Node.js入门和企业级项目开发

第5天课堂笔记

讲师：邵山欢

日期：2017年11月10日

# 一、后期课程的学习方法

2014年：只需要会简单的JavaScript基础，DOM操作，jQuery，HTML5、CSS3，就能找到工作。

2014年的夏天行业巨变：React、Vue这些MVVM框架，让很多C/S架构的项目变为B/S项目，网页项目变得越来越大。单页面应用开始火热。**前端开发工程师现在的主要工作变为开发Dashboard系统**。



如果跟不上了，一定先把现在阶段的课程学完，然后再复习之前的，可以选择留级。现在的前端，值得你用大半年的时间学习。

工作的时候，**90%以上的同学是搞React和Vue技术栈，写组件的**。

**NodeJS对于React和Vue项目的作用：提供有数据库支持的RESTful风格的API接口**。比如我们做一个《宠物店消费管理系统》，有会员管理功能、积分卡功能，能够记录每个主人有什么宠物、消费记录，此时就需要NodeJS和mongoose提供数据库的功能和一套RESTful API的接口。

现在说说React和Vue到底是干嘛的？

**传统的jQuery编程最麻烦的事情就是数据和DOM的一致性问题。数据变化了，视图也要写代码让它变化**。



能不能自动变化？？

能！Vue和React解决了这个问题：**Angular、Vue和React能够让数据变化的时候，DOM自动变化**。

# 用户信息更改页面的制作

## 2.1 用户信息的拉取接口

用户的基本信息：

* email（一旦注册，不能更改）
* 昵称（全站不能相同）
* 一句话简介（默认：这家伙很懒，什么都没有留下）
* 头像
* 密码（加密之后的密码）

只要是用户的基本信息，就要更改User这个schema。

var mongoose = require("mongoose");

var schema = new mongoose.Schema({

"email" : String , //email

"password" : String , //密码，加密之后的密码

"nickname" : String , //昵称

"introduction" : {

"type" : String,

"default" : "这家伙很懒，什么都没有留下"

}, //简介

"avatar" : String //数据库中不保存图片，只保存图片的文件名。

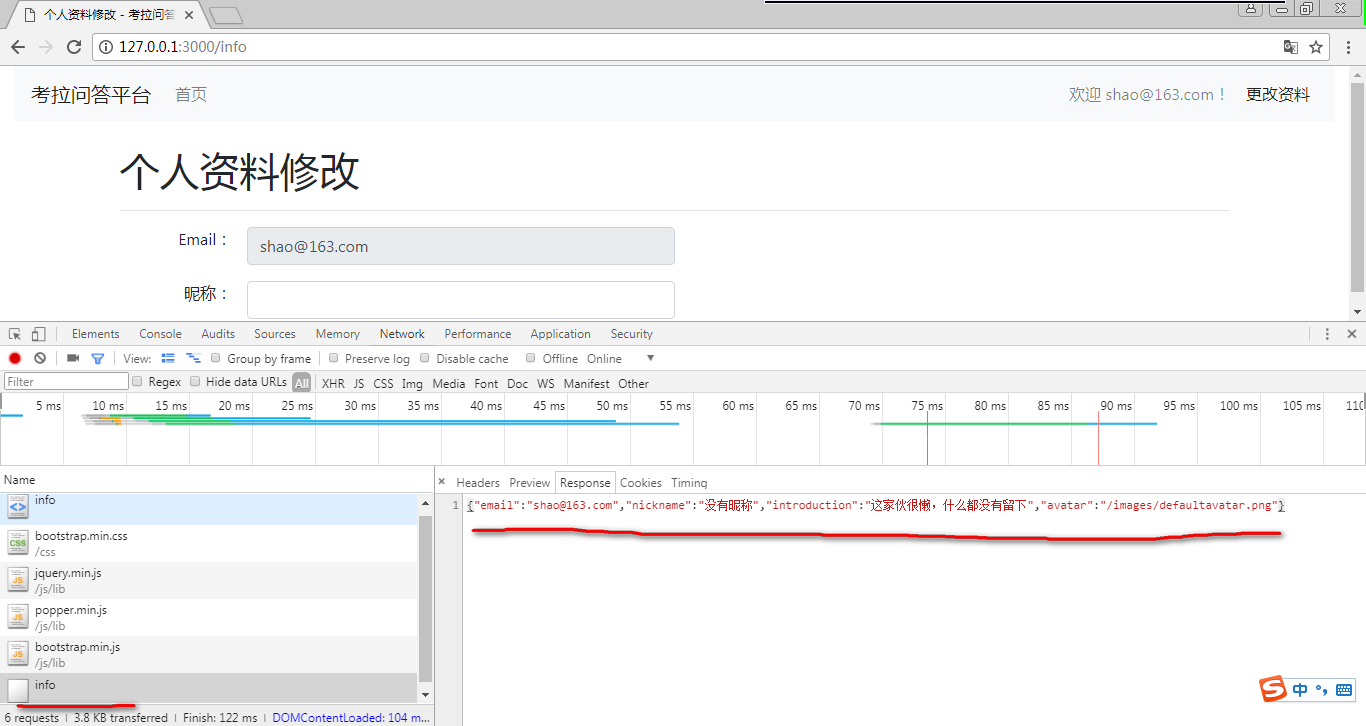
});

