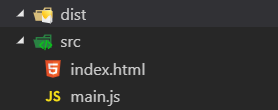
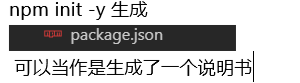
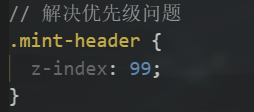
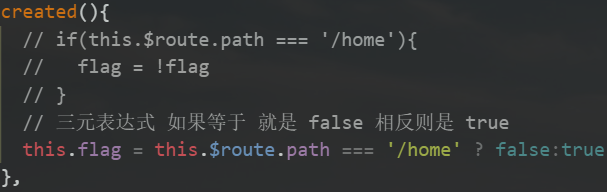
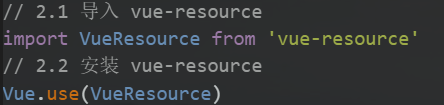
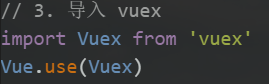
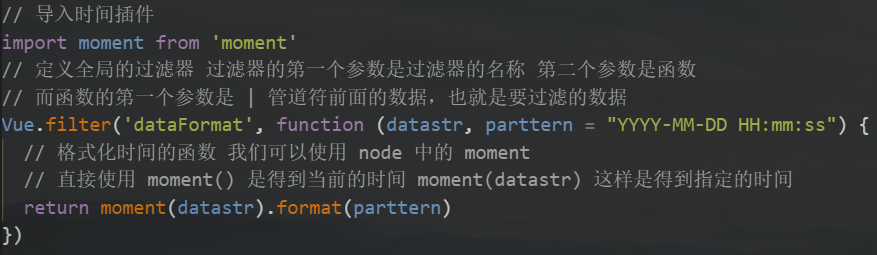
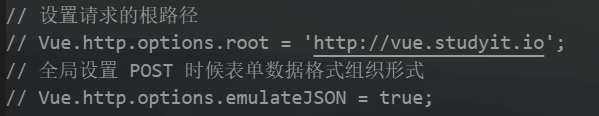
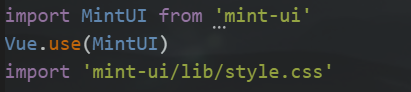
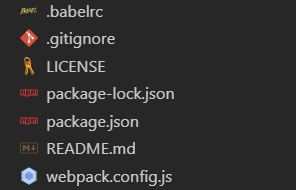
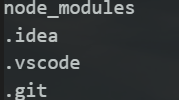
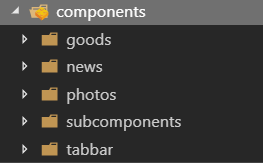
# 项目总结

个人收获经验总结：

在一个项目中，首先要做到是项目目录结构的清晰、完整，结构要让人更容易读；其次，在写代码的时候，格式的正确，变量名和函数名要见名知意，在另一个地方使用的时候，避免出现拼写错误，所以，推荐去原地复制；另外，在项目的伊始，要考虑到哪些组件是可能会被重复调用的，重复调用的要把此组件作为子组件；最后，我觉得在一个项目中最重要的能力是解决问题以及分析问题的能力，因为表面工作绝大多数人都能够做好，处理问题的能力却不见的，这方面对于我来说要加强。因为日后的工作中，不管是做什么，考上了公务员或还是当了个前端工程师，都要记得这一点：你的工资是取决于你的不可替代性，做人做事亦是如此。

个人对项目总结：

1. 初始化项目
   1. 构建项目基本的目录结构
      1. 
   2. 初始化webpack
      1. 
      2. 做各种配置，详情看onenote 总结
      3. Webpack 是一个集成的打包工具，非常重要
2. APP.vue 界面注意
   1. 这是首页，一般移动端app首页会分成三个部分，顶部Header区域、中间路由组件切换区域以及底部Tabbar区域
   2. 注意：在每个.vue文件中，要注意template模板字符串中必须要有一个根元素，我们可以换一种方式理解，就是每个template中都需要一个container，这样也方便我们在scss中使用嵌套方式写css
   3. 很多东西都是有现成的，我们只需要把现成的东西拷贝过来，最不理想的结果是自己写代码实现一些通用的功能，因为别人写的已经很完善了，各种方面都考虑齐全。移动端最常用的两个组件库是 mint-UI 和MUI 这两个组件库极大的减少了我们的代码量，让我们更加专注于业务逻辑，这就是我们使用已经封装好的东西带给我们的好处
   4. 注释是必须要的，这是程序员的良好的行为习惯，也是增加了代码的可读性，这一点在这之后就不重复了
   5. 代码问题：
      1. 优先级问题：在你检查完元素之后，发现组件之间没有padding但是依旧重叠在一起，这就是优先级问题
         * 
      2. 在vue实例的生命周期中，使用最多的是created和mounted，前一个表示在vm实例被创建就起作用的时候，即同时和页面一起被创建、后面则是有一些组件在创建过程中需要引入相应的js，这时候就需要等页面中的所有的dom渲染好之后再挂载到vm中
      3. 在简单的if判断语句中，如果用三元表达式就会显得更简洁，可读性也更高
         * 
      4. Vm实例中监听路由变化可以用watch
         * 
      5. 首页的返回键问题，这时就要区别$router和$route的不同，详情见onenote vue总结拔高区
3. Main.js 代码总结
   1. 这是项目所有js的入口文件
   2. 注意导入vue的一些功能是有先后的
      1. 
      2. 
      3. 
   3. 定义全局的过滤器也在此页面中定义，node中提供了时间插件moment，简化了我们自定义时间过滤器
      1. 
   4. 为了我们更方便的请求数据，在main.js中我们可以这样做
      1. 
      2. 在以后的请求中就要加上/开头
   5. 导入Vuex之后要写个一个new Vue.Store 并且要注意 Store一定是大写开头
      1. 其中有三个参数，第一个参数是state 相当于data，如果调用state中的参数时，可以这样用 this.$store.state.\*\*\*，state中的数据只推荐读取
      2. 第二个参数就是mutations相当于methods，这个里面是定义函数的，另外每个方法的第一个参数都是state，最多可以有两个参数，如果修改之后，一定要通过mutations的方法来修改数据this.$store.commit(‘方法名称’，‘唯一个参数’)
      3. 第三个参数是getters相当于计算属性，也相当于filters，引用的话是this.$store.getters.\*\*\*
   6. 导入相应mintui要注意同时还要导入相应的css
      1. 
      2. 当然我们也可以按需导入mintui组件，但是这样就不能实现懒加载的小图标
      3. 懒加载使用步骤详情看mintui官网
4. Router.js 代码总结
   1. 第一步导入vue-router
   2. 所有的路由组件都是在这里导入
   3. 
   4. 最后一步要把路由对象暴露出去
5. 完成了三大页面的基本构建之后，在根目录下还需要配置一些其余的文件
   1. 
   2. .gitignore是上传git的时候需要忽略的文件，因为文件没必要而且会使项目冗杂
      1. 
   3. LICENSE是一个主流的开源协议
   4. READMA.md是项目的说明文件
   5. Webpack.config.js是webpack的配置文件
6. 把所有的组件都写在一个components中
   1. 

这就是我通过此次手机商城项目的浅总结，难点，主要是在于购物车中的数据几个地方要同时改变，这首先要用到vuex这个公共的仓库，其次，因为我获取不到数据，这点对我来说不好实现，等我学完node.js我再考虑来实现这一功能。