Vue2+Vue-router+Axios 斗鱼项目

[1. 初始化项目 3](#_Toc6260)

[2.克隆到本地 3](#_Toc31256)

[3.切换到工作目录 3](#_Toc10510)

[4.初始化vue项目或者使用第一次初始化好的项目 3](#_Toc16378)

[5.安装依赖 4](#_Toc29054)

[6.导入assets资源 4](#_Toc5381)

[7.下载lib-flexible插件 4](#_Toc29035)

[8.引入lib-flexible 5](#_Toc31161)

[9.下载axios插件 5](#_Toc28920)

[10.安装axios 5](#_Toc24754)

[11.跑起来 5](#_Toc16209)

[12.发现报错 5](#_Toc20345)

[13.打开浏览器预览 5](#_Toc15014)

[14.配置一下API的映射 6](#_Toc1827)

[15.修改了环境配置，需要重新启动 6](#_Toc5326)

[16.打开App.vue，写一个请求测试一下API是否能够访问 6](#_Toc31455)

[17.修改后保存，打开浏览器测试 6](#_Toc8741)

[18.清理默认项目中的文件 7](#_Toc19411)

[19.src目录里面新建页面目录pages 7](#_Toc11382)

[20.新建Home.vue页面组件 7](#_Toc31994)

[21.修改根路由 8](#_Toc20954)

[22.src目录下新建一个bus.js 8](#_Toc26339)

[23.在App.vue中注册一个自定义事件showSide 9](#_Toc30716)

[24.实现一个侧边栏导航（无代码） 10](#_Toc29316)

[25.新建SideMenu组件 10](#_Toc24104)

[26.新建头部组件AppHeader 11](#_Toc7031)

[27.新建Loading组件 12](#_Toc27907)

[28.新建public.js导出一个对象 13](#_Toc32100)

[29.实现Home头部 14](#_Toc6374)

[30.App.vue中使用SideMenu组件 14](#_Toc3422)

[31.到目前为止公共头部，加载组件，侧边栏菜单就已经出来了，效果如下： 16](#_Toc5169)

[32. 全局添加两个过滤器 17](#_Toc26488)

[34. Home加载列表 19](#_Toc14608)

[35. 到目前为止首页内容可以显示出来了 21](#_Toc6641)

[36. 添加回到顶部组件 21](#_Toc3717)

[37. 使用BackTop组件 23](#_Toc25234)

[38.查看返回顶部效果 24](#_Toc10940)

[39. 新建GameList组件 24](#_Toc14397)

[40. 新建后退组件 25](#_Toc1259)

[41. 新建分类页面 26](#_Toc2496)

[42. 把分类页面添加到路由中 28](#_Toc6992)

[43. 新建RoomList页面 29](#_Toc9316)

[44. 添加对应的路由 32](#_Toc13838)

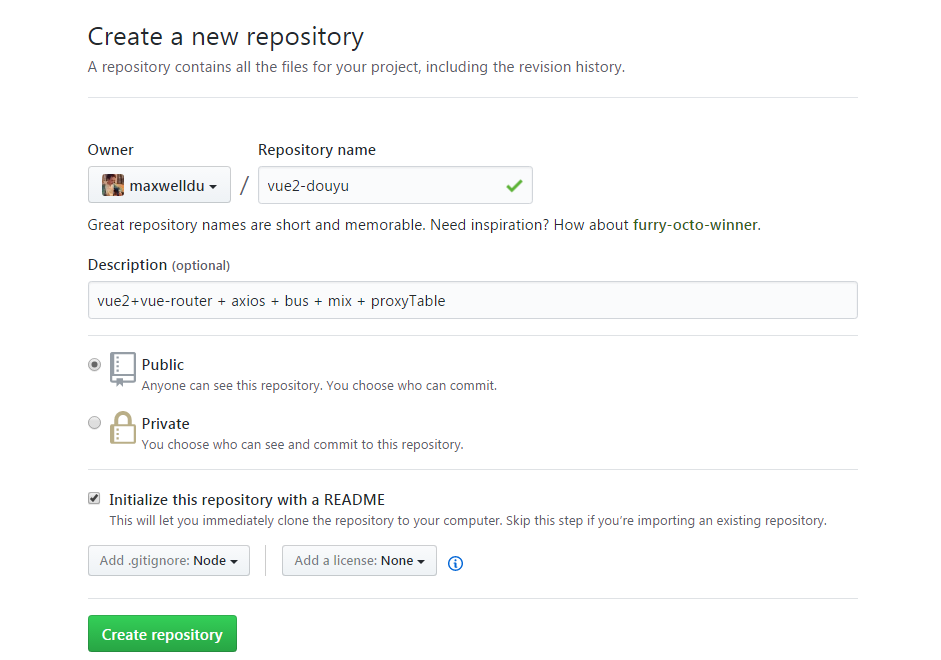
[45. 房间列表效果 32](#_Toc2001)

[46. 添加图片懒加载功能 33](#_Toc5619)

## 1. 初始化项目

在做一个项目之前我们先在Github上面创建一个项目，这样便于团队合作

这里我们新建一个项目，写上描述，公开，初始化README.md, gitignore选择Node, 然后点击创建



## 2.克隆到本地

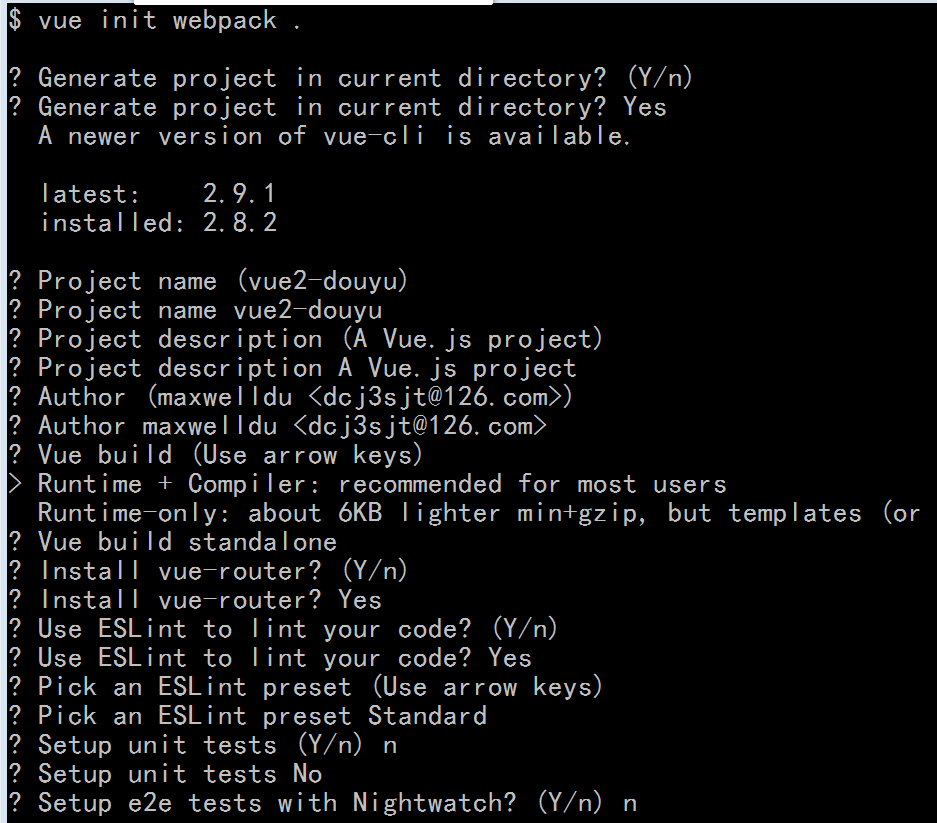
git clone 仓库地址

## 3.切换到工作目录

cd vue2-douyu

## 4.初始化vue项目或者使用第一次初始化好的项目

vue init webpack .



