基于 Vue+MongoDB 的博客 App 设计与实现

伍兴月 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生活分享类博客 App。本 App 基于“记录生活，记录美好”的理念，参考学习了当下热门博客 App 的优点，丢弃他们的缺点，构建了一款操作简单，页面简洁的博客 App。该项目采用 Vue 来构建前端页面，使用 mongoose 搭建服务器，采用 mongoDB 存储数据，是一款高效便捷的轻量级博客 App。在本 App 中，用户可以体验点击查看文章详情，在发表页写文章，修改个人信息，点赞评论，添加好友等功能。

【关键词】：App 开发；VueNode.js；MongoDB

1 引言

在互联网高速发展的潮流下，“手机族”，“低头族”等人群源源不断的涌现，这使得人与人之间的交流越来越少，更多的人患有社交障碍。为了顺应时代潮流，需要给用户提供一个能够记录生活，抒发情感，交到更多朋友的平台。本 App 提供了登录/注册，文章管理，用户管理，评论点赞，消息管理等功能。本文将对选用主要技术，可行性分析，用户需求分析以及功能模块几大部分归纳介绍本 App 的实现流程。

2 主要技术选型介绍

2.1 Vue.js

Vue.js 作为前端主流框架，与其他框架的不同之处在于，Vue 采取自下而上增量开发模式,它的核心库是视图层，使得它易于与其他库或者项目结合。同时，Vue 经由简单的 API 便可以实现对响应数据的绑定以及组件复用。所以我们也可以说 Vue 是一套响应式系统，它采用 MVVM 构架，把 view 的状态和行为抽象化，将视图层和业务逻辑分离,以此来实现数据数据的双向绑定。

2.2 Element UI 组件库介绍

Element UI 是基于 Vue 框架开发的 UI 组件库，它不依赖 Vue，但却是当前和 Vue 配合在前端页面开发比较好的一套 UI 库，其提供了多种类的组件资源，可以帮助开发者高效快速的构建页面。Element UI 使用简单，只需根据文档步骤进行安装，导入需要样式即可。

2.3 MongoDB 数据库介绍

MongoDB 作为基于分布式存储文件存储的数据库，数据以 JSON 等多种形式存储[1]。MongoDB 的使用场景灵活，它结合了关系数据库和非关系数据库的特点，数据结构以 key-value 键值对的形式进行存储。Key 作为文档的唯一标识，可以用来快速检索数据，通过键可以快速匹配其所对应的值，可以用来快速检索数据，value 可以存储较多类型的数据，这种键值对的存储形式通常我们叫做 BSON,其具有查询快，简单的特点。

2.4Mongoose 模块介绍

Mongoose 是一个文档对象模型库，通过与 Node 相连接，可以对 MongoDB 进行数据驱动，其对 node 原生的 mongoDB 模块进行优化封装，提供更简便的接口[2]，开发者使用封装好的接口进行项目开发，能够缩减开发中的代码量，提高开发效率。

2.5 Ajax 介绍

Ajax 作为一种跨平台跨浏览器的技术，它得到全部主流浏览器的支持和认同，能够更好，更快的创建 Web 应用。通过前端与服务器的小部分数据交换，Ajax 可以使网页实现异步更新，这样，就能够让网页不需要重新刷新页面而实现对 App 的部分数据进行更新，提高用户的满意度。

3 可行性分析

3.1 开发模式可行性

本 App 以前后端分离架构模式为核心。前端选用 Elment UI 组件和 Vue 框架、共同构建页面，后端使用 mongoose 搭建服务器，数据由 MongoDB 数据库进行存储。在这个模式下，大大提升了开发的效率，让后端只负责数据库操作的业务逻辑和为前端提供数据接口，而前端页面只展示交互逻辑，这使得前端有较大的页面改动，后端也不会有什么影响，对于同一个数据接口，我们页面可以定制多个版本，同样在这个模式下的后端报错也不会直接反映到前端，让错误接收较为友好。这样的开发模式，使得前后端职能明晰，前端通过 axios 请求后端 API，而后端负责具体业务逻辑，这样使得前端的逻辑变重，前端可以做大部分的数据处理工作，减小服务器的压力。同时可以提升用户体验感。因此，开发上可行。

