# 项目分析

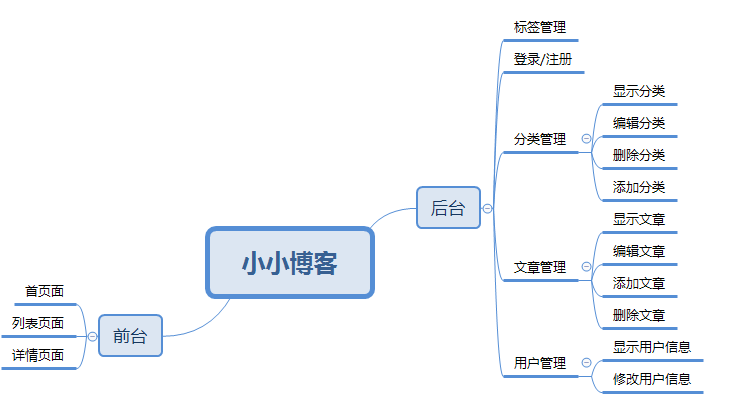
## 做什么

小博客

后台：管理员可以进入到后台中，发表文章，普通用户是不能进入。

前台：主要提供浏览功能，对所有的用户开放。

大家可以了解一下思维导图工具：xmind 工作中可能会用到。

功能如下：  


## 技术选型

后端：Node.js + Express + MongoDB + EJS

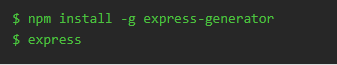
前端：Bootstrap + HTML5 + CSS3

使用服务端渲染，有利于SEO。

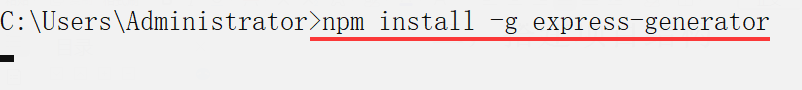
# 搭建项目结构

## 使用epress脚手架创建项目结构

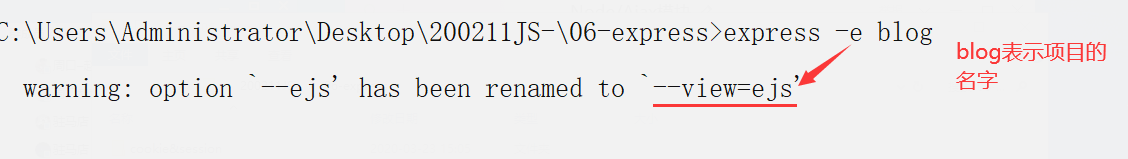
全局安装express:



