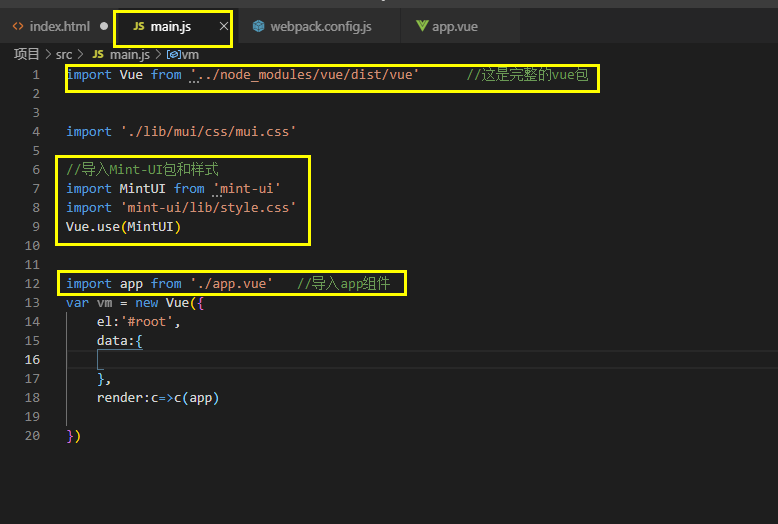
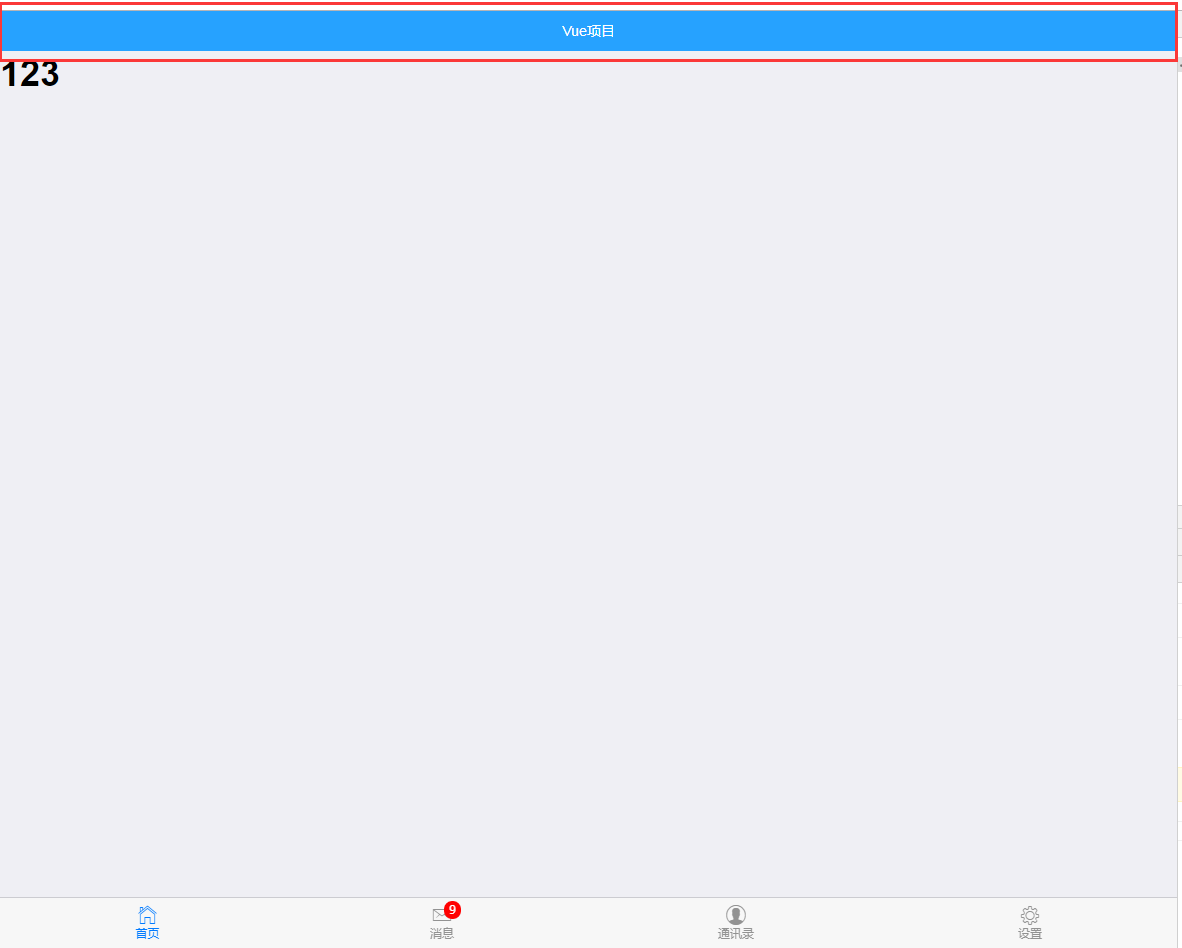
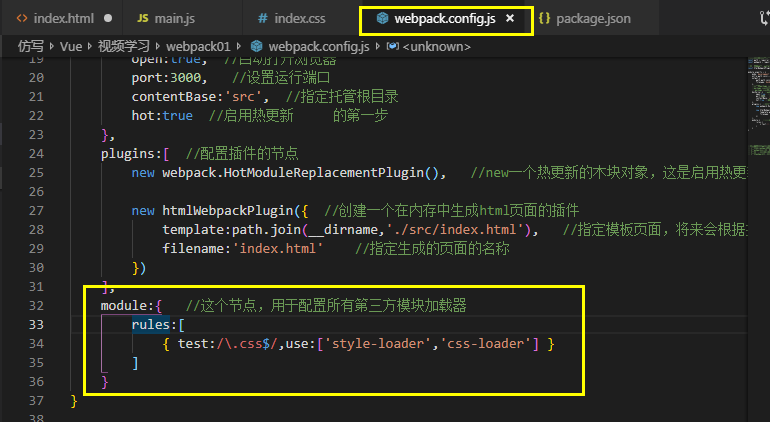
1. 新建项目文件夹里的文件
2. 在main.js里面导入各种包和组件（.vue文件），Mint-UI是基于vue.js的，所以直接导入包即可（导入方式可以参考官方文档）：

