






This repository ▾

Search or type a command 🔍

ExploreGistBlogHelp

 Macro-Bin   

IC  nswbmw / N-blog

Watch ▾92

★ Star387

Fork245

HomePagesHistory

New Page

番外篇之——使用 _id 查询

Edit Page

Page History

Clone URL

我们知道，MongoDB 会自动为每个文档添加一个特殊的 `_id` 键，这个 `_id` 键的值是经过特殊计算的长度为 24 的字符串（详见《MongoDB 权威指南》），因此保证了每个文档的 `_id` 都是独一无二的。那我们可不可以使用 `_id` 键查询一个独一无二的文档呢？当然可以，这也是设计 `_id` 的原因所在。

使用 `name`、`day`、`title` 查询一篇文章有个小 bug，即不能在同一天发表相同标题的文章，或者说发表了相同标题的文章后只能返回最近发表的那篇文章。使用 `_id` 就可以很好的避免这个 bug。

下面我们举例使用 `_id` 代替使用 `name`、`day`、`title` 来查询一篇文章，即将：

```
app.get('/u/:name/:day/:title')
```

修改为以下形式：

```
app.get('/p/:_id')
```

打开 `post.js`，在最上面添加：

```
var ObjectID = require('mongodb').ObjectID;
```

将

```
Post.getOne = function(name, day, title, callback) {
```

修改为

```
Post.getOne = function(_id, callback) {
```

并将 `Post.getOne()` 内两处的

```
"name": name,
"time.day": day,
"title": title
```

都修改为

```
"_id": new ObjectID(_id)
```

打开 `index.js`，将 `app.get('/u/:name/:day/:title')` 修改如下：

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

注意：我们将文章页面的路由修改为 `app.get('/p/:_id')` 而不是 `app.get('/u/:_id')` 是为了防止和上面的用户页面的路由 `app.get('/u/:name')` 冲突，况且，`p` 也代表 `post`，表示发表的文章的意思。

打开 `index.ejs`，将

```
<p><h2><a href="/u/<%= post.name %>/<%= post.time.day %>/<%= post.title %>"><%= post.title %></a></h2>
```

修改为

```
<p><h2><a href="/p/<%= post._id %>"><%= post.title %></a></h2>
```

现在，运行你的博客并发表一篇文章，从主页点击标题进入该文章页面，就变成了以下的 `url` 形式：

```
http://localhost:3000/p/52553dcd5bb408ec11000002
```

注意：`MongoDB` 数据库中是以以下形式存储 `_id` 的：

```
"_id" : ObjectId("52553dcd5bb408ec11000002")
```

我们可以直接使用 `post._id` 从数据库中获取 `_id` 的值（字符串），但在查询的时候，要把 `_id` 字符串包装成 `MongoDB` 特有的 `ObjectId` 类型。

读者可依此类推，自行将剩余的工作完成。

Last edited by nswbmw, 15 days ago