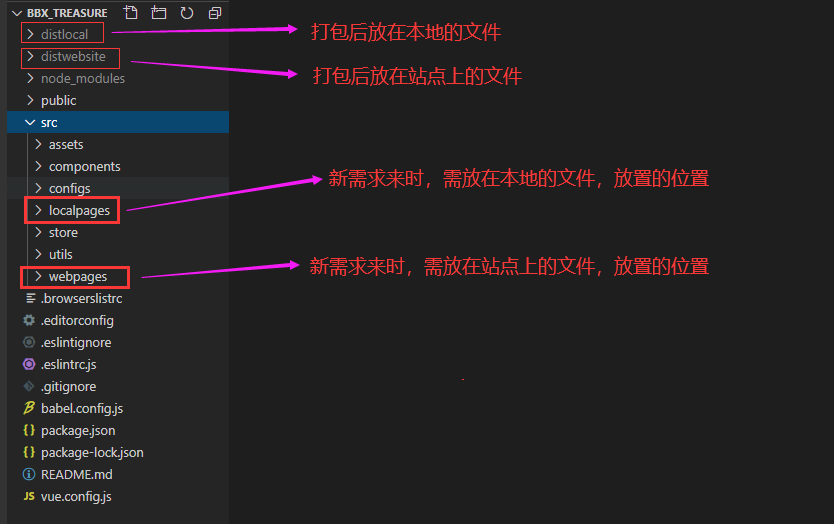
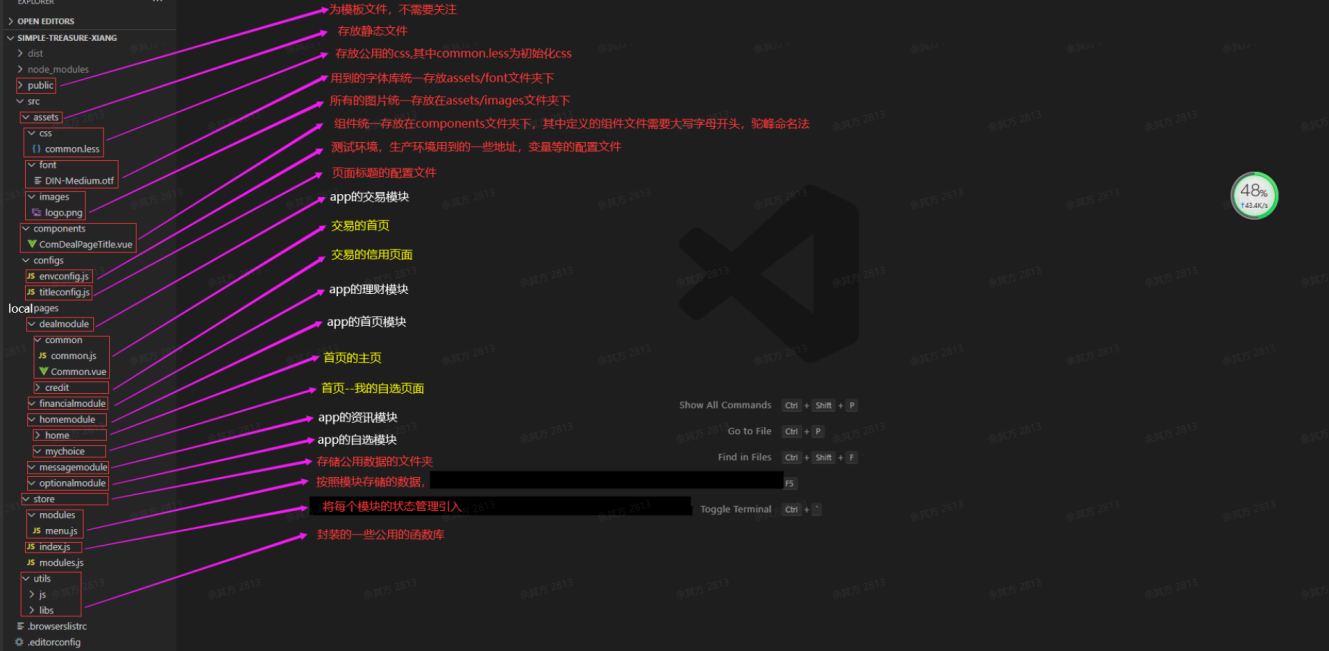
**bbx\_treasure介绍说明**