3.2 技术可行性

Vue 作为前端三大主流框架之一，它由国人开发，官方文档明了清晰，比 React,Angular 简单易学。其对模块友好，代码格式方面不像 Angular 那样局限，可以使用 NPM,Bower 或者 Dou 安装，使其应用场景更加灵活高效，其组件功能强大，可解耦，可复用，可重新组合的特点，使得开发时能够节省时间，提高效率，同时易于团队维护。

Node.js 基于 JavaScript 语法，打破了 JavaScript 只能运行在浏览器的局限，使得前后端编程环境统一。因此，只需掌握 JavaScript 语法便可学会 Node 后端开发。

MongoDB 作为一个面向文档的数据库，它直接存取 BSON,使用场景的灵活性高，非常适合实时，插入，查询和更新。其优势明显，丰富的文档易于开发者迅速上手，提高开发效率，方便开发者高效快速开发。

综上，技术可行。

3.3 经济可行性

针对于本次开发，开发场景不限，普通的个人笔记本电脑作为开发工具便可，使用本机进行测试，基本 chrome 浏览器和 cmd 控制台就能输出测试和查看报错结果。因此，经济上可行。

4 需求分析

眼下，互联网时代的不断发展进步，使得网络成为人们学习，生活中的必需品，不断的影响和充实我们的日常生活。为了顺应时代潮流，构建一个博客平台来充分表达我们的思想，展现个人才能，抒发内心情感，拓宽人们的娱乐生活方式势在必行。庞大的网民是我们的目标受众，目标受众覆盖各个年龄阶层，因此，App 设计旨在页面简洁，操作简单，可读性高，能够满足各年龄阶层的市场需求。本次 App 设计开发，参考当下热门同类型 App 设计配色，把需求大致分为：用户登录/注册，用户管理，好友管理，文章管理，评论点赞，消息管理等。

5 模块设计

5.1 功能模块

本 App 主要分为用户管理，文章管理，消息管理，个人信息页面等功能进行模块开发设计，目的是给用户提供一个简洁，实用的生活分享类博客 App。当用户未登录时可以查看权限开放的最新文章，但不能进行评论，点赞，发表文章，查看好友等功能。部分功能未登录用户禁止使用。模块设计如图 1 所示：

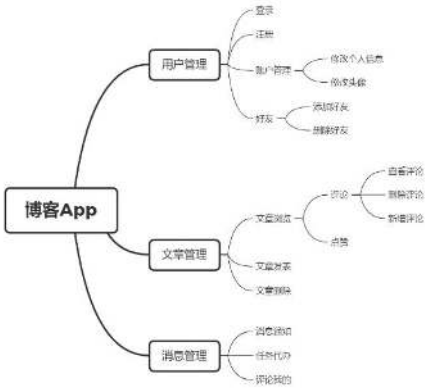

图 1 功能模块设计

5.2 功能描述

5.2.1 用户注册/登录

（1）注册操作只针对新用户（已注册用户直接登录即可）。

（2）用户在登录界面点击注册按钮进行注册。

（3）用户进行注册操作时，后台会发送验证码到注册邮箱。用户填写验证码，与服务器生成验证码相匹配时，对数据库进行添加数据，返回操作成功状态码。

（4）用户进行登录操作时，将表单数据发送给服务器，服务器接收到前端请求后，查找数据库字段，当邮箱与密码匹配成功时，则登录成功，匹配不成功，返回错误提示信息。

（5）用户登录功能关键代码如下：

后端接口设计：

```
router.post("/login",(req,res)=>{
        User.findOne(req.body,(errs,data)=>{
            if (data){
            req.session.user_id=data._id;
                  return res.json({
            ret:1,
            data:{
            user_id: data._id
                    }
                  })
            }else{
                 return res.json({
            ret: 0,
                   data: errs
                 })
               }
            })
      })
```

