Component构造器 | 文档 (qq.com)

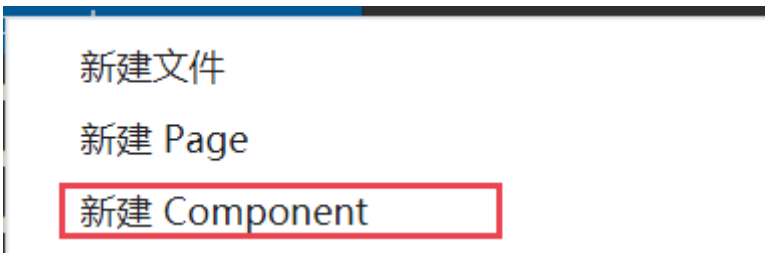
组件生命周期 | 微信开放文档 (qq.com)

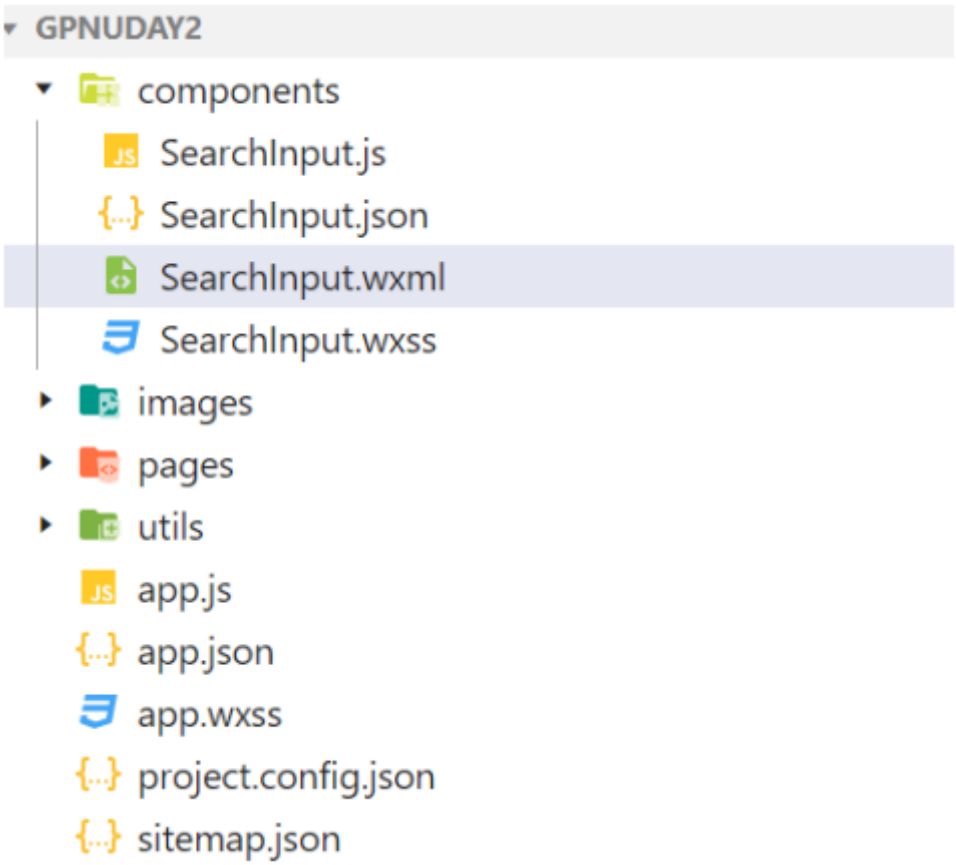
组件模板和样式 | 微信开放文档 (qq.com)

Component 构造器 | 微信开放文档 (qq.com)