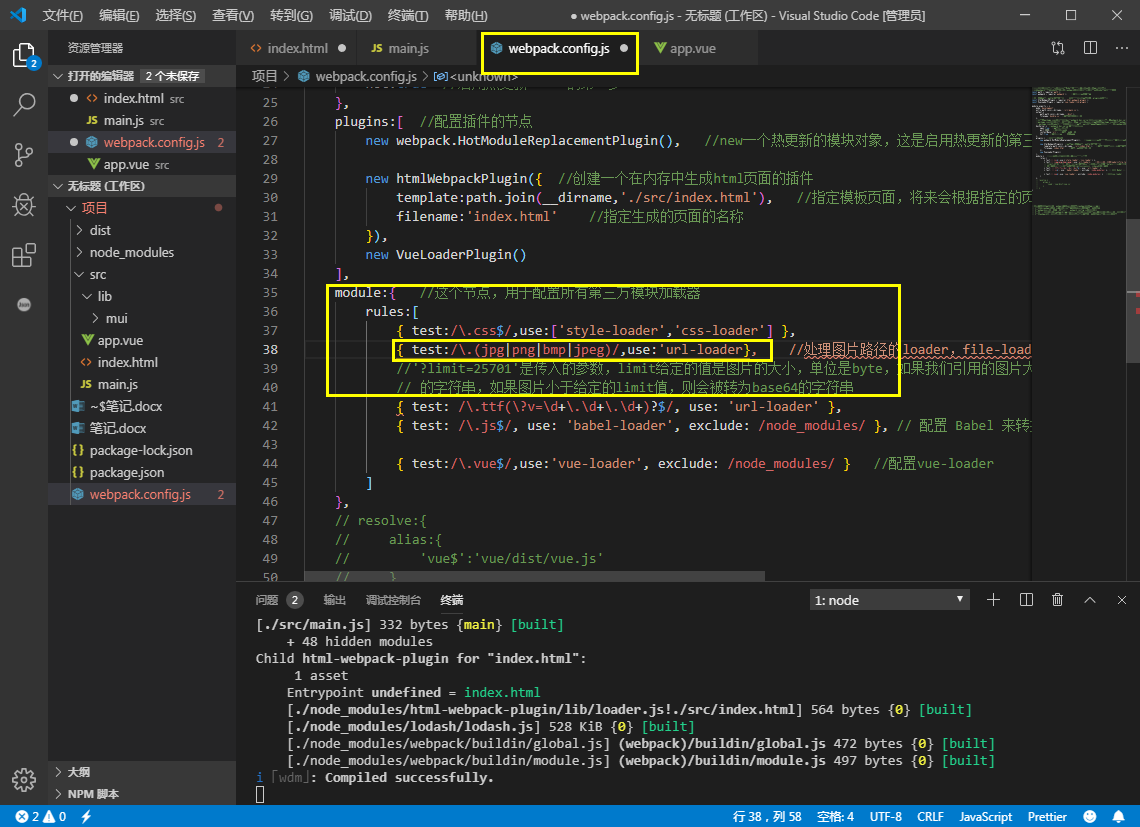


头部使用的Minu-UI中的header组件：

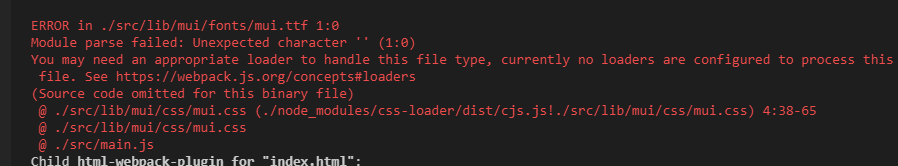


3、导入mui的包，包里要安装处理css和url的loader，css的loader：npm install style-loader css-loader，url的loader：配置文件里匹配和处理规则：

