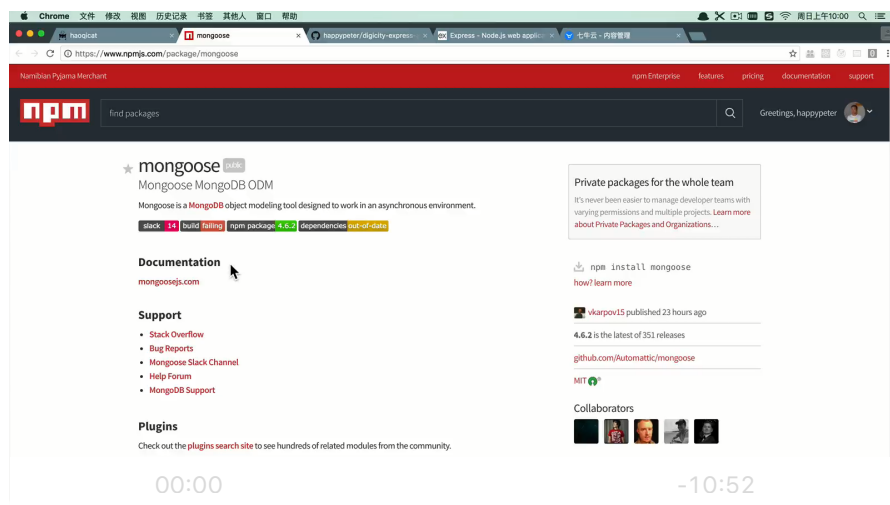


[首页](#) [目录](#)[退出](#)

## 第一章：后端搭建 Express API 服务

创建一个简单的  
Express 应用

MongoDB 数据库简  
单操作

# Express 对话 MongoDB

如何在 Express 应用和 MongoDB 数据库之间进行通信是本门课程要解决的问题。

## 安装 Mongoose 包

通过 **mongoose** npm 包，让 Express 应用和 MongoDB 建立连接

```
npm install --save mongoose
```

## 连接 MongoDB 数据库

打开 `index.js` 文件，首先添加一行：

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

导入 `mongoose` 功能模块，然后添加一行代码：

```
mongoose.connect('mongodb://localhost:27017/')
```

通过 `Mongoose` 提供的 `connect()` 方法连接到运行在本地的 `react-express-api` MongoDB 数据库。

然后，加入如下代码，判断连接是否成功：

```
var db = mongoose.connection;
db.on('error', console.log);
db.once('open', function() {
  console.log('success!')
});
```

这里的 `mongoose.connection` 会映射到数据库 `react-express-api`，若此时 MongoDB 数据库没有启动，则运行本项目的话，在命令行中会报告错误信息：

```
{ [MongoError: failed to connect to server []]
  name: 'MongoError',
  message: 'failed to connect to server [local...]
```

参考文档 [Mongoose 入门](#)

## 构建 Post Model

新建一个文件 `models/post.js`，首先添加两行代码：

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

导入 `mongoose` 功能模块以及调用它提供的 `Schema()` 接口创建一个新的 `schema`，每个 `schema` 会映射为 MongoDB 数据库中的一个 `collection`（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。下面代码就创建了一个名为 `PostSchema` 的 `schema`，并规定所映射的集合将包含三个字段：`category`、`title` 和 `content`，并且每个字段只能存储字符串类型的数据，其中 `title` 字段中存储的数据不能为空。

```
const PostSchema = new Schema({
  category: { type: String },
  title: { type: String, required: true },
  content: { type: String }
},
{ timestamps: true }
);
```

选项 `timestamps` 的值设置为 `true`，则自动给所映射集合添加 `createdAt` 和 `updatedAt` 两个字段。

虽然定义了一个 `schema`，但是 `Mongoose` 还不知道

这个 schema 到底映射成数据库中的哪个集合，所以还得把一个 schema 转换成一个 **model** 之后，根据 model 的名字，Mongoose 会自动查找到这个 schema 在数据库中对应的集合。

最后再添加一行代码：

```
module.exports = mongoose.model('Post', PostSchema);
```

通过 Mongoose 的 **model()** 方法把一个 schema 编译成一个 model，一个 model 实例会对应映射集合中的一条记录，这个 **model()** 方法的第一个参数 **Post** 则是映射集合名字的单数形式，所以 **PostSchema** 映射集合的名字是 **posts**。上述代码还把构建成的 **Post Model** 导出供外部其它文件使用。

现在 **posts** 数据集合虽然已经有模有样了，但是若没有数据存入 **posts** 数据集合中的话，本项目所使用的数据库 **react-express-api** 中是不存在 **posts** 集合的，所以我们接下来要做的工作就是构建一条 **post** 记录并存入数据库，这样 **posts** 集合就会真正存在了。

## 构建一条记录

打开文件 **index.js**，添加代码：

```
var Post = require('./models/post');
```

```
db.once('open', function() {  
  var post = new Post({title: 'mongoose usage'  
  post.save(function(err){  
    if(err) console.log(err);  
  })  
  console.log('success!');  
});
```

首先导入 Post model，然后创建一个新的 model 实例 `post`，其对应 `posts` 集合中的一条记录，最后数据保存到数据库。

注意：因为使用了异步操作方法 `save()`，导致在终端报告警告信息：

```
Mongoose: mpromise (mongoose's default promise)
```

解决办法是在连接 MongoDB 数据库

`mongoose.connect(...)`；之前，添加一行代码：

```
mongoose.Promise = global.Promise;
```

详情请参考 [promises](#)

欢迎添加 Peter 的微信：happypeter1983

冀ICP备15007992号-3