初始化好了会卡住，按Ctrl+C 退出

## 5.安装依赖

npm i 或者 yarn

## 6.导入assets资源

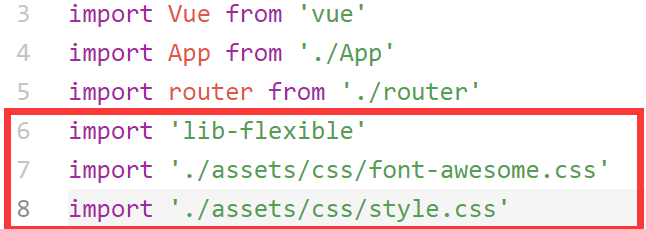
放到src/assets目录中

## 7.下载lib-flexible插件

npm i -S lib-flexible

## 8.引入lib-flexible

在 src/main.js中

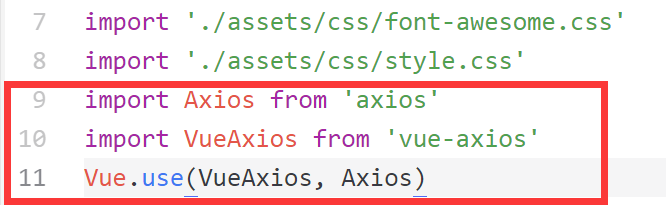


## 9.下载axios插件

npm i -S axios vue-axios

## 10.安装axios

在main.js中加入这三行:



## 11.跑起来

npm run dev

## 12.发现报错

npm i 或者 yarn 重新装一下，然后再启动

## 13.打开浏览器预览

localhost:8080查看效果

## 14.配置一下API的映射

config/index.js中的proxyTable:



