基于 Vue 和 Express 的博客系统设计和实现

曾海阳 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】随着互联网的迅速发展，博客成为当今主流的网络交流方式。通过博客系统，用户能够记录美好的生活，分享丰富多彩活动。通过参考当前主流的几种博客系统，并对博客系统用户的需求进行分析，设计并实现了一款操作简单、用户体验良好的博客系统。本系统前端采用主流的 Vue 框架，后端采用 Express 框架，数据库采用 Mysql。本系统中用户通过登录进入系统，在首页可以查看或筛选博客文章，在发表页面发表自己的文章，在消息页面查看关注的动态，在“我的“页面对个人信息和博文进行管理。

【关键词】Vue ； Express； Mysql

1 引言

在当今互联网时代，生活和网络已经融为一体，网络之间的交流也成为人与人之间的主要交流方式，博客这种媒体工具也在互联网中占据了重要的地位，为了顺应互联网时代的潮流，提升用户体验，本系统设计并实现了登录 / 注册、用户信息管理、评论点赞、文章管理、博文推荐等功能。本文将对该系统进行技术的选型、需求分析、系统模块设计几大部分进行分析和归纳总结，同时介绍本系统实现的思路和流程。

2 主要技术介绍

2.1 Vue.js 框架

Vue 框架 [1] 具有轻巧、高性能、组件化等多个优点，是一套构建用户界面的渐进式框架。它采用 MVVM 模式、数据响应式系统，实现了数据驱动视图，视图驱动页面。通过这种双向数据绑定的响应式设计，减少了开发者对 Dom 的操作，从而提高了系统的性能。

2.2Elemnt UI 组件库

Element-Ui 是一个组件库，是基于 Vue2.0 设计并开发的，在目前现有的 Vue 组件库中，Element UI 使用人数多、社区活跃、组件库完整丰富，为前端开发大大减少了代码量，从而提高开发效率。

2.3 Express 后端框架

Express[2] 具有灵活、简洁的两大特性，它包含了 Node 原生所有的 API，同时进行了更好的封装，从而使开发人员可以更加容易的使用 Node.js。Express 提供各种模块，可以快速地搭建一个具有完整功能的网站后台。

2.4Mysql 数据库

MySQL 是一种关系型数据库管理系统。主要的优点：1. 运行速度快。 2. 使用成本低。MySQL 是开源的，且提供免费版本，对大多数用户来说大大降低了使用成本。3. 使用容易。其复杂程度较低，易于使用。

2.5Axios 网络模块

Axios 是基于 Promise[3] 和用于浏览器和 Node.js 的 HTTP 客户端，同时 Axios 具有很多优点，支持 Promise，同时能够转换请求数据和响应数据，Axios 会自动把请求后的数据转换为 JSON 格式，这样减少了前端处理数据的代码量。

3 需求分析

3.1 登录注册

新用户没有账号可以通过注册功能注册唯一的账号，再进行登录。老用户可以直接通过账号登录系统。

3.2 个人信息管理

用户登录成功后，可以查看自己的个人信息，同时可以修改自己的个人信息和用户密码，点击收藏夹显示用户收藏的博文，可以查看用户的关注用户，还能查看自己的已经发表的博文基本信息，也可以退出登录。

3.3 博文查看

对于登录成功的用户可以查看谁评论和点赞了我的文章。

3.4 发表博文

只针对于登录成功的用户，用户可以在该界面编写博文然后发表。

3.5 博文点赞评论收藏

用户登录成功后可以对博文进行点赞、评论、收藏。

3.6 首页

可以展示不同类别的博文，点击分类菜单进行对博文进行筛选。对于登录成功的用户是可以看见关注用户的博文发表动态。

4 系统模块

4.1 系统功能模块

功能模块分为，用户登录模块、个人中心、内容管理 3 大模块。具体的模块功能看结构图：

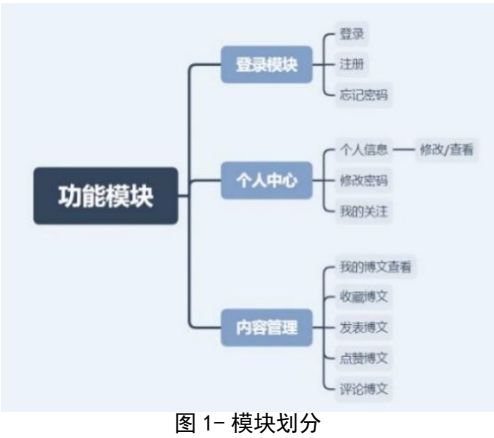

图 1- 模块划分

4.2 前端系统设计

4.2.1 登录

1. 登录流程如下：

页面渲染成功之前通过Axios模块给后端发送请求，后端查询用户表把数据返回给前端。

前端接收用户表数据后，把数据存入 vuex。

用户信息填写完成后点击登录，系统会把输入的数据和 vuex 的数据进行匹配，匹配成功后进入到个人中心，同时登录成功就会替换 vuex 里登录成功用户的信息。

忘记密码，可以点击“忘记密码“进入密码找回界面，通过输入注册时的密保问题答案，系统对该问题答案进行验证，验证正确后进入下一步填写新的密码，最后确认更改。

前端给后端发送网络请求，后端拿到更改后的数据然后把最新的数据存入数据库，成功后返回成功信息，前端再提示返回登录界面。

登录功能关键代码如下：

前端主要代码：

```
beforeMount(){
axios({
method:”post”,
url:”http://127.0.0.1:8080/user”,
}).then((res)=>{
this.$store.commit(“addUser”,res.data);
return
})
}
addUser(state,data) {
state.user = data;
},
```