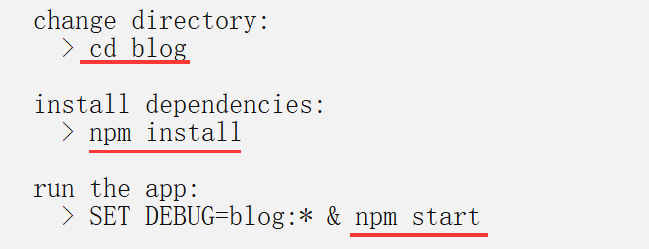
安装如下：



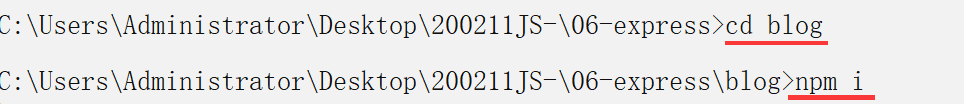
安装OK后，就可以使用脚手架创建项目，如下：

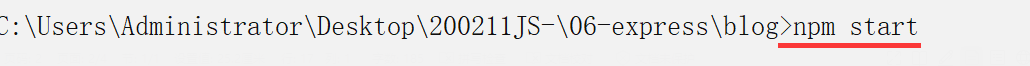


根据提示：

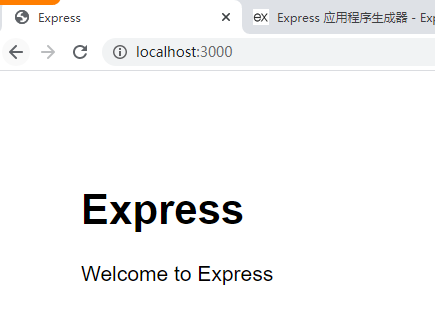


操作如下：





端口默认是3000，通过浏览器访问如下：



通过脚手架创建的项目 目录结构如下：

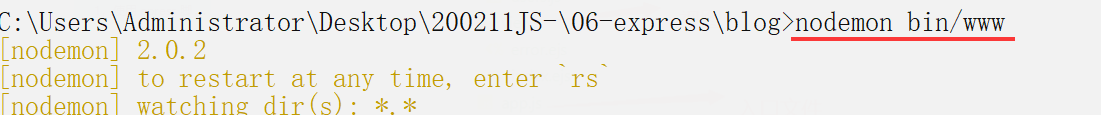


启动项目：

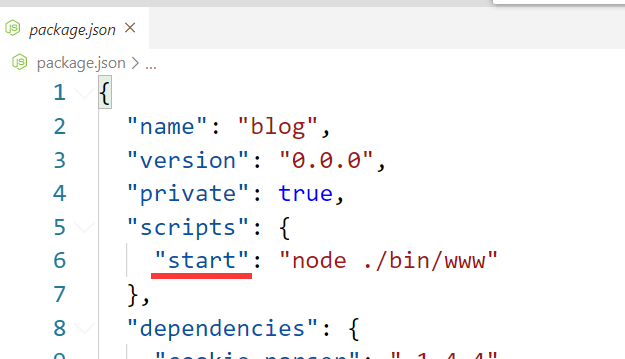
1. node bin/www nodemon bin/www

2）npm run start

如下：



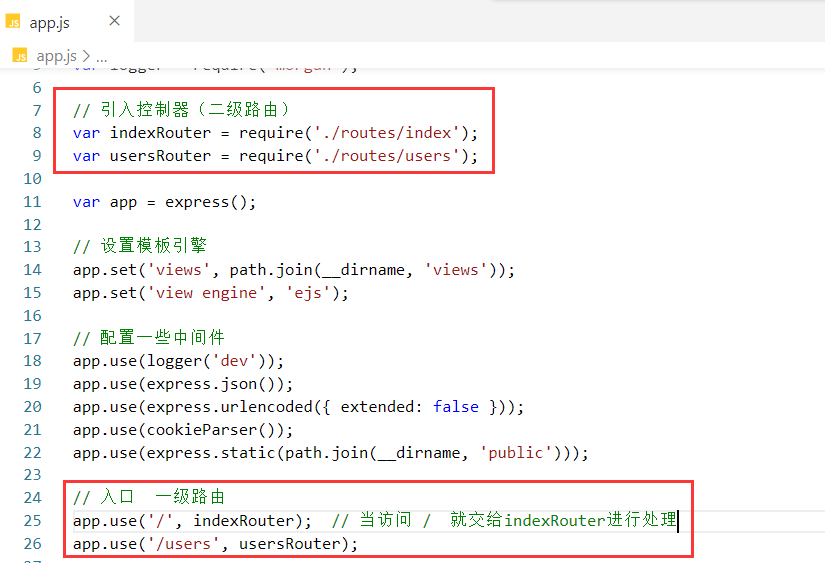
脚手架在package.json中配置好了一个命令，如下：



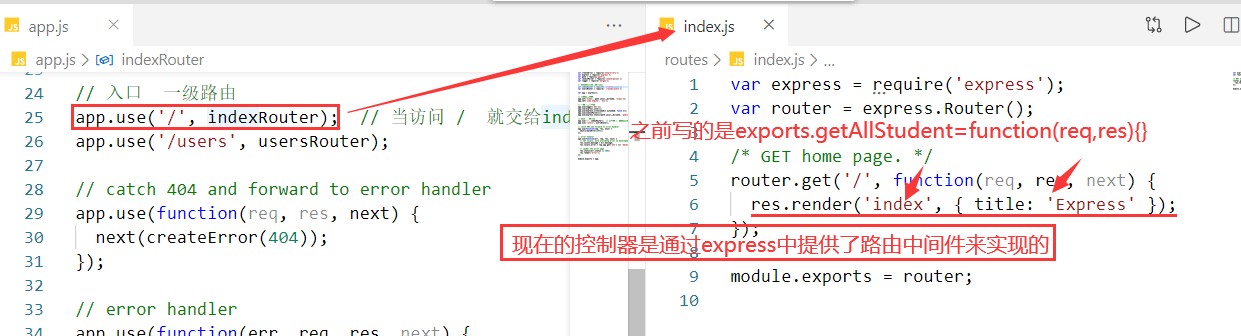
