






This repository ▾

Search or type a command 🔍

ExploreGistBlogHelp

 Macro-Bin   

 nswbmw / N-blog

Watch ▾92

★ Star387

Fork245

HomePagesHistory

New Page

第11章 增加文章检索功能

Edit PagePage HistoryClone URL

现在我们来给博客增加文章检索功能，即根据关键字模糊查询文章标题，且字母不区分大小写。
首先，我们修改 header.ejs，在 `</nav>` 前添加一行代码：

`<form action="/search" method="GET"><input type="text" name="keyword" placeholder="SEARCH" class="search" /></form>`

在 style.css 中添加一行样式：

`.search{border:0;width:6em;text-align:center;font-size:1em;margin:0.5em 0;}`

打开 post.js，在最后添加如下代码：

```
//返回通过标题关键字查询的所有文章
Post.search = function(keyword, callback) {
  mongodb.open(function (err, db) {
    if (err) {
      return callback(err);
    }
    db.collection('posts', function (err, collection) {
      if (err) {
        mongodb.close();
        return callback(err);
      }
      var pattern = new RegExp("^.*" + keyword + ".*$", "i");
      collection.find({
        "title": pattern
      }, {
        "name": 1,
        "time": 1,
        "title": 1
      }).sort({
        time: -1
      }).toArray(function (err, docs) {
        mongodb.close();
        if (err) {
          return callback(err);
        }
        callback(null, docs);
      });
    });
  });
};
```



```
//返回通过标题关键字查询的所有文章
Post.search = function(keyword, callback) {
  mongodb.open(function (err, db) {
    if (err) {
      return callback(err);
    }
    db.collection('posts', function (err, collection) {
      if (err) {
        mongodb.close();
        return callback(err);
      }
      var pattern = new RegExp("^.*" + keyword + ".*$", "i");
      collection.find({
        "title": pattern
      }, {
        "name": 1,
        "time": 1,
        "title": 1
      }).sort({
        time: -1
      }).toArray(function (err, docs) {
        mongodb.close();
        if (err) {
          return callback(err);
        }
        callback(null, docs);
      });
    });
  });
};
```

注意：我们通过 `pattern` 定义了包含关键字 `keyword` 的正则，若 `keyword` 字符串的开头或结尾包含特殊字符（比如说 `*`）则需转义。

修改 index.js，在 `app.get('/u/:name')` 前添加如下代码：

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

```
});
```

在 `views` 文件夹下新建 `search.ejs`，添加如下代码：

```
<%- include header %>
<ul class="archive">
<% var lastYear = 0 %>
<% posts.forEach(function (post, index) { %>
  <% if(lastYear != post.time.year) { %>
    <li><h3><%= post.time.year %></h3></li>
    <% lastYear = post.time.year } %>
    <li><time><%= post.time.day %></time></li>
    <li><a href="/u/<%= post.name %>/<%= post.time.day %>/<%= post.title %>"><%= post.title %></a></li>
  <% }) %>
</ul>
<%- include footer %>
```

注意：目前为止，你会发现 `tag.ejs` 和 `search.ejs` 代码完全一样，因为我们都用相同的布局。这也突出了模版的优点之一，可以重复利用，但我们这里并没有把三个文件用一个代替，因为每一个文件的名字代表了不同的意义。

现在，我们给博客添加了文章检索功能。

Last edited by nswbmw, a month ago