**一.使用的技术栈：**

vue@2.6.12, 组件库使用vant@2.11.1, vuex@3.4.0, less@3.0.4，eslint@6.7.2，jquery@1.12.4



**二.项目目录介绍：**



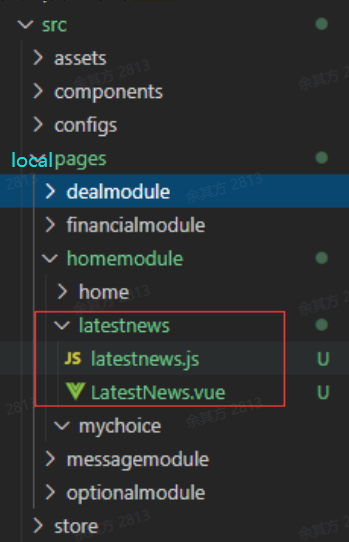
**三.做新需求时如何新建一个页面？**

**（这里以新建放在本地首页--最新资讯页面为例）**

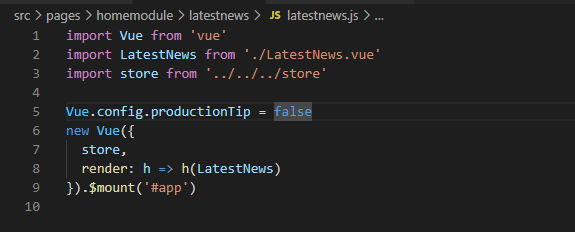
提示：放在站点上的代码新建时，放在src/webpages目录下

1.在src/localpages/homemodule文件夹下新建一个latestnews文件夹

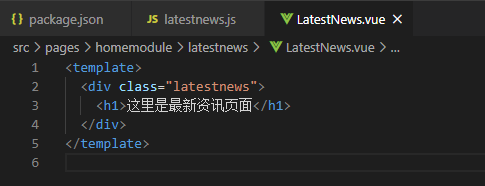
2.在src/localpages/homemodule/latestnews文件夹下新建一个latestnews.js文件和LatestNews.vue