运行命令：npm run start 其中run可以不写 直接npm start

## 快速分析通过脚手架创建出来的项目

**入口分析：**



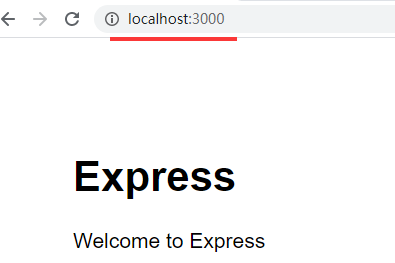
看控制器：



看视图：



浏览器得到的是服务端渲染好的有真实数据的页面，如下：



上面的逻辑是没有Models参与的。因为还没有进行数据库的操作。

为了让大家感受到MVC的经典思想，修改文件夹名字，如下：



到此，分析流程就OK了。

express脚手架，并没有帮我们做太多的事情，但是后面讲vue react时，它们也有自己的脚手架，它们的脚手架帮我们做了非常多的事件。

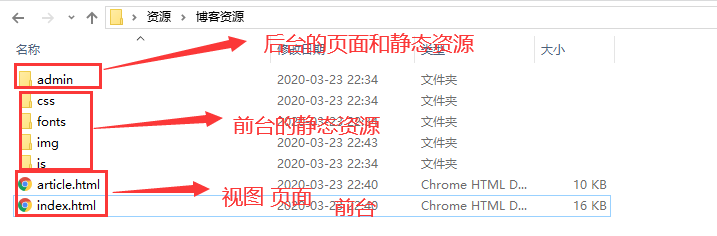
## 3，渲染前台首页面

现在我们要拿到前端程序员写好的页面，前端程序员只需要写好页面交给后端就OK了，我们会通过服务端渲染，把渲染好的页面扔给浏览器就OK了，也就是说，前端程序员写的页面，也是跑在我们同一个域名，同一个端口下，换句话说，此时，是没有跨域问题的。

