

Design and Implementation of Blog System based on Node.js

Ziqing TANG Zhengde BAO Yawen TANG

School of Computer and Software, Jincheng College, Sichuan University, Chengdu, 611731

Abstract

Blog is also known as the diary of human nature, with self-management, irregular update content as a feature, with the update of the network age blog form content is also constantly enriched. With the development of node.js technology, the writing of server-side programs has gradually begun to rely on JavaScript. This paper introduces the development of blog system by combining node.js technology.

Key Words

Node.js, Express, Mongo. Blog

DOI:10.18686/jsjxt.v1i2.688

基于 Node.js 的博客系统的设计与实现

唐子晴 鲍正德 唐娅雯

四川大学锦城学院计算机与软件学院, 四川成都, 611731

摘要

博客又称人性的日记, 以自我管理、不定时更新内容为特色, 随着网络时代的更新换代博客的形式内容也在不断丰富。随着 Node.js 技术的发展, 服务器端程序的编写渐渐开始依靠 JavaScript, 本文结合 Node.js 技术对博客系统开发进行了介绍, 浅析了如何通过 Node.js 技术用 Express 与 MongoDB 数据库建立博客系统, 实现了登录注册功能、发布功能、留言功能、信息检索功能等。

关键字

Node.js; Express; MongoDB; 博客

1.引言

在互联网发展之初, 便衍生出了网络社交, 科技的进步同时引导了网络社交的蓬勃发展, 以中国为例, 截止 2018 年 12 月的数据统计, 中国的互联网社交使用率占全国总人口的 50.3%。博客 (Blog) 流行于 2000 年, 起初只是个人写的网络日记, 经过不断地更新迭代之后, 如今的博客内容多样, 既可以当成用户个人的心情记录日记, 也可用于和网友分享心得、技术、新闻等干货文章, 网友在文章下方留言与博主取得进一步联系和交流。博客随着时代的发展潮流不断地更新增加功能和内容, 也越来越具有个性化特征, 所以它在网络时代一直占有一席之地。Node.js 天生代码简洁加上有效率, 并且可以通过 V8 引擎在服务器端快速搭建并且运行一

个项目, 这种支持高并发的优势正好适用于博客系统的搭建。

2.Node.js 特点

2.1 非阻塞异步 I/O

I/O 是指数据的读取和写入, 异步 I/O 和同步 I/O 总是被拿来区分理解, 同步 I/O 指系统访问数据是一步一步进行的, 就算进程会被阻塞, 也要等到访问完该数据。相反的, 非阻塞异步 I/O 会在系统访问数据时, 选择不死等, 而是接着做后面的事情, 等到系统访问完数据之后, 再用回调函数来处理。可用一个简单的生活例子来解释, 某餐馆有一位服务员, 每当客人来店时, 服务员会给客人递上菜单, 给客人时间让他思考自己的菜

单,与此同时,服务员会用这位客人点餐的空档,服务另一位客人,而不是选择死等第一位客人点菜而让第二位客人等待,这样做的好处是节约了大量的时间和资源,在同等时间内大大提高了工作效率。

2.2 单线程

单线程通俗的解释就是它的线程有且只有一个,前面的执行好了才能执行下一个。单线程的优点是因为线程少,所以降低了开发成本,但随之即来的缺点就是如果线程中出现一个错误,就会使整个进程停止运行。

2.3 事件驱动

事件驱动机制又称“事件环”机制,指在执行回调函数时,如果有新事件加入,可以在处理了新事件之后又返回执行回调函数,它确保了 Node.js 的正常运行。

3. 相关技术

3.1 Express

Express 是一个框架,用于开发 web,博客系统的框架就可由此搭建。Express 的用法是把模块安装在 blog 的根目录下,以供工程文件所需,然后初始化 blog 项目。

