第一个坑:

最终发现是缺少 vue-loader , 照官网敲一遍就知道了



除非你使用Vue公司的模板编译自己的分叉版本的高级用户，你应该安装vue-loader并vue-template-compiler在一起：

npm install -D vue-loader vue-template-compiler

vue-template-compiler必须单独安装的原因是您可以单独指定其版本。

每次vue发布新版本时，都会同时发布相应版本vue-template-compiler。编译器的版本必须与基础vue包同步，以便vue-loader生成与运行时兼容的代码。这意味着**每次升级vue项目时，都应升级vue-template-compiler以匹配它。**

[＃](https://vue-loader.vuejs.org/guide/#webpack-configuration)**的WebPack配置**

Vue Loader的配置与其他加载器略有不同。除了适用vue-loader于任何带扩展名的文件的规则外，请.vue确保将Vue Loader的插件添加到webpack配置中：

// webpack.config.js

const VueLoaderPlugin = require('vue-loader/lib/plugin')

module.exports = {

module: {

rules: [

// ... other rules

{

test: /\.vue$/,

loader: 'vue-loader'

}

]

},

plugins: [

// make sure to include the plugin!

new VueLoaderPlugin()

]

}

bunble 的安装

npm install -D babel-core babel-loader

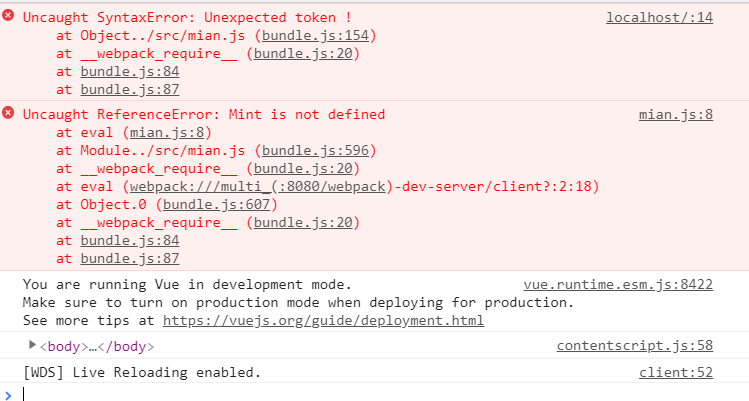
// webpack.config.js -> module.rules

{

test: /\.js?$/,

loader: 'babel-loader'

}

**2** 

**安装了 html-webpack-plugins,并且配置好**

**安装了babel.仍旧出现问题**

**未将**

**未将index.html中的script的bundle注释掉**

**3 mint-ui is not defined**

npm install babel-plugin-component -D

### 按需引入

借助 [babel-plugin-component](https://github.com/QingWei-Li/babel-plugin-component)，我们可以只引入需要的组件，以达到减小项目体积的目的。

首先，安装 babel-plugin-component：

npm install babel-plugin-component -D

然后，将 .babelrc 修改为：

{

"presets": [

["es2015", { "modules": false }]

],

"plugins": [["component", [

{

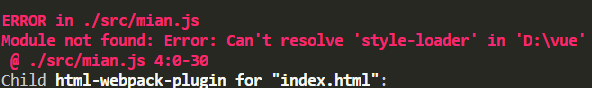
"libraryName": "mint-ui",

"style": true

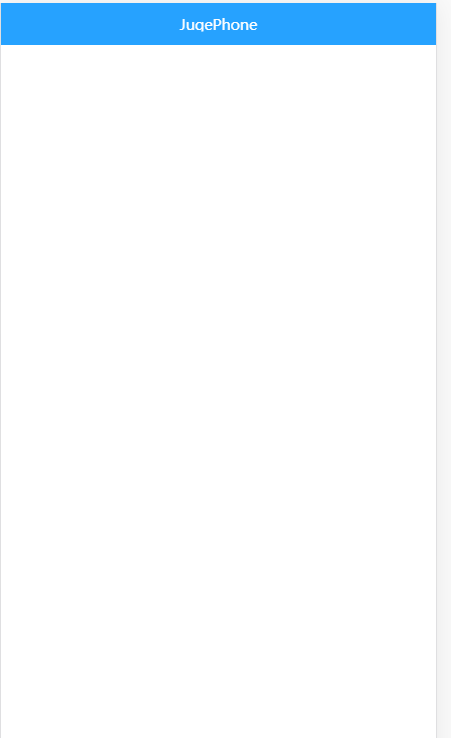
}

]]]

