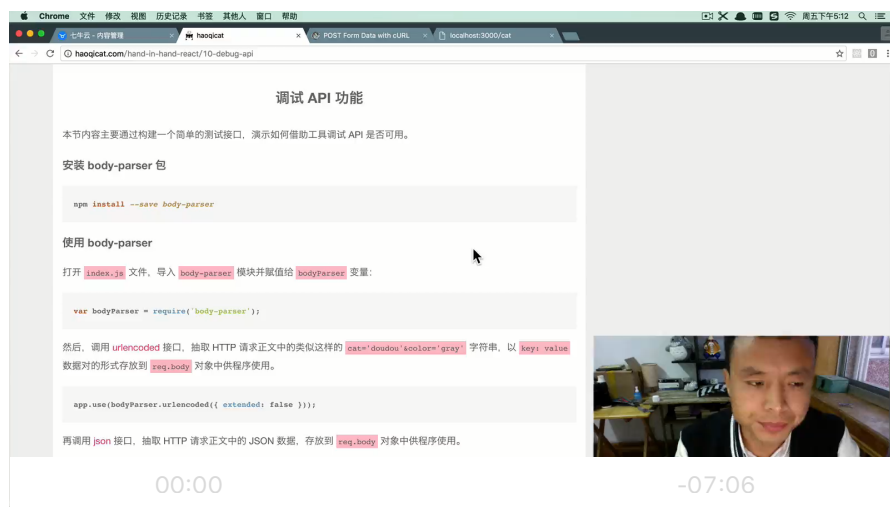


[首页](#) [目录](#)[退出](#)

Mongoose 连接两座
孤岛之间的桥梁

第二章：后端实现用户 认证接口

借助 Mongoose 保存
用户信息到
MongoDB 数据库

调试 API 功能

本节内容主要通过构建一个简单的测试接口，演示如何借助工具调试 API 是否可用。

安装 body-parser 包

```
npm install --save body-parser
```

使用 body-parser

打开 index.js 文件，导入 body-parser 模块并赋值给 bodyParser 变量：

```
var bodyParser = require('body-parser');
```

然后，调用 `urlencoded` 接口，抽取 HTTP 请求正文中的类似这样的 `cat='doudou'&color='gray'` 字符串（例如普通的 form 提交，就属于这种情况），以 `key: value` 数据对的形式存放到 `req.body` 对象中供程序使用。

```
app.use(bodyParser.urlencoded({ extended: false }));
```

再调用 `json` 接口，抽取 HTTP 请求正文中的 JSON 数据，存放到 `req.body` 对象中供程序使用。

```
app.use(bodyParser.json());
```

构建 API

新建 `server/routes.js` 文件，实现一个简单的名为 `/cat` 的接口：

```
module.exports = function(app) {  
  app.post('/cat', function(req, res) {  
    return res.json({  
      cat: req.body.cat,  
      color: req.body.color  
    });  
  });  
}
```

调用 `app.post()` 接口构建一个响应 HTTP POST 请求的 `/cat` 接口，并调用 `res.json()` 接口发送 JSON 数

据响应 HTTP 请求。其中 `req.body` 可以得到客户端 HTTP 请求发送过来的数据，这就得归功于安装的 `body-parser` 中间件了。

加载 API

打开 `server/index.js` 文件，导入刚才编写的路由模块

```
var routes = require('./routes');
```

然后在 `app.listen(3000, ...)` 代码之前添加下面一行代码，让定义的 API 生效：

```
routes(app);
```

调试 API

先来创建一个 `test.html` 内容如下

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>test</title>
</head>
<body>
  <form action="http://localhost:3000/cat" me
    <input type="text" name="color" />
    <input type="text" name="cat" />
    <input type="submit">
```

```
</form>
</body>
</html>
```

这样，可以测试 form 提交的这种形式了。

然后用 curl 这个命令行工具，来模拟客户端 JS 提交 JSON 数据的情形：

```
curl -H "Content-Type: application/json" -X POST
```

命令输出为：

```
{"cat": "doudou", "color": "gray"}%
```

接口调试工具 morgan

首先安装 **morgan** 包：

```
npm install --save morgan
```

morgan 是用于 Node.js 应用的 HTTP 请求日志中间件，它可以把 HTTP 请求信息按照一定的格式在终端中打印出来，或者是保存到文件中，方便接口调试。

然后，打开 server/index.js 文件，导入 morgan 模块并赋值给 logger 变量：

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

创建一个新的 `morgan` 日志中间件函数，并调用 `app.use` 加载它，代码如下：

```
app.use(morgan('dev'));
```

`morgan` 支持多种日志输出格式，上面代码中采用的是 `dev` 格式，默认日志信息会打印到终端窗口中。

欢迎添加 Peter 的微信：happypeter1983

冀ICP备15007992号-3