# 网络模块封装

## 网络请求模块的选择-axios



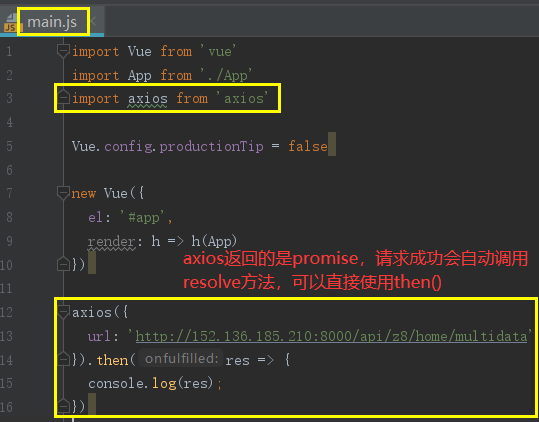
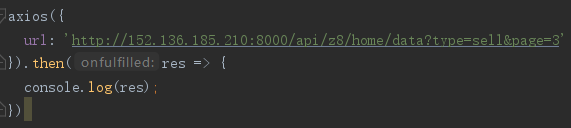
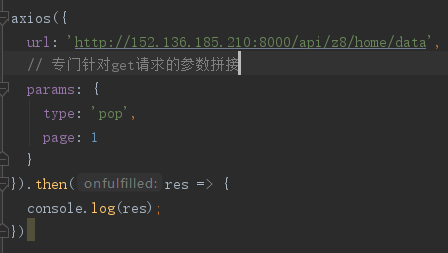
## Axios框架的基本使用



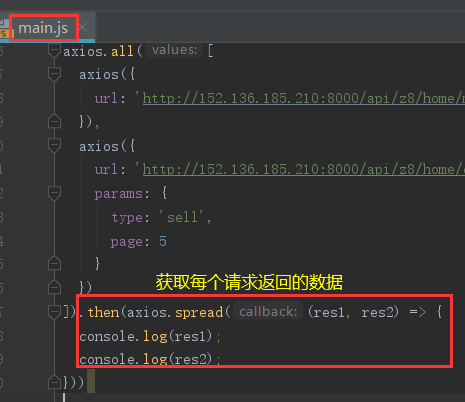
1. 创建项目：vue init webpack leaenaxios



1. 安装axios框架：npm install axios --save
2. Axios最简单的使用：导入框架，发送请求，成功则返回结果

## Axios发送并发请求

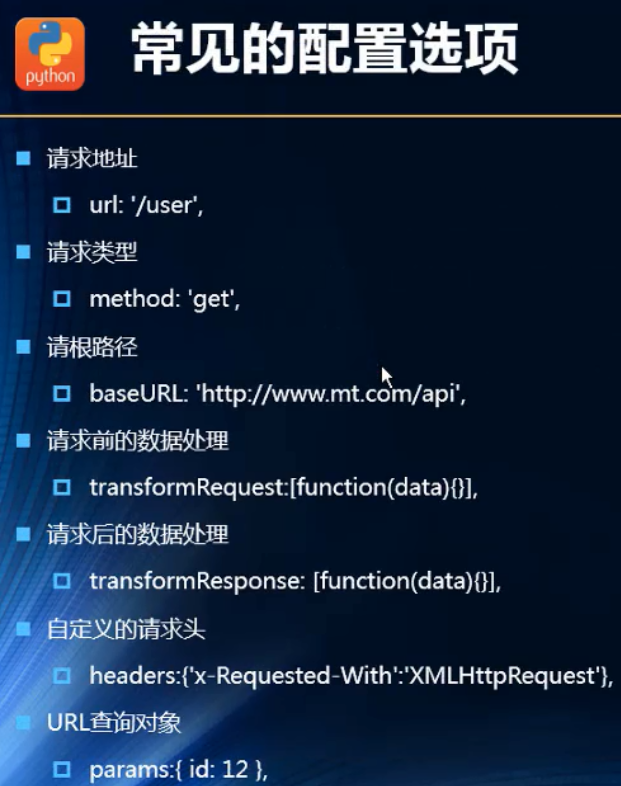
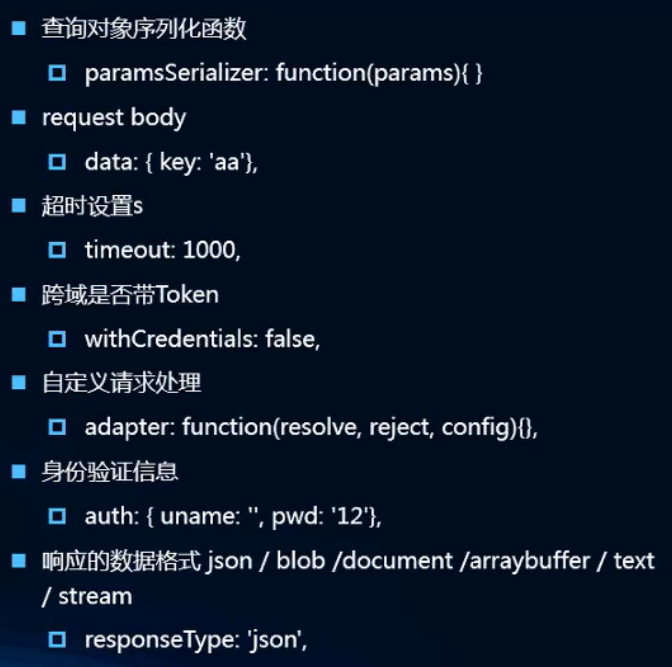