1.新建自定义组件文件

在根目录新建 components 目录，然后右键新建自定义组件目录 SearchInput

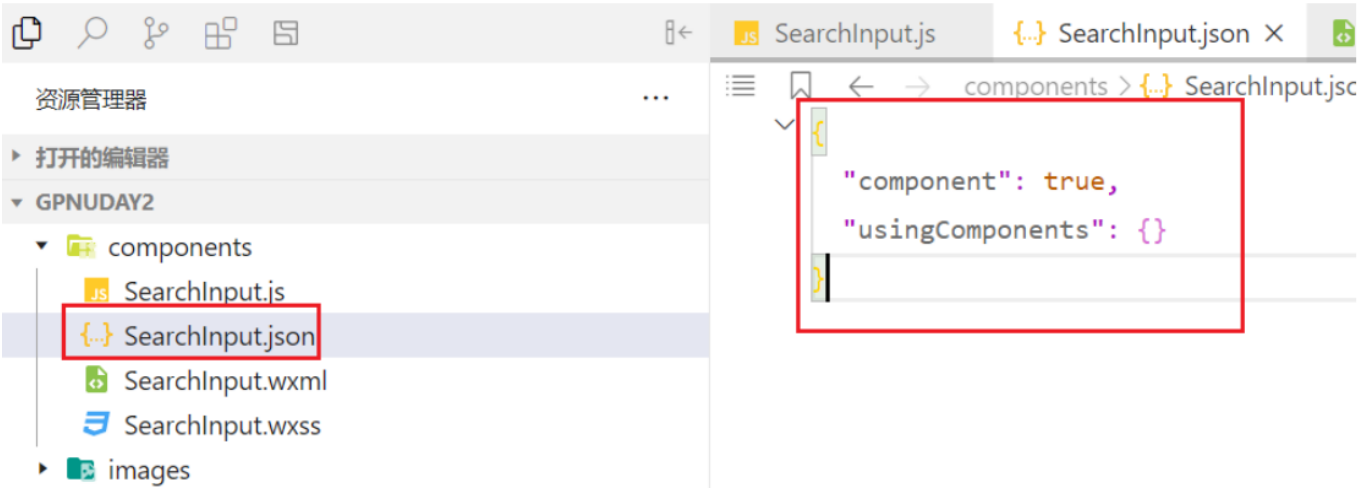




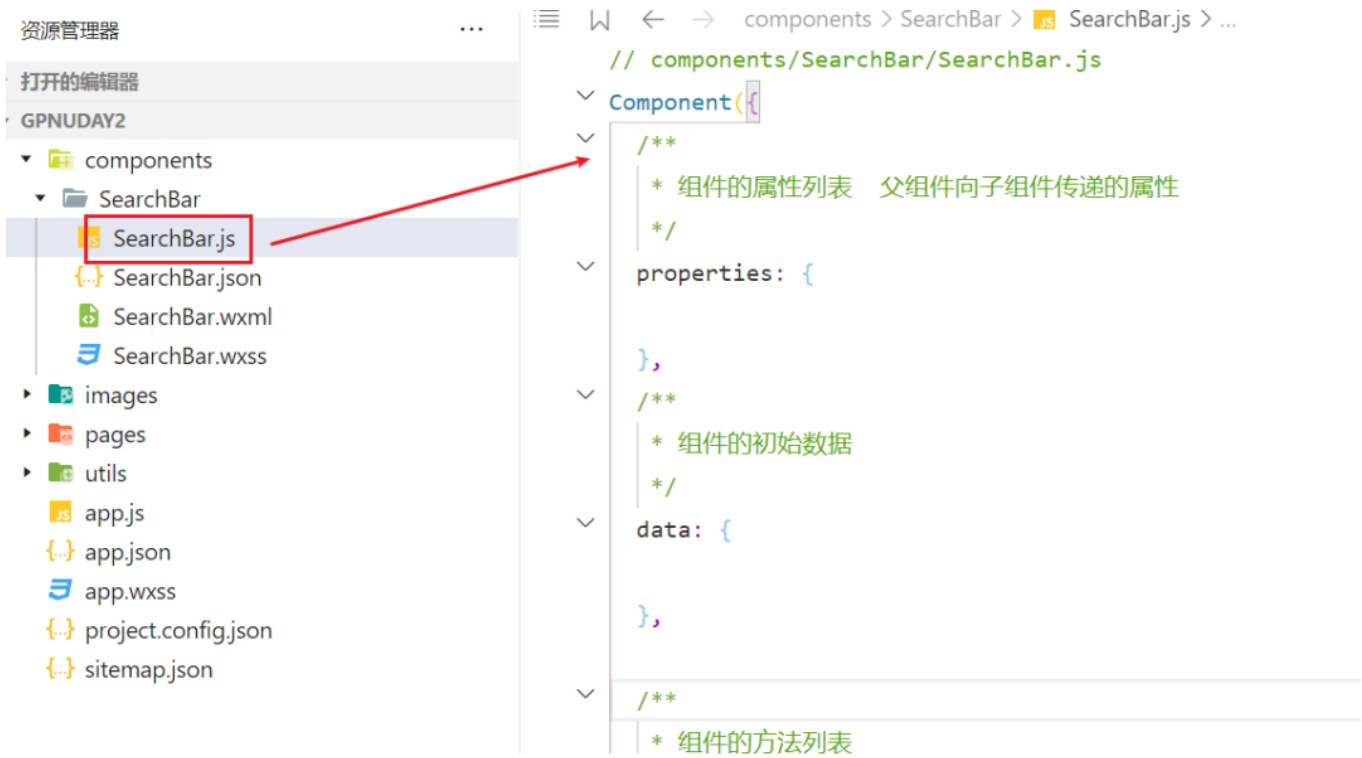
2.配置组件json 文件

这里配置Component 构造器 代表 开启构造器

一个自定义组件也可以视为一个页面，故页面也可以使用 `Component` 构造器构造，拥有与普通组件一样的定义字段与实例方法，其必要配置项（ `json` ）与正常自定义组件一致，即需要有`component: true`字段。



3.配置组件js文件



4. 在SearchInput.wxml 文件中写入内容

```
<view class="search_input">
    <navigator url="/pages/logs/logs" open-type="navigate"> 搜索 </navigator>
</view>
```

这里跳转logs页注意情况，如果在我们的tabBar（app.json）配置了按钮，使用navigator跳转不生效，建议换个地址跳转

5.在SearchInput.wxss 配置样式

```
.search_input {
    height: 90rpx;
    padding: 10rpx;
    background-color: var(--themeColor);
}

.search_input navigator {
    height: 100%;
    display: flex;
    justify-content: center;
    align-items: center;
    background-color: #fff;
    border-radius: 15rpx;
    color: #666;
}
```

app.wxss

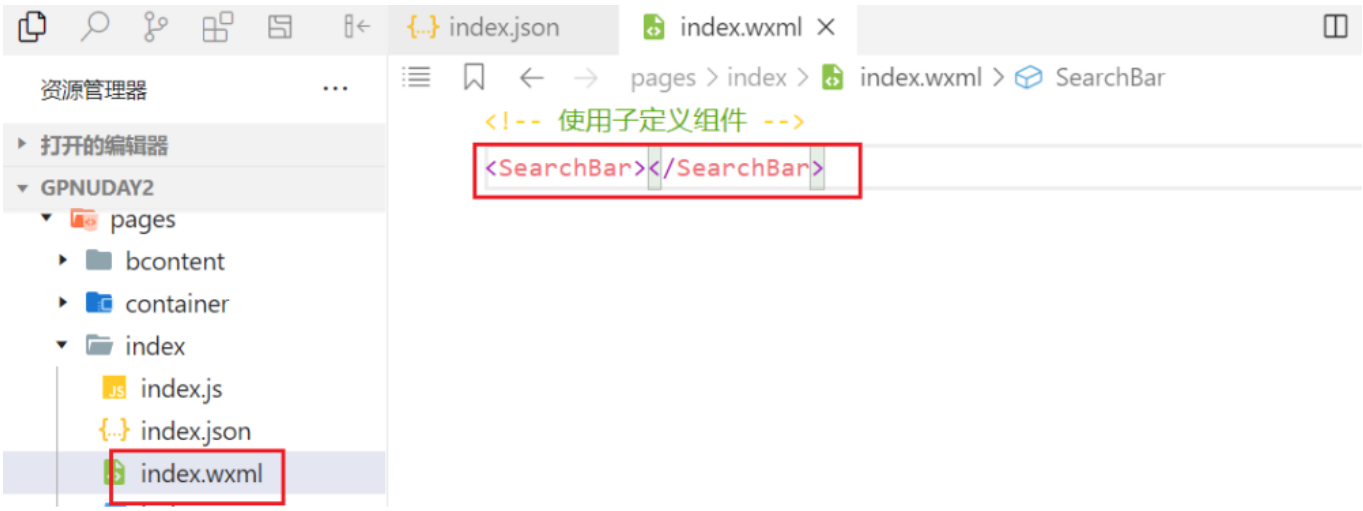
```
page{
    /**定义主题颜色**/
    --themeColor:#eb4450;
}
```