3.2 MongoDB

MongoDB,分布式文档存储数据库,是用 c++写的非关系型数据库(NoSQL)的一种,它的存在解决了 web 的数据存储性能不足的问题,并且具有很强的延展性。。现在网络信息铺天盖地,数据库需要存储的数据也是数不胜数,因此对数据库的要求更是百尺竿头更进一步。SQL 是数据库中的老大哥,从 1998 年开始便服务于大众,但身处如今的大数据时代,SQL 越来越不能满足大众的需求,缺点也在不断地被放大,与此同时 NoSQL (Not Only SQL) 开始崛起, MongoDB 作为其中的一种,深受用户的喜爱。

4. 系统分析

博客系统主要是为用户提供一个信息的交流和共享网络平台,用户既可以在博客上建立自己的朋友圈,发布个人心情日记,和朋友分享自己的生活,又能就某个问题发表文章与广大网友探讨并共享解决问题的方法。

4.1 需求分析

程序的设计以满足用户的需求为基础,只有用户体验度高的系统才能维持一个程序的长久的生存周期。需求分析首先需要了解用户对于即将开发的这个程序有哪些需求和具体的要求,然后将这些信息进行整理和分析,汇总成一份需求规格说明书,后期的工作就是根据需求规格说明书确定软件需要拥有哪些功能并进行程序设计。^[1]该博客具有以下功能:

4.1.1 登陆注册

进入博客后用户有两种状态,分别是登录和登出。^[2]用户在打开网站后,不登录也可以浏览博客的内容,但是很多操作都会受到限制。在登录状态的用户又分为发布者和访问者,发布者有权利对文章进行发布和增删查改的操作,访问者只能访问发布的文章但不能编辑,也可以留言,这样较好的保障了发布者对自己博客的权利。用户在进入博客后,首先需要注册一个身份 ID 以成为该博客系统中的一员之后。已经有了账号的用户,在输入框输入账号和密码跳转到登陆状态即可,如果没有账号,按提示注册用户名和密码之后可登陆。

4.1.2 发布区

发布区这部分包括了发表文章,删除文章和修改文章三个部分。用户随时可以发表文章,在文章发表了之后,如果对文章的质量和 content 觉得不满意了,可以通过修改功能重新修改编辑这篇文章,直到满意为止,当然,用户在任何时候都能删除这篇文章。

4.1.3 留言区

博客除了能发布文章之外,还有一个特色的功能就是留言。用户能给博主留言,留言的时间和内容会显示在留言区,而此篇文章的作者除了能看到上述内容之外,还有回复留言或者删除留言的管理权限,以保障自身的权利。留言区让博客增加了人气,成为了用户之间交流的桥梁。

4.1.4 文章检索区

为了提高用户的体验,特地增加了文章检索的功能。很多用户都会有这样的烦恼:在偶然间看过一篇文章,过段时间后想再次访问却因为忘记了文章的标题或

者作者而再也找不到文章了。针对这一普遍存在的情况,此博客系统增加了模糊检索功能,用户可根据作者、标题或文章的部分内容,并按相关度进行排序,由此查找到目标文章。

4.2 系统功能设计

根据上述的需求分析,确定出了博客的功能结构图。如图1所示。

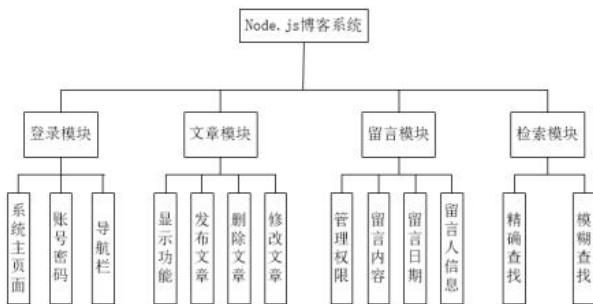


图1 功能结构图

4.3 数据库设计

该博客系统采用 MongoDB 作为数据库,首先在官网下载 MongoDB,然后用 node-mongodb-native 驱动模块连接数据库,并且需要一个专门来保存信息的根目录 js 文件。

```
//连接数据库
mongoose.connect('mongodb://localhost:27017/articles');

var Schema = mongoose.Schema;
var blogSchema = new Schema({
  title: String,
  content: String,
  time: String
});
var Blog = mongoose.model('Blog', blogSchema);
```