## Axios的配置信息相关

URL前缀都是一样、重复





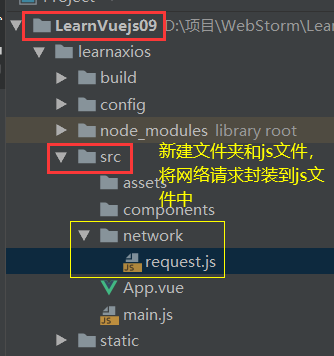
 

## Axios的实例和模块封装

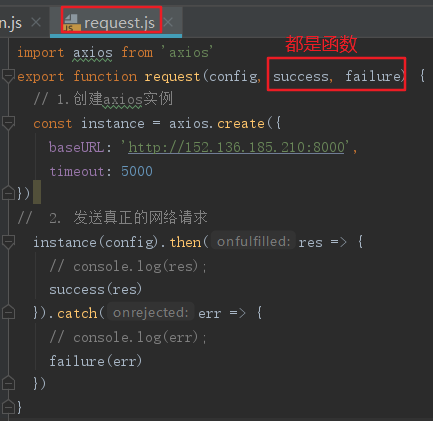
### 创建axios实例

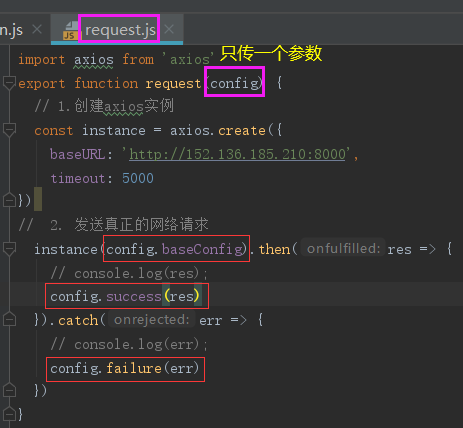
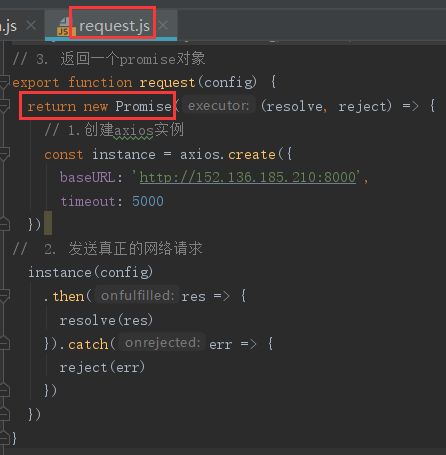
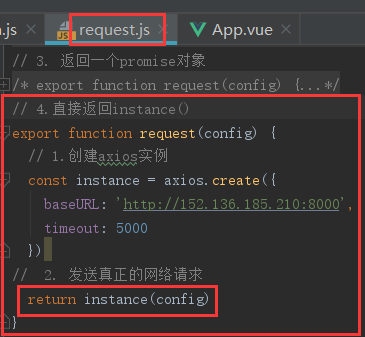
使用create()