6.引入自定义组件

以在 /pages/index/index 父组件中引入子组件为例

a.在父组件json 文件的usingComponents 中导入组件

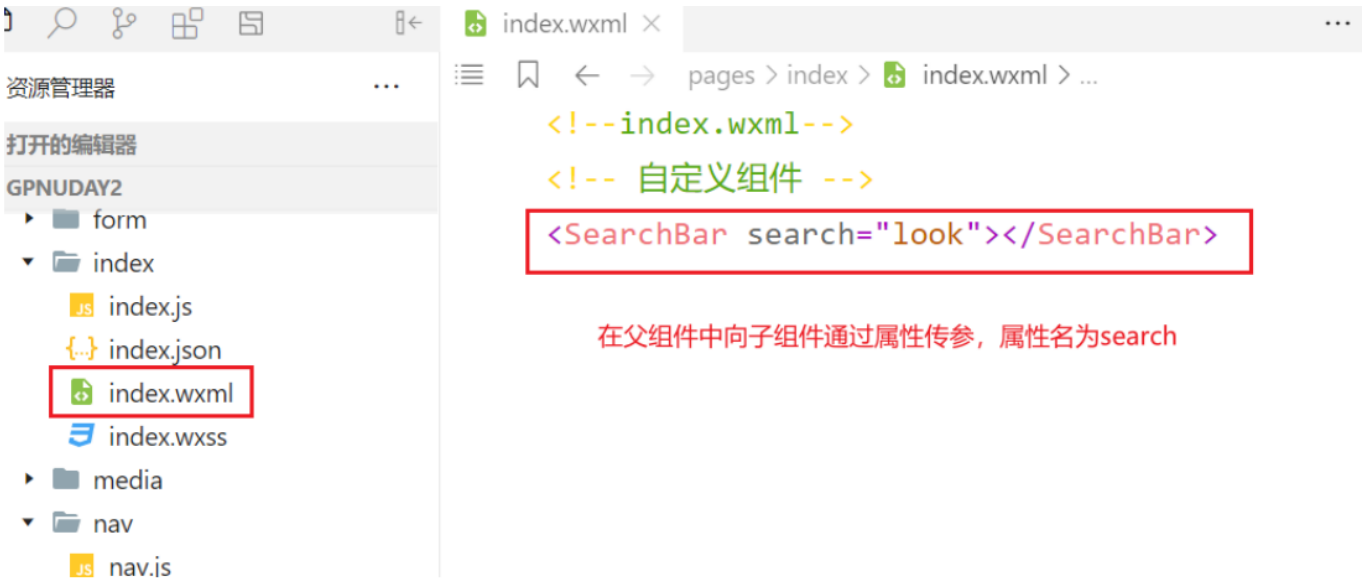
b.在父组件wxml 文件中以组件名称作为标签使用组件



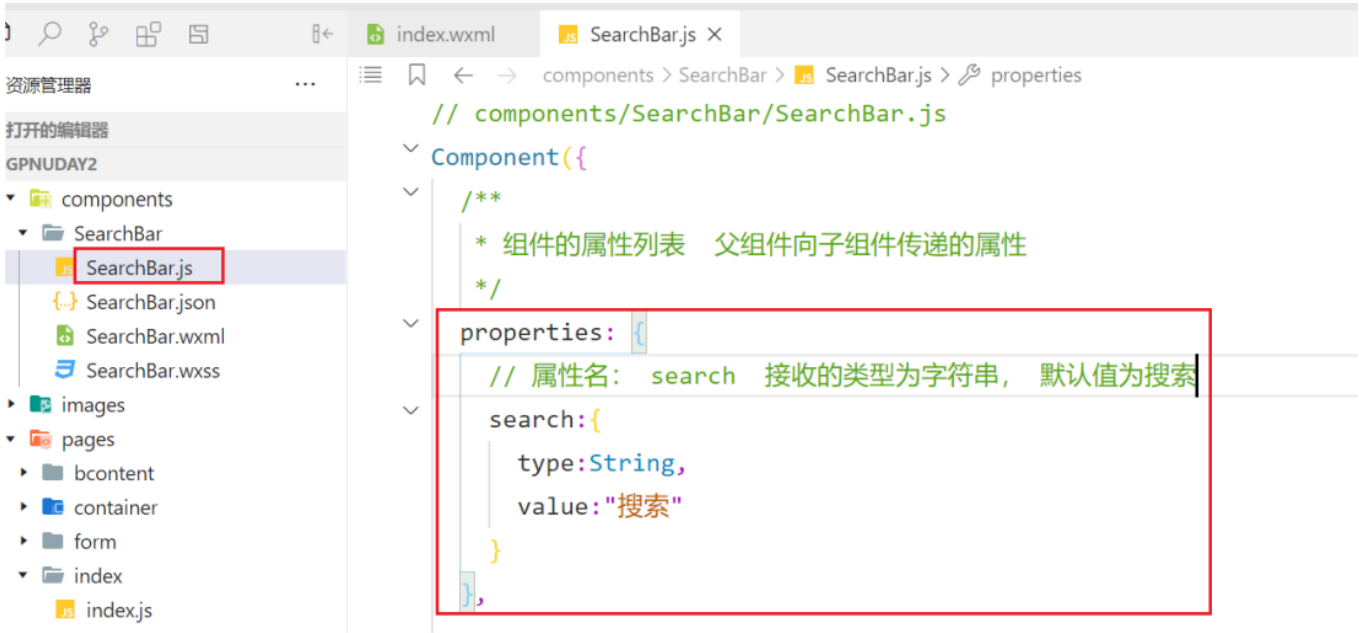
7.组件参数传递 - 父传子

注：与VUE父子组件传参原理一样