latestnews.js文件内容如图：

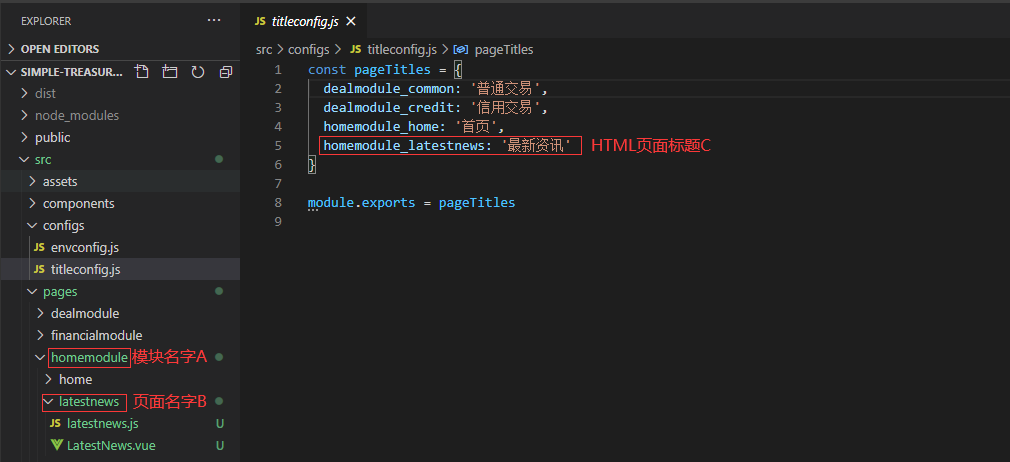


LatestNews.vue文件内容如图：

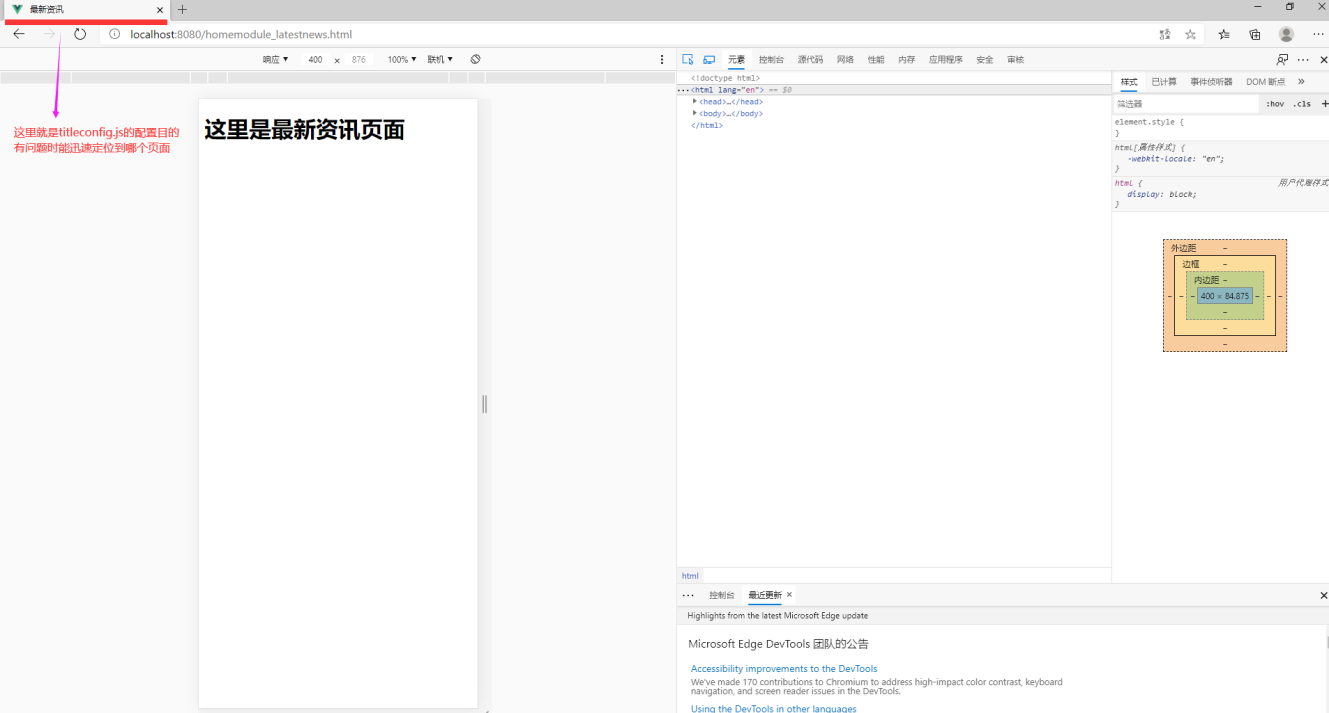


3.在src/configs/titleconfig.js文件配置页面title标题

标题key值配置规则是 key = '模块名字A' + '\_' + '页面名字B'