}

**4** 

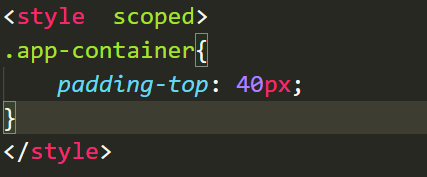
**安装 style-loader 问题解决**



**mint 组件的引入**



**5 报错 去除**<style lang="scss" scoped>



**6 .gitignore**

**忽略文件**

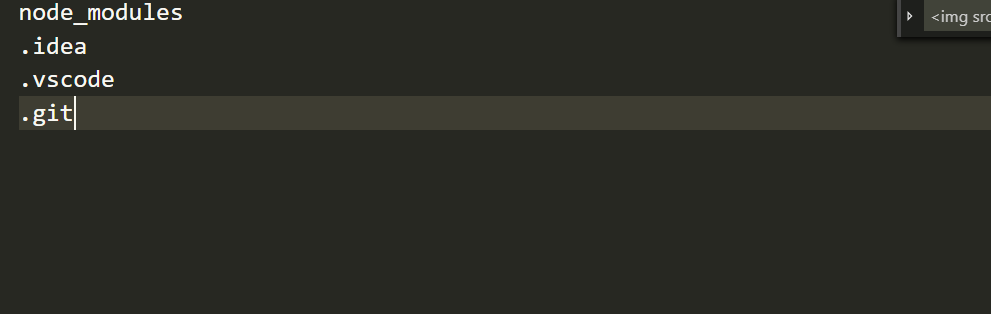
**1 新建.gitignore 文件**

**node\_modules**

**.idea (webstorm)**

**.vscode**

**.git**



**2 新建一个readme.md**



**3 MIT开源协议**

**LICENSE**

**git init**

**git status 展示所有文件的状态**

**git add . 添加追踪**

**git status 再次展示**

**git commit -m “init my project”**

**上传到远程仓库**

**1 github add.**

**2 git commit -m “提交信息”**

**3 git push**

**安装 bootstrap 和jquery**

npm install jquery@1.12.4 //jquery建议安装1.X.X的版本

npm install bootstrap@3.3.7

**webpack.config.js中**

plugins: [

new webpack.ProvidePlugin({

$: "jquery",

jQuery: "jquery", //之前之所以报错是把这里的jQuery 写成了jquery

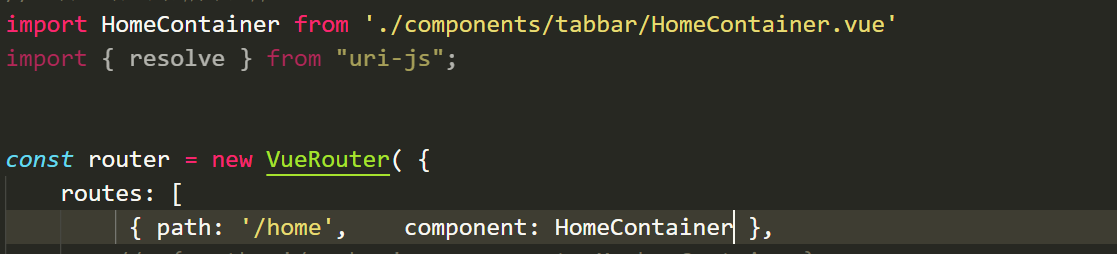
})

]