在父组件中的子组件标签添加属性，给子组件传递数据



在子组件 js 文件中通过 properties 接收，可以指定接收数据类型



在子组件 wxml 文件中的内容替换成变量

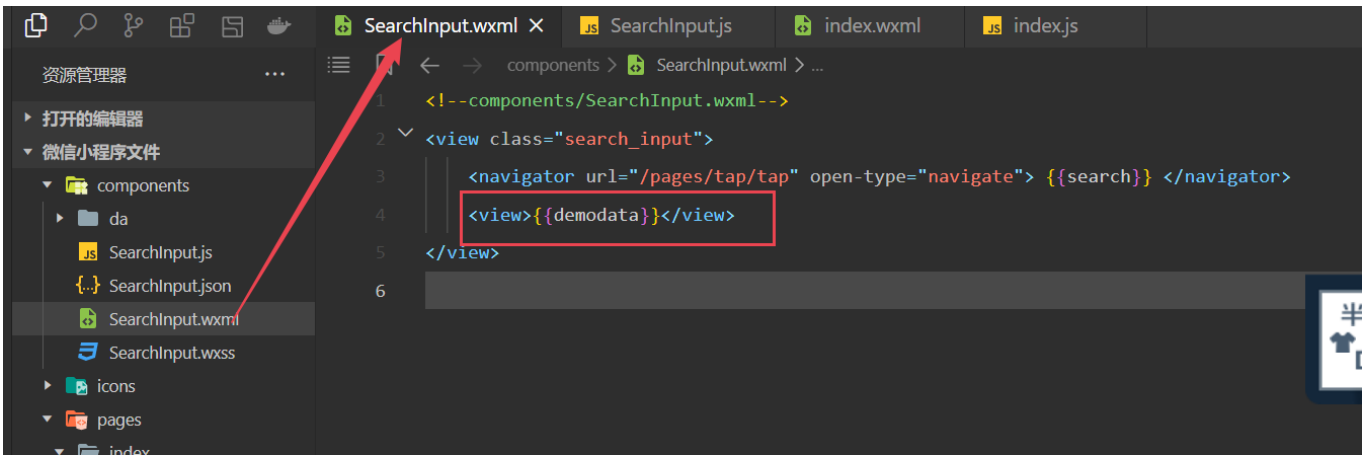
效果如下：

8.父传子进阶

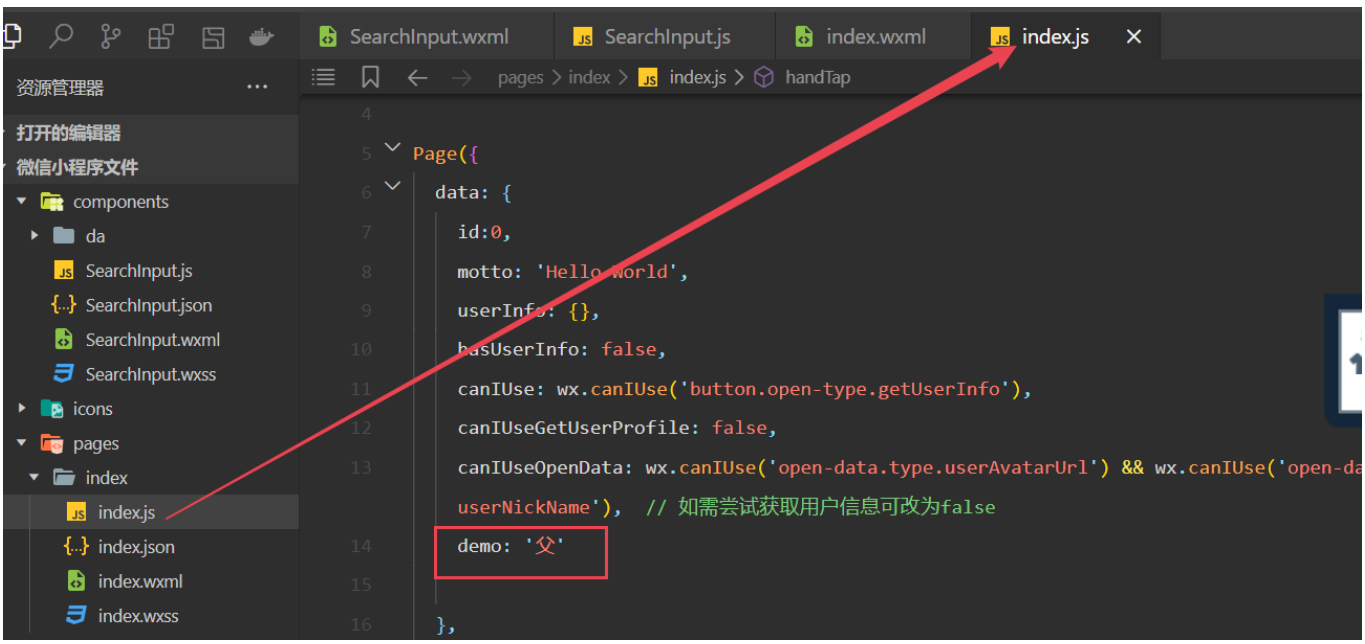
子传父：https://blog.csdn.net/XH_jing/article/details/111563278

