

[首页](#) [目录](#)[退出](#)

借助 Mongoose 保存
用户信息到
MongoDB 数据库

安全的存储用户密码

认证方式对比

JWT 知识科普

实现用户认证接口

上节课程中，我们已经知道了如何构建和调试 API，
本节课程主要构建本案例要用到的用户认证接口

构建用户认证 API

新建 `server/routes.js` 文件，导入 `User` 模型并赋值给 `User` 变量：

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

接下来定义用户认证接口，将实现的接口名称为
`/auth/login`：

```
module.exports = function(app) {  
  app.post('/auth/login', function(req, res)  
    User.findOne({ username: req.body.username  
      if(err) { return console.log(err); }  
      if(!user) { return res.status(403).json({  
        user.comparePassword(req.body.password,  
          if(err) { return console.log(err); }  
          if (!isMatch) { return res.status(403)  
            return res.json({  
              user: {name: user.username}  
            });  
          });  
        });  
      });  
    })  
  }  
}
```

用户从客户端向服务器提交其 用户名 和 密码，服务器端通过 `body-parser` 中间件把客户端传送过的数据抽取出来并存放到 `req.body` 中，这样我们就可以通过 `req.body.username` 获取到用户名。然后在 MongoDB 数据库中查找这个用户，若查找过程中出错，则打印错误信息到终端；若数据库中不存在这个用户，则向客户端响应错误信息；若数据库中存在这个用户，则验证客户端提交的密码

`req.body.password` 是否与用户保存在数据库中的密码匹配。若密码不匹配，则向客户端返回错误信息；若密码匹配，则给客户端返回用户信息。

定义 `comparePassword` 方法

打开 `models/user.js` 文件，在 `UserSchema.pre()`

方法之后，添加代码：

```
UserSchema.methods.comparePassword = function(password, cb) {
  bcrypt.compare(password, this.password, function(err, isMatch) {
    if (err) { return cb(err); }
    cb(null, isMatch);
  });
}
```

Schema.methods Bcrypt#compare()

测试用户认证接口

使用 curl

```
curl -H "Content-Type: application/json" -X POST http://localhost:3000/auth/login
```

通过 Chrome 插件 **Advanced REST client** 测试刚才编写的认证接口。

打开 **Advanced REST client** API 调试工具，向 `http://localhost:3000/auth/login` 接口发送 POST 请求，设置 `Content-Type` 请求头标识：

```
Content-Type: application/x-www-form-urlencoded
```

相应地向服务器发送的数据格式，如下所示：

```
username='billie'&password='cccccc'
```

之前我们已经保存了一条用户名为 `billie`，密码为 `cccccc` 的信息到 MongoDB 数据库中，所以能够得到正确的响应数据：

```
{
  "user": {
    "name": "billie"
  }
}
```

可以试着向服务器端提交不同的测试用户数据，观察响应信息的变化。

欢迎添加 Peter 的微信：happypeter1983

冀ICP备15007992号-3