## 15.修改了环境配置，需要重新启动

Ctrl+C 然后 npm run dev

## 16.打开App.vue，写一个请求测试一下API是否能够访问



## 17.修改后保存，打开浏览器测试

localhost:8080 打开控制台查看是否能够输入一个数组对象，如果可以表示成功请求，否则尝试重启和修改配置

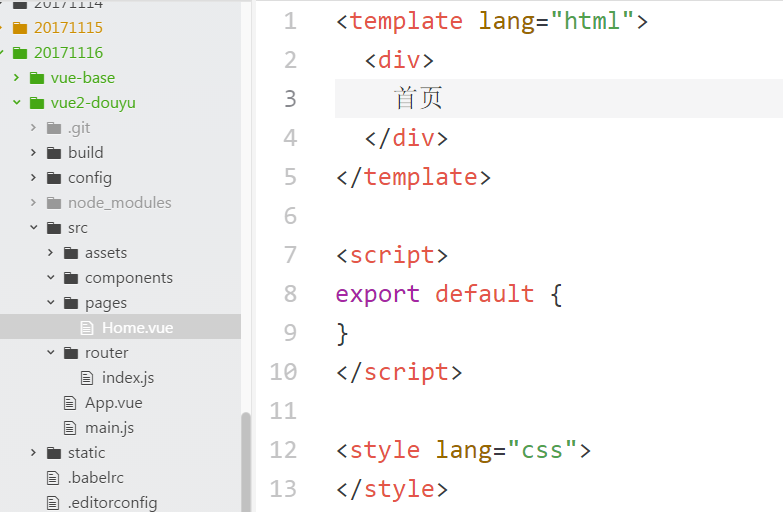
## 18.清理默认项目中的文件

删除src/components/HelloWorld.vue

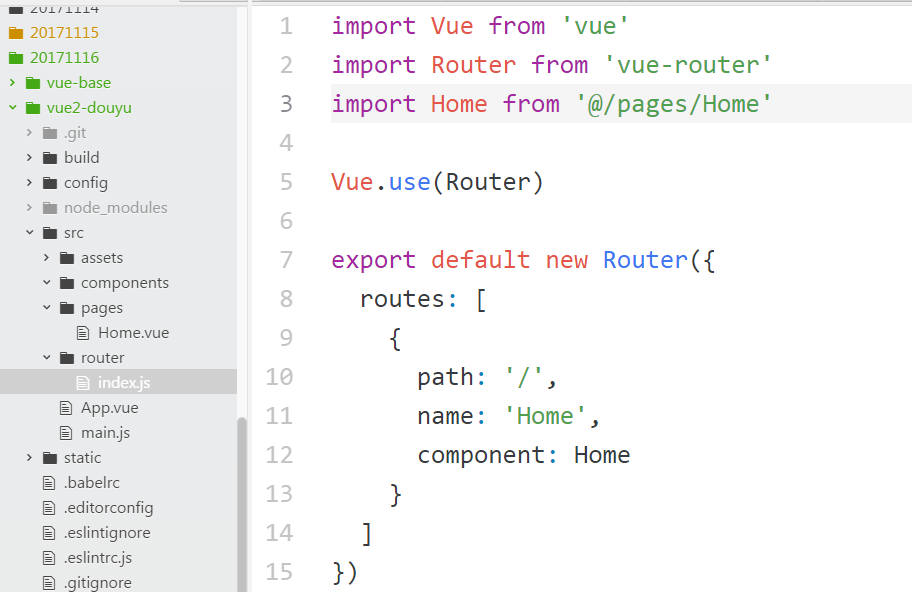
此时会报错

## 19.src目录里面新建页面目录pages

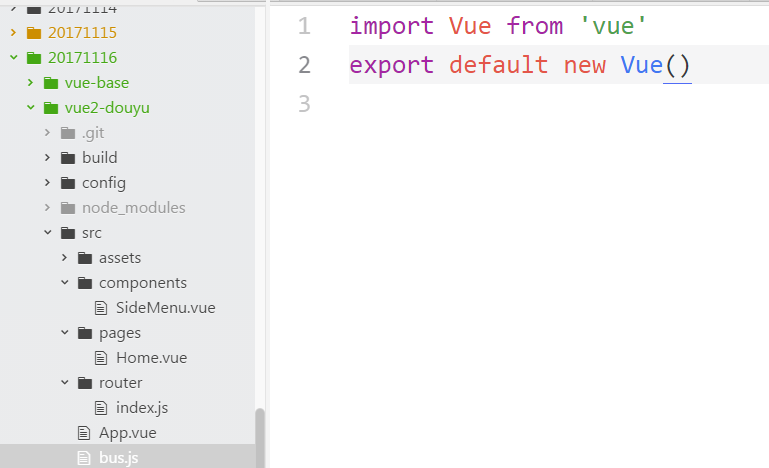
## 20.新建Home.vue页面组件



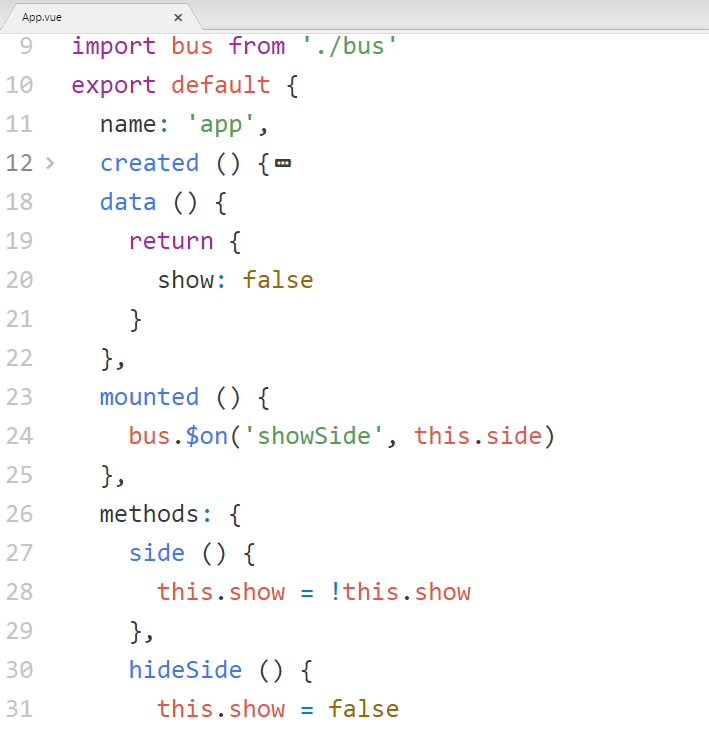
## 21.修改根路由



## 22.src目录下新建一个bus.js



## 23.在App.vue中注册一个自定义事件showSide



## 24.实现一个侧边栏导航（无代码）

## 25.新建SideMenu组件





<style lang="css">

.side-menu {

background: rgba(10,10,10,.3);

height: 100%;

position: fixed;

width: 100%;

top: 0;

padding-top: 44px;

z-index: 11;

}

.side-menu ul {

width: 70%;

background: #282828;

height: 100%;

border-top: 1px solid #222;

}

.side-menu ul li {

height: 50px;

border-bottom: 1px dotted #333;

font-size: 14px;

line-height: 50px;

padding: 0 30px 0 20px;

color: #9a9a9a;

}

.side-menu ul li i {

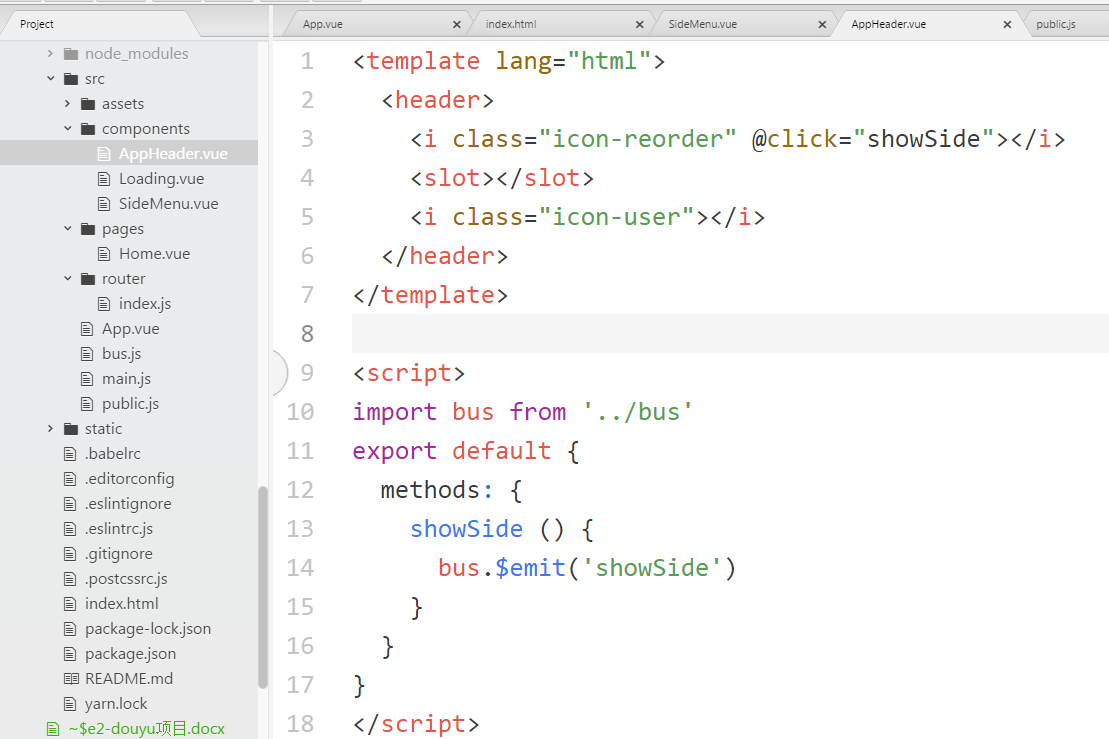
float: right;

line-height: 50px;

}

</style>

## 26.新建头部组件AppHeader



<style lang="css" scoped>

header {

height: 44px;

background: #333;

position: fixed;

left: 0;

right: 0;

top: 0;

z-index: 100;

padding: 0 15px;

color: #fff;

line-height: 44px;

font-size: 16px;

}

header i {

color: #999;

}

.title {

margin-left: 15px;

display: inline-block;

}

.icon-user {

float: right;

line-height: 44px;

}

</style>

## 27.新建Loading组件



<style lang="css">

.loading {

height: 100%;

position: fixed;

z-index: 10;

width: 100%;

background: #062734;

opacity: .4;

}

.loading img {

width: 100%;

height: auto;

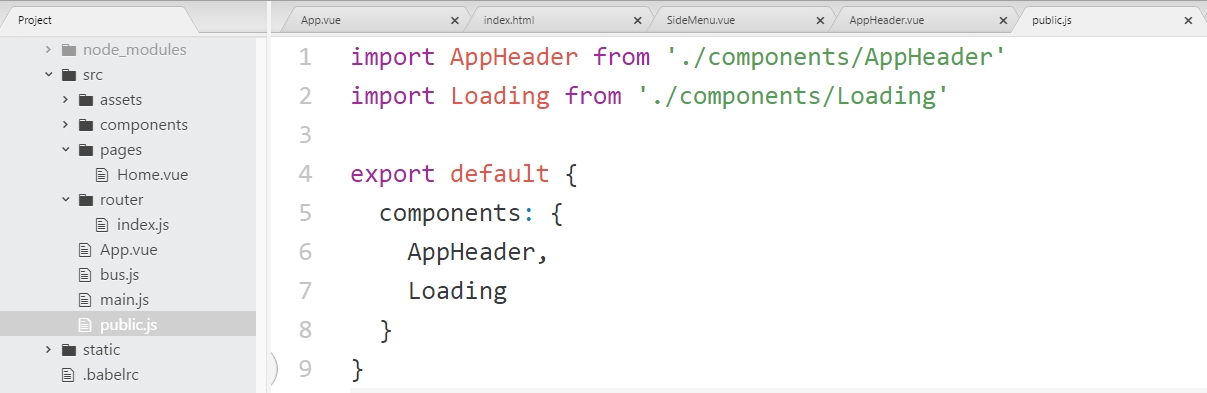
position: absolute;

top: calc(50% - 140px);