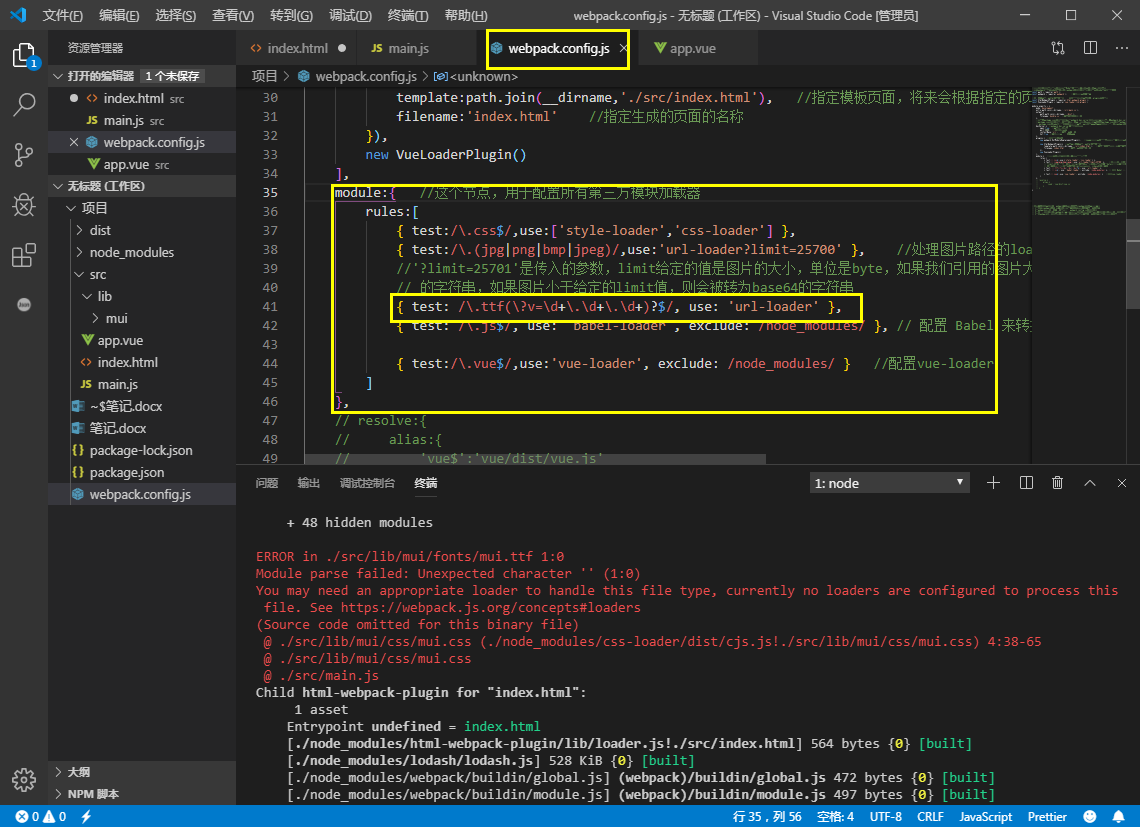


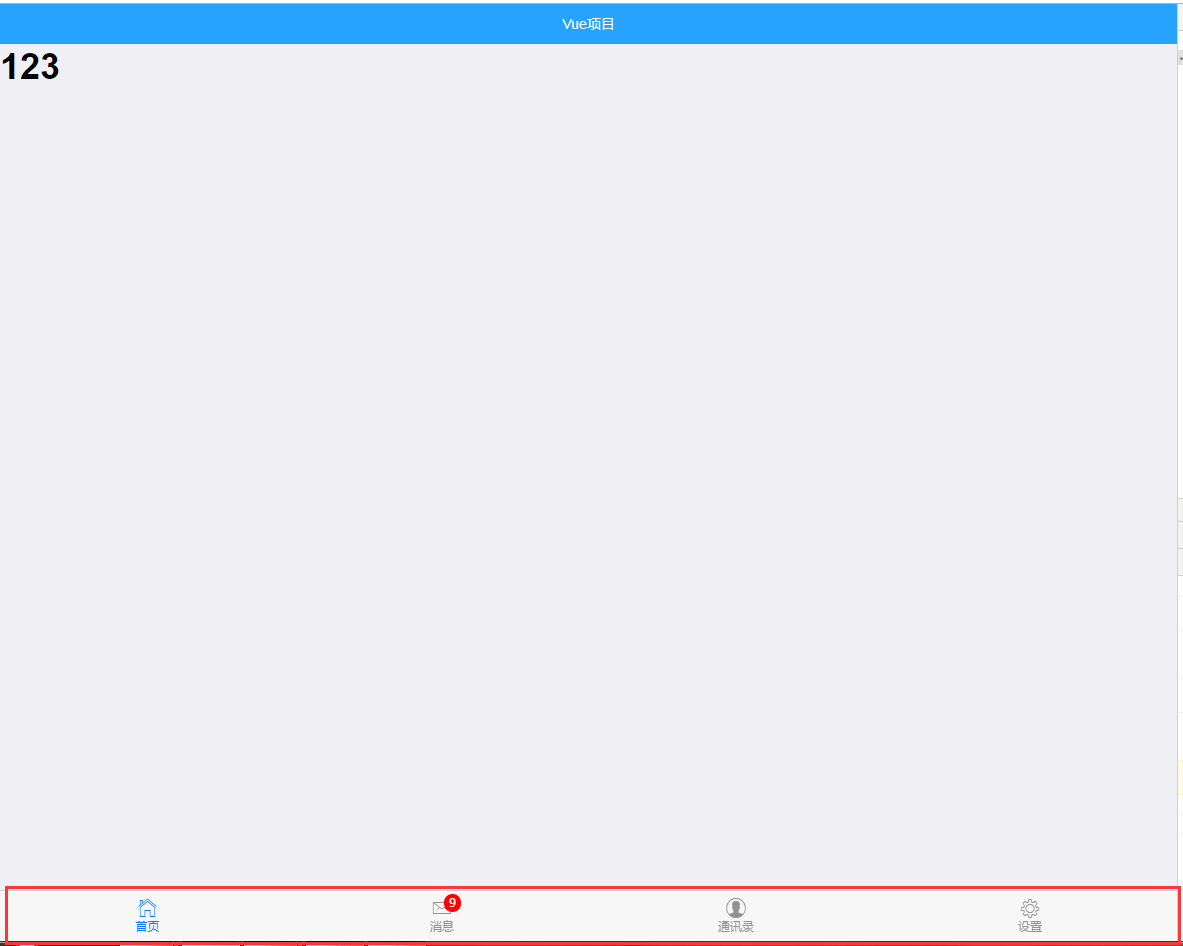


当执行程序时，报错：（遇到的问题）



是因为浏览器解析不了ttf的组件，所以要用url-loader解析，在配置文件里添加解析规则：

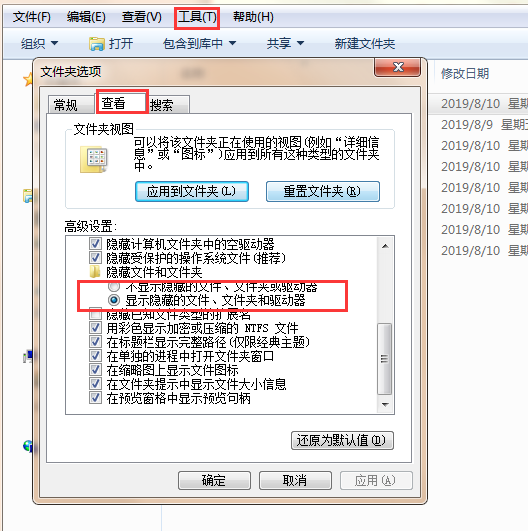




4、将项目托管到git中，先要安装git

项目忽略文件：.gitignore

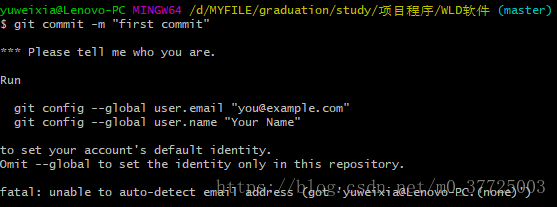
安装完git后，在本地项目中初始化：git init，生成一个.git文件，文件夹是默认隐藏这个文件的，在文件夹中显示出隐藏文件：（遇到的问题）



将项目上传到github中，https://github.com/sliceland/vue-webpack-Mint-UI-mui-.git（上传代码的地址）：

将项目提交到本地仓储中： 参考：https://blog.csdn.net/zamamiro/article/details/70172900

本地文件夹项目中打开git命令（选择git bash），输入git init（把项目文件夹变成git可管理的仓库） ->git add . （把项目添加到仓库，可通过git status查看项目状态）->git commit -m “init my project(自己取个名字)” -> 出现以下提示时，按要求输入邮箱和名字即可 -> 然后再git commit -m “init my project”

