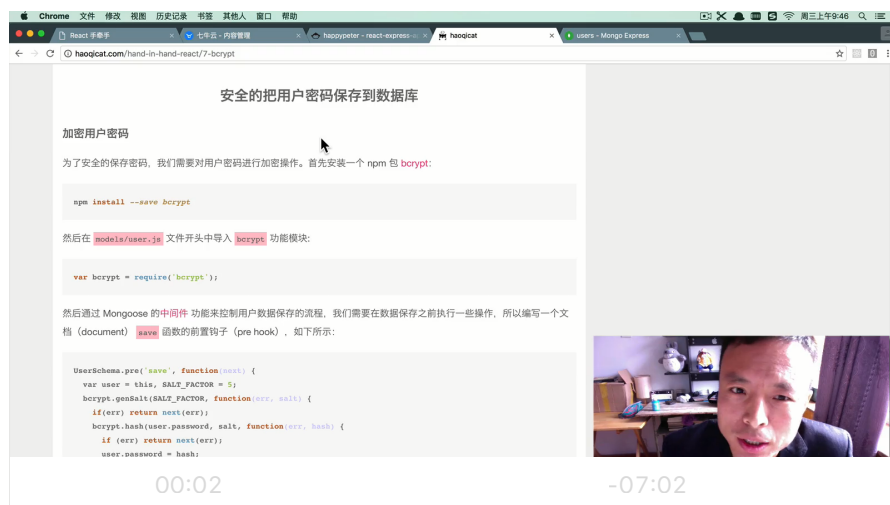


[首页](#) [目录](#)[退出](#)[奋仲磊 Nodemon](#)[MongoDB 数据库入门](#)[MongoDB 数据库图形化管理工具](#)[Mongoose 连接两座孤岛之间的桥梁](#)

## 安全的把用户密码保存到数据库

### 加密用户密码

为了安全的保存密码，我们需要对用户密码进行加密操作。首先安装一个 npm 包 **bcrypt**:

```
npm install --save bcrypt
```

然后在 `models/user.js` 文件开头中导入 `bcrypt` 功能模块:

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

然后通过 Mongoose 的中间件 功能来控制用户数

据保存的流程，我们需要在数据保存之前执行一些操作，所以编写一个文档（document） save 函数的前置钩子（pre hook），如下所示：

```
UserSchema.pre('save', function(next) {  
  var user = this, SALT_FACTOR = 5;  
  bcrypt.genSalt(SALT_FACTOR, function(err,  
    if(err) return next(err);  
    bcrypt.hash(user.password, salt, function(  
      if (err) return next(err);  
      user.password = hash;  
      next();  
    });  
  });  
});
```

调用 bcrypt 提供的 API 加密用户密码。上述代码需要放置在 `module.exports = ...` 语句之前。

## 测试用户密码是否加密

打开 `index.js` 文件，更改一下测试用户的信息，代码如下：

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

然后通过 `mongo-express` 工具查看一下数据库中的数据，会发现数据库 `users` 集合中记录了用户 `billie` 的信息，并且密码经过了加密处理。

欢迎添加 Peter 的微信：happypeter1983

冀ICP备15007992号-3