2.引入bootstrap  
安装：npm install --save bootstrap  
在入口文件main.js添加：

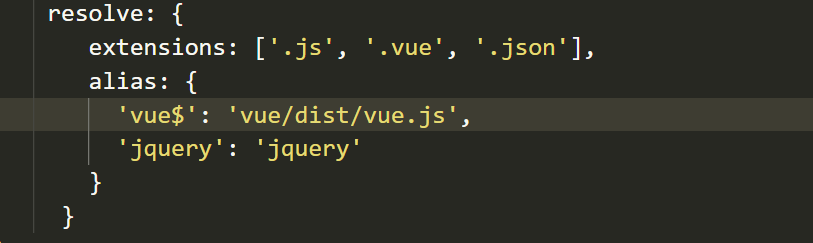
import 'bootstrap/dist/css/bootstrap.min.css'

import 'bootstrap/dist/js/bootstrap.min.js'





**引入bootstrap, 去掉”@”: resolve(‘src’)**



**css之类的文件用 直接import + ‘路径’ 就可以了**

**js 文件需要 import x from ‘xx’**

**配置路由的包:**

// 1.1 导入路由的包

import VueRouter from 'vue-router' ;

// 1.2 启用路由

Vue.use(VueRouter);

// 1.3 导入自己的路由模块

import router from './router'

*var* vm = new Vue( {

el: '#app',

render: *c* *=>* c (app),

// 1.4 挂载路由对象到vue实例上

router

})

**在路由模块中要导入路由**

import VueRouter from "vue-router";

*var* router = new VueRouter( {

rules: []

})

**一开始无显示.a标签改为 router-link 是因为没有把路由暴露出来**

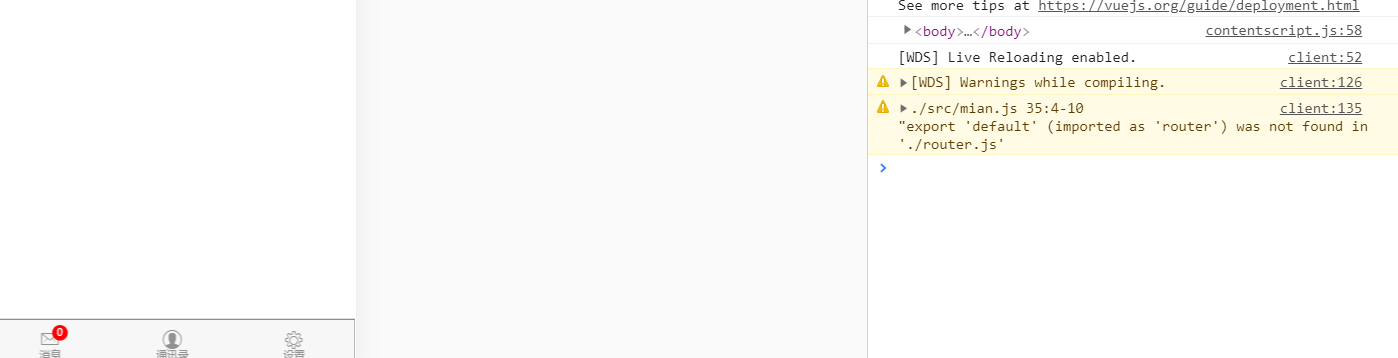
import VueRouter from "vue-router";

*var* router = new VueRouter( {

rules: []

})

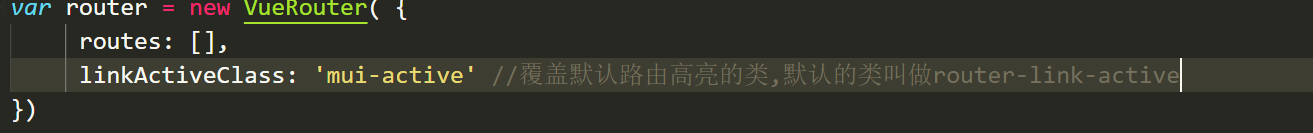
export default router;