4.查看新建页面的效果：<http://localhost:8080/homemodule_latestnews.html>



**四.组件的懒加载：**

问题一：为什么使用组件开发

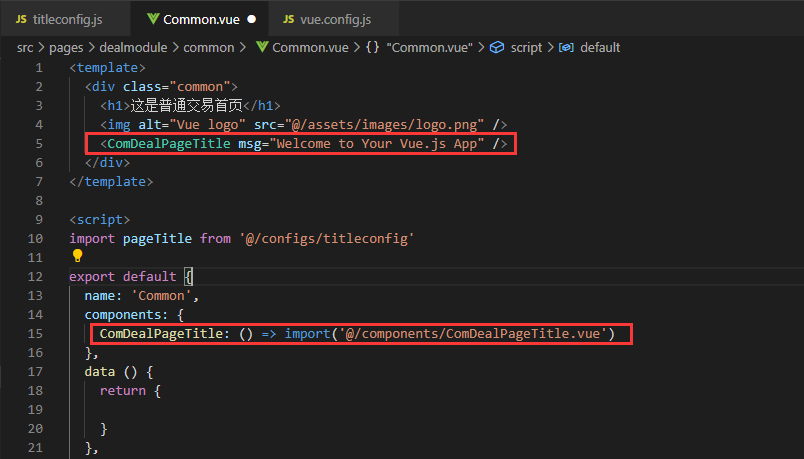
减少代码体积，提高开发效率，避免重复劳动

问题二：为什么使用懒加载？

答：使页面加载速度更快，为给客户更好的客户体验。尤其一个页面tab很多时，建议使用懒加载

问题三：如何使用懒加载？

(两行核心代码如图)



**五.父子组件的传值：**

父组件：



子组件：



**六.vuex store状态存储的使用(非必须，当页面tab很多，组件拆分的很小时建议使用)：**

1.往store里存值--方法一

import { mapActions } from 'vuex'

methods: {

changeState() {

this.setPlayMode(data)

},

...mapActions([

'navClickAction'

]),

...mapMutations({

setPlayMode

}),

},

// store中使用

actions: {

navClickAction ({ commit, state }, payload) {

commit('SET\_MENU', payload)

},

},

mutations: {

SET\_MENU(state, menu) {

state.leftMenuItem = menu

}

}

2.往store里存值--方法二

this.$store.commit('setTreeList', res) // 同步

this.$store.dispatch('navClickAction', data) // 异步

const mutations = {

setTreeList(state, data) {

state.allTree = data;

}

}

3.使用store里存的值--方法一

this.$store.state.allTree

4.使用store里存的值--方法二

import { mapState } from 'vuex'

computed: {

...mapState({

count: (state) => state.allTree

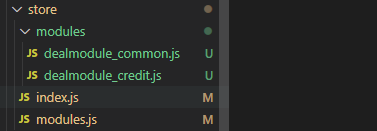
})

}

5.状态管理分模块后如何传值

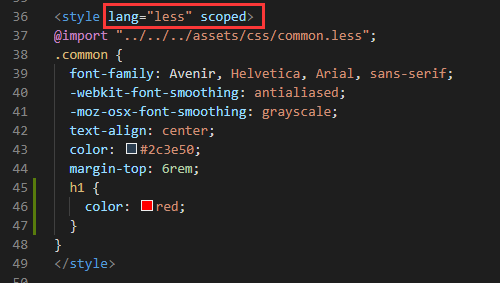
this.$store.dispatch('dealmodule\_common/actionsPhoneType', 'IOS')

this.$store.commit('dealmodule\_common/changePhoneNum', '18004568888')