### axios模块封装

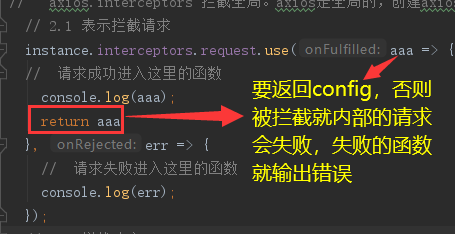
为了导出的扩展性，写js文件时不使用export default

1. 传入3个参数，其中success和failure是回调函数

1.  
2.  
3.  

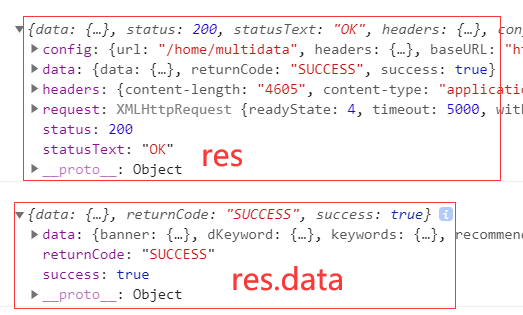
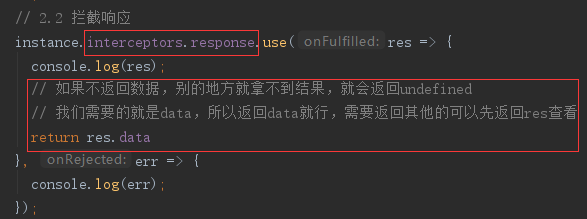
## Axios的拦截器的使用