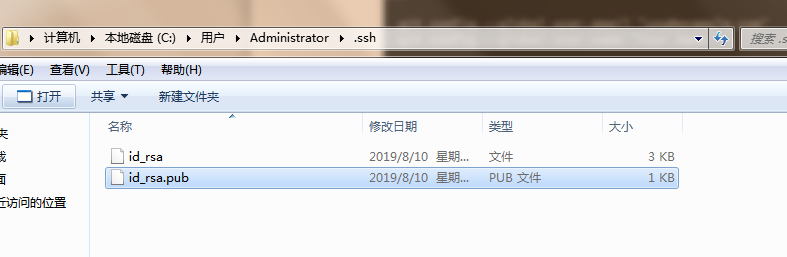


以上，项目就在本地仓库上了（注：git commit是把项目提交到仓库，-m后面引号里是本次提交的注释内容）

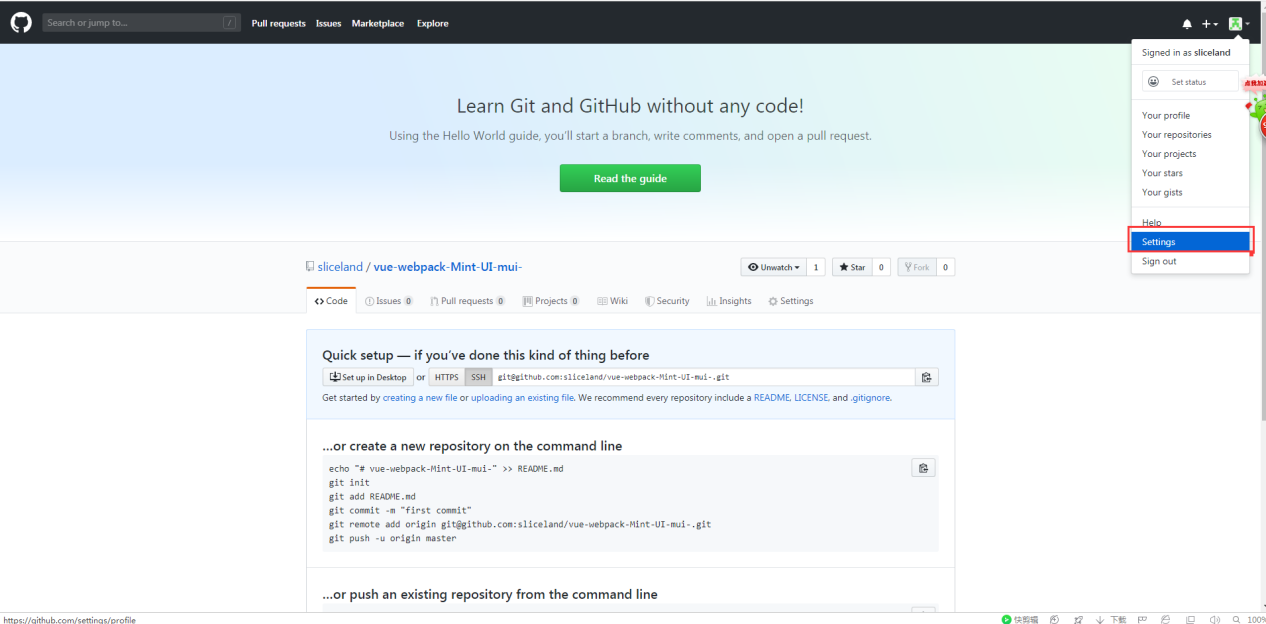
      由于本地Git仓库和Github仓库之间的传输是通过SSH加密的，所以连接时需要设置一下：

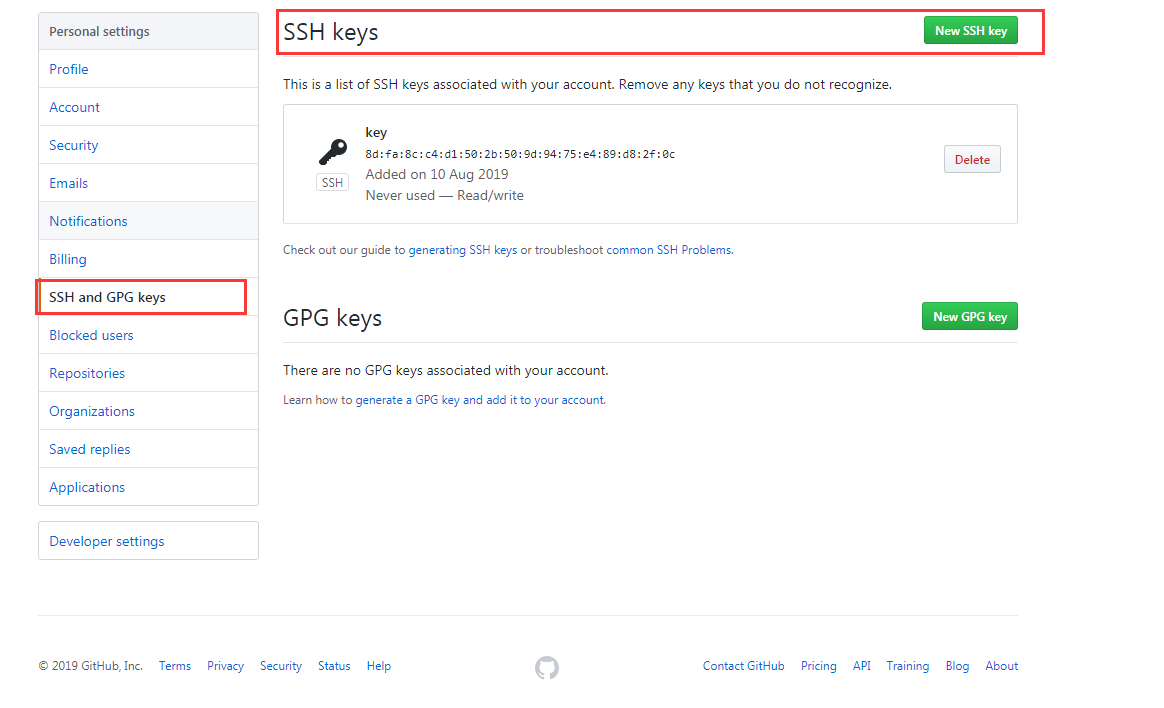
先安装公钥：打开git bash命令符，输入$ ssh-keygen -t rsa -C ["youremail@example.com"](mailto:\"youremail@example.com\")