module.exports = mongoose.model("User" , schema);

个人资料页的修改，是建立在用户已经登录的情况下！

**所有的页面信息：涉及安全的内容，用模板引擎；如果普通信息，要用Ajax。**

**只要涉及Ajax，你的服务器必须要开一个GET请求的接口，这个接口可以读取用户的所有信息。接口：**



## 2.2 头像上传

头像的上传很简单，因为formidable天生支持文件的上传，用files来接收就行了。

图片的上传有两种形式：

* 同步上传 : 必须有form表单，用submit按钮来提交
* 异步上传 ： 用ajax来提交表单（实际上不是ajax，是猫腻，下午说）

这里使用同步上传，一个能够上传图片的表单，必须有enctype属性：

<form action="/uploadavatar" method="post" enctype="multipart/form-data">

<input type="file" **name="avatar"**/>

<input type="submit" />

</form>

后端的中间件基本不用写什么东西，只需要写uploadDir即可。

//处理上传

exports.uploadavatar = function(req,res){

//得到前端提交的表单信息

var form = new formidable.IncomingForm();

//设置上传文件夹

**form.uploadDir = path.resolve(\_\_dirname , "../uploads");**

//保留拓展名

**form.keepExtensions = true;**

form.parse(req , function(err , fields , files){

res.send("123");

});