什么时候有跨域问题，前端它的页面没有交给后端，它自己放到的一个服务器，然后，通过ajax向后端要数据，此时前端和页面和后端的服务不在同一个服务器上面，此时就有跨域问题了，你要知道跨域问题，就由于浏览器造成的，浏览器了为安全，浏览器不让你访问不同域名下面的资源。所以说产生域名条件：1）浏览器 2）Ajax 3）前后端代码不在同一台服务器。后面学习vue， react ，那个时候，写项目是前后端分离开发，前端是一个服务器，后端也是一个服务器，此时你的前端使用ajax要后端要数据，就会有跨域问题。

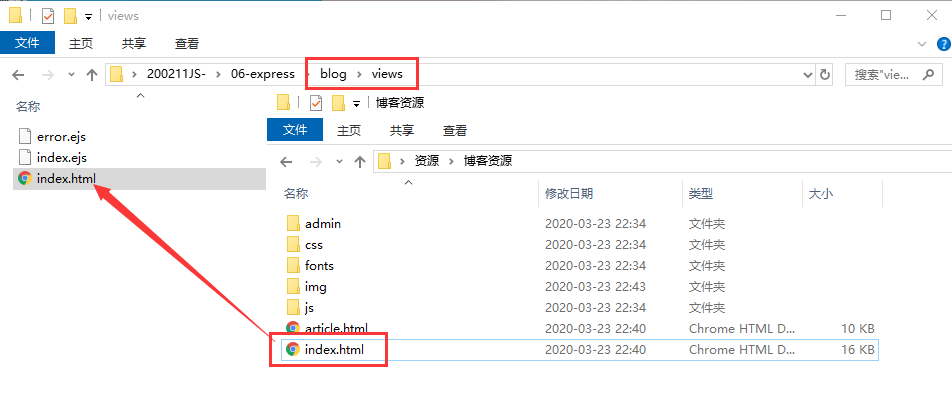
如何解决跨域问题？--- 后面讲~

得到前端程序员写好的页面，如下：

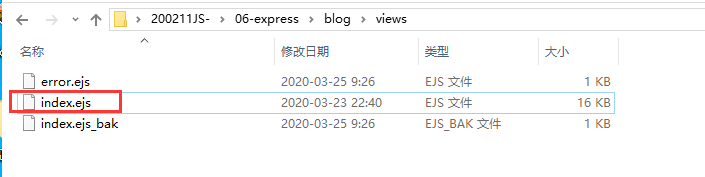


先载入前台首页面，就是index.html页面，步骤如下：

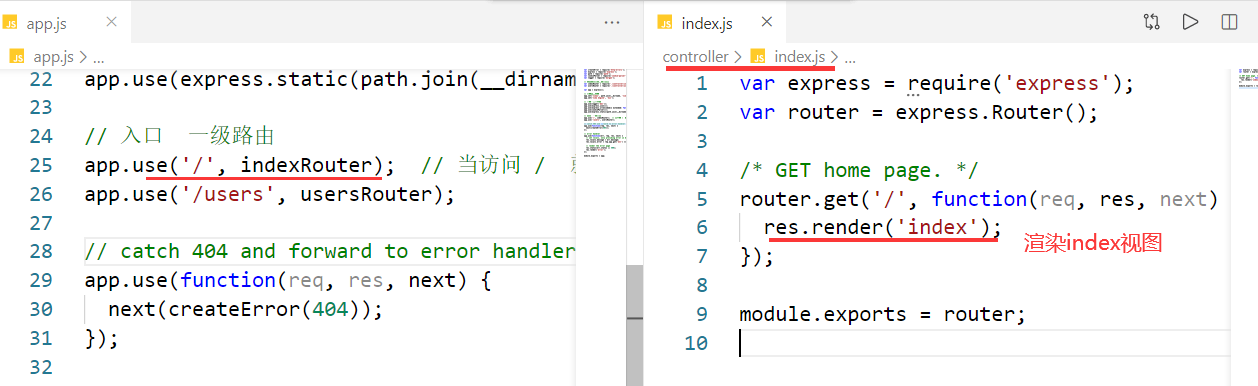
1）把前台的首页面放到blog项目下面的views目录下面



2）把名字改成ejs，如下：



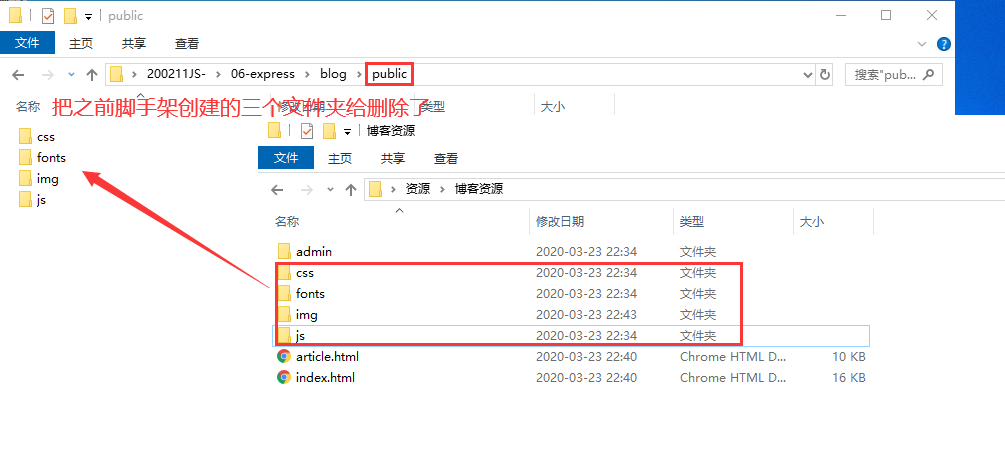
3）前面我们已经分析过了页面渲染流程了：



1. 访问之 如下：



上面的之所以没有出现样式，是由于我们没有托管静态资源，我们托管静态资源，如下：



再次访问之，如下：



为什么ok ?