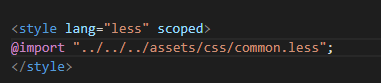
**七.style样式的使用：**

需加上lang="less" scoped

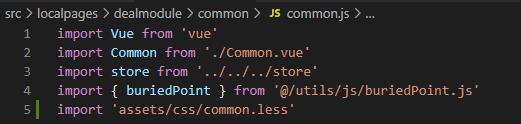


**八.如何引入初始化样式**

方法一：



方法二：



**九.如何启动本地服务**

需放在本地的文件：npm run serve:local

需放在站点的文件：npm run serve:web

**十.如何打包代码**

打包放在本地的文件：npm run build:local 会生成一个distlocal文件夹

打包放在站点的文件：npm run build:web 会生成一个distwebsite文件夹

**十一.打包好的文件夹distlocal， distwebsite如何放到develop分支**

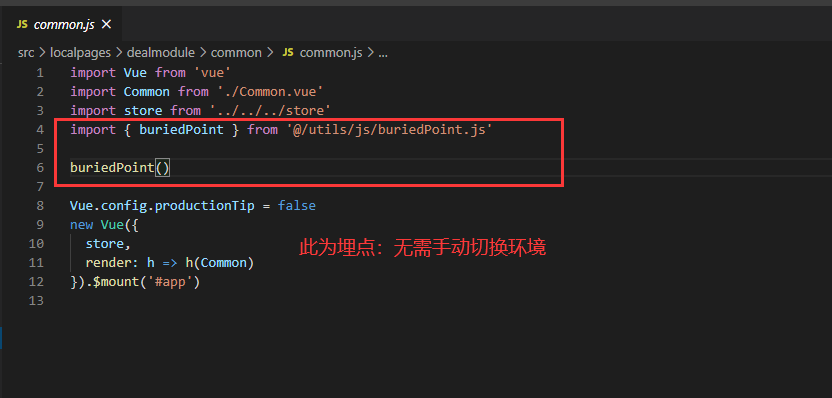
打包好的distlocal文件直接替换AppCfg\CfgBeta\webApp\app下的distlocal文件夹