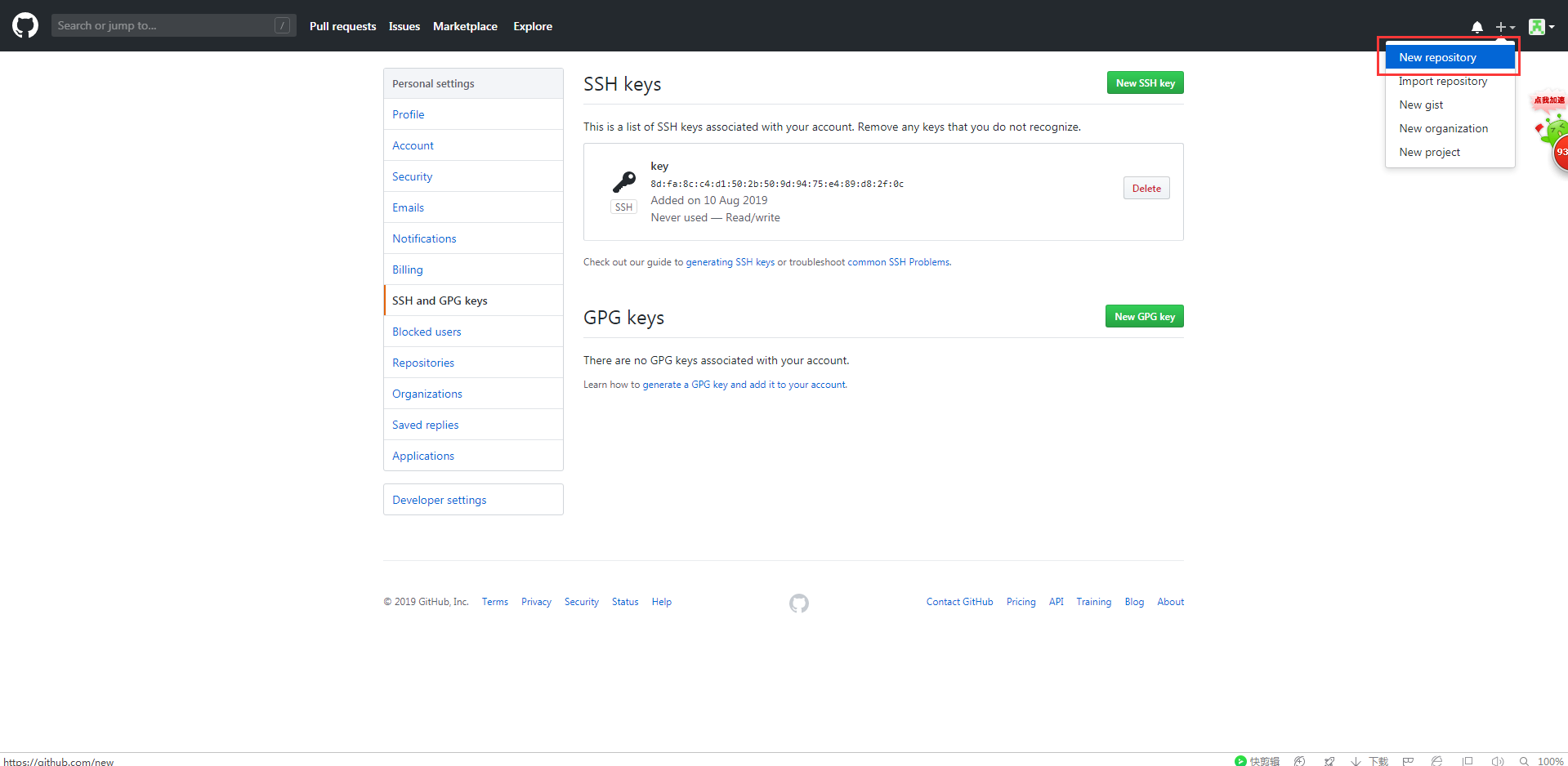
然后去下面这个文件夹下复制公钥：

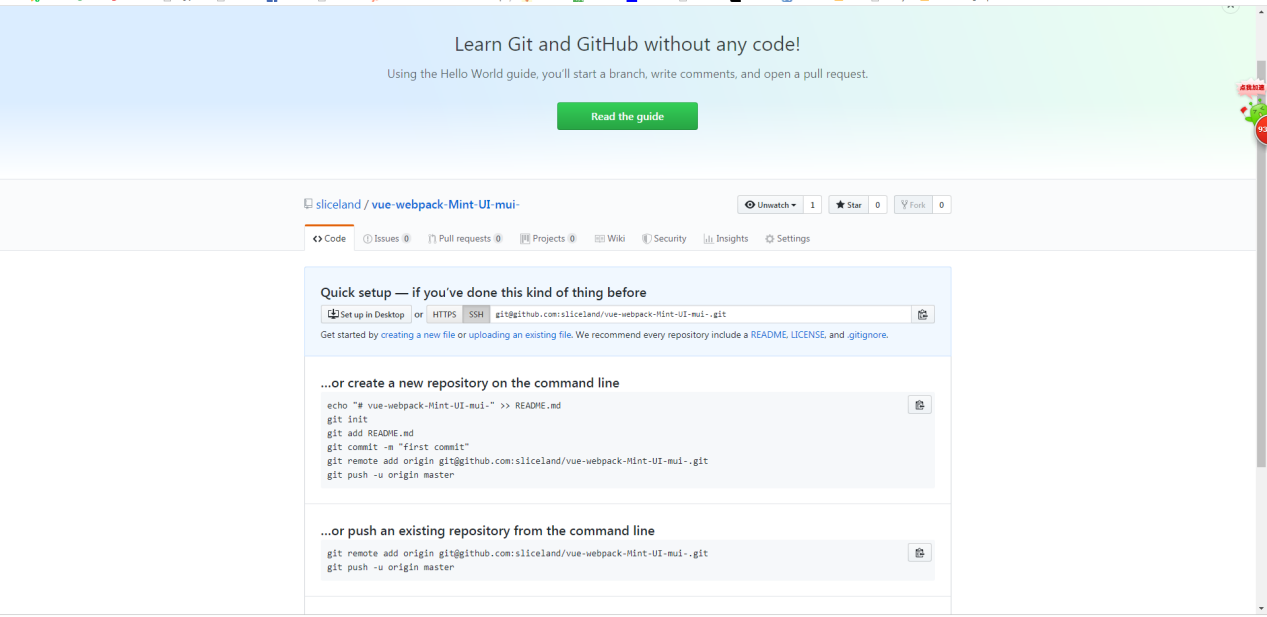


到github上去配置仓库：





Github上建仓库：



关联本地和github上的仓库，在本地项目的git bash命令中输入git remote add origin <https://github.com/sliceland/vue-webpack-Mint-UI-mui-.git>

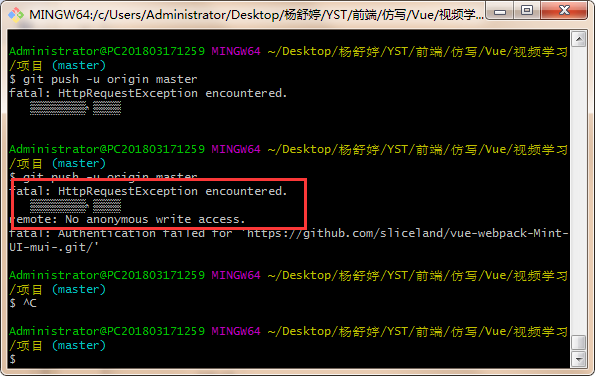
关联好之后我们就可以把本地库的所有内容推送到远程仓库（也就是Github）上了，通过：