### 请求拦截



### 响应拦截



## 项目开发-项目创建和GitHub

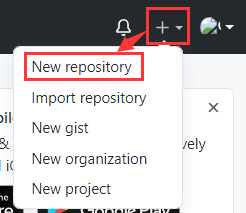
### 创建项目

1. 创建脚手架3项目：vue create supermall



### GitHub托管项目代码

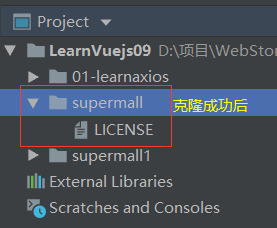
1. 登鹿GitHub
2. 创建仓库



1. 将项目和远程仓库连接起来



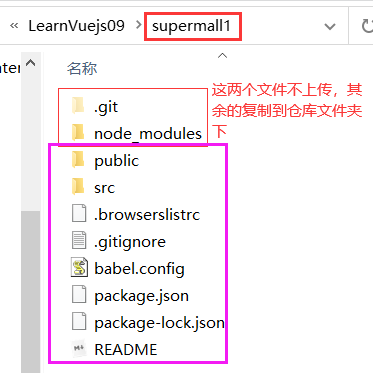
然后克隆仓库，把复制的地址拷贝粘贴

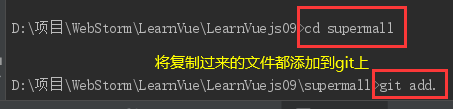
 

1. 将项目的代码放到仓库中。

方法一：手动将项目文件夹下的东西放到仓库文件夹下

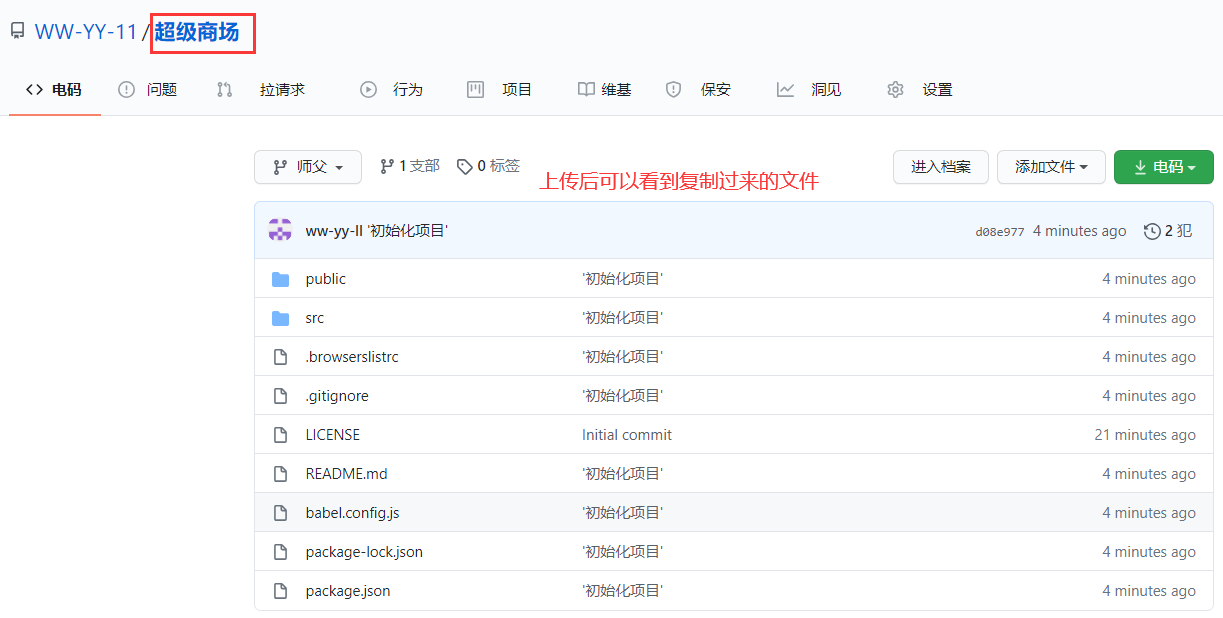


1）

2） 

3）

4）



之后在项目开发中，修改了代码，只需要【git add】【 git commit】【 git push】，就可以把修改后的代码上传了

方法二：不手动拷贝本地文件



1. 新建有仓库

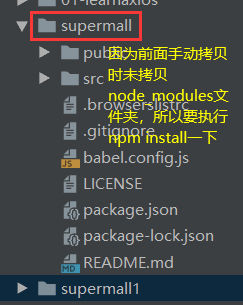
 

2）

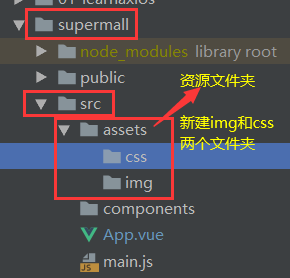
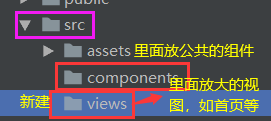
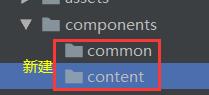
3）

代码也成功上传到仓库了

## 项目开发-划分目录结构

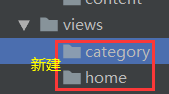
安装之后就有node\_modules文件夹了

最后删除supermall1

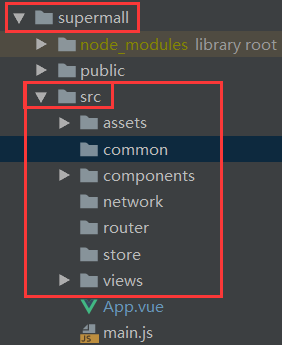
1. 删除不需要的文件，如HelloWorld.vue文件和用到它的配置
2. 
3.  

common文件夹：里面放的组件不仅是当前项目可以使用的，下一个项目使用时复制过去也能使用。相当于放完全公共的组件。

content文件夹：放和当前项目业务相关的公共组件。

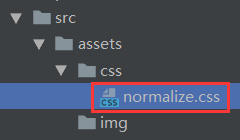
单独写对应视图的相关逻辑代码

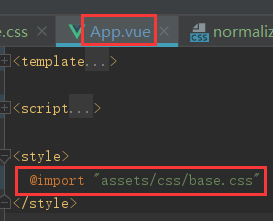
1. 新建router文件夹：放和路由相关的
2. 新建store文件夹：放和状态相关的
3. 新建network文件夹：放和网络、请求相关的
4. 新建common文件夹：放公共的js文件，如常量文件、工具类



