



This repository

Search or type a command

ExploreGistBlogHelp

Macro-Bin

nswbmw / N-blog

Watch92Star387Fork245

第4章 实现用户页面和文章页面

Edit PagePage HistoryClone URL

现在，我们来给博客添加用户页面和文章页面。

所谓用户页面就是当点击某个用户名链接时，跳转到：`域名/u/用户名`，并列出了该用户的所有文章。
同理，文章页面就是当点击某篇文章标题时，跳转到：`域名/u/用户名/时间/文章名`，进入到该文章的页面（也许还有该文章的评论等，以后再做介绍）。

在开始之前我们需要做个工作，打开 `post.js`，将 `Post.get` 修改为 `Post.getAll`，同时将 `index.js` 中 `Post.get` 修改为 `Post.getAll`。
在 `post.js` 最后添加如下代码：

```
//获取一篇文章
Post.getOne = function(name, day, title, callback) {
  //打开数据库
  mongodb.open(function (err, db) {
    if (err) {
      return callback(err);
    }
    //读取 posts 集合
    db.collection('posts', function (err, collection) {
      if (err) {
        mongodb.close();
        return callback(err);
      }
      //根据用户名、发表日期及文章名进行查询
      collection.findOne({
        "name": name,
        "time.day": day,
        "title": title
      }, function (err, doc) {
        mongodb.close();
        if (err) {
          return callback(err);
        }
        //解析 markdown 为 html
        doc.post = markdown.toHTML(doc.post);
        callback(null, doc);//返回查询的一篇文章
      });
    });
  });
};
```

简单解释一下：

- `Post.getAll`：获取一个人的所有文章（传入参数 `name`）或获取所有人的文章（不传入参数）。
- `Post.getOne`：根据用户名、发表日期及文章名精确获取一篇文章。

下面我们来实现用户页面和文章页面。

打开 `index.js`，将 `app.get('/')` 内的 `Post.get(null, function (err, posts) {` 修改为：

```
Post.getAll(null, function (err, posts) {
```

在 `app.post('/upload')` 后添加如下代码：

```
app.get('/u:name', function (req, res) {
  //检查用户是否存在
  User.get(req.params.name, function (err, user) {
    if (!user) {
      req.flash('error', '用户不存在!');
      return res.redirect('/');//用户不存在则跳转到主页
    }
    //查询并返回该用户的所有文章
```

```
Post.getAll(user.name, function (err, posts) {
  if (err) {
    req.flash('error', err);
    return res.redirect('/');
  }
  res.render('user', {
    title: user.name,
    posts: posts,
    user : req.session.user,
    success : req.flash('success').toString(),
    error : req.flash('error').toString()
  });
});
});
});
```

这里我们添加了一条路由规则 `app.get('/u/:name')`，用来处理访问用户页的请求，然后从数据库取得该用户的数据并渲染 `user.ejs` 模版，生成页面并显示给用户。

接下来我们添加文章页面的路由规则。

在 `app.get('/u/:name')` 后添加如下代码：

```
app.get('/u/:name/:day/:title', function (req, res) {
  Post.getOne(req.params.name, req.params.day, req.params.title, function (err, post) {
    if (err) {
      req.flash('error', err);
      return res.redirect('/');
    }
    res.render('article', {
      title: req.params.title,
      post: post,
      user: req.session.user,
      success: req.flash('success').toString(),
      error: req.flash('error').toString()
    });
  });
});
});
```

最后，我们创建用户页面和文章页面的模版文件。

在 `views` 文件夹下新建 `user.ejs`，添加如下代码，同时也将 `index.ejs` 也修改成如下代码：

```
<%- include header %>
<% posts.forEach(function (post, index) { %>
  <p><h2><a href="/u/<%= post.name %>/<%= post.time.day %>/<%= post.title %>"><%= post.title %></a></h2></p>
  <p class="info">
    作者: <a href="/u/<%= post.name %>"><%= post.name %></a> |
    日期: <%= post.time.minute %>
  </p>
  <p><%= post.post %></p>
<% }) %>
<%- include footer %>
```

在 `views` 文件夹下新建 `article.ejs`，添加如下代码：

```
<%- include header %>
<p class="info">
  作者: <a href="/u/<%= post.name %>"><%= post.name %></a> |
  日期: <%= post.time.minute %>
</p>
<p><%= post.post %></p>
<%- include footer %>
```

现在，我们给博客添加了用户页面和文章页面。示例：

用户页面



文章页面



Last edited by nswbmw, a month ago