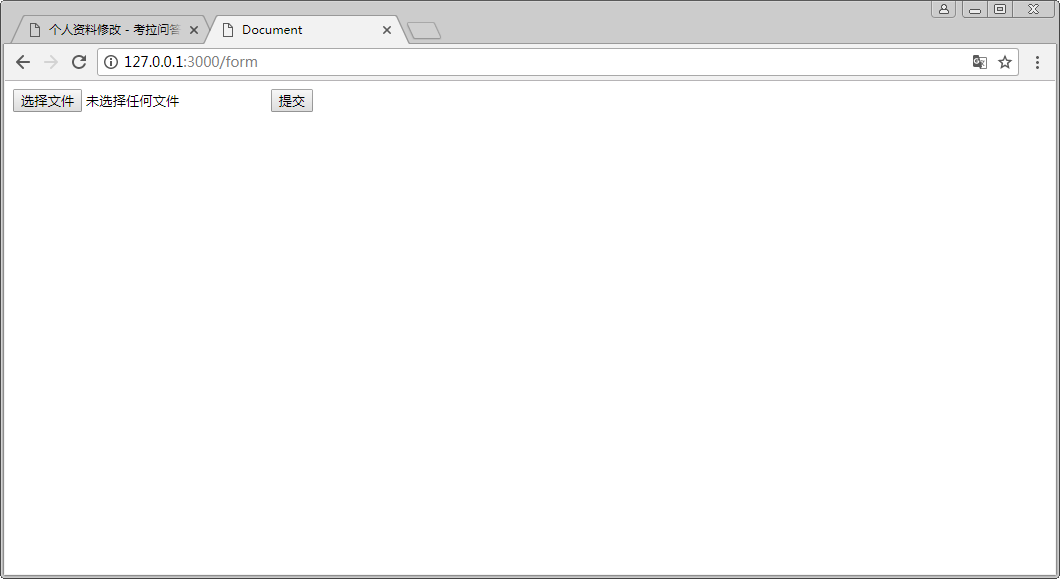
}

**我们的<form>标签有一个特点，当你点击submit按钮的时候，会自动跳转到提交到的那个页面去！**

**此时如何阻止跳转呢？可以内嵌一个iframe小窗口，小这个“小电视”中，呈递流程页面**。



所以我们单独做一个/form的路由，这个页面专门做一个表单：



把这个/form放到iframe中。这样的话，当form被submit的时候，大的URL不会跳转。跳转发生杂小电视里面。

bootstrap中很方便做一个弹出层。要有一个按钮：

<button **data-target="#avatarModal" data-toggle="modal"**>更改头像</button>

一个模态框的div

<div class="modal fade" **id="avatarModal"** tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalLabel">Modal title</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">...</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

<button type="button" class="btn btn-primary">Save changes</button>

</div>

</div>

</div>

</div>

## 2.3 头像的剪裁界面的开发

**如何知道用户上传的是什么图片呢？**是在formidable的files变量中。

配合url模块的parse()方法和正则表达式中的match方法，可以轻松提炼上传文件的文件名。

form.parse(req , function(err , fields , files){

//提炼出刚刚上传的图片的文件名

var pathname = url.parse(files.avatar.path).pathname.match(/\/(upload\_.+)$/)[1];

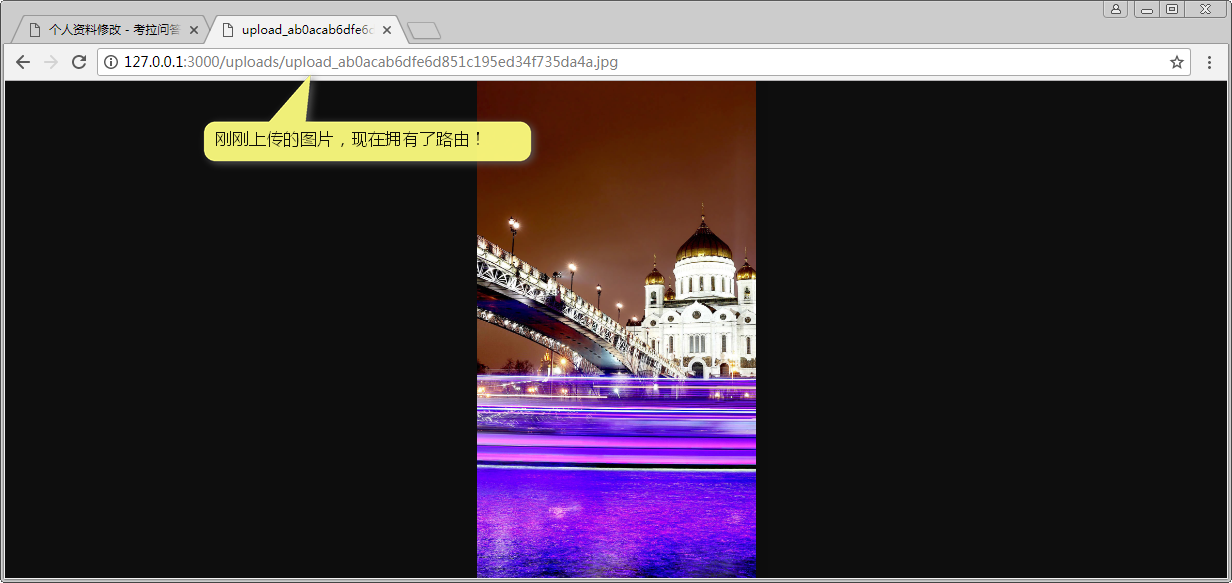
console.log(pathname);

});

开路由！将uploads文件夹也静态化出来。所以我们改变app.js文件，增加：

//静态化uploads文件夹

app.use("/uploads" , express.static("uploads"));