## 项目开发-css文件的引入

1. 对不统一的样式进行统一性规范，即初始化。

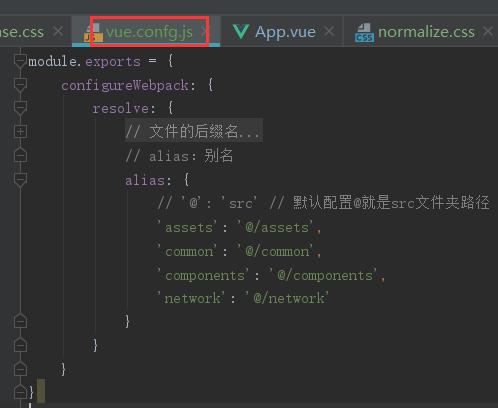
直接复制文件到css文件夹下

1. 然后再创建一个base.css文件复制另一个项目中的base.css中的所有样式到这里
2. 引入样式表才能生效

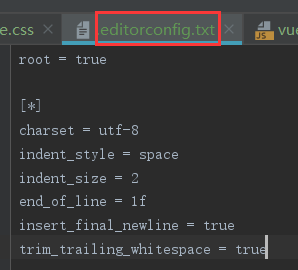
## 项目开发-vue.config和editorconfig

对文件路径进行设置别名，就不需要写很多【../../../】类似这样的路径，脚手架3中配置文件别名：

1. 在项目文件夹的根目录下，创建新的【vue.config.js】文件

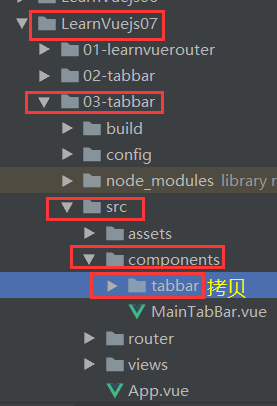
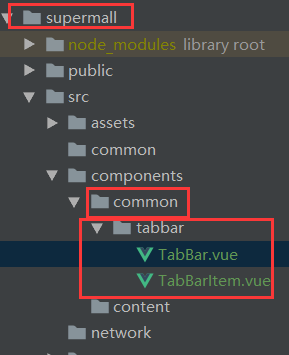
1. 在项目文件夹的根目录下，新建【.editorconfig.txt】文件：统一代码风格，如每行代码缩进2个字符还是4个字符

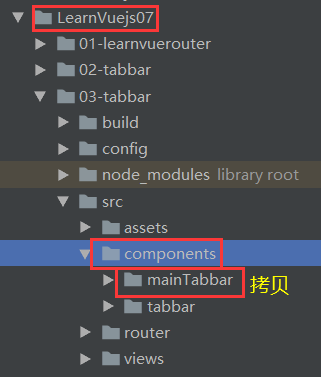
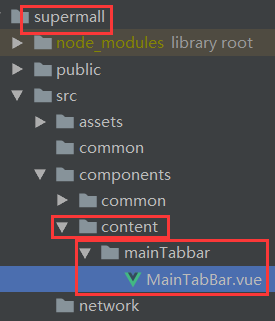