}

</style>

## 28.新建public.js导出一个对象



## 29.实现Home头部

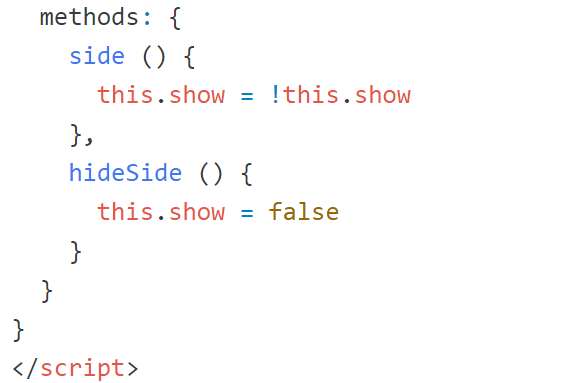




## 30.App.vue中使用SideMenu组件







<style>

.side-enter-active,.side-leave-active,

.side-enter-active ul,.side-leave-active ul{

transition:all .4s linear;

}

.side-enter,.side-leave-active{

opacity: 0;

}

.side-enter ul,.side-leave-active ul{

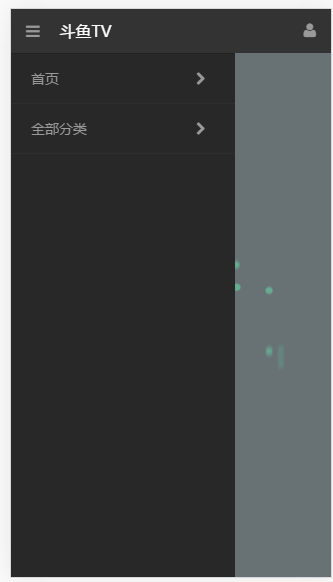
transform:translateX(-50%);

opacity:0;

}

</style>

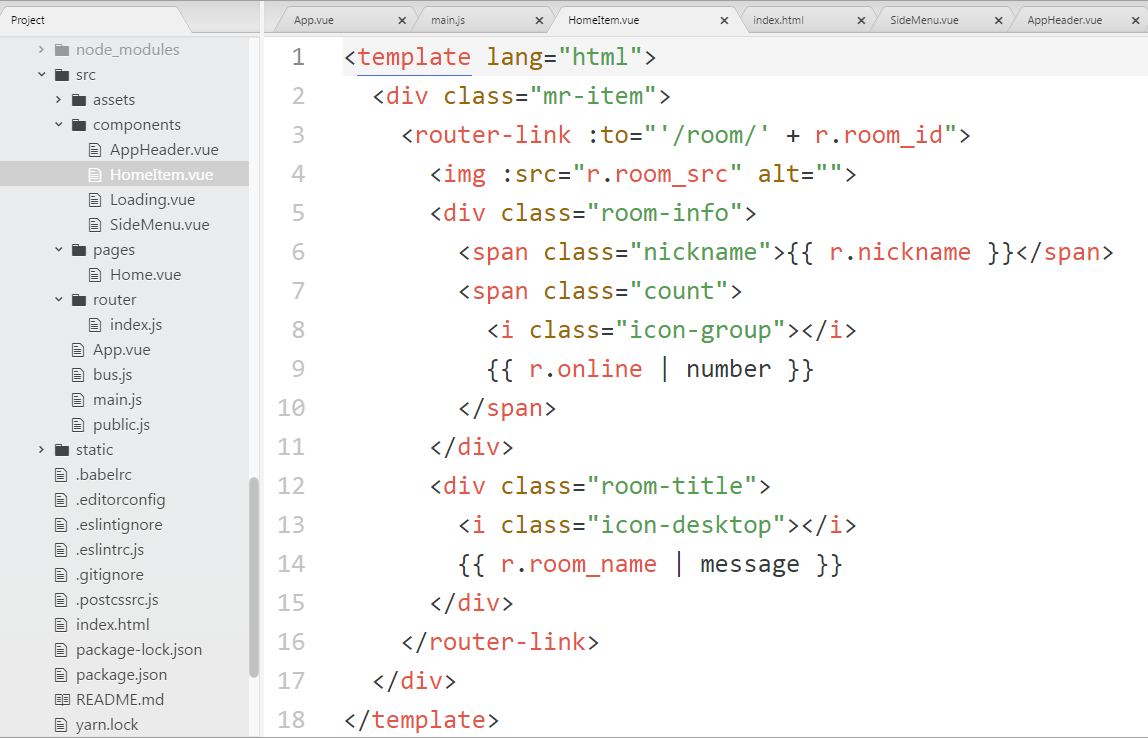
## 31.到目前为止公共头部，加载组件，侧边栏菜单就已经出来了，效果如下：

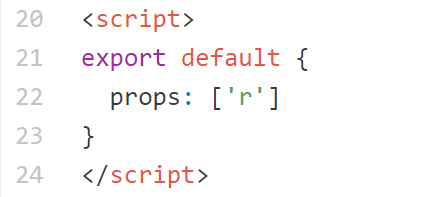


## 全局添加两个过滤器



1. 新建HomeItem组件





<style lang="css">

.mr-item {

margin-top: 10px;

float: left;

width: 4.4rem;

margin-right: .3rem;

position: relative;

}

.mr-item img {

width: 100%;

height: 2.6rem;

border-radius: 5px;

}

.room-info {

position: absolute;

bottom: 33px;

color: #fff;

padding: 0 5px;

left: 0;

right: 0;

overflow: hidden;

background: rgba(10,10,10,.5);

line-height: 24px;

border-bottom-left-radius: 5px;

border-bottom-right-radius: 5px;

}

.room-info .count {

float: right;

}

.room-title {

line-height: 30px;

}

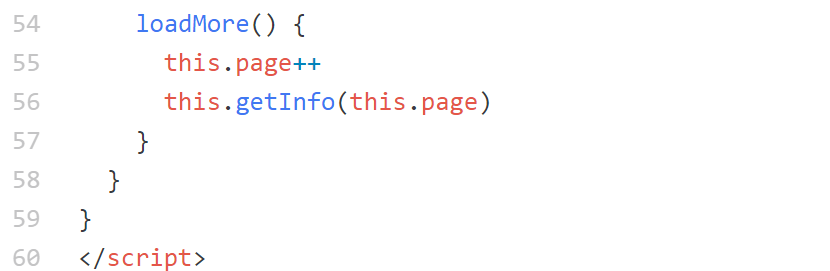
</style>

## Home加载列表









<style lang="css">

.mr-content {

padding: 44px 0 0 .3rem;

overflow: hidden;

