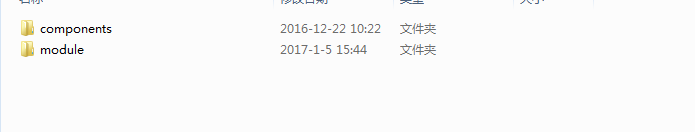
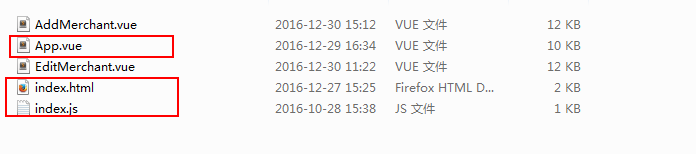
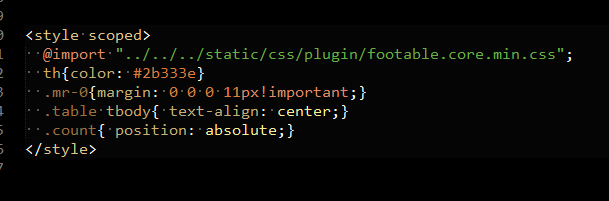
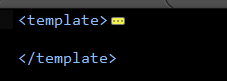
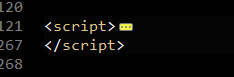
后台框架使用说明

1. 目前后台框架主要运用了Bootstrap，用Vue模板去渲染数据。
2. 运用webpack进行打包压缩加版本号，并实时监控页面变化,使用命令npm run dev启动每个项目的目录如上图所示（除dist外，README.md是介绍，通过npm install安装的node\_module中有些配置需要修改）
3. build和config文件主要是webpack打包和端口的信息配置
4. dist是打包以后才会生成的文件（npm run build执行此命令）
5. node\_modules是webpack的依赖模块，在node中使用npm install命令进行安装
6. static文件夹主要放的是一些静态资源。（CSS+JS+IMG+JSON+fonts等）
7. src 主要放的是你当前项目文件,一般是由组件+模块组成

components存放的就是你需要的组件

module就是当前需要的模块模块里面建好对应模块的文件名称即可。每个模块里面包含的文件其中红框标识出来的是每个模块都必要的

APP.vue里面由三部分构成

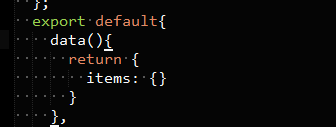
1. 样式表
2. 模板
3. Js逻辑处理

**Vue父子级的传值**

1. 父页面向子页面传值
   1. 例如： <editmerchant v-ref:profile></editmerchant>

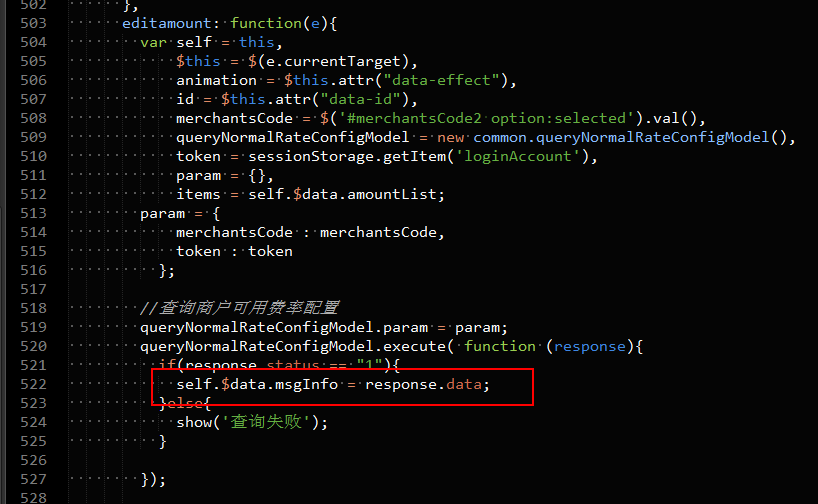
父页面通过赋值self.$refs.profile.items = items[i];把父页面的值放到items中存储，