访问静态资源，应该是按下面的方式进行访问，如下：

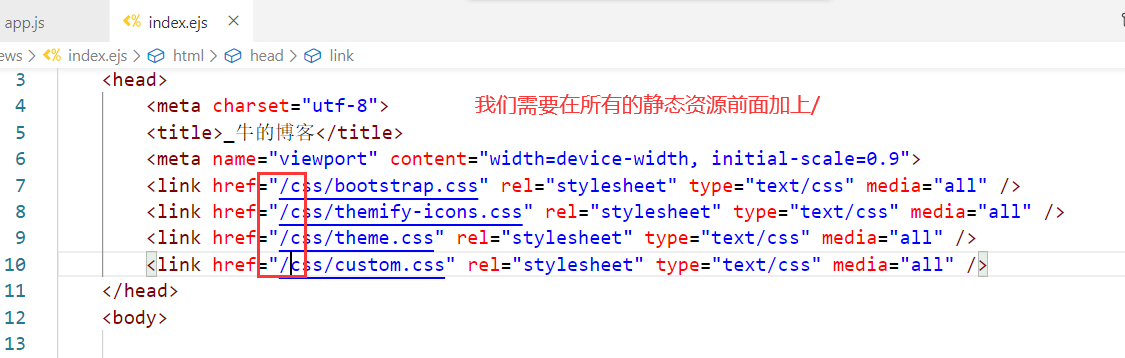


但是你看我们的代码如下：

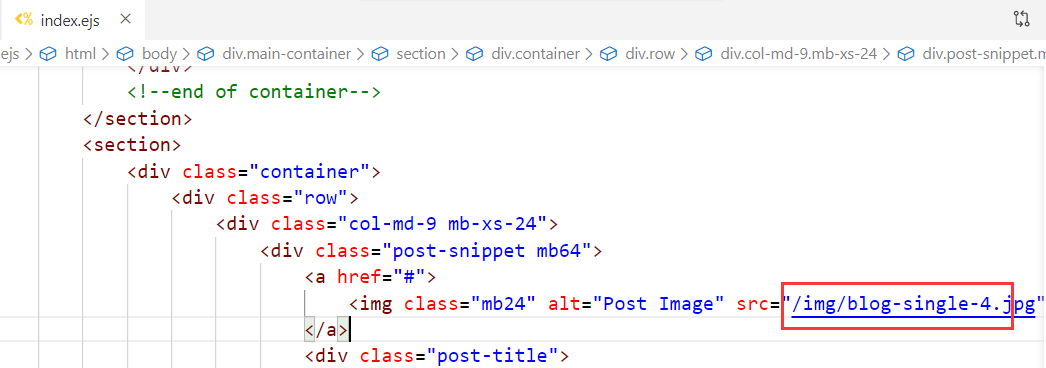


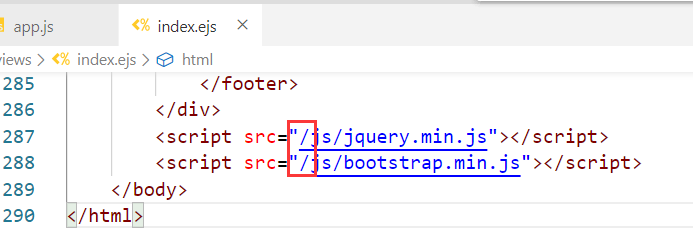
**没有加/ 为什么刚才的样式起作用了？**

答：浏览器有时候会给你加上去，但是有时候不给你加/ 所以说，虽然我们刚才访问时有样式了，但是为了保险起见，我们都以/打头。



**一定要看整个页面，都要加上/ 。**



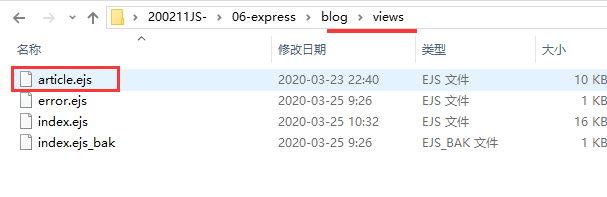


到此，前台首页面就被渲染了，如下：



## 4，载入前台的详情页面

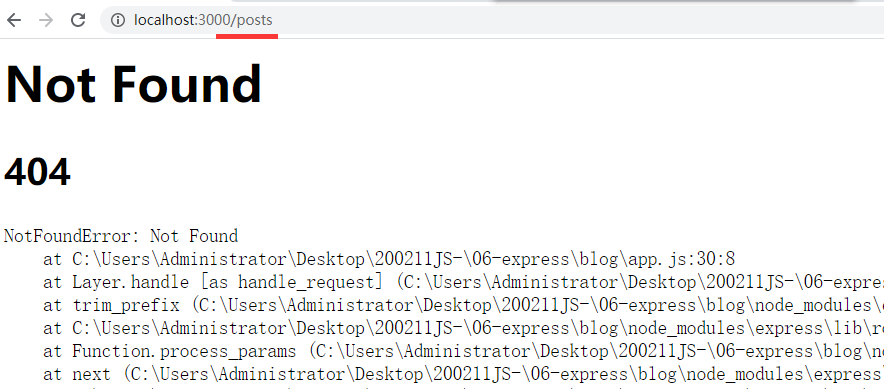
同理，和上面的一样，载入前台的详情页面，如下：



赶紧在页面中 把使用到 静态资源 的位置加上/，如下：

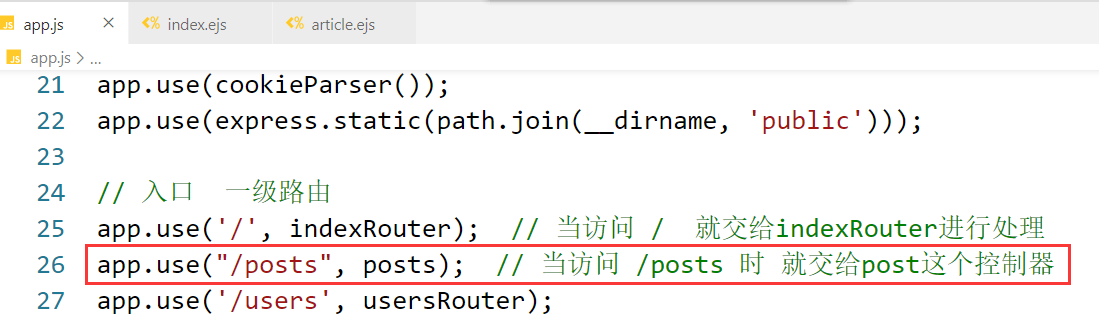


规划路由，当我们访问http://locahost:3000/posts时，我们想访问到文章的详情页面，但是访问效果如下：

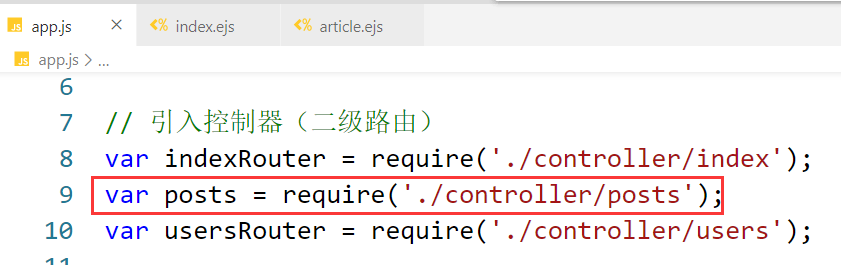