}

.load-more {

margin: 10px;

text-align: center;

}

.load-more span {

display: inline-block;

line-height: 30px;

padding: 0 20px;

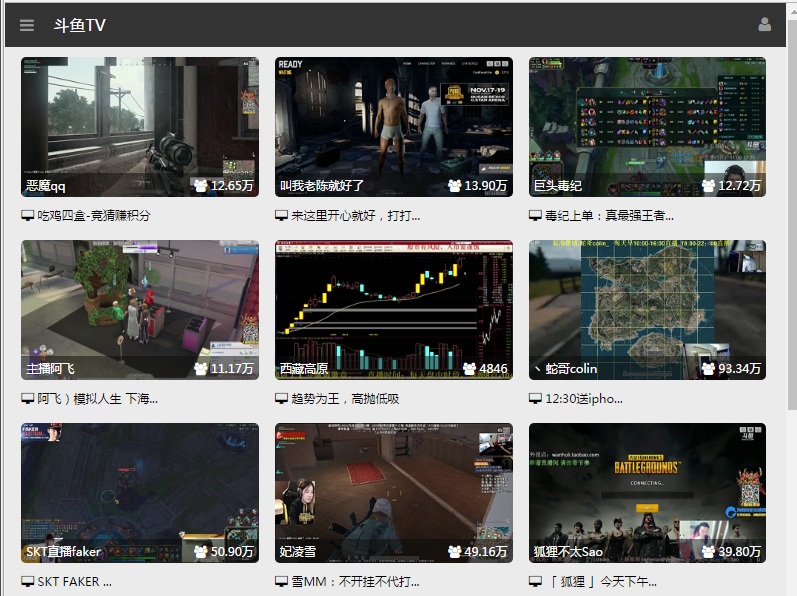
border-radius: 10px;

border: 1px solid #000;

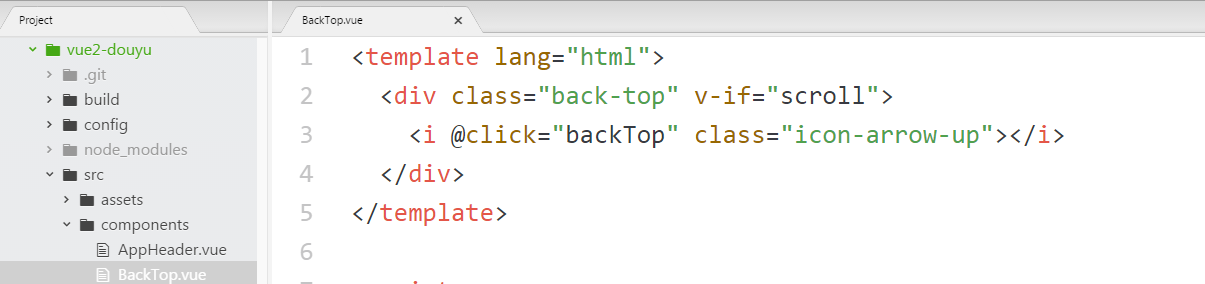
}

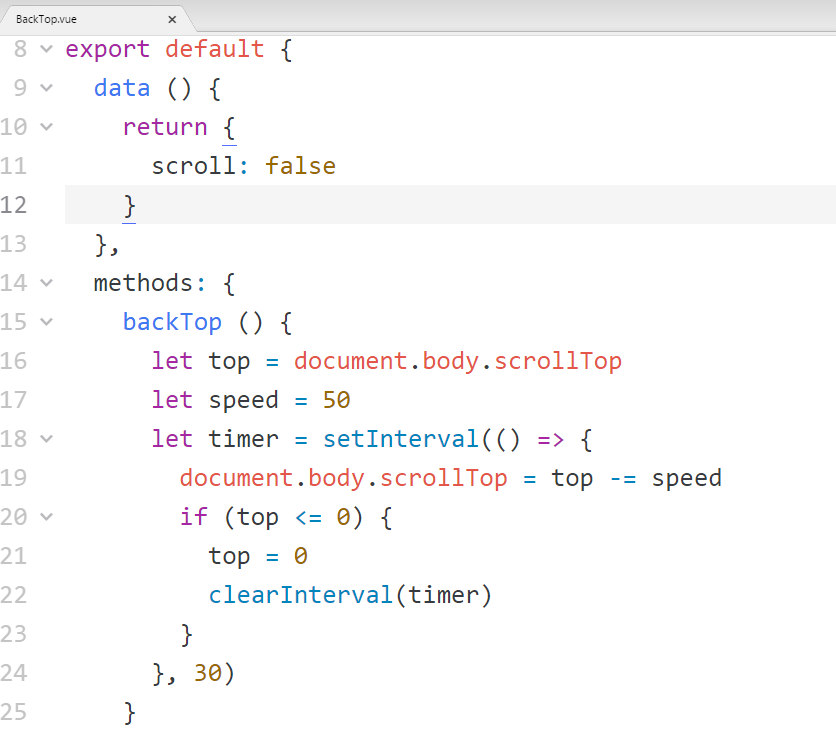
</style>

## 到目前为止首页内容可以显示出来了



## 添加回到顶部组件







<style lang="css">

.back-top {

position: fixed;

right: 20px;

bottom: 40px;

}

.back-top i {

font-size: 20px;

display: inline-block;

padding: 10px;

border: 1px solid #ccc;

color: #ccc;

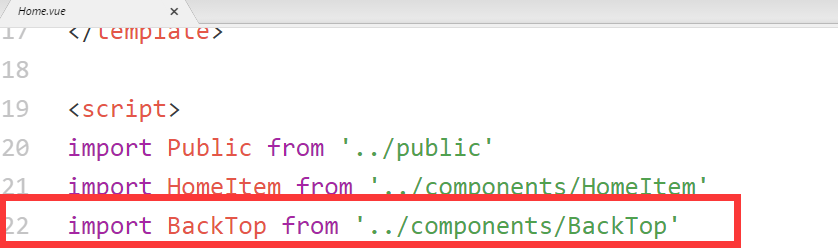
border-radius: 50%;

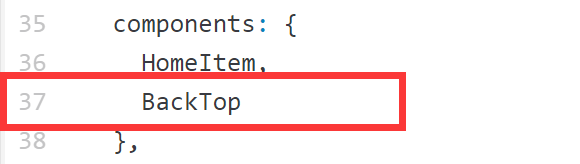
background: #fff;