打包好的distwebsite文件直接替换AppCfg\other\webSite\webApp\app下的distwebsite文件

整体直接替换会自动分辨出改动的了哪些文件，未改动的不会发生变化，（规划好分支的基础上）无需区分哪些需要静默哪些不需要静默

**十二.如何区分生产测试环境这里以埋点为例**

通过调用张颖封装的方法：xcscWebGetCacheData



**十三.注意事项：**

1.封装组件时，不可存在异步调接口维护的数据

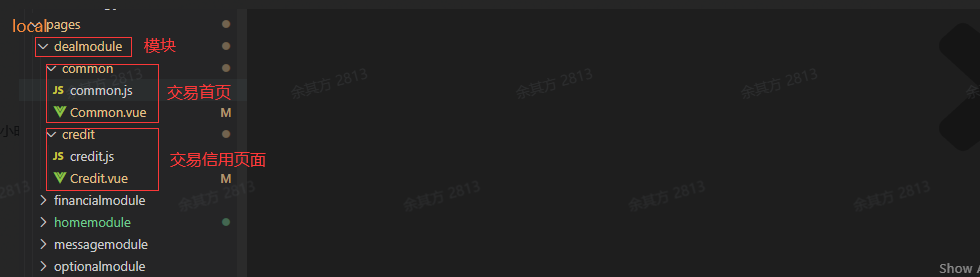