我们应该？ 肯定写路由，写控制器，控制器去渲染文章详情页面，步骤如下：

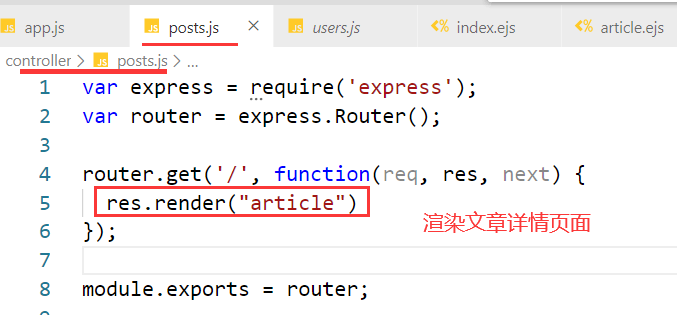
1. 在入口文件中写一级路由，如下：



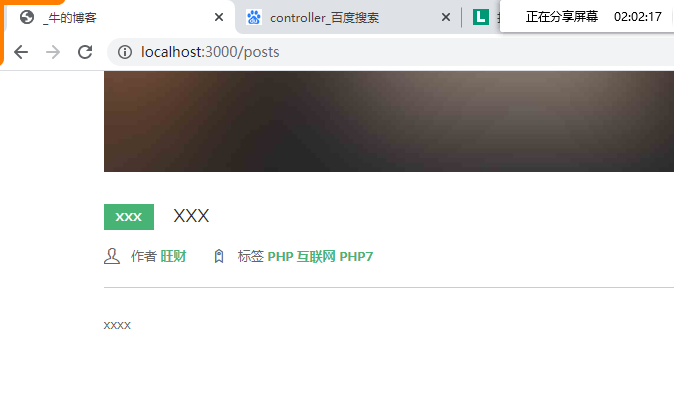
1. app.js导入这个posts控制器



1. 创建post控制器，如下 ：



1. 访问之，如下：



到此，文章详情页面就渲染OK了~

# 载入后台页面

## 1，分析

然后，载入后台的页面，思路和前面是一样的。如下：

1. 把Html文件放到views下面，改后缀为ejs
2. 将所有的后台的静态资源，放到public下面
3. 针对后台不同的视图，写不同的路由

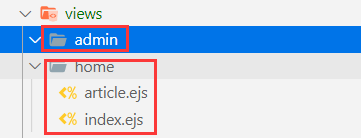
但是如果这样做，你有没有觉得有点小问题？

不足之处：把前台的视图和后台的视图 把前台的静态资源和后台的静态资源都混到一块了，后期肯定不利于维护，不利于二次开发。所以我们要进行前后台的分离。

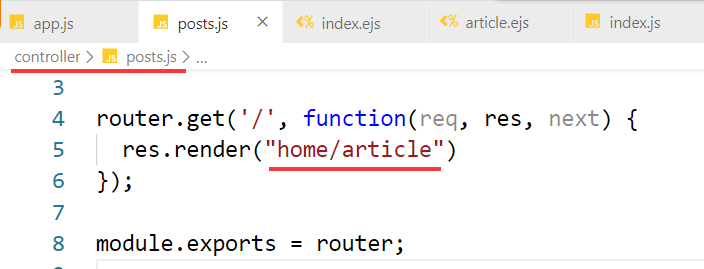
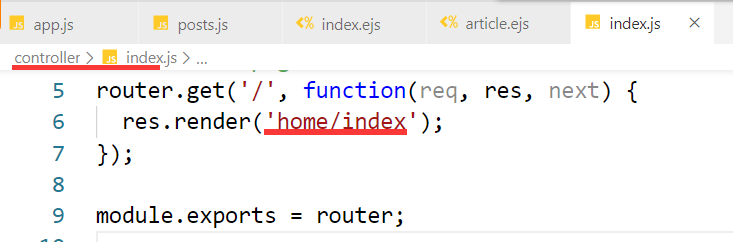
我们做的，就是创建不同的文件夹，把它们放到不同文件夹中，如在views下面，创建两个文件夹，一个是admin 一个是home。大家都这样命名。