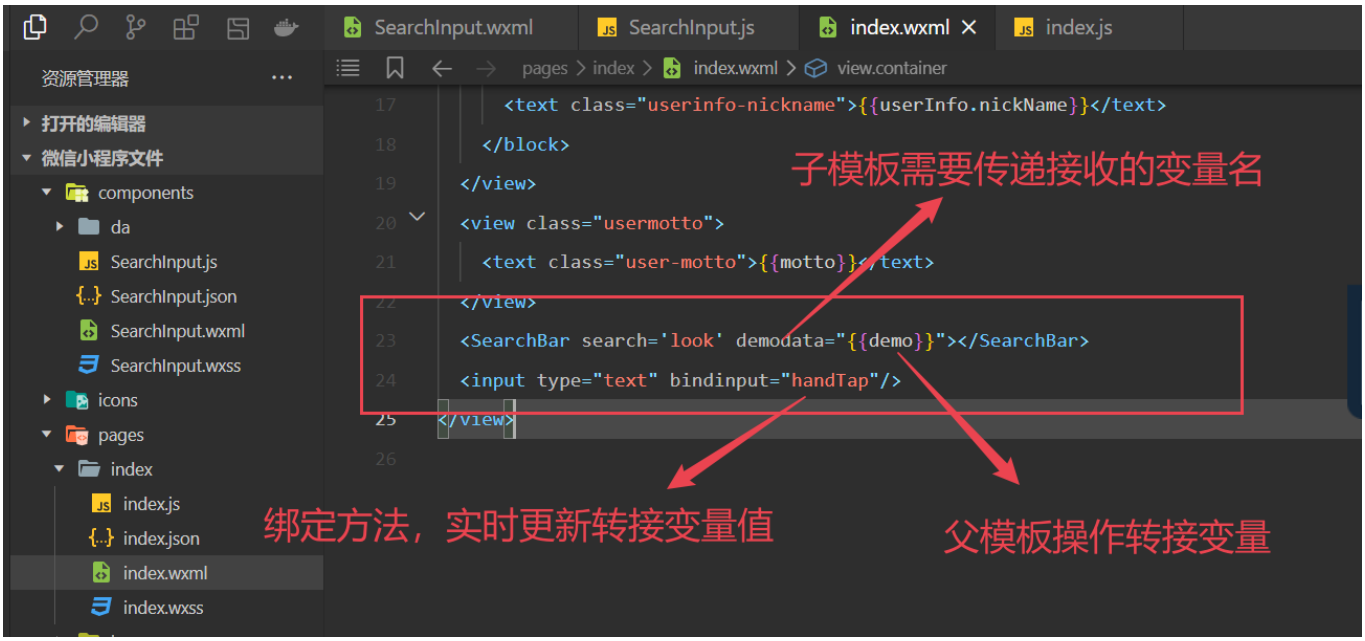
1.子组件定义接收变量demodata

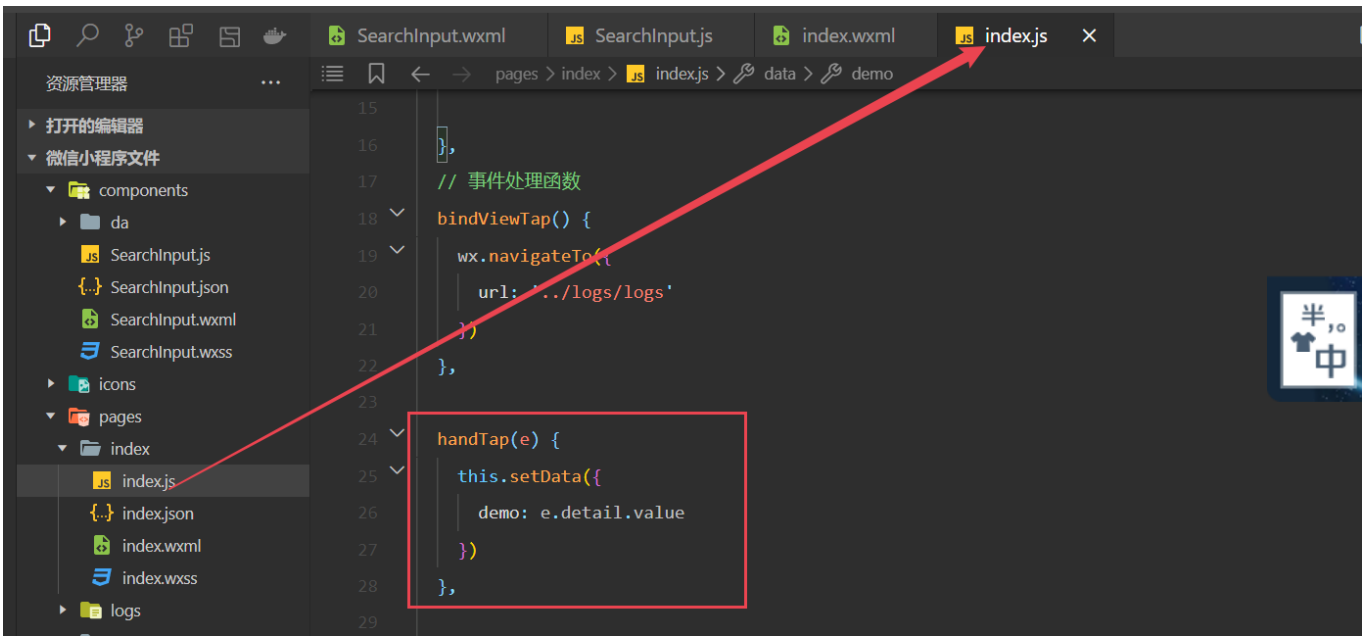
2.子组件渲染

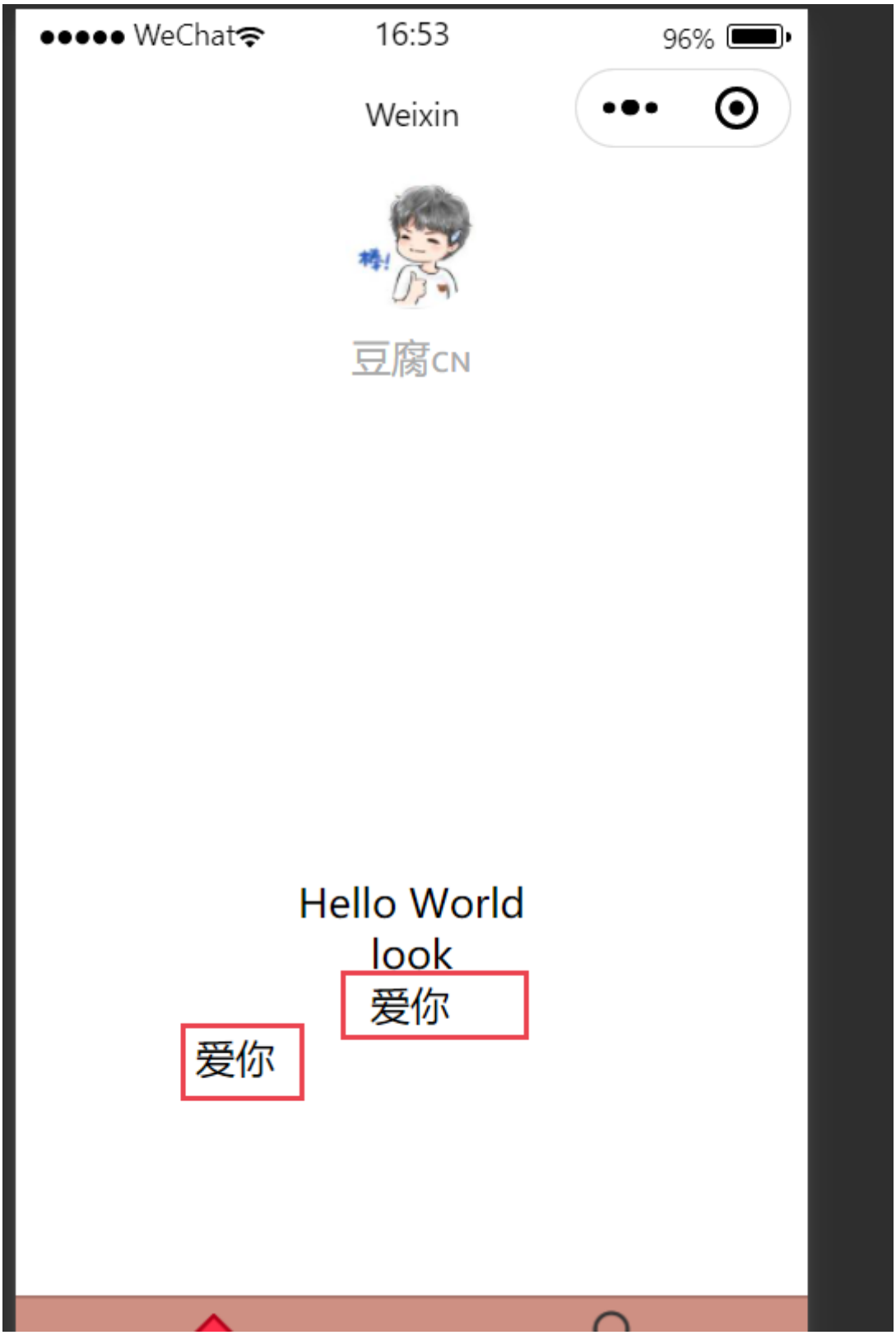


3.父组件定义操作变量



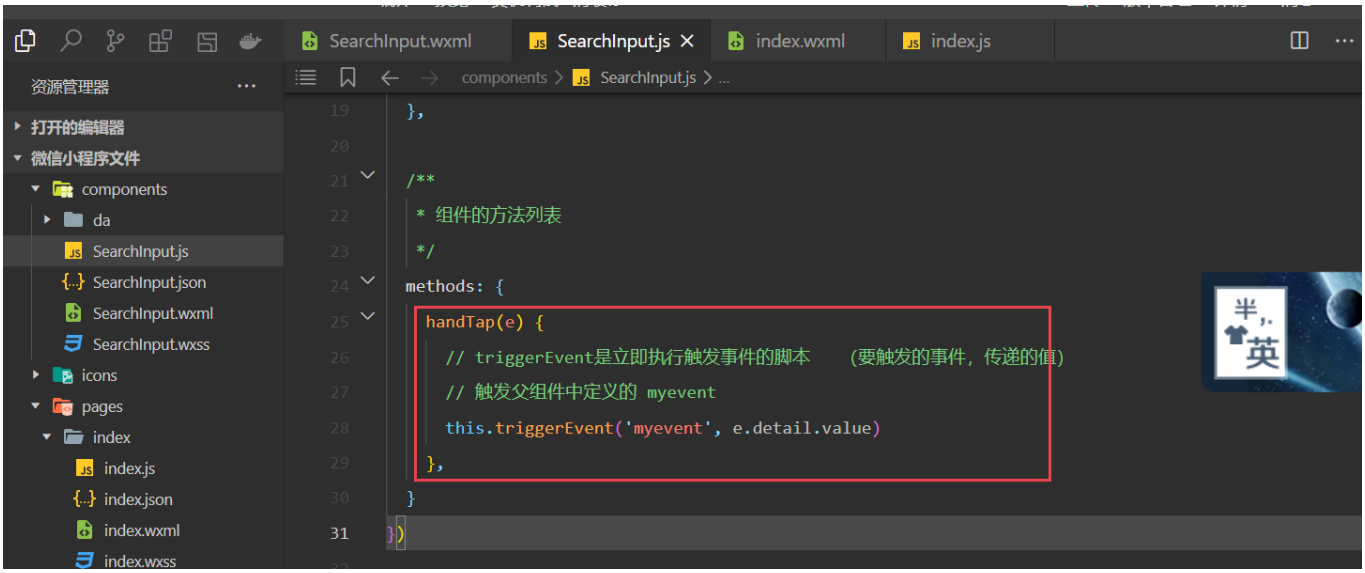

SearchInput.wxml
SearchInput.js
index.wxml
index.js
资源管理器
打开的编辑器
微信小程序文件
components
da
SearchInput.js
SearchInput.json
SearchInput.wxml
SearchInput.wxss
icons
pages
index
index.js
index.json
index.wxml
index.wxss
pages > index > index.wxml > view.container
<text class="userinfo-nickname">{{userInfo.nickName}}</text>
</block>
</view>
<view class="usermotto">
<text class="user-motto">{{motto}}</text>
</view>
<SearchBar search='look' demodata="{{demo}}"></SearchBar>
<input type="text" bindinput="handTap"/>
</view>
子模板需要传递接收的变量名
绑定方法，实时更新转接变量值
父模板操作转接变量


SearchInput.wxml
SearchInput.js
index.wxml
index.js
资源管理器
打开的编辑器
微信小程序文件
components
da
SearchInput.js
SearchInput.json
SearchInput.wxml
SearchInput.wxss
icons
pages
index
index.js
index.json
index.wxml
index.wxss
logs
pages > index > index.js > data > demo
},
// 事件处理函数
bindViewTap() {
wx.navigateTo({
url: '../logs/logs'
})
},
handTap(e) {
this.setData({
demo: e.detail.value
})
},