后端代码接口实现：

```
app.post( ‘/user’, function (req, res) {
conno.query( ‘SELECT *FROM `user`’, function (err, result) {
if (err) {
console.log(“用户表查询错误”, err);
res.send(“false”);
return
}
res.send(result)
})
})
```

4.2.2 注册

1. 注册流程如下：

用户通过页面提示填写昵称、账号、密码、性别、年龄基本信息。

每个表单都绑定 blur 事件对输入信息的判断是否正确，完成填写信息无误后点击注册，给后端发送网络请求。

后端接收前端表单数据，再把新的用户数据存入数据库，成功后给前端返回成功后的信息。

前端收到成功信息后再提示用户点击返回登录按钮，通过路由跳转回到登录界面。

4.2.3 个人中心：

1. 页面渲染流程如下：

在登录成功时会把登录成功的用户基本信息保存在 vuex 中，个人中心页面再对部分数据进行展示。

2. 个人信息修改流程：

点击修改个人信息按钮，路由携带登录成功用户信息参数跳转到修改界面。

再填写表单基本数据，密码的修改会进行验证，首先输入原始密码，系统会把输入的原密码和数据库的原密码进行比对，正确后才能进行修改密码操作。

最后确认修改，系统把新数据传到后端。后端再把数据更新到数据库中，成功后页面提示用户修改成功，2s 自动跳回登录界面。

3. 收藏夹功能流程：

在收藏夹界面渲染成功之前会执行 boforeMount 钩子函数，向后端发送数据，后端接收数据对 collect 表进行查询得到收藏博文 id 的数组，再通过遍历数组查询 message 表，再把数据返回前端。

前端得到数据后，通过 v-for 指令对数据进行渲染。

4. 我的博文功能流程：

点击我的博文，在我的博文页面渲染之前通过 boforeMount 钩子函数发送网络请求。

后端通过用户 id 查询数据库中 message 表，再把数据传给前端。

前端接收数据后把数据放入 vuex 中，再用 v-for 指令渲染页面。

我的博文可以进行删除，点击删除，会删除 vuex 对应的数据，同时更新页面，给后端发送网络请求，通过博文 id 进行删除该博文。

4.2.4 消息

1. 消息功能流程如下：

消息页面在页面渲染成功之前，判断 vuex 是否存在登录成功用户的信息，同时结合 v-if 显示成功或不成功的页面。

如果登录成功，通过 beforeMount 给后端发送网络请求，获取评论、点赞、关注的动态信息，同时显示消息条数。

4.2.5 首页

1. 关注内容显示流程：

进入关注子页面时，系统判断 vuex 内是否含有登录成功用户的数据，没有页面就提示用户还未登录。

登录成功的用户，进入关注子页面，把登录成功用户的 id 传给后端。

后端通过用户 id 对数据库 concern 表进行查询关注用户 id，再通过遍历关注用户 id 对 message 表进行查询相关数据，再把数据返回给前端。

前端接收数据通过 v-for 指令把数据渲染到界面。

2. 分类功能流程如下：

系统在发表博文的时候对博文进行了分类，同时首页分类功能可对博文进行分别展示。

点击某一个类别按钮，系统通过获取到此类别名，同时通过网络请求把类别名传给后端，后端通过对 message 进行筛选，再把数据返回给前端。

前端拿到此类别的博文数据，然后用 v-for 指令把

数据渲染到界面。

图 2- 首页

4.2.6 博文页面

1. 博文渲染流程：

在此页面渲染成功之前，首先从路由获取到要展示的博文 id，然后发送网络请求给后端。

后端通过博文 id 再查询该博文的信息，把博文相关的数据返回到前端。

前端接收数据后把数据存入 vuex 中，页面再从 vuex 中取对应的数据进行渲染。

2. 评论功能流程：

点击评论弹出输入框，提示输入评论内容，输入完成后点击确认。

前端获取到评论的日期、评论内容、评论用户数据，再把这些数据传到后端，后端得到数据分别插入到 comment 表和 comment_info 表，后端操作成功后返回成功信息。

前端接收到成功信息再把评论的基本信息放入 vuex 中。

3. 收藏功能流程：

在博客页面渲染之前，通过路由传来的博文 id 去和用户的收藏博文表进行匹配，判断是否已经收藏，再通过 v-if 显示对应的图标。

点击收藏图标对博文进行取消收藏和收藏。

前端判断是取消收藏还是收藏，再把博文数据、用户数据传给后端。

后端再对数据库的收藏表进行添加或删除。

5 总结

本系统采用 Vue.js 前端框架和 Express 后端框架、数据库 Mysql 进行开发，目前已经实现了博客系统大部分功能。在交互上，界面简洁美观、用户体验较好。对于本系统一些细节还值得改进，比如字体设计以及少量操作流程的设计，在后续的过程中，会根据实际情况进行改进优化，打造一个简洁、美观、操作简单、可用性高的博客系统。

【参考文献】

[1] 范博涛 • Vue.js 前端开发实战：2020 年 4 月第一版：02-03

[2] 钟小平 • Node.js 开发实战教程：慕课版，2020 年 8 月第一版：190-192