}

</style>

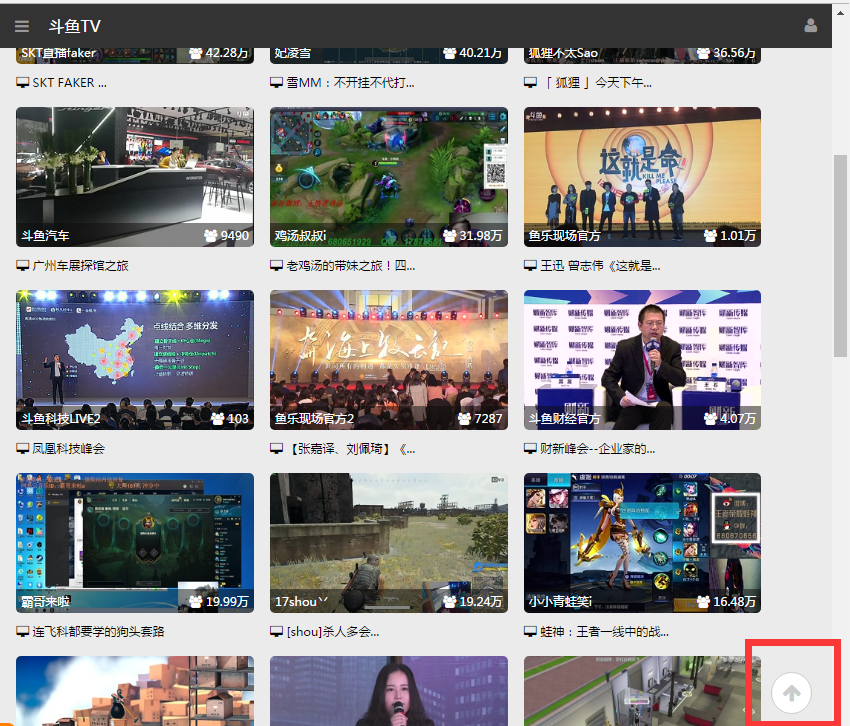
## 使用BackTop组件







## 38.查看返回顶部效果



## 新建GameList组件



<style lang="css" scoped>

li {

flex-basis: 33.3333%;

border-right: 1px solid #ccc;

border-bottom: 1px solid #ccc;

text-align: center;

padding: 20px 0;

}

li:nth-child(3n) {

border-right: none;

}

li img {

width: 2rem;

height: 2rem;

border-radius: 50%;

}

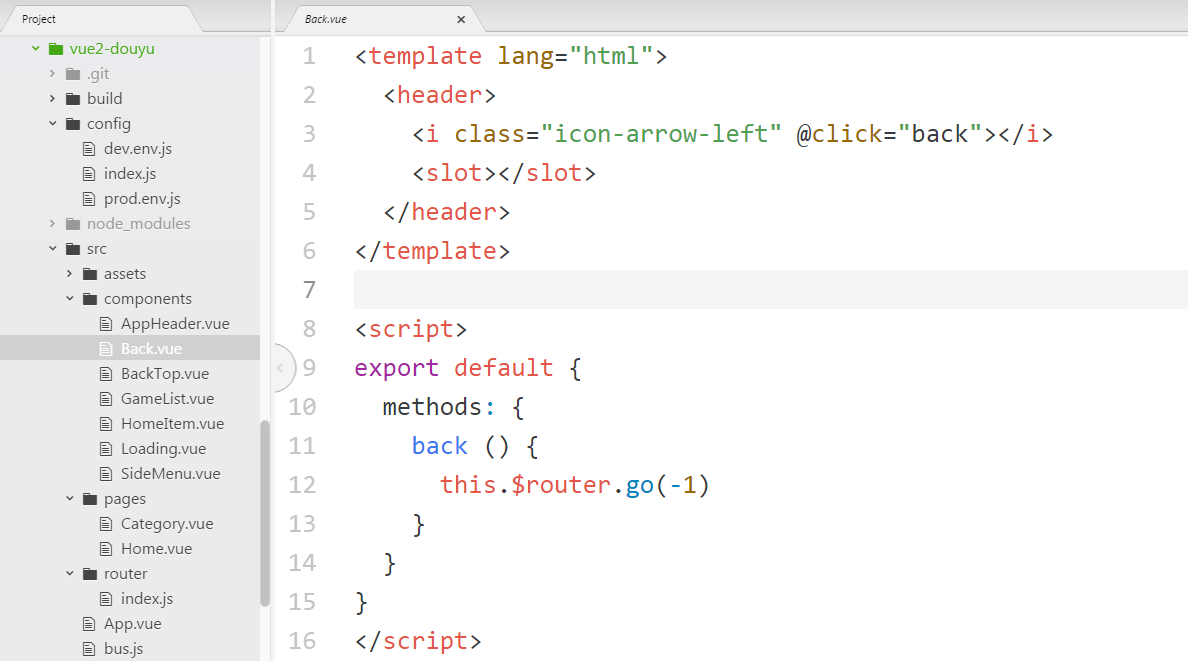
li p {

margin-top: 10px;

}

</style>

## 新建后退组件



<style lang="css" scoped>

header {

height: 44px;

background: #333;

position: fixed;

left: 0;

right: 0;

top: 0;

z-index: 100;

padding: 0 15px;

color: #fff;

line-height: 44px;

font-size: 16px;

}

header i {

color: #999;

}

.title {

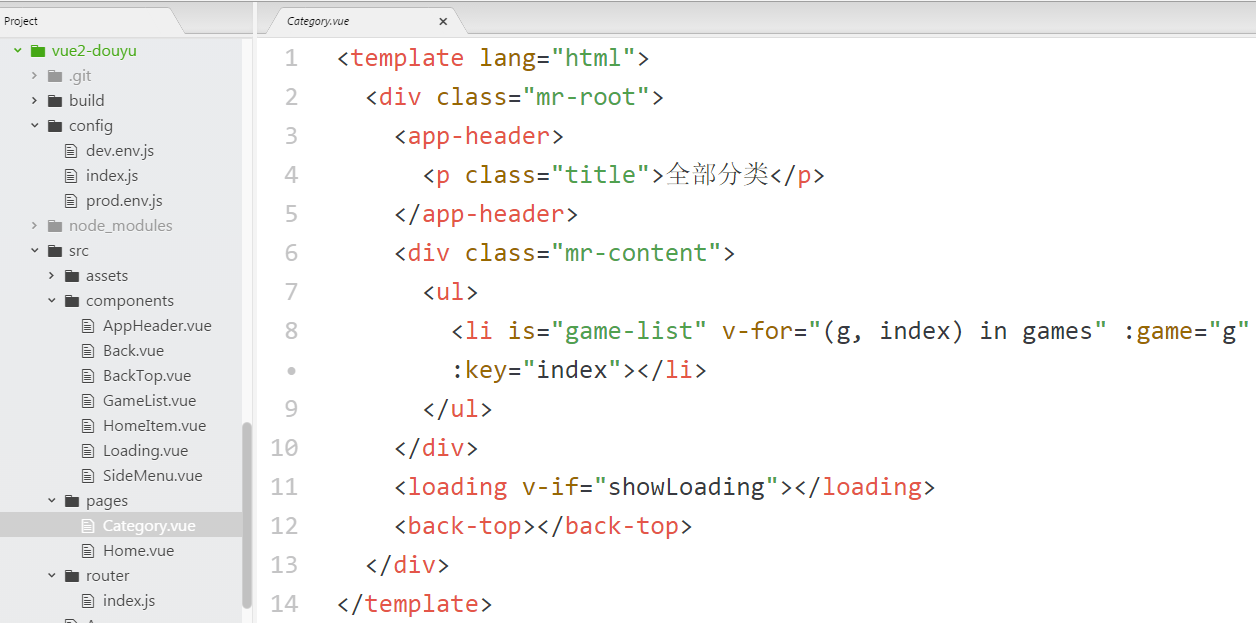
margin-left: 15px;

display: inline-block;