## 项目开发-tabbar引入和项目模块划分

运行项目npm run serve 

1. 将learnvuejs07文件夹下，封装过的tabbar拷贝到当前项目中。

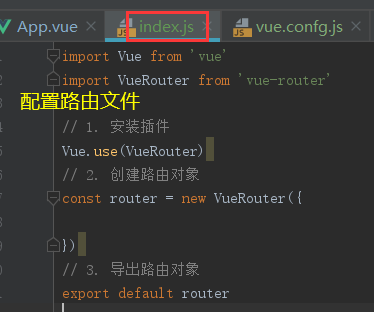
1. 更改图片路径，使用别名路径，使用~



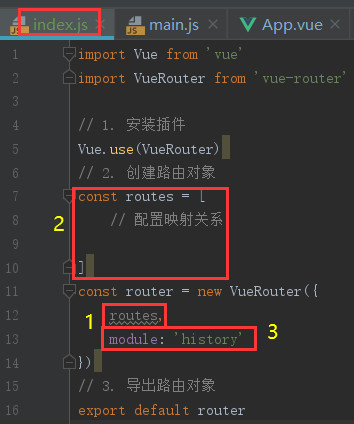
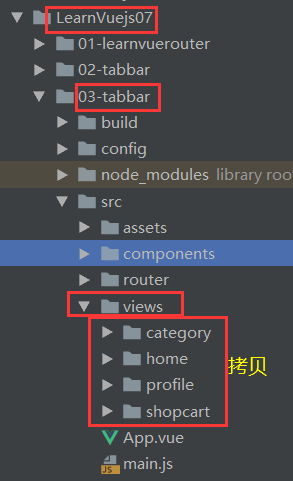
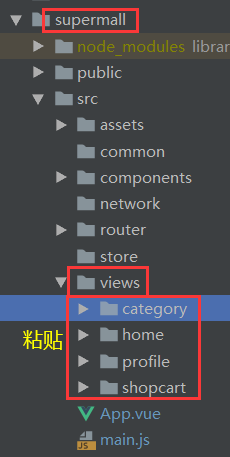
1. 使用tabbar组件

而每个tabbar都对应路由的页面切换，但是项目中没有路由相关的文件，所有先安装npm install vue-router –save

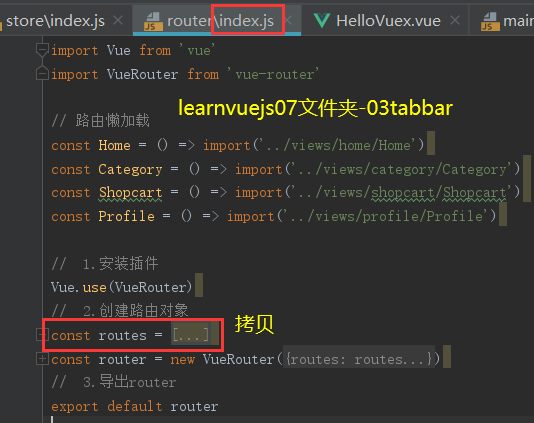


1.  

更正

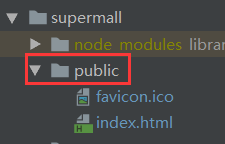
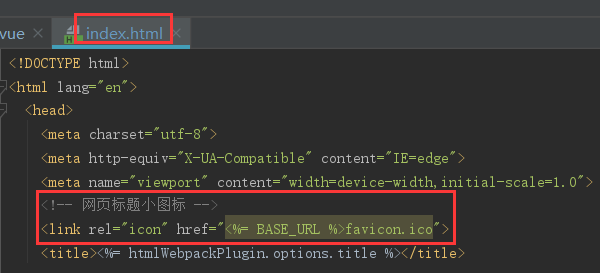
1. 配置路由映射关系

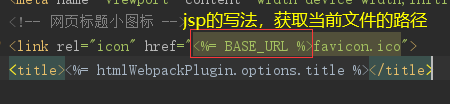
1. 项目模块划分：tabbar -> 路由映射关系

## 项目开发-小图标的修改以及路径问题

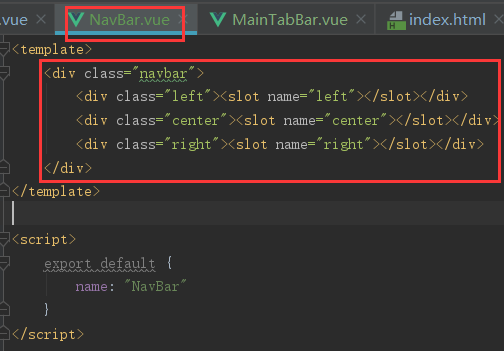
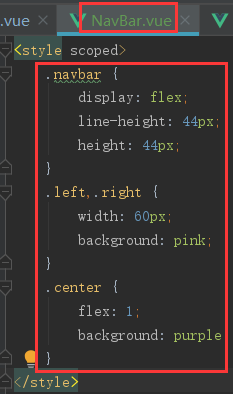
1. 小图标的修改