2.组件封装时，首字母要大写，采用驼峰命名法

3.书写样式时必须加上scoped

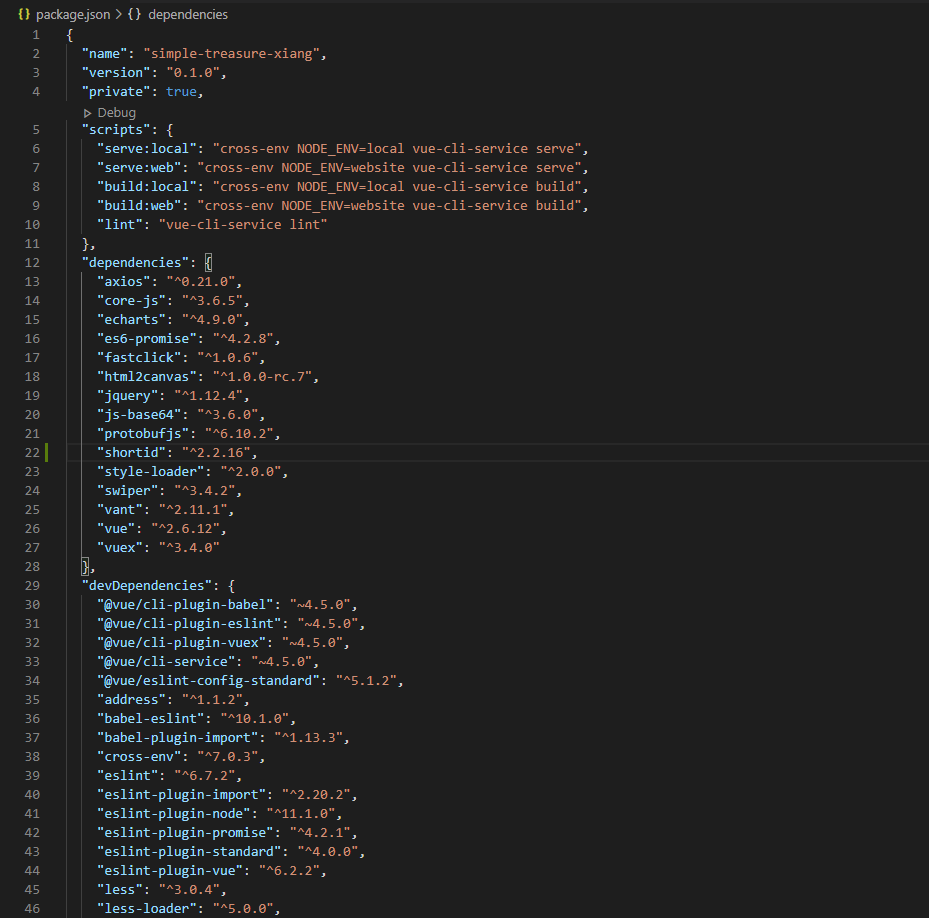
4.为保证代码风格统一规范，易于维护，不可关闭eslint开关

5.localpages 和 webpages文件夹下的模块及页面层级不可随意更改

6.执行npm install时会提示tsclient的问题，需把通达信封装的tsclient拷贝到node\_modules里去



package.json的内容为：