该博客系统的 E-R 图包含实体有用户、文章和留言,如图2所示。^[3]

5. 功能实现

博客系统在根据两个不同状态呈现不同界面,未登录前有登录页和注册页,登陆后显示的是发布区、留言区和检索页面。

5.1 登陆注册

因为游客状态是受限制的,所以为了得到完整的用户使用权,需要先注册登录。为了实现用户注册的信息存入数据库的用户文档,我们用 Users.prototype.save 来存储用户信息,再用 Users.get 实现用户信息读取。^[4]为了提高用户账号的安全性,使用了 crypto 生成密码的 md5 值来对用户设置的密码进行加密处理,达到提高安全指数的效果。

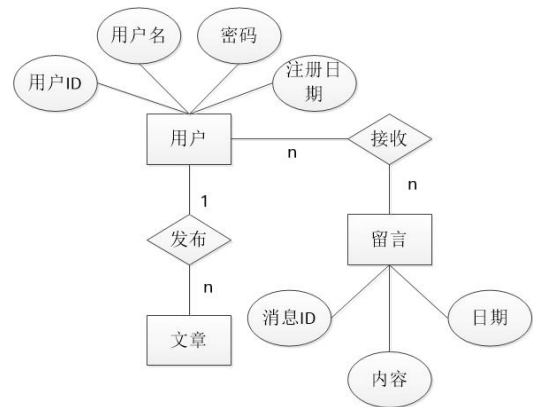


图2 系统 E-R 图

```
var md5=crypto.createHash('md5'),
    password
    =
md5.update(req.body.password).digest('hex');
var newUser=new newUser({
  name:req.body.name,
  password:password,
  email:req.body.email
});
```

5.2 首页

首页是博客系统的门面,用户打开博客之后映入眼帘的就是首页,所以首页需要囊括博客系统的主要内容。首先是导航条,用户可以通过导航条直达想去的页面,效果可通过 html+css 来实现。其次首页需要显示用户发布过的文章标题和部分内容,这里需要用到的是数据库,通过 request 请求数据库的内容,数据库收到请求后作出 response 响应,把数据返回给网页。

```
app.get('/', function (req, res) {
  //加载数据库内容
  Blog.find({}, function (err, data) {
    if (err) {
```

```

        console.log(err);
    };
    //将数据库数据返回给网页
    res.render('index', {
        contents: data
    });
});
});

```

5.3 发布区

5.3.1 添加功能

添加功能就是用户写好文章后可以在这个页面发表出去。要实现这个功能需要向服务器传递数据，所以要使用 `post` 函数。因为博客是即时性的，所以还要用 `date` 函数获取时间和日期。

```

app.get('/post', function (req, res) {
    res.render('post', {});
});
//添加数据
app.post('/post', function (req, res) {
    //获取表单数据
    var time = new Date();
    var Time = {
        Year: time.getFullYear(),
        Month: time.getMonth() + 1,
        Day: time.getDate(),
        Hours: time.getHours(),
    }
    var date = (Time.Year + "-" + Time.Month + "-" +
    Time.Day + " " + Time.Hours + ":" + Time.Minutes + ":"
    + Time.Seconds);
    console.log(date);
    var blog = new Blog({
        title: req.body.title,
        content: req.body.content,
        time: date
    });
    //保存到数据库
    blog.save(function(err, res) {});
    //返回主页

```

```

        res.redirect('/');
    });

```

5.3.2 删除功能

用户在某些时候也会想删除自己发布的文章，所以就出现了删除功能。同样我们用 `post` 函数，打开数据库根据用户 ID 查找并删除文章。

```

//删除数据
app.get('/delete', function (req, res) {
    //获取 id
    var id = req.query.id;
    //通过查询 id 删除
    Blog.findByIdAndRemove(id, function (err) {
        if (err) {
            console.log(err);
        };
        //返回主页
        res.redirect('/');
    });
});