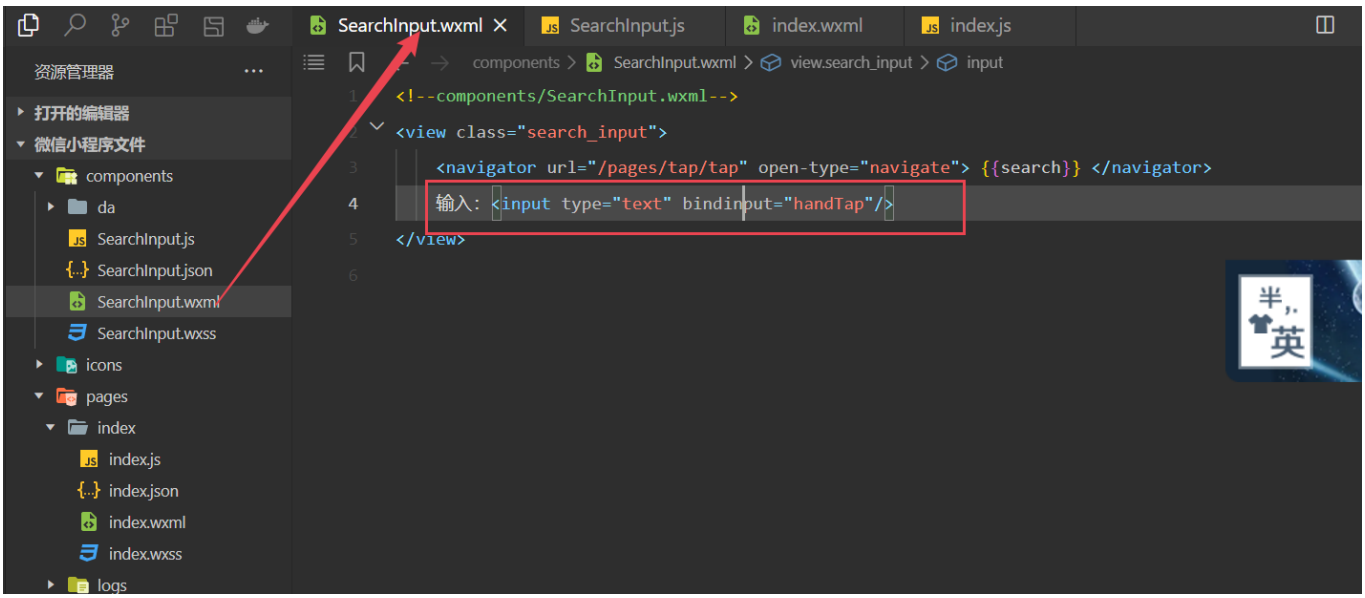


9.子传父

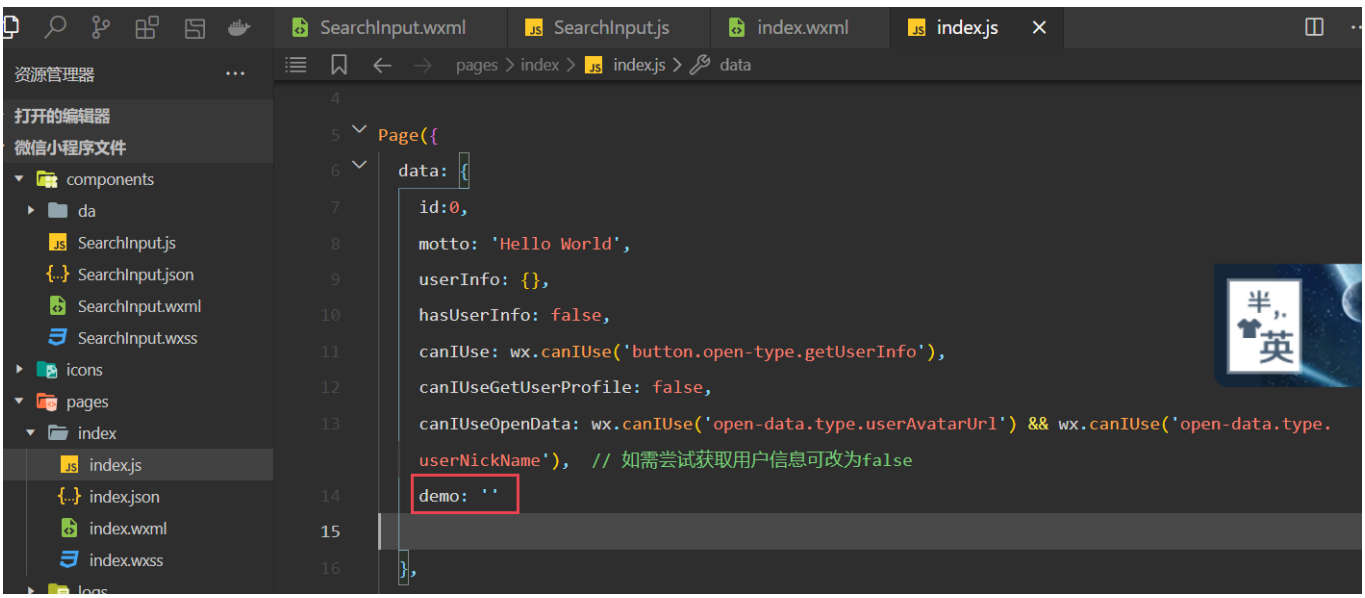
首先在子组件js中建立一个方法，在出触发后调用父组件方法 传递子组件数据



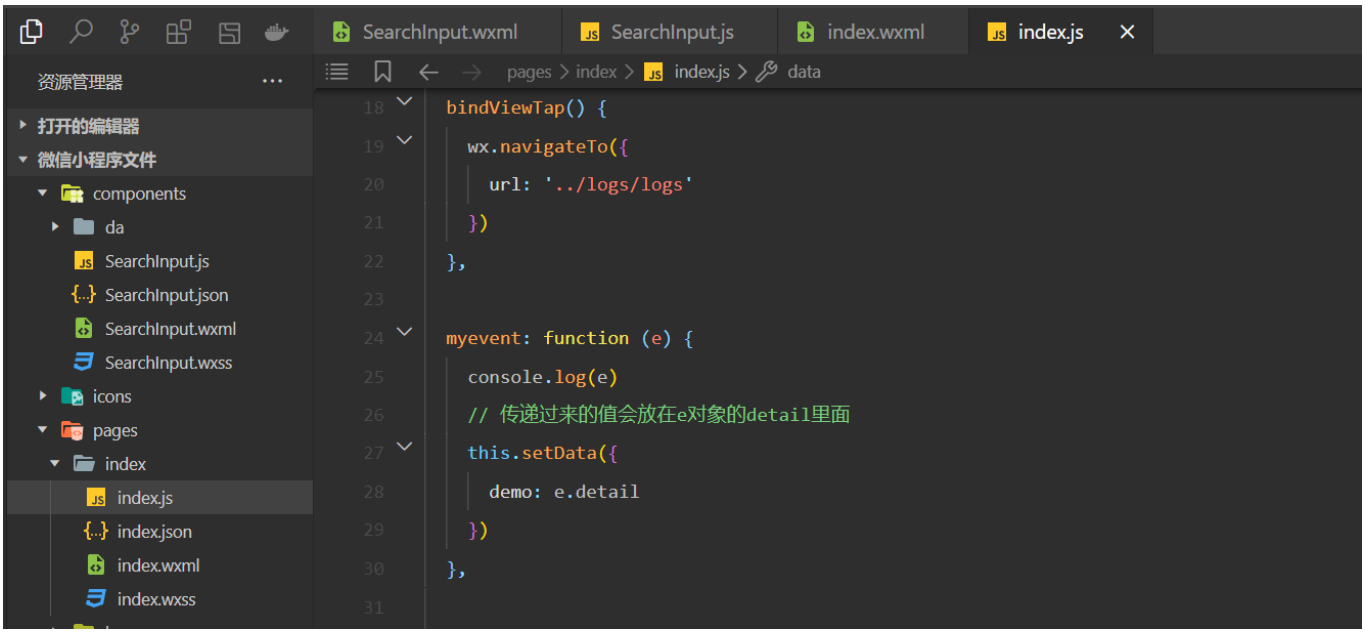
子组件建立输入框，同时绑定input事件（输入时触发）



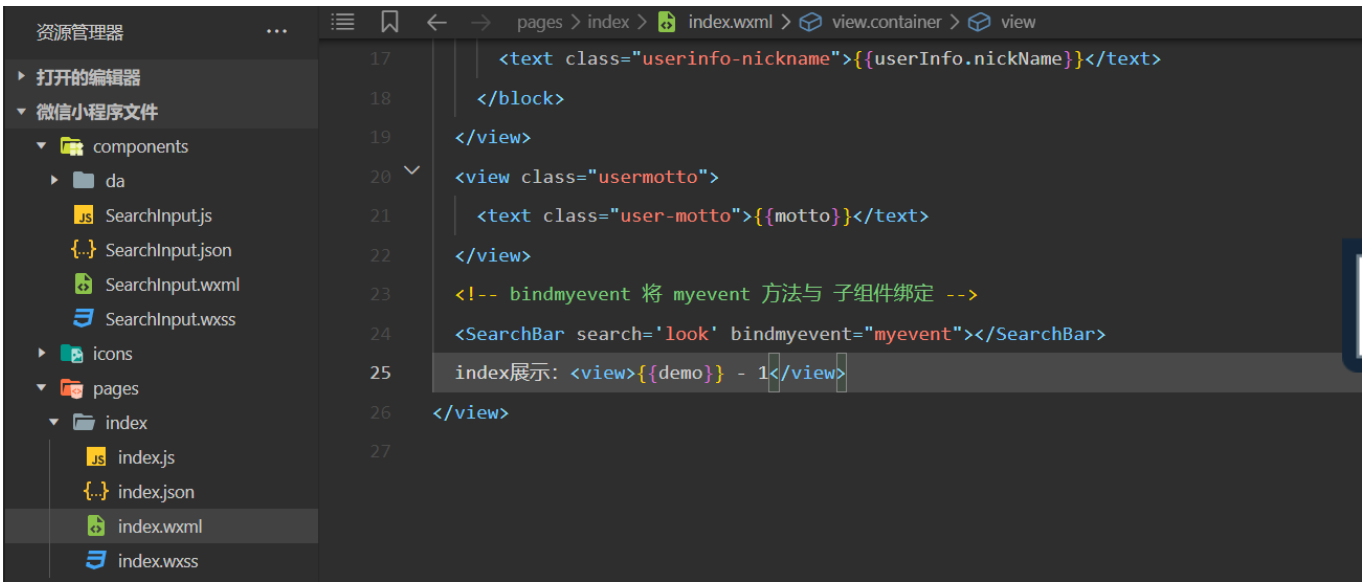
父组件建立接收变量



父组件建立方法事件-将子组件调用此方法传递过来的数据处理


SearchInput.wxml
SearchInput.js
index.wxml
index.js
资源管理器
pages > index > index.js > data
打开的编辑器
微信小程序文件
components
da
SearchInput.js
SearchInput.json
SearchInput.wxml
SearchInput.wxss
icons
pages
index
index.js
index.json
index.wxml
index.wxss
bindViewTap() {
wx.navigateTo({
url: '../logs/logs'
})
},
myevent: function (e) {
console.log(e)
// 传递过来的值会放在e对象的detail里面
this.setData({
demo: e.detail
})
},

父组件渲染展示


资源管理器
pages > index > index.wxml > view.container > view
打开的编辑器
微信小程序文件
components
da
SearchInput.js
SearchInput.json
SearchInput.wxml
SearchInput.wxss
icons
pages
index
index.js
index.json
index.wxml
index.wxss
<text class="userinfo-nickname">{{userInfo.nickName}}</text>
</block>
</view>
<view class="usermotto">
<text class="user-motto">{{motto}}</text>
</view>
<!-- bindmyevent 将 myevent 方法与 子组件绑定 -->
<SearchBar search='look' bindmyevent="myevent"></SearchBar>
index展示: <view>{{demo}} - 1</view>
</view>