在裁切页面，怎么显示刚刚上传的图片呢？

当用户上传完毕图片之后，将文件名存储到session中，然后呈递cut.ejs模板，通过字典将图片的网址传入。

处理上传的时候存入session：

//处理上传

exports.uploadavatar = function(req,res){

//得到前端提交的表单信息

var form = new formidable.IncomingForm();

//设置上传文件夹

form.uploadDir = path.resolve(\_\_dirname , "../uploads");

//保留拓展名

form.keepExtensions = true;

form.parse(req , function(err , fields , files){

//提炼出刚刚上传的图片的文件名

var pathname = url.parse(files.avatar.path).pathname.match(/\/(upload\_.+)$/)[1];

//将这个文件夹名存入session！

**req.session.avatarurl = pathname;**

//跳转页面

res.redirect("/cut");

});

}

在裁切页面使用这个session:

//呈递裁切页面

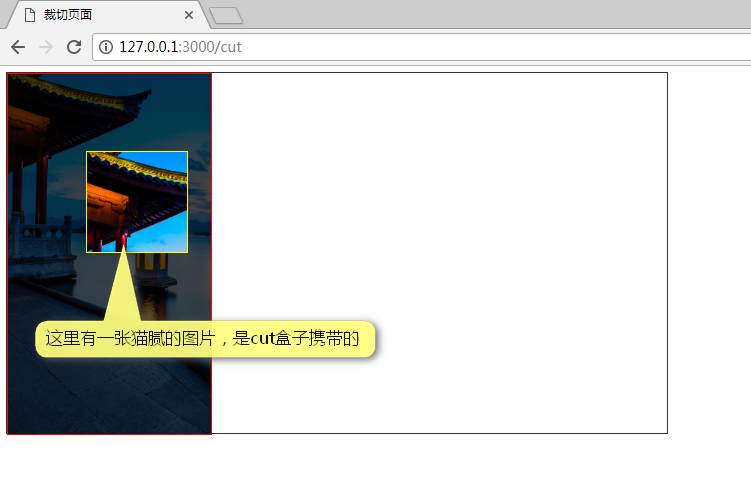
exports.showcut = function(req,res){

res.render("cut" , {

**"avatarurl" : req.session.avatarurl**

});

}



jQuery-ui有什么功能：

* Draggable ：拖拽
* Droppable : 拖放
* Resizable : 可更改尺寸
* Selectabel : 可被选择
* Sortable : 可更改顺序

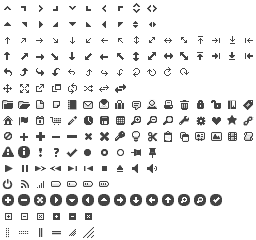
我们的cut小框框可以被更改尺寸。此时使用jquery-ui提供的：

$("#cut\_rect").resizable();

但是一定要注意，必须引用jquery-ui的样式表。

**<link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">**

css文件夹中必须有图片



jQuery中有一个trigger()的方法，可以模拟别的元素的事件。

## 2.4 使用gm裁剪图片

gm就是GraphicsMagick的缩写。

http://www.graphicsmagick.org/

开源的库真的很值得尊敬，比如割绳子、愤怒的小鸟都是依靠box2d这个库。

可以看这个演示：http://yuehaowang.github.io/demo/box2d\_linkage/

安装软件，

下一步、下一步无脑安装。

**然后将你的安装目录C:\Program Files\GraphicsMagick-1.3.22-Q16设置为环境变量！**

设置环境变量成功之后，打开CMD就能使用gm命令了

gm -version



命令行里面就能裁剪图片：

gm convert -crop 100x100+50+60 1.jpg 3.jpg

node.js如何操作它？此时就要安装npm包gm。

安装依赖：

cnpm install --save gm

程序中引包：

var gm = require('gm');

使用：

gm(avatarurl).crop(w,h,x,y).write(avatarurl, function (err) {

console.log("裁剪成功！");

});

**iframe内嵌的页面可以用js调用外部的DOM；但是反之不行。**

调用方法：

$("#avatarModal" , **window.parent.document**)