$ git push -u origin master

由于新建的远程仓库是空的，所以要加上-u这个参数，等远程仓库里面有了内容之后，下次再从本地库上传内容的时候只需下面这样就可以了：

$ git push origin master

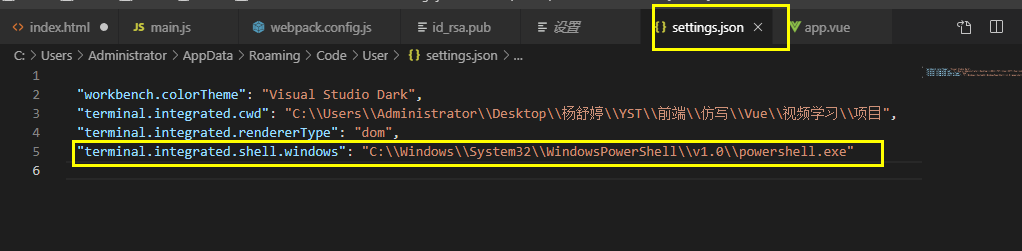
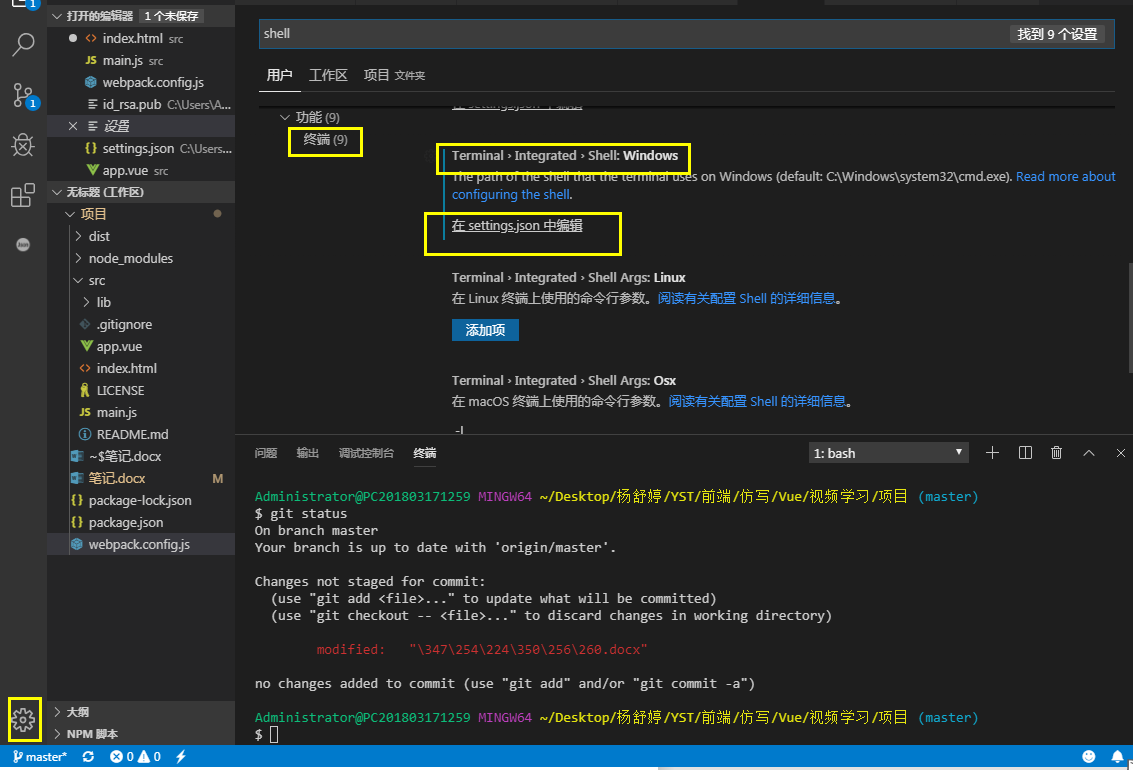
当出现以下错时，

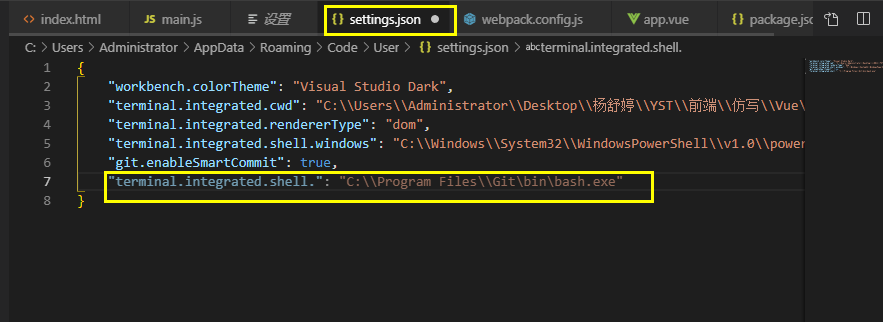


安装

Vs code终端中无法执行git命令的问题：

在终端中配置bash：





#用命令行把代码提交到Github

1、git add .

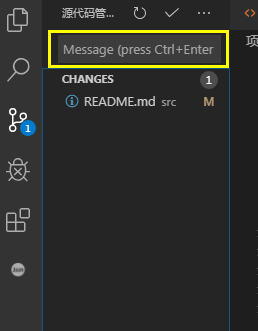
2、git commit -m "提交信息"

3、git push origin master

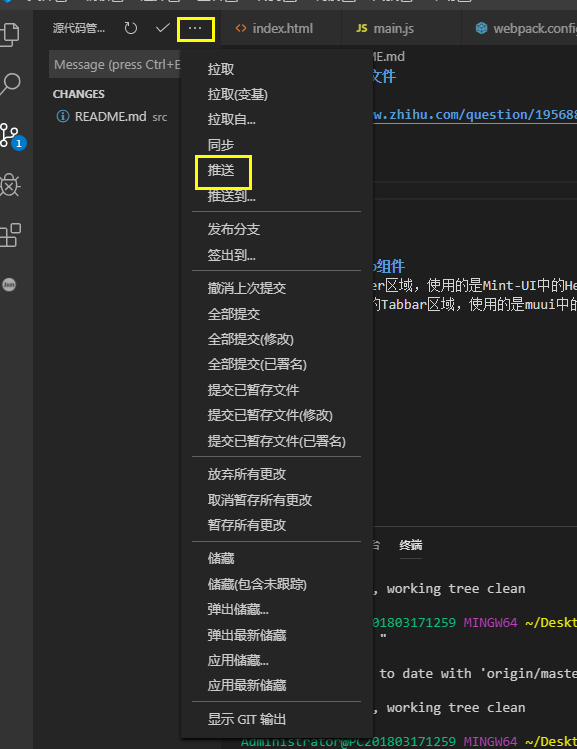
#或者利用vs code的可视化工具修改本地代码并提交到Github：

当修改了项目中某个文件时，如下图的提示

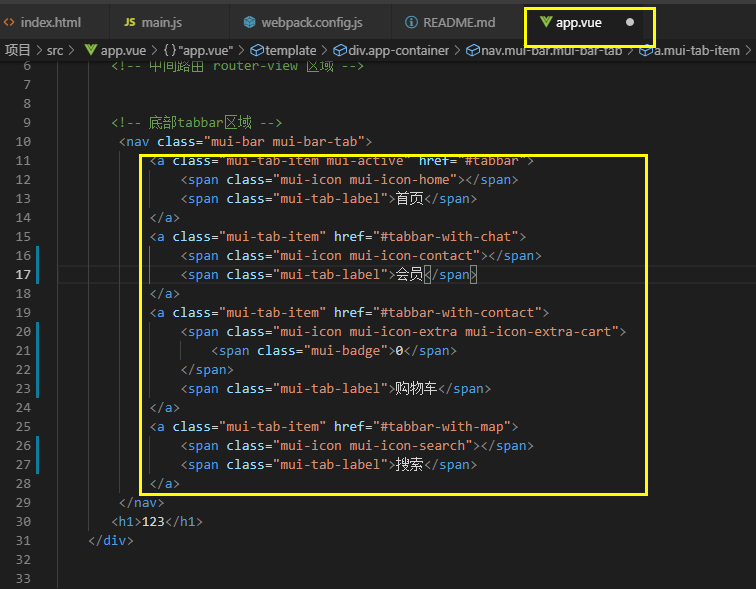


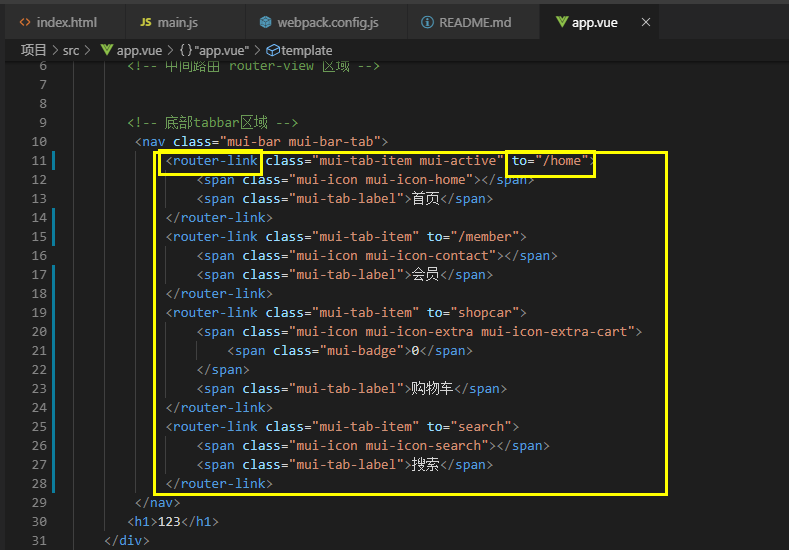
然后在中输入修改的说明文字，相当于git commit -m "提交信息" 引号中的说明，然后点击，现在就修改了本地仓库中的内容，

