果搜索框的内容存在于历史记录, 就删除之前的旧纪录, 将新搜索值重新放入数组首部。

网站首页的所有文章会按照时间顺序来排序, 通过 `blogger_choice()` 把最近日期的排在前面。首页默认会显示前几行字以便用户自行选择是否为自己感兴趣的内容, 点击想要阅读的内容可进入该文章页面, 查看全部内容。

文章可以一键分享, 分享的渠道有: 微信, 微博, 复制链接, 二维码。

4.4 消息模块

该模块用于处理用户消息, 把用户未读的消息显示在消息界面。

在消息界面中, 已登录用户可以查看目前收到的消息, 包括评论, 点赞, 收藏, 关注。

该页面通过 `v-for` 来循环输出动态变化的消息, 遍历数据库中评论数据渲染到 `ul` 的 `li` 标签中。

为了页面美观度, 引入三个 `Vant` 组件, 类名为“like”与“follow”分别作为收藏, 点赞与关注的图标, 还有一个小红点样式图标作为区分评论已读还是未读的标志, 当用户点击评论进入相应博文后, 把此评论对应的索引加入到状态库 `index` 队列中, 通过 `v-if` 判断此索引已存在, 隐藏改评论对应的图标, 表示已读。

前端把请求的评论数量保存为 `comment_info`, 把关注数量保存为 `concern_info`, 在获取收藏数据后调用 `then()` 获取点赞数据, 两个数据用数组的方法合并保存为 `state` 中的 `info`, 合并是因为之后收藏点赞的提示是放到同一个页面中, 显示数据长度时需要合并。当页面渲染的时候, 判断是否有数据, 没有数据代表是第一次渲染, 直接把获取到的数据放入状态库, 有数据代表页面已经渲染过, 不是第一次渲染, 此时把新数据放入状态库, 并改变相应数据的长度, 例如评论数据的长度为 `comment_info_num`。保存数据的长度是为了之后显示有多少条消息。

点击全部已读按钮后调用 `read()`, 保存当前评论数量, 调用 `len()`, 把 `len` 即评论数量置为 0, 同时把点赞, 收藏数量置为 0, 此时在 `v-for` 中无评论数据, 点赞收藏与关注也无数据, 未读消息提示数量是收藏与点赞数量加上关注与评论的数量, 此时也为 0。

4.5 评论模块

该模块是存储用户评论文章的相关数据。

评论模块加大了用户之间的互动性, 用户可在博文内容下的评论区中发表评论, 收到的来自其他用户的评论在消息界面中查看。

4.6 收藏与点赞模块

该模块是存储用户收藏与点赞文章的相关数据。

用户在收藏博文后, 可在用户中心查看自己已经收藏的博文。后端通过数据库中博文收藏表中博文 ID 与用户 ID 是否对应来判断用户是否收藏了博文。用户点击收藏后, 前端会把博文 ID, 用户 ID, 用户名, 用户头像, 收藏日期, 博文题目, 与博文内容存入收藏表中。

收藏与点赞的区别是,用户看不到收藏数,但是在点赞前可以看到点赞数,用户收藏文章后可以在自己的收藏夹中再次查看,但是无法查看自己点赞过的文章。点赞可以作为文章热度的一个重要参考值,系统向用户推送时,也会优先推送点赞数高的文章。

5 结束语

本系统采用 B/S 架构,使用模块划分的方法,完成了用户登陆与注册,文章浏览与发表,点赞与关注等功能,经过测试验证,本系统具有界面简洁,功能清晰,操作简单的特点,能够为用户提供良好的在线阅读与交友的体验。本系统目前存在的问题是博文推荐和热榜是按照热度值来计算的,热度值只有点赞与收藏相加,没有考虑浏览量与博主粉丝数量,不够准确。后续会把点赞,收藏,浏览量与博主粉丝数量通过计算来得到更精确的热度值,给用户更好的阅读体验。

【参考文献】

- [1] 罗路腾, 王贵鑫. 基于 Springboot 的博客网站的设计与实现 [J]. 科学技术创新, 2019(33):64-66.
- [2] 鞠宏军, 林涛. 基于 Spring Boot 的博客系统的设计与实现 [J]. 电脑知识与技术 2019, 15(33).