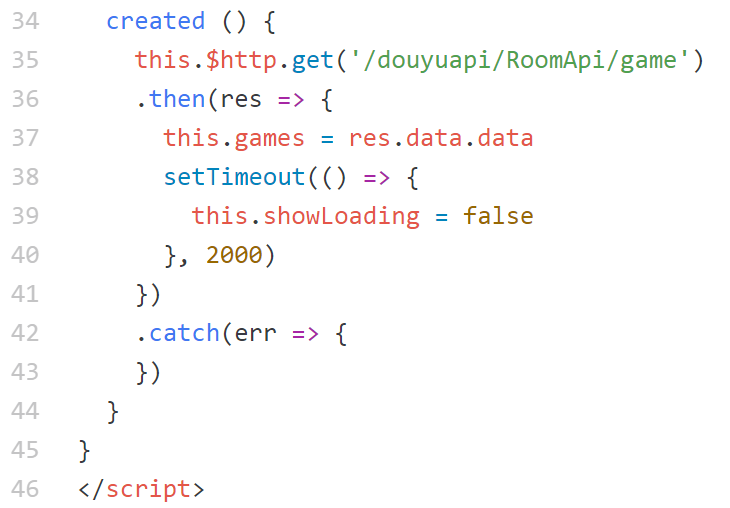
}

</style>

## 新建分类页面







<style lang="css" scoped>

.mr-content{

padding: 44px 0 0 0;

}

.mr-content ul{

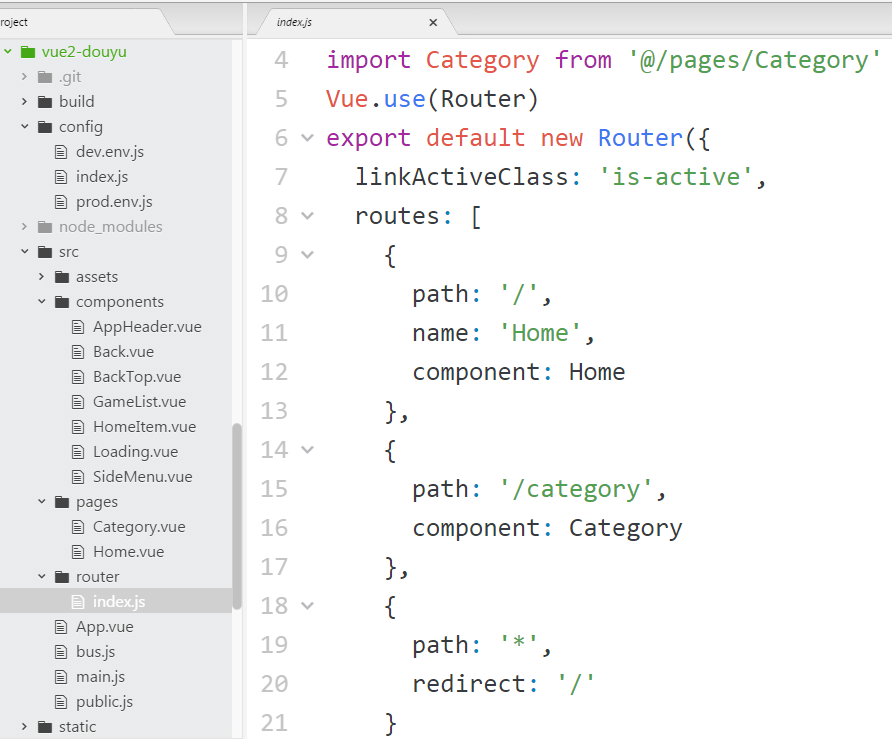
display: flex;

flex-wrap: wrap;

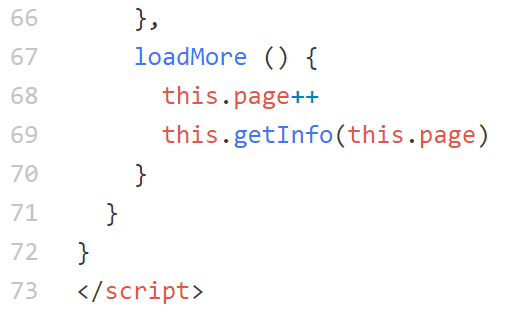
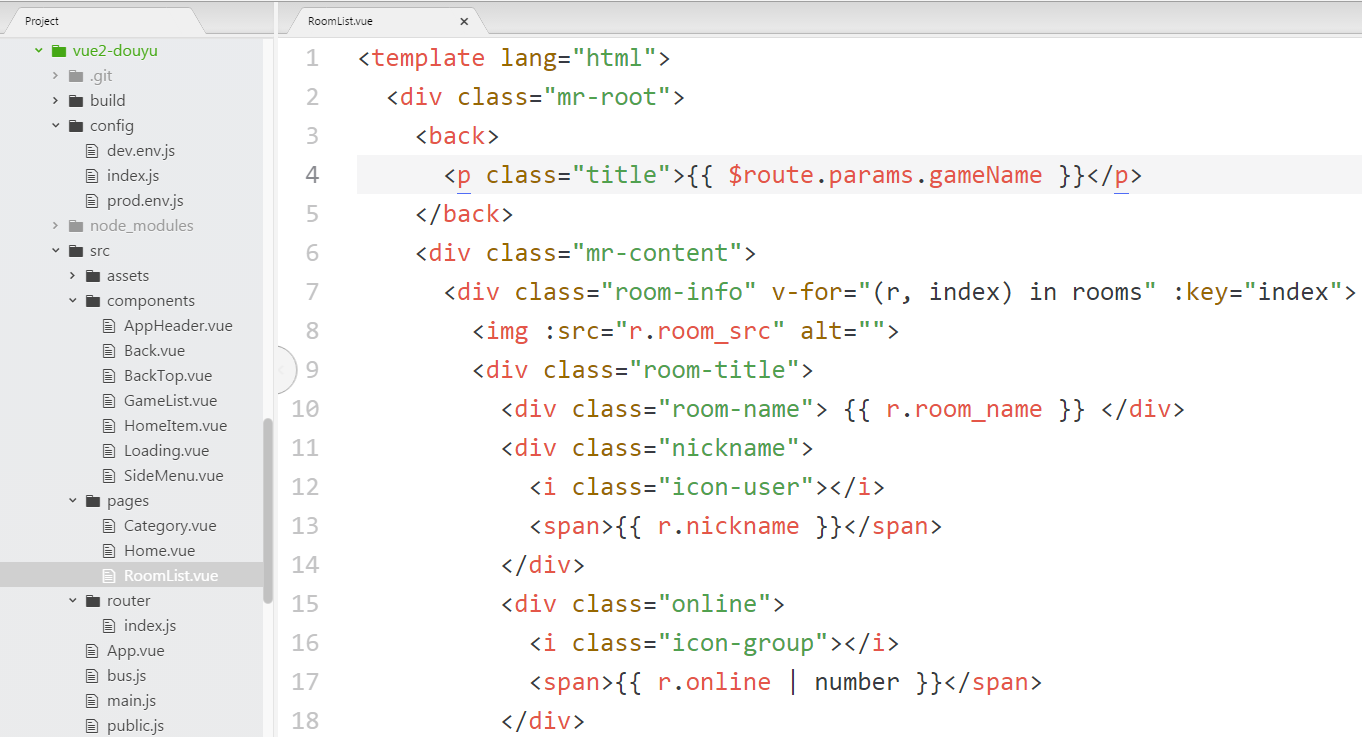
}