**切换的时候高亮显示 默认是active**



mui-active 去除这个类,并在路由对象中 添加覆盖



**创建路由组件**



**路由 – 路由组件**

**每个组件 里都是模板写法**

**路由暴露之后 router.js 暴露到 外部 , main.js 导入路由组件**



**在app.vue 中为什么不显示 路由组件里的内容**

**router-view 的坑**

**# 不用加 ,这方面还没有理解哈希 路由的关系**

<nav class="mui-bar mui-bar-tab">

      <router-link class="mui-tab-item" to="/home">

        <span class="mui-icon mui-icon-home"></span>

        <span class="mui-tab-label">首页</span>

      </router-link>

      <router-link class="mui-tab-item" to="#/member"> //这里的#是不能加的,app.vue要和router.js 里path 中的 路径对应

        <span class="mui-icon mui-icon-contact"></span>

        <span class="mui-tab-label">通讯录</span>

      </router-link>

**router.js中**



**App.vue 中**



**第二部分 整体**

**2.1 首页轮播图**

**mint-ui**

## 引入

import { Swipe, SwipeItem } from 'mint-ui';

Vue.component(Swipe.name, Swipe);

Vue.component(SwipeItem.name, SwipeItem);

## 例子

基础用法

<mt-swipe :auto="4000">

<mt-swipe-item>1</mt-swipe-item>

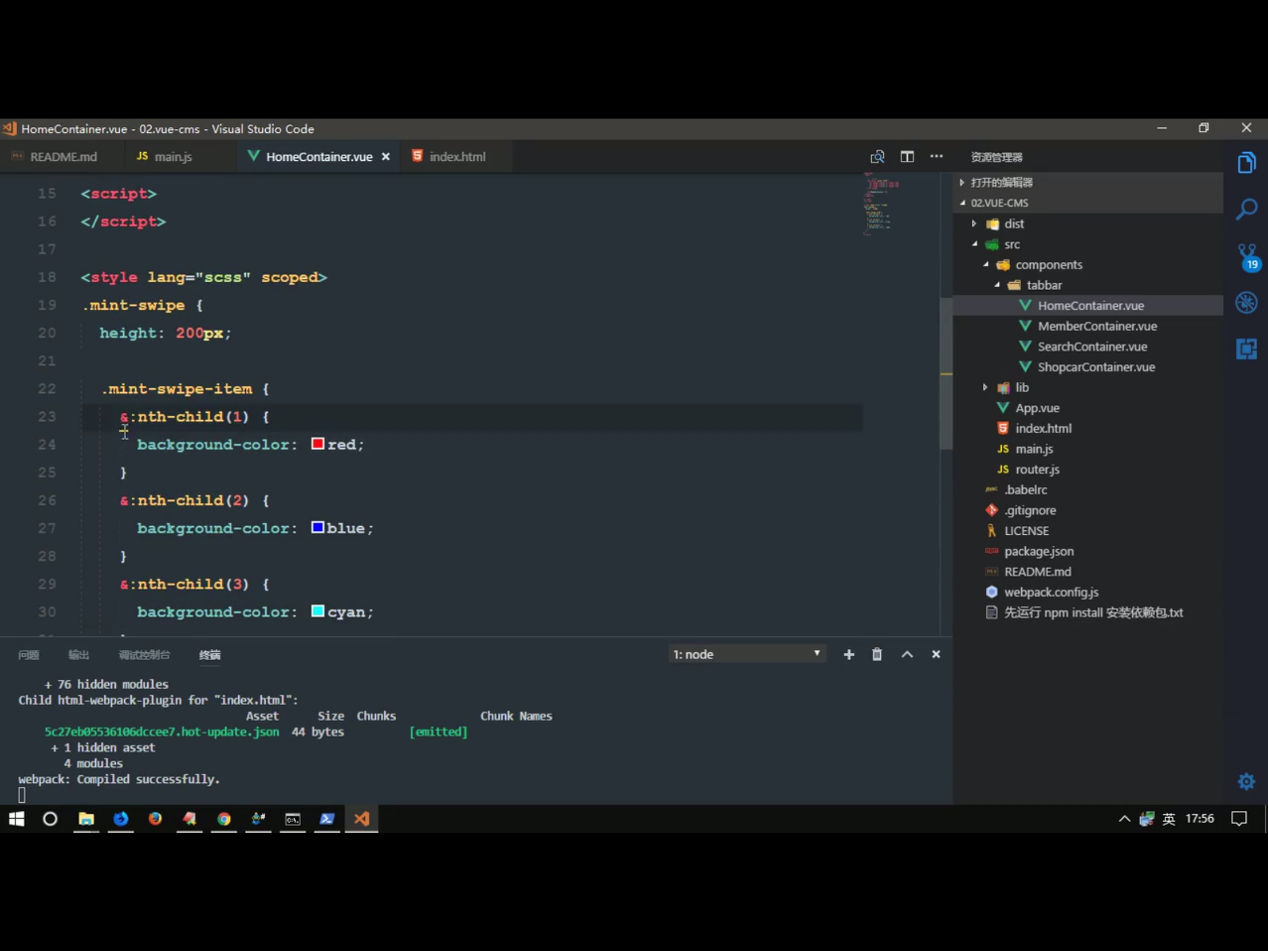
<mt-swipe-item>2</mt-swipe-item>

<mt-swipe-item>3</mt-swipe-item>

</mt-swipe>

**默认情况下,轮播图是没有高度的,要给他一个高度 覆盖类**

**scss写法**

****

**导入vue-resource 管理数据**

// 导入vue-resource

import VueResource from 'vue-resource';

Vue.use(VueResource);

**在各个组件写各个数据获取的方法,这个是跟后端连接的**

<script>

