

[首页](#) [目录](#)[退出](#)

MongoDB 数据库入门

MongoDB 数据库图形化管理工具

Mongoose 连接两座孤岛之间的桥梁

借助 Mongoose 保存用户信息到 MongoDB 数据库

从本节课程开始，我们将介绍关于网站用户认证相关的知识点了，演示让用户通过输入用户名和密码实现认证的过程。本节课程主要完成的任务是利用 Mongoose 保存用户名和密码到 MongoDB 数据库中。

构建 User Model

在把数据存入 MongoDB 数据库之前，我们先了解三个概念：数据库（database）、集合（collection）和文档（document）。这三者的关系是一个数据库中包含多个集合，一个集合中又包含多条文档。每条

文档中存储了具体的数据。现在本案例使用的数据库是 `react-hand-in-hand`，接下来我们需要创建一个集合，才能保存数据。在前面课程中，我们知道了可以用 `mongo` 或者 `mongo-express` 创建集合，那么在 Express 应用中 Mongoose 是如何完成这个任务的呢？

新建一个文件 `models/user.js`，导入 `mongoose` 模块并赋值给 `mongoose` 变量：

```
var mongoose = require('mongoose');
```

然后调用 `Mongoose#Schema()` 接口，得到一个 `Schema` 对象类型：

```
var Schema = mongoose.Schema;
```

每个 `Schema` 实例对象会映射为 MongoDB 数据库中的一个集合，同时还能定义所映射集合包含的字段，以及字段的类型等规范。接下来，创建一个新的 `Schema` 实例对象 `UserSchema`：

```
var UserSchema = new Schema();
```

给 `UserSchema` 所映射的集合添加两个自定义的字段：`username` 和 `password`，规定了每个字段存储数据的类型，并对存储数据进行了简单的校验：

```
var UserSchema = new Schema(  
  {  
    username: { type: String, unique: true, },  
    password: { type: String, required: true  
  }  
);
```

其中，`type` 指定存储数据类型，`String` 代表字符串类型；`unique` 校验存储数据是否唯一，属性值为 `true`，则代表该字段存储的数据不能重复；`required` 校验存储数据是否为空，属性值为 `true`，则代表该字段存储的数据不能为空。

另外，再给 `UserSchema` 所映射的集合添加两个字段：`createdAt` 和 `updatedAt`，这两个字段不需要手动添加，添加一行代码就搞定了：

```
var UserSchema = new Schema(  
  {...},  
  { timestamps: true }  
);
```

`timestamps` 属性值为 `true` 的时候，会自动添加 `createdAt` 和 `updatedAt` 两个字段。

创建 User 模型

上面我们总是提到 `UserSchema` 所映射的集合，那它到底映射的是哪一个集合呢？下面一行代码会给出答案：

```
module.exports = mongoose.model('User', UserSchema);
```

通过 `Mongoose#model()` 方法把一个 Schema 实例对象转换成一个 `Model`，传递给这个方法的第一个参数即为将转换成的 `Model` 的名字。

Mongoose Model 命名规范是名词单数形式并且首字母为大写字母，Mongoose 会把 Model 名字小写并复数化，比如上述代码中，Mongoose 会把 `User` 变成 `users`，从而能对应 MongoDB 数据库中的 `users` 集合。若 Model 的名字是不可数名词，则 Mongoose 不会在 Model 名字的末尾添加 `s` 字母。请参考文章 [Mongoose 从未告诉过你的](#)

现在，我们就知道了 Mongoose 通过模型化一个 Schema 对象实例，就能与 MongoDB 数据库中的集合对应起来了。接下来，就是搞清楚 Mongoose 是如何把数据保存到数据库集合中的。

存储一条用户信息到数据库

打开 `server/index.js` 文件，首先导入刚才定义的 `User` 模型：

```
var User = require('./models/user');
```

当 Mongoose 连接 MongoDB 数据库成功之后，我们创建一个 `User` 模型的对象 `user` 并初始化，其对应

着 MongoDB 数据库 `users` 集合中的一条文档

```
db.once('open', function() {  
  var user = new User({  
    username: 'peter',  
    password: 'aaaaaa'  
  });  
});
```

然后调用 `Model#save()` 方法保存新创建的用户信息到数据库:

```
db.once('open', function() {  
  ...  
  user.save();  
});
```

保存文件后, 在命令行中输入 `mongo` 命令, 启动 MongoDB 数据库的命令行操作界面, 执行下面命令:

```
show dbs  
use react-hand-in-hand  
show collections  
db.users.find()
```

或者启动 MongoDB 数据库的图形化管理工具

`mongo-express`, 会发现刚才新创建的用户信息已经保存到数据库中了。不过, 你可能会觉得用户密码保存的不妥当, 用户密码没有加密保存, 下节课程我们

就解决这个问题。

欢迎添加 Peter 的微信: happypeter1983

冀ICP备15007992号-3