</style>

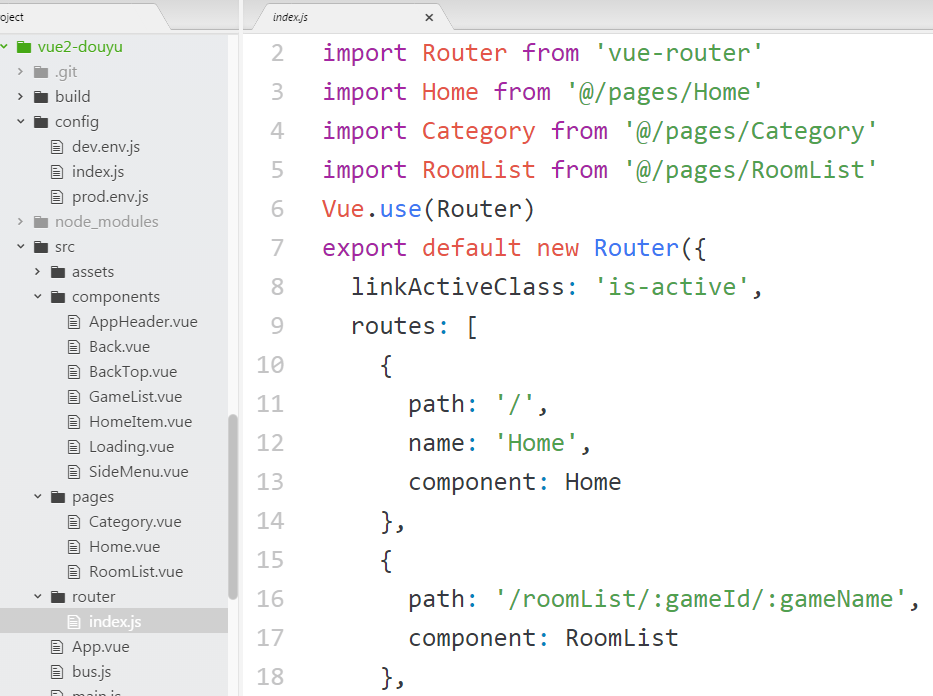
## 把分类页面添加到路由中



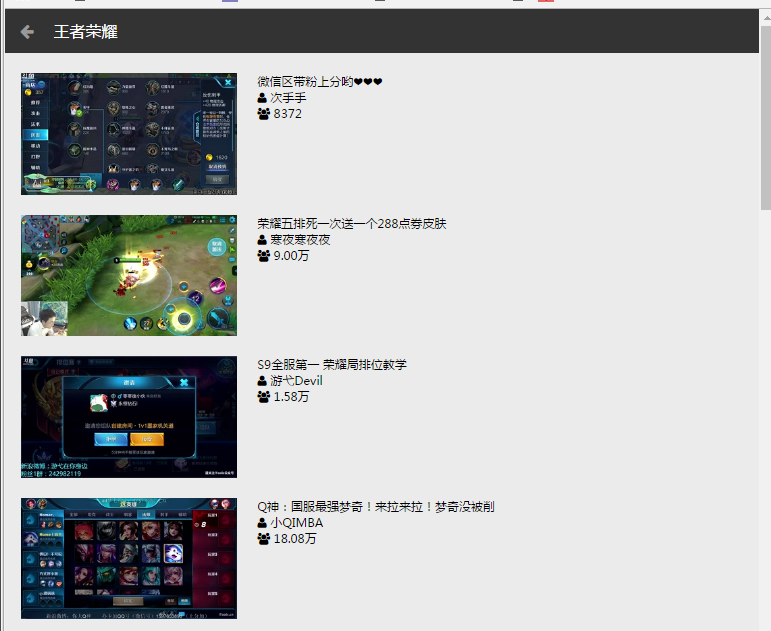
## 新建RoomList页面



## 添加对应的路由



## 房间列表效果



## 添加图片懒加载功能

安装模块：npm i -S vue-lazyload

安装为Vue的插件：

import VueLazyload from 'vue-lazyload'

Vue.use(VueAxios, Axios)

Vue.use(VueLazyload, {

preLoad: 1.3,

error: 'assets/error.png',

loading: require('@/assets/loading-spinning-bubbles.svg'),

attempt: 1

})

使用：

在用到img图片的地方，把:src换成v-lazy: 示例：

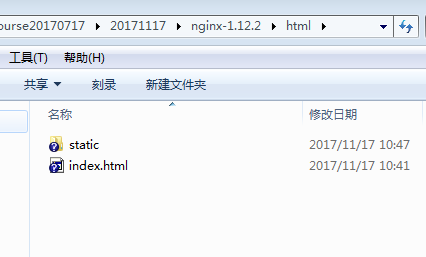
<img v-lazy="r.room\_src" alt="">

## 项目使用nginx在windows上面部署上线

npm run build

下载Nginx服务器程序：<http://nginx.org/en/download.html>

将vue生成的目标文件dist目录中的文件全部拷贝，粘贴到nginx程序的html目录中



双击nginx启动

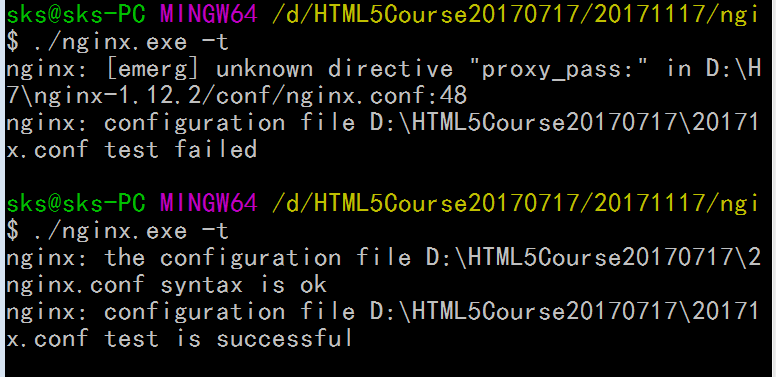
浏览器打开 localhost 查看效果

## 配置nginx的反向代理

修改nginx的配置文件(添加反向代理的指令)：



修改之后检查一下配置文件是否OK



如果发现返回的是syntax is ok则没问题，否则就是配置文件修改错了，重新修改

关闭nginx进程

重新启动nginx进程

打开localhost