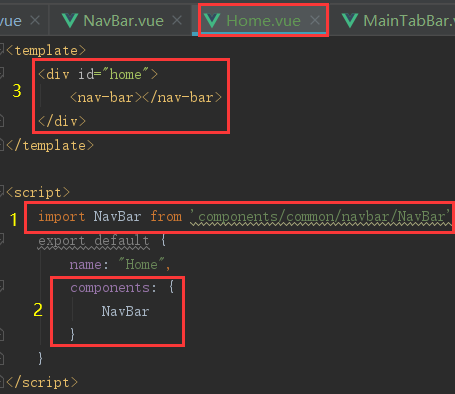
 

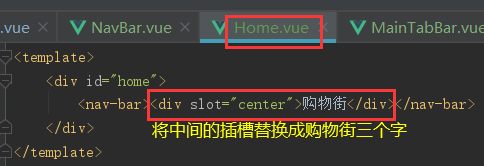
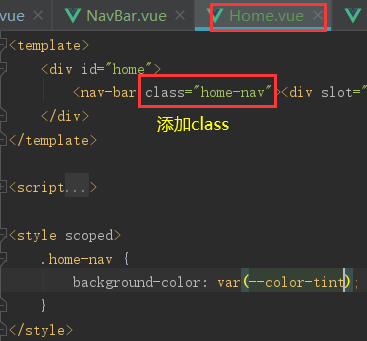
1. 路径问题



## 首页开发-首页导航栏的封装

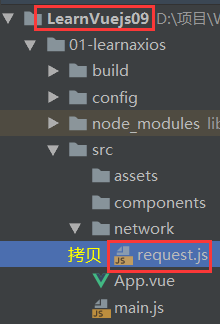
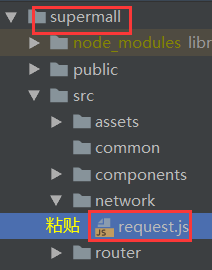
  

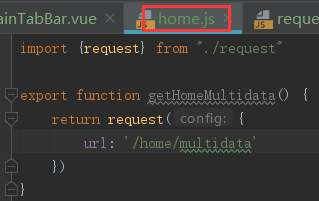
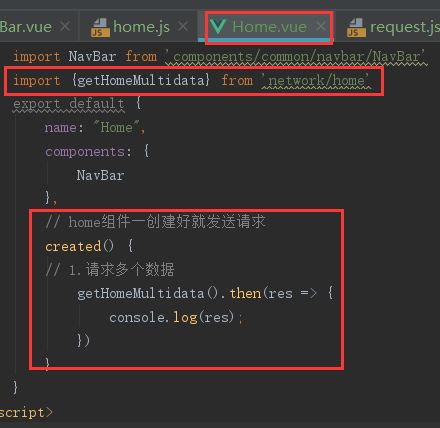
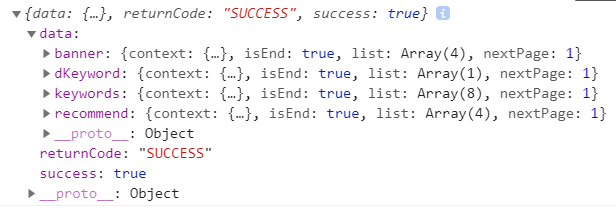
页面结果

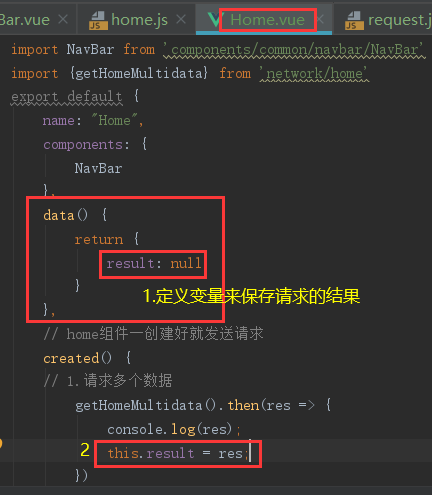
1. 
2. 替换背景色使用base.css中定义的变量设置背景色

## 首页开发-请求首页的多个

1. 发送请求，获取所有的数据。

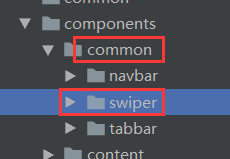
  

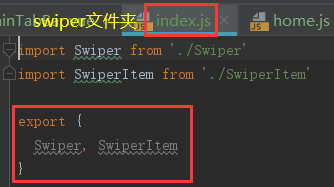
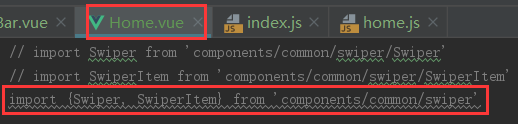
1. 安装axios【npm install axios –save】
2. 新建js文件，来封装首页发送的所有请求 
3. 页面请求结果
4. 将请求的数据保存起来。因为函数执行完毕后里面的数据会被销毁，箭头函数也是。