* admin: 表示后台
* home：表示前台

操作如下：



把前台的视图放到home中，我们需要去改变渲染它的路径，如下：



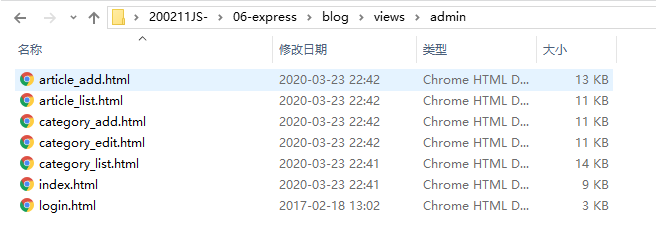
到此，先不要着急往后写，先测试一下，上面的改完后，是否可以访问，如下：



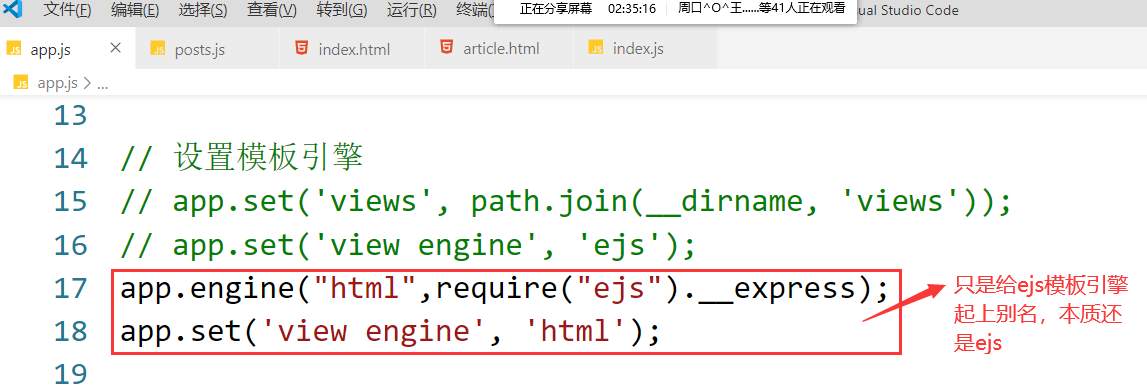
OK了，接着再向后写。

## 2，载入后台的首页面

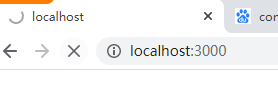
把后台的页面，放到views下面的admin文件夹下面，如下：

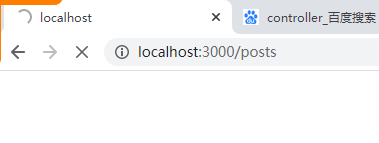


按理说，需要把后缀都改成ejs，页面可能非常多，改成ejs也不方便，所以说，在ejs中可以指定文件后缀，如下：

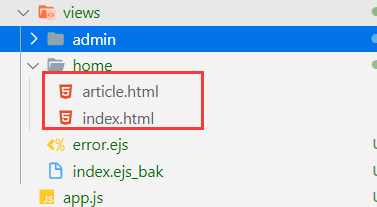


再次访问前台页面，如下：





需要把之前的改成ejs后缀的视图重新改成html，如下：



测试之，如下：



接着，就是实现载入后台页面。但是后台页面非常多，载入时，先把路由规划好了，如下：

前台路由：

[http://localhost:3000/](http://localhost:3000/posts)  前台首页面

<http://localhost:3000/posts> 前台文章详情页面

后台路由：

<http://localhost:3000/admin/> 后台首页面

<http://localhost:3000/admin/>cats 后台显示分类页面

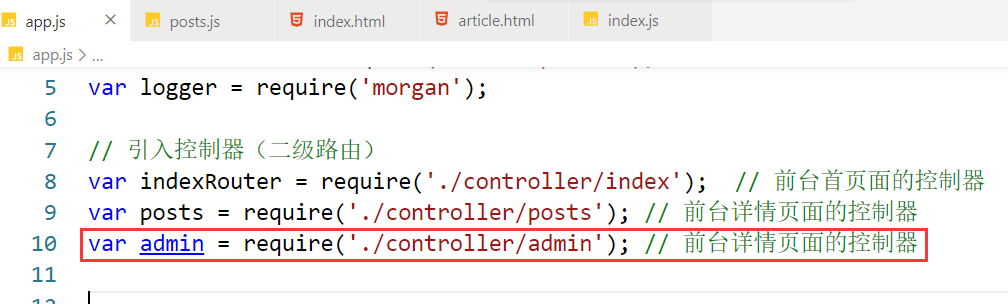
<http://localhost:3000/admin/>cats/add 后台添加分类页面

<http://localhost:3000/admin/>posts 后台显示文章页面

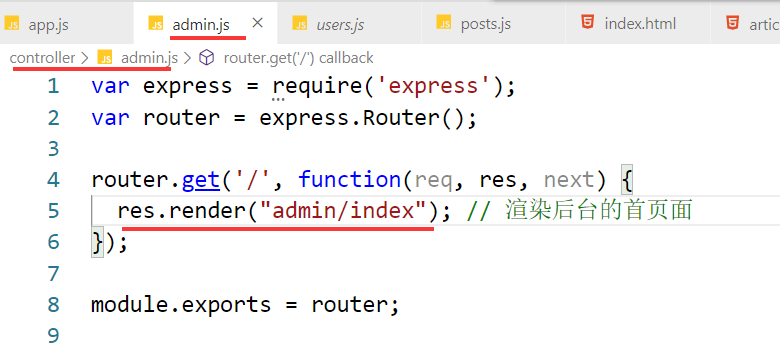
<http://localhost:3000/admin/>posts/add 后台添加文章页面

在app.js中，创建一级路由，如下：



引入admin这个控制器，如下：  


接着创建这个控制器，如下：



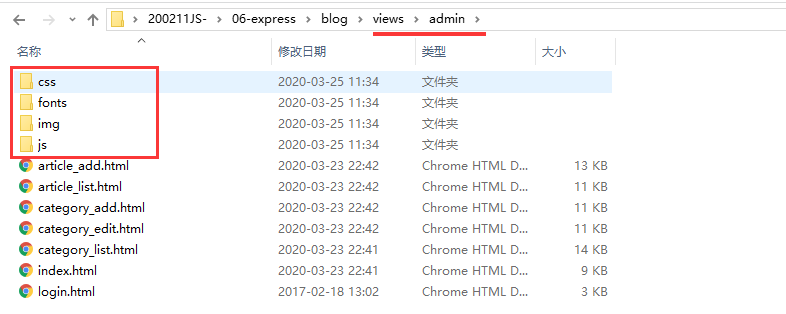
访问之，如下：



原因没有样式，静态资源没有进行托管，此时我们就进行托管了，如何托管，方案如下：

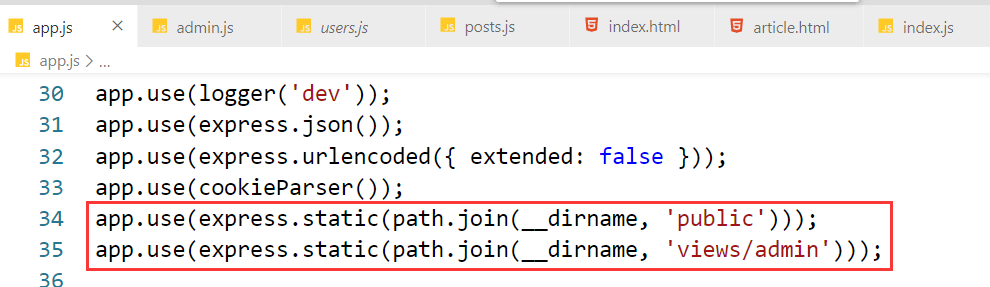
方案一：在public下面，创建两个文件并，一个是admin，一个是home。把前台的静态资源放到的Home中，把后台的静态资源放到amdmin中。这个方案肯定是OK了。