```

5.3.3 修改功能

用户修改自己的文章也是常有的事，所以博客加入了修改文章的功能。首先需要请求修改数据，然后通过响应把修改数据的值返回给网页。

```

app.get('/update', function (req, res) {
    var id = req.query.id;
    Blog.findById(id, function (err, result) {
        if (err) {
            console.log(err);
        };
        res.render('update', {
            content: result
        });
    });
});

```

用户在修改了文章之后还需要更新修改之后的日期。

```

app.post('/update', function (req, res) {
    var id = req.query.id;
    var time = new Date();
    var Time = {

```

```

        Year: time.getFullYear(),
        Month: time.getMonth() + 1,
        Day: time.getDate(),
        Hours: time.getHours(),
    }
    var date = (Time.Year + "-" + Time.Month + "-"
+ Time.Day + " " + Time.Hours);
    console.log(date);
    var blog = {
        title: req.body.title,
        content: req.body.content,
        time: date
    };
    Blog.findByIdAndUpdate(id, blog, function (err)
{
    if (err) {
        console.log(err);
    };
    res.redirect('/');
});
});
});

```

5.4 留言功能

为了增强博客的趣味性，留言功能是必不可少的。在留言的过程中，用 comments 组建把用户输入的留言储存到数据库中，留言完成之后用 res.redirect('back')可以返回页面并显示留言。^[5]

```

Comment.prototype.save=function(callback){
    var name=this.name,
        day=this.day,
        title=this.title,
        Comment=this.Comment;

    HTMLAllCollection.update({
        "name":name,
        "time.day":day,
        "title":title

```

```

    },{
        $push: {"comments":Comment}
    },function(err){
        mongodb.close();
        if(err){
            return callback(err);
        }
        callback(null);
    });
};

```

5.5 文章检索功能

用户可根据自己的想法和习惯自行选择模糊或精确搜索，用 search 函数可实现该操作。用模糊检索的原因是找出很多相关的文章，让用户根据自己的需求自行选择查看。

```

app.get('/search',function(req,res){
    Post.search(req.query){
        if(err){
            req.flash('error',err);
            return res.redirect('/');
        }
        res.render('search',{
            posts:posts,
            user:req.session.user,
            success:req.flash('success').toString(),
            error:req.flash('error').toString()
        });
    });
});

```

6.结束语

博客作为一个社交媒体平台，以网络为载体，记录着人们的交流互动、分享着人们的经验之谈、收藏着人们的心情日记，而人们对于博客的功能、界面的需求也精益求精。Node 作为一个年轻的前端框架，后台语言，各种文档和库发展还不完全完善，但是它能高并发连接服务器，并逐渐让 Javascript 与 PHP 等服务端语言媲美，让博客的共享与分享精神以更生动的形式、更完善的功能深入人心。



图3 博客界面效果图

参考文献

- [1]李贇华.江西萍乡市公安情报信息管理系统的设计与实现[D].福建:厦门大学,2015.
- [2]宋万刚.基于 Web2.0 的辽宁科技学院知识管理系统的设计与实现[D].四川:电子科技大学,2011.
- [3]尚云龙.关于网络博客系统的设计与实现[D].四川:电子科技大学,2013.
- [4]罗承林.嵌入式 Web 远程监控技术在污染源自动监控系统中的应用[D].广东:华南理工大学,2008.
- [5]陈雷.天津通信公司金色会员俱乐部管理信息系统的 Web 实现 [D]. 天津 : 南开大学 ,2004.

DOI:10.7666/d.y804689.

作者简介

第一作者：唐子晴（1997-），女，汉，四川省广安市，本科，四川大学锦城学院，研究方向：前端技术开发、信息管理与信息系统。

第二作者（通讯作者）：鲍正德（1989-），男，汉，黑龙江哈尔滨，研究生，四川大学锦城学院，研究方向：电子商务。

第三作者：唐娅雯（1999-），女，汉，四川省资阳市，本科，四川大学锦城学院，研究方向：信息管理、J2EE