export default {

data(){

return {}

},

methods: {

getLunbotu(){ //获取轮播图数据的方法

this.$http.get('http://vue.studyit.io/')

}

}

}

</script>

**自己设置 url 地址 接口失败**



**yarn add i --save-exact axios vue-axios**

**添加这个并引入 更失败**

**最终导入vue-axios 不报错**

// 导入axios

import axios from 'axios';

import VueAxios from 'vue-axios'

Vue.use(VueAxios,axios);

**新错是接口地址失效,404**

**稀里糊涂就行了??**

**在 homecontainer中 设置一个接口 回顾自己是去除了vue-resource**

export default {

data(){

return {

lunbotuList: [] //保存轮播图的数组

}

},

created() {

this.getLunbotu();

},

methods: {

getLunbotu(){

this.$http.get(" https://www.easy-mock.com/mock/5cab1e59c0af8d1b182a4120/smart/lunbo", ).then(*response* *=>*{

if (response.status === 0){

// 获取轮播图成功

// this.lunbotuList = response.body.data;

Toast (' 加载轮播图chengg ')

}

})

.catch(

*error* *=>*{

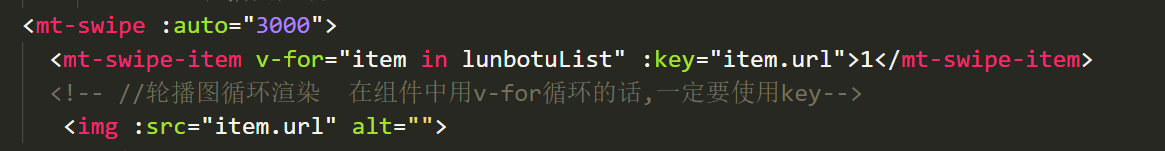
//获取轮播图失败

Toast (' 加载轮播图失败')

}

)

**轮播图获取之后要 循环绑定**



**这里占一个坑,以后自己搭建 接口 这个要学习easy mock 里的语法**

**组件中的循环绑定要用到key ,要用到表达式计算,需要用到v-bind,即 : src**

**2.2 九宫格**

**改样式**

**2.3 进入首页 让其默认到轮播图首页即 /home 路由重定向**

**在 routes 中进行路由规则匹配**

{ path: '/', redirect: '/home'},

**2.4 路有切换动画**

**在App.vue中用transition 包裹着router-view**

<transition>

<router-view></router-view>

</transition>

**2.4.1**

**为路由设置动画类**

**overflow-x**