方案二：把静态资源放到views下面的， 这种方案用的也比较多

我们就使用方案二，如下：  


以前托管静态资源是托管的Public目录，现在你把静态资源放到views下面的amdin文件夹下了，肯定是托管不了的，如何解决？

答：把views下面的admin也托管了，如下：



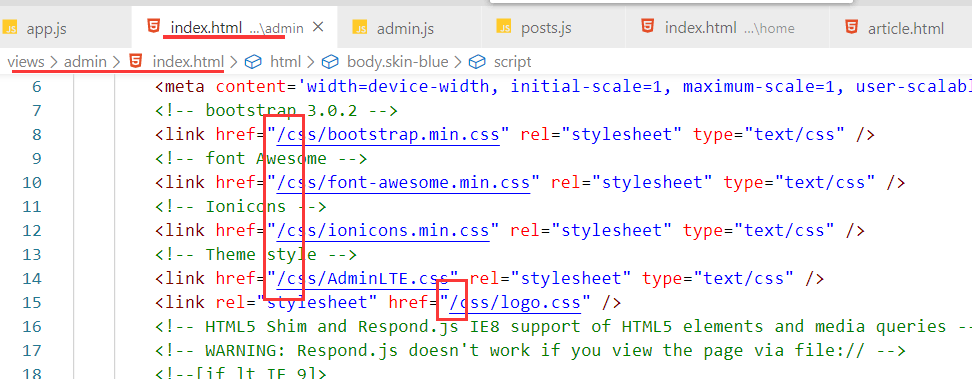
访问之，如下：



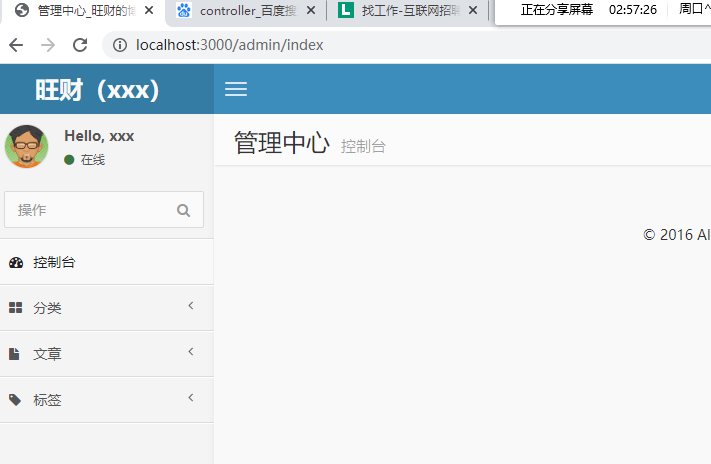
**还是不行，为什么？**

需要加/

加/ 如下：



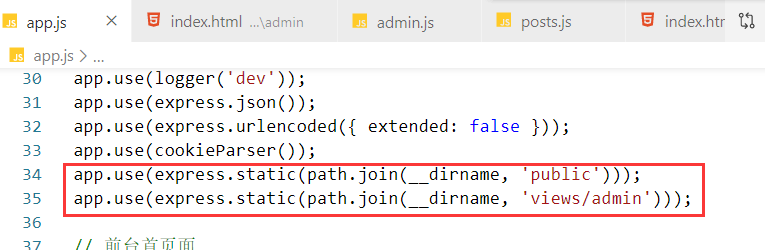
访问之，如下：



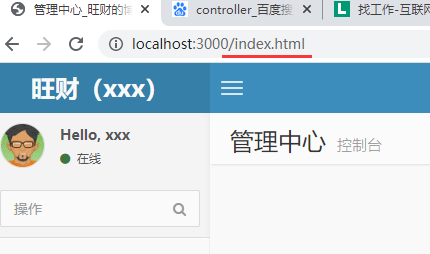
上面的访问后台的首页面，就OK了，但是此时你再去访问前台的首页面，如下：



原因是我们托管静态资源时，写代码如下：

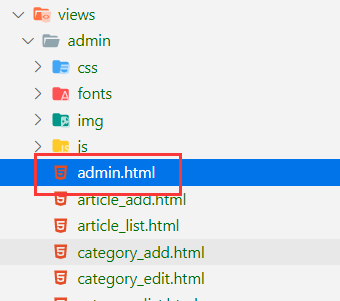


它托管了，amdin，也就意味着，你admin下面的文件都可以访问了，我们不写文件名，默认它会访问index.html，以说，访问如下：

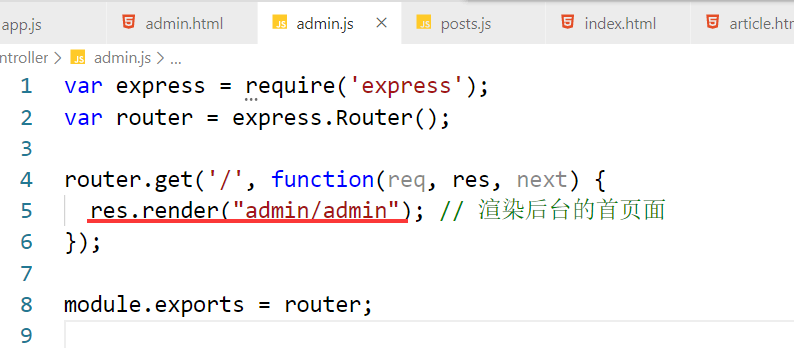


此时访问到的静态资源 的index.html 不是通过控制器渲染的前台的首页面，就没有走控制器。

解决：只需要把后台的index.html 换个名字 叫admin.html



还要告诉控制器，你渲染时，还是渲染admin.html，如下：“



访问之，OK OK OK ~

## 载入分类管理的页面

同现，我们要渲染分类管理页面和文章管理页面，路由规划，如下：

<http://localhost:3000/admin/>cats 后台显示分类页面

<http://localhost:3000/admin/>cats/add 后台添加分类页面

<http://localhost:3000/admin/>posts 后台显示文章页面

<http://localhost:3000/admin/>posts/add 后台添加文章页面

## 载入文章管理的页面

# 实现分类的增删改查

# 实现文章的增删改果

# 后台的登录控制

# 前台渲染数据