提交到Github中：

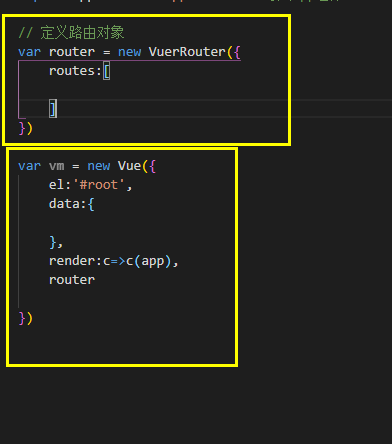


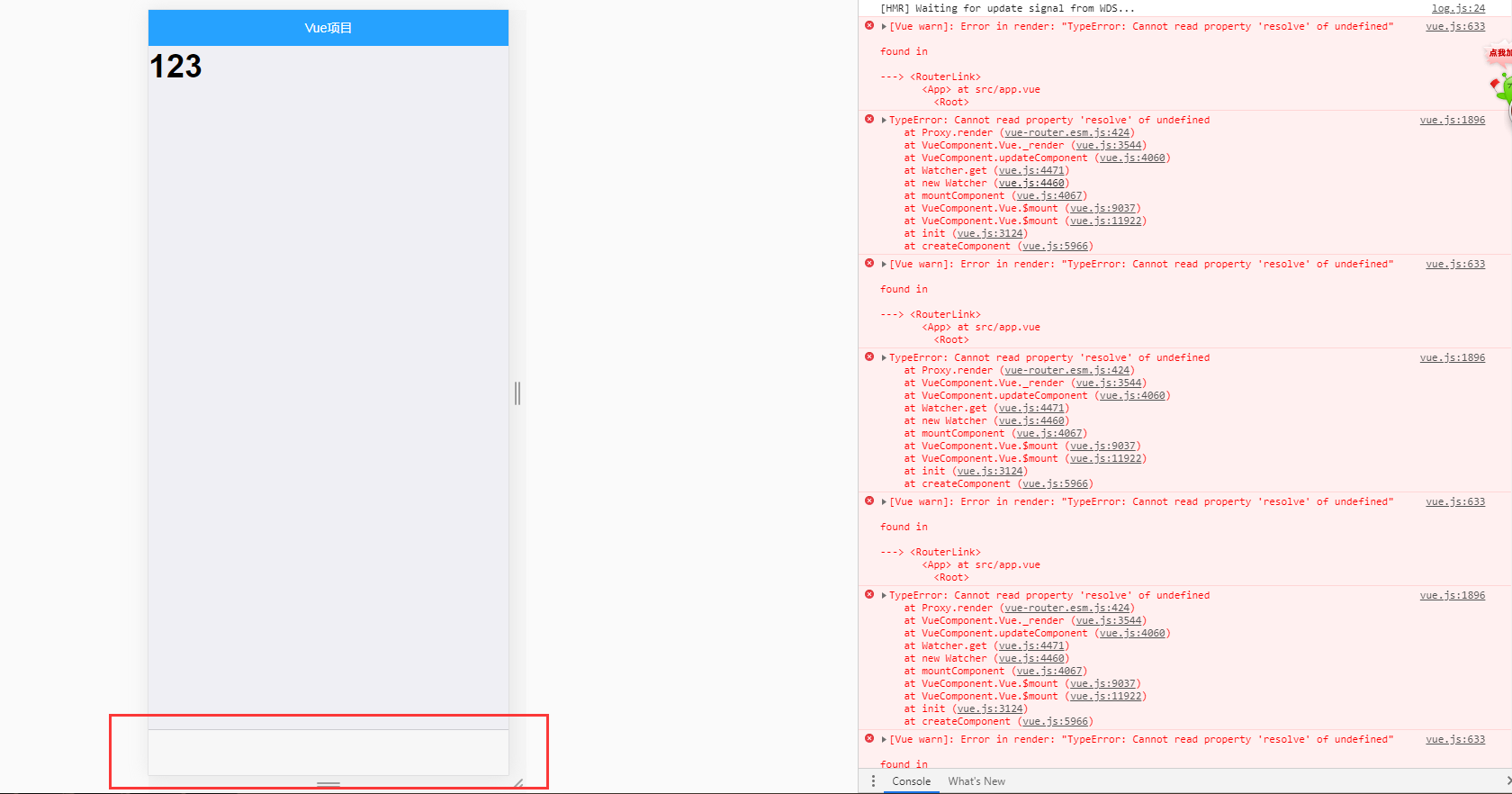
创建路由实例对象后，把底部的Tabbar的a标签改造成router-link标签：