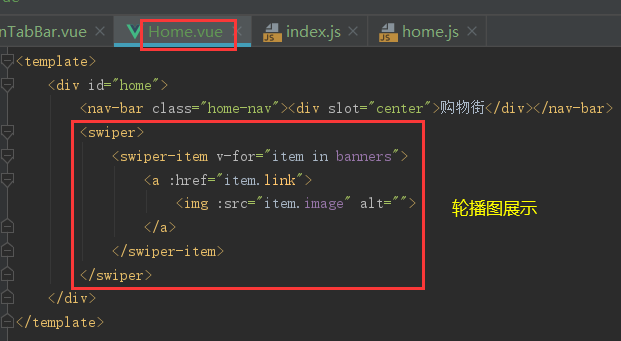
 

## 首页开发-轮播图的展示

先封装组件，再展示

1. 拷贝封装好的轮播图组件过来
2. 因为swiper文件夹下，index.js文件导出时是导出对象，所有导入时可以合在一个对象里面

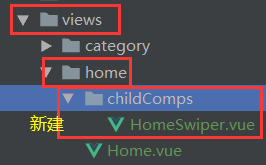
  

1. 

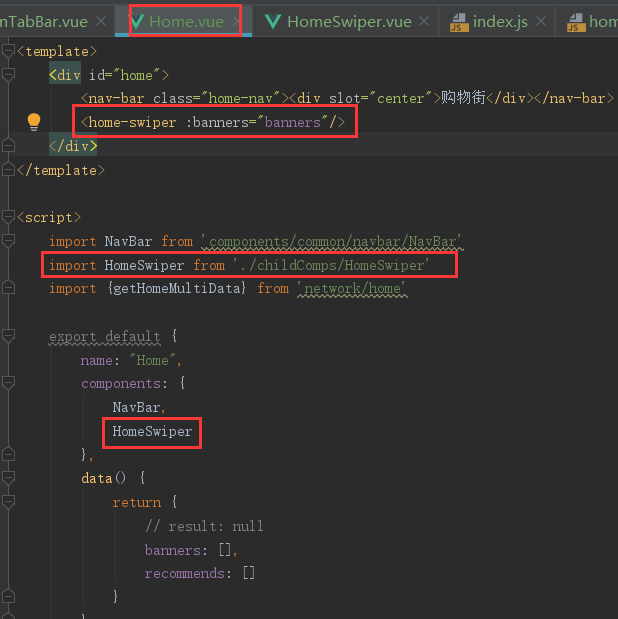
页面运行结果

1. 封装轮播图部分代码

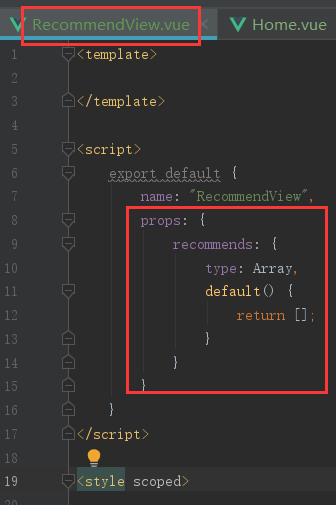
新建childComps文件夹，新建HomeSwiper.vue文件

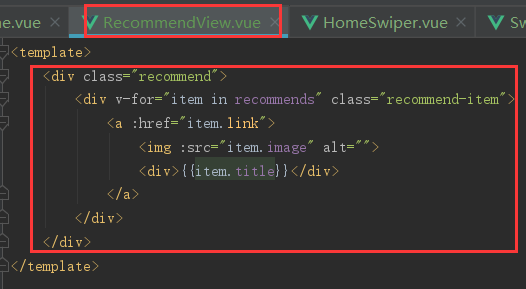


把Home.vue中swiper部分代码剪切放到HomeSwiper.vue文件中

## 首页开发-推荐信息的展示



页面结果：