前端 axios 请求：

```
        axios.post("/api/login",this.loginForm)
    .then(res=>{
        if (res.data.ret==1&&res.status==200) {
        this.$message({
                message:"登录成功!",
                type: "success"
        });
        this.$router.push("/");
        }else{
        this.$message.error("登录失败!");
            }
      });
```

5.2.2 用户管理

（1）该功能面向已登录用户。用户可以对个人信息进行修改保存。

（2）用户点击确定按钮提交修改表单，表单数据会发送给后台，后台查询数据后进行更新，并发送成功状态码给前端，前端页面自动刷新并显示新信息。

5.2.3 文章管理

（1）该功能面向已登录用户。用户可以在 App 内进行相应文章操作等功能体验。

（2）用户进入首页时，自动查询数据库，返回查询数据并显示前 8 条信息

（3）用户在发表页面进行文章发表，提交文章信息后，数据发送到后台，后台进行数据添加，并发送成功状态码给前端，前端页面自动刷新并显示文章。

（4）用户在任意文章界面均可进行删除操作，点击删除按钮后，删除请求发送到后台，后台进行文章 id 匹配，删除，并返回成功状态吗给前端，前端自动刷新页面

5.2.4 点赞评论

（1）该功能面向已登录用户。用户可以在任意文章下面进行点赞，评论。

（2）用户在点赞，评论时，发送点赞评论请求给后台，后台进行数据添加，并返回成功状态码给前端，前端进行新数据的显示。

（3）评论功能关键代码：

后端接口设计：

```
    var addComment=new Comment(datas);
      addComment.save(function (errs,data){
            let [ret,msg]=[0,"评论失败"]
              if (!errs){
          Comment.find({_id:data._id})
          .populate({path:'author',select:'iduse
        ridavatat username
        introduction'}).exec().then(function(user){
            ret=1;
            res.json({ret:ret,ata:user})
    })
        }else{
            res.json({ret:ret,msg:msg,data:data})
      }
     })
})
```

5.3 部分项目成果展示

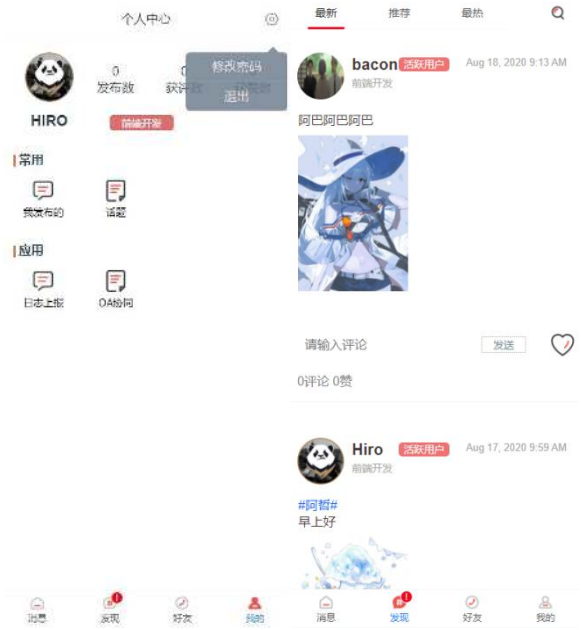

图 2 项目展示

结语

本文简述了基于 Vue.js 框架开发博客 App 的实现流程，选取 Vue.js+MongoDB+Mongoose 技术，通过 Vue 搭建页面，同时使用 Mongoose 模块搭建服务器，以 MongoDB 作为数据存储实现 App 功能开发。目前该 App 仍有需要改进完善的地方，在后续过程中将根据使用情况不断进行优化更新，致力于为用户提供一个简洁，高效的博客平台，提高用户满意度的博客平台。

参考文献：

[1] 钏涛，薛永军. 关于分布式存储应用技术的应用[J]. 电子技术与软件工程，2016，No.76(02):210.

[2] 张钊源,刘晓瑜,鞠玉霞.Node.js 后端技术初探[J].中小企业管理与科技(上旬刊),2020(08):193-194.