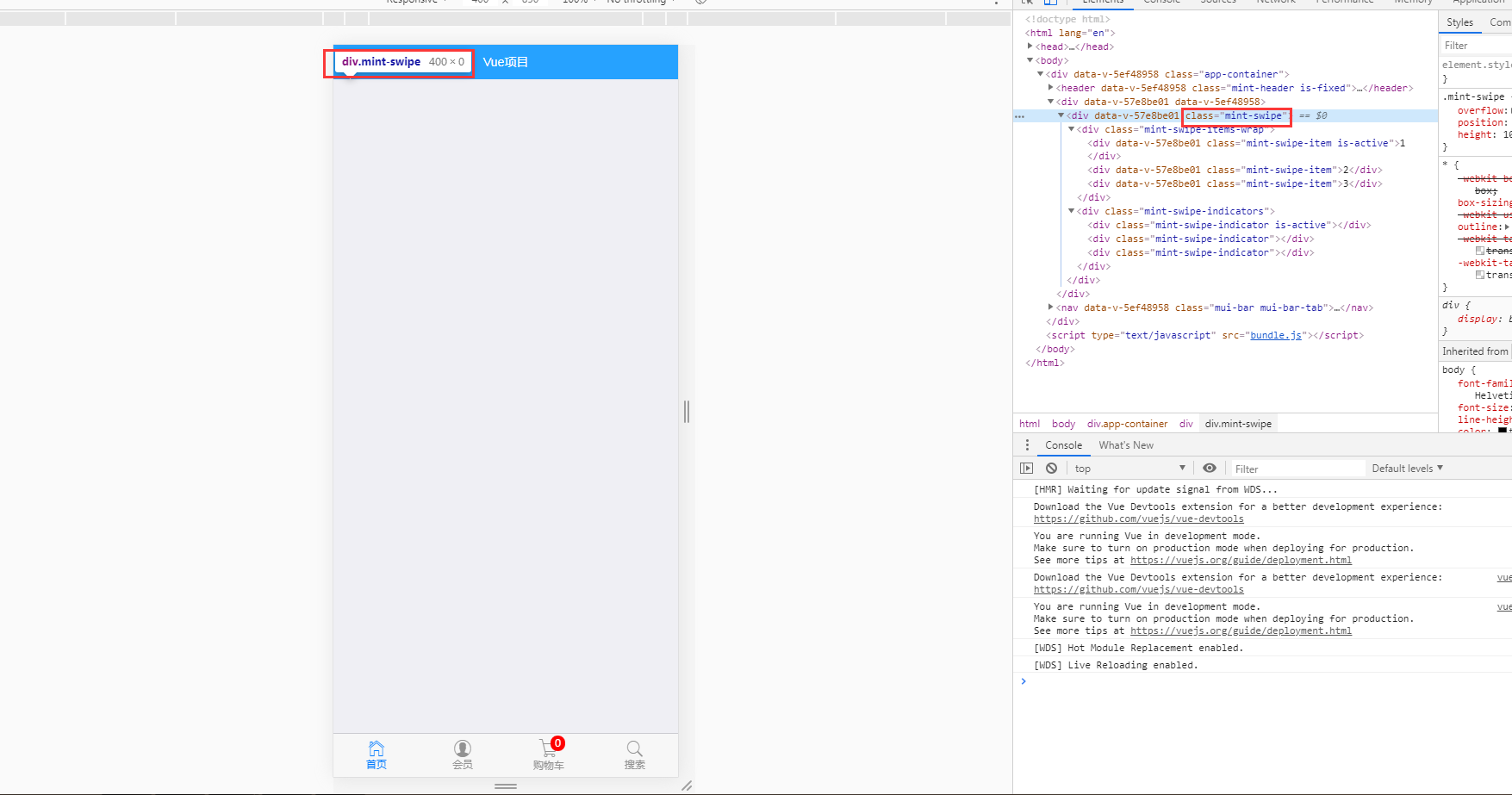


注意：路由对象一定要在vue实例对象之前，否则报一大堆错：





中间展示路由的部分，使用的是Mint-UI中的swipe组件，当在首页添加了轮播图组件后，没有内容，是因为这个组件本身的高度是为0的，所以要在该轮播图所属类中定义这个类的高度：



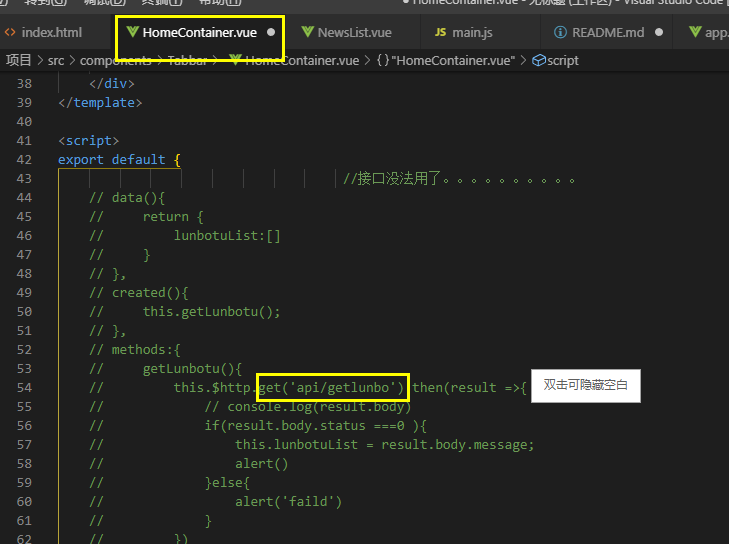
首页轮播图区域的图片宽高是100%，显示效果良好

使用vue-resource获取数据：先安装vue-resource，npm install vue-resouce -S

可以全局配置vue-resource的接口：



在后面需要使用接口地址的时候，直接请求地址即可：

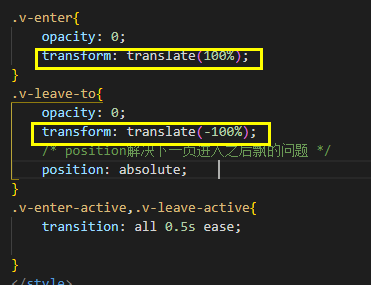


切换Tabbar的时候，样式设置好后出现的问题：

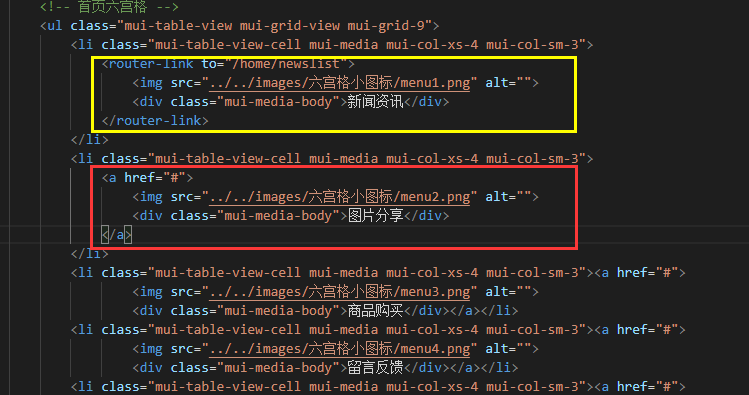
1. 切换tabbar的时候，header和tabbar会出现滚动条，动画进入时有一段空白期



1. 中间router-view切换的时候，上一个组件出去，下一个组件从下面进来
2. 下一个组件进来的时候飘的问题



把六宫格的a链接改造成路由的时候，a标签改成router-link：



新闻资讯使用mui中的media-list组件

新闻资讯列表改造成路由，实现点击跳转到新闻详情页：

GoodsInfo页面的轮播图区域，数据加载后的结果如下：



效果不是很好，应该让图片的高度是100%，宽度自适应

首页的轮播图和商品详情页的轮播图，显示效果不一样，原因是首页的轮播图宽高显示为100%，这种情况再商品详情页的轮播图不适用，商品详情页的高可以是100%，宽应为自适应

解：自定义一个属性，让使用轮播图的调用者，手动指定是否为100%的宽度

遇到的问题：





这是用getBoundingClientRect方法获取宽高，在商品详情页实现点击“加入购物车时”，小球不能跳到底部tabbar的购物车徽标里时出现的错误，TypeError是因为找不到用操作DOM方法获取底部tabbar的购物车徽标，



在控制台里输出b时，返回值是null，所以会出现TypeError错误

教程里的写法，无效：

el.style.transform = 'translate(${xDist}px,${yDist}px)';

百度后的写法：

el.style.transform = "translate("+xDist+"px, "+yDist+"px)";