（注：在一个父页面中有多个v-ref时，profile是你可以定义的名字，但是注意要和$refs调用的名称保持一致。）

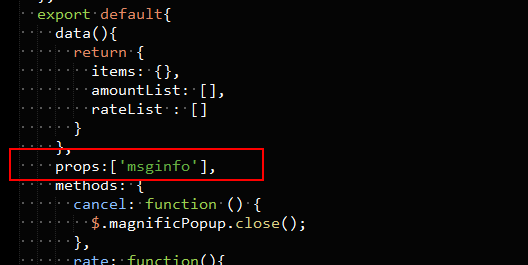
在子页面中直接调用items的方法属性即可。但是必须在初始化数据的时候return items: {}，不然没有预期效果。

* 1. 在父页面中绑定事件，调用子页面的方法，子页面通过props来触发。

父：<editamount v-ref:profile2 :msginfo="msgInfo" ></editamount>

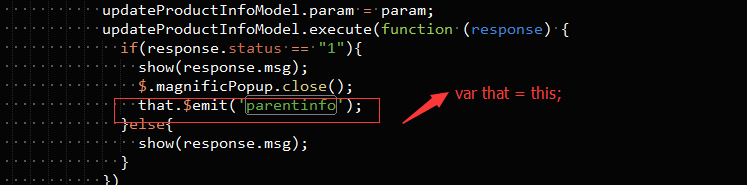


子：



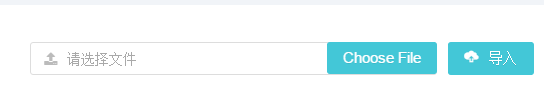
1. 子页面向父页面传值（子页面修改完成，父页面的数据不刷新得到更新）

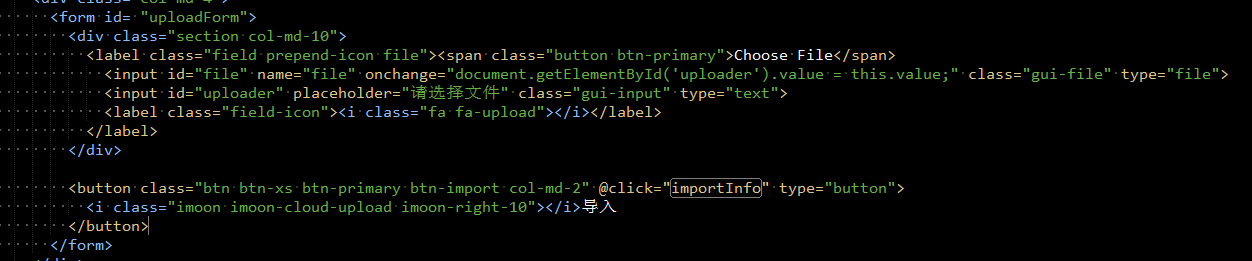
父页面通过v-on绑定对应的事件C:\Users\BAOWEN~1\AppData\Local\Temp\1484792691(1).png

子页面使用$emit 这样就实现了子页面传值给父页面啦。

**导入导出**

其实导入导出不应该写在这里面，但是，好多项目都遇到了，那就顺便写进来吧

1. 导入

导入是通过表单来做的，所以必须要使用FormData（使用FormData的最大优点就是我们可以异步上传一个二进制文件。）



1. 导出
2. 如果是get方式可以直接把下载的链接写到href里面 如：<a href=”xxx.com”></a>这种方式的局限性是不适合传很多参数。
3. 可以通过form表单解决。

同时再也不怕需要传递很多个参数啦O(∩\_∩)O（截图部分是主要代码）

目前想到的就是这些，有不对的地方，请欢迎指正。好让我继续改进。哈哈，谢谢大家花时间来瞅瞅，第一次，写的不太好，请大家多理解。