**百宝湘develop分支老代码升级迭代**

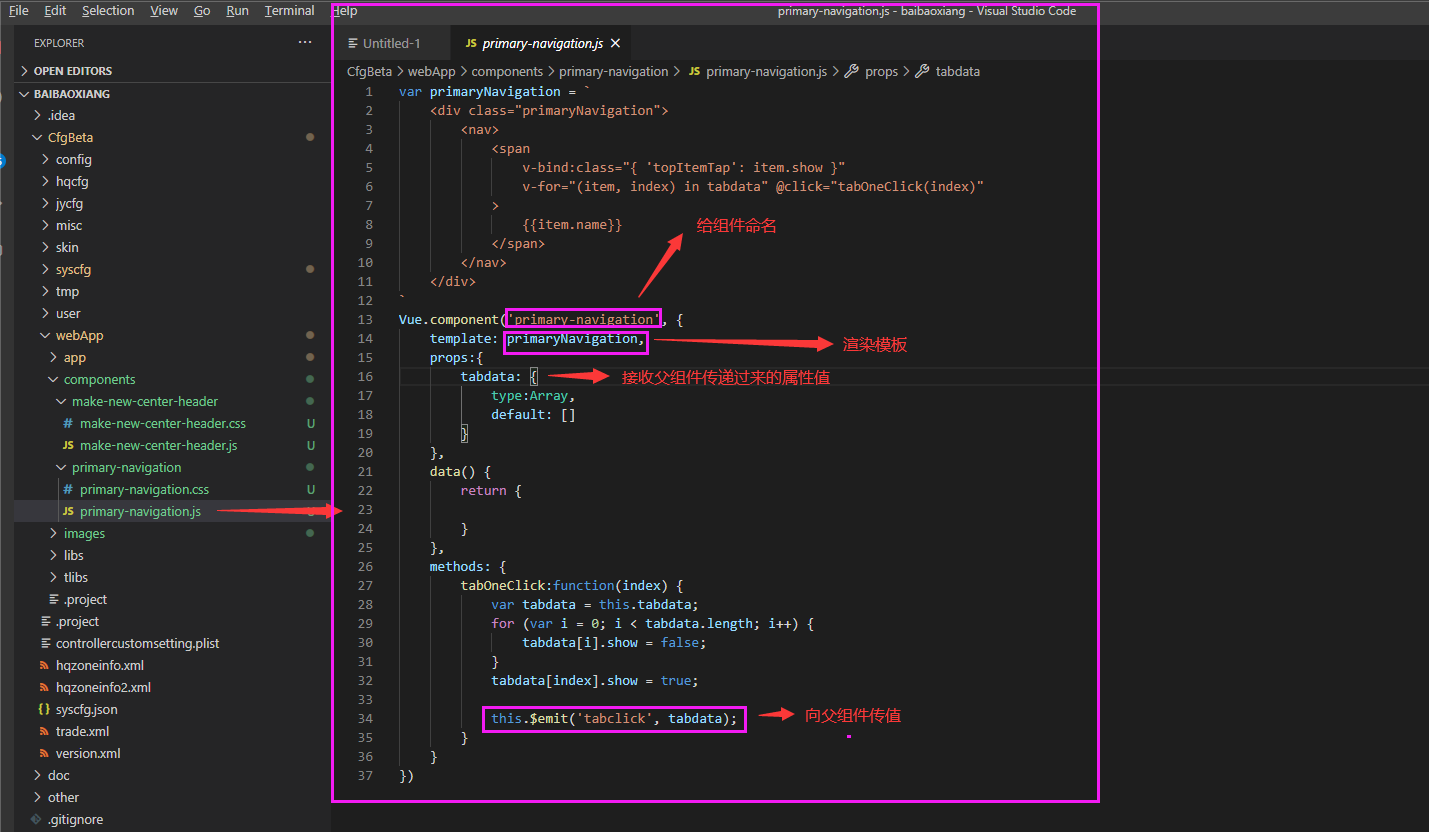
**一.需改动的位置：**



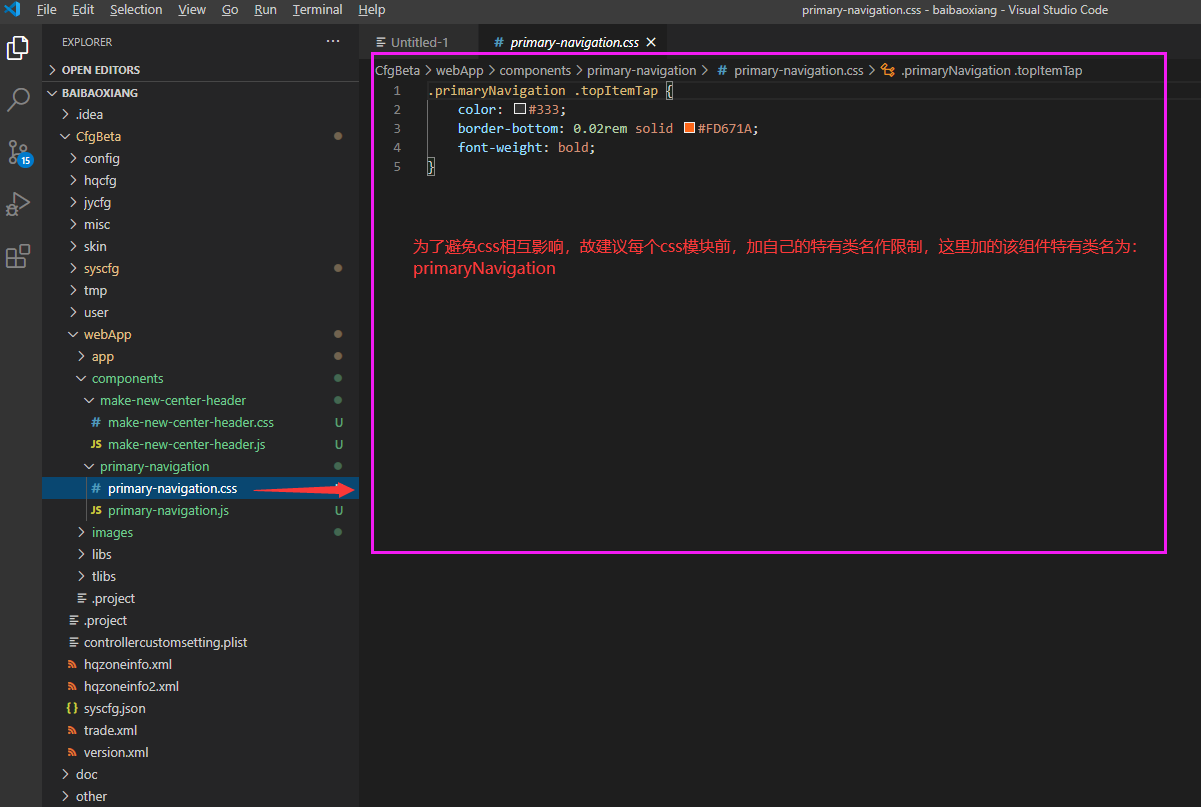
**二.develop分支如何组件化开发**

举例说明：如何使用组件B

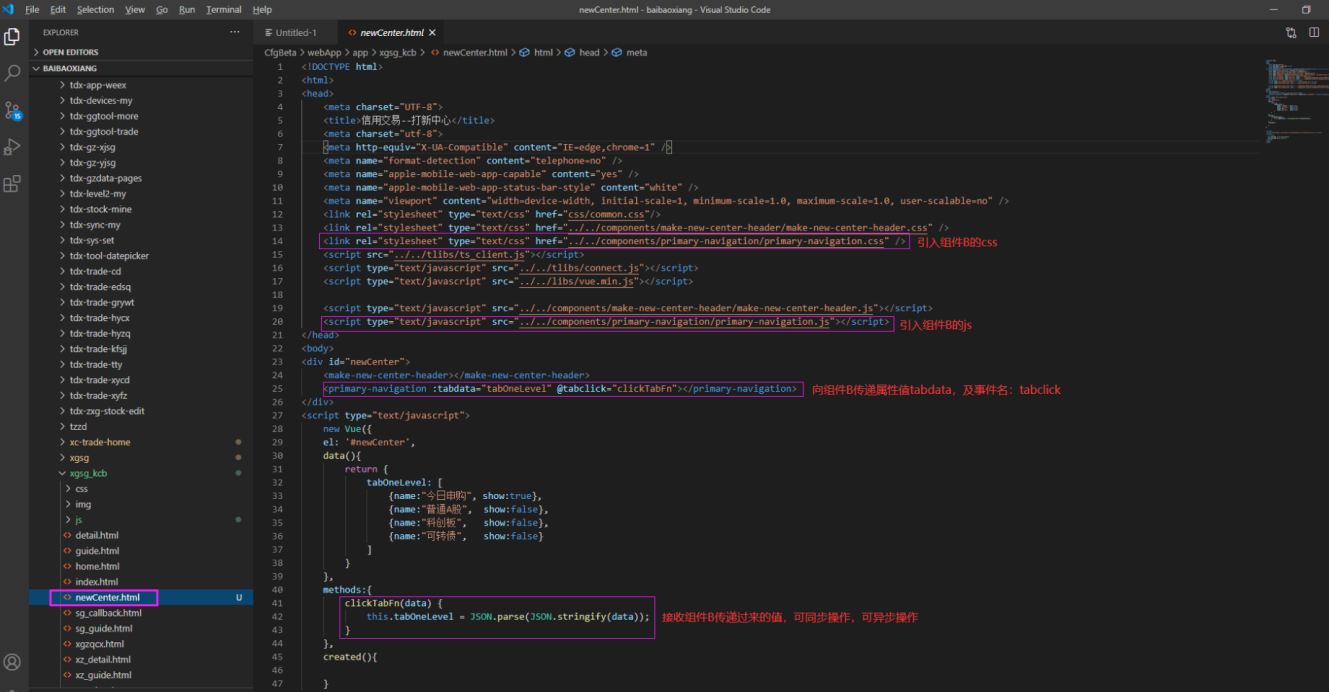
1. 组件B的js内容



2. 组件B的css内容



3. 父组件如何使用组件B



父组件向子组件传递属性时，属性要小写

解决低版本兼容性：

