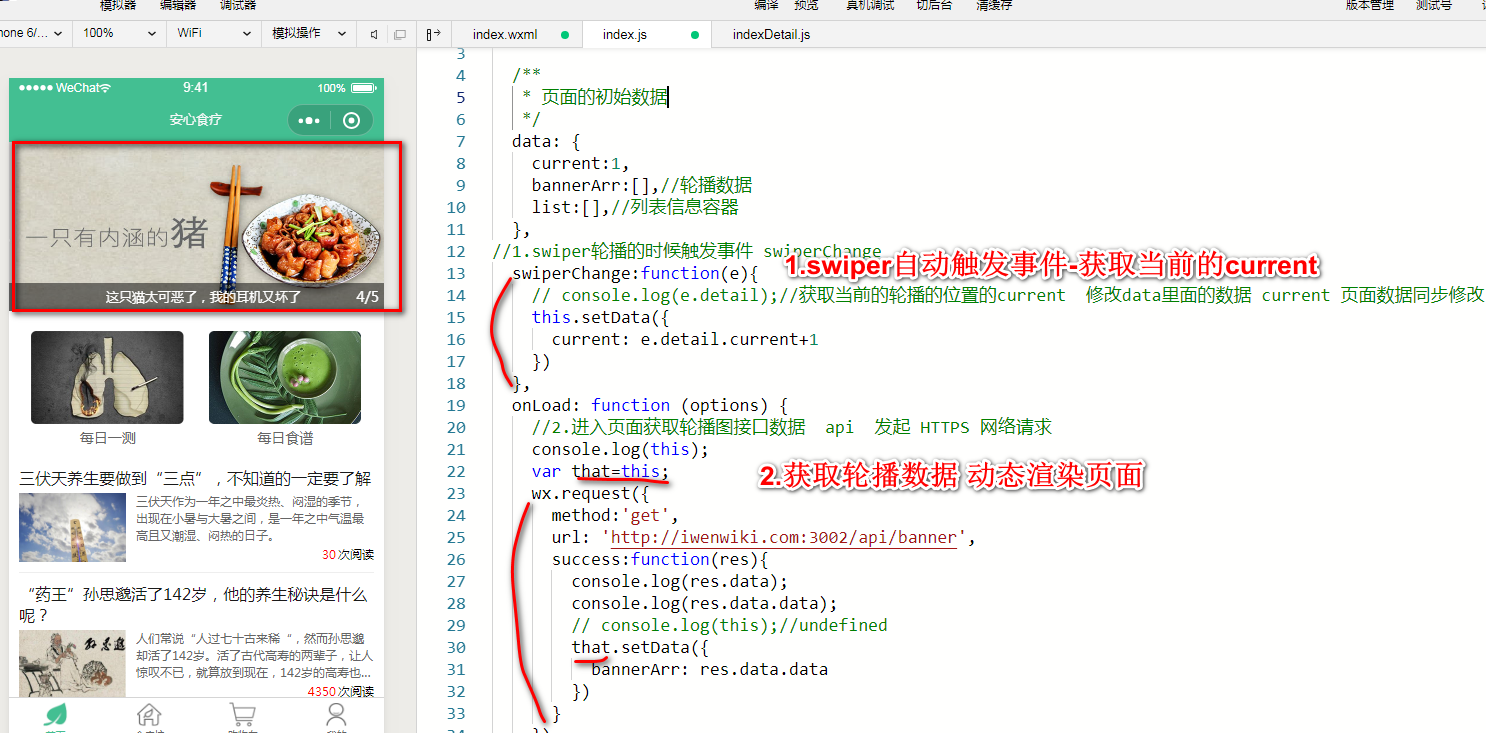
## 1.轮播图





## 2.首页列表页面跳转

页面跳转 <navigator url='xxx/xxx?id=123'></navigator>

页面跳转 小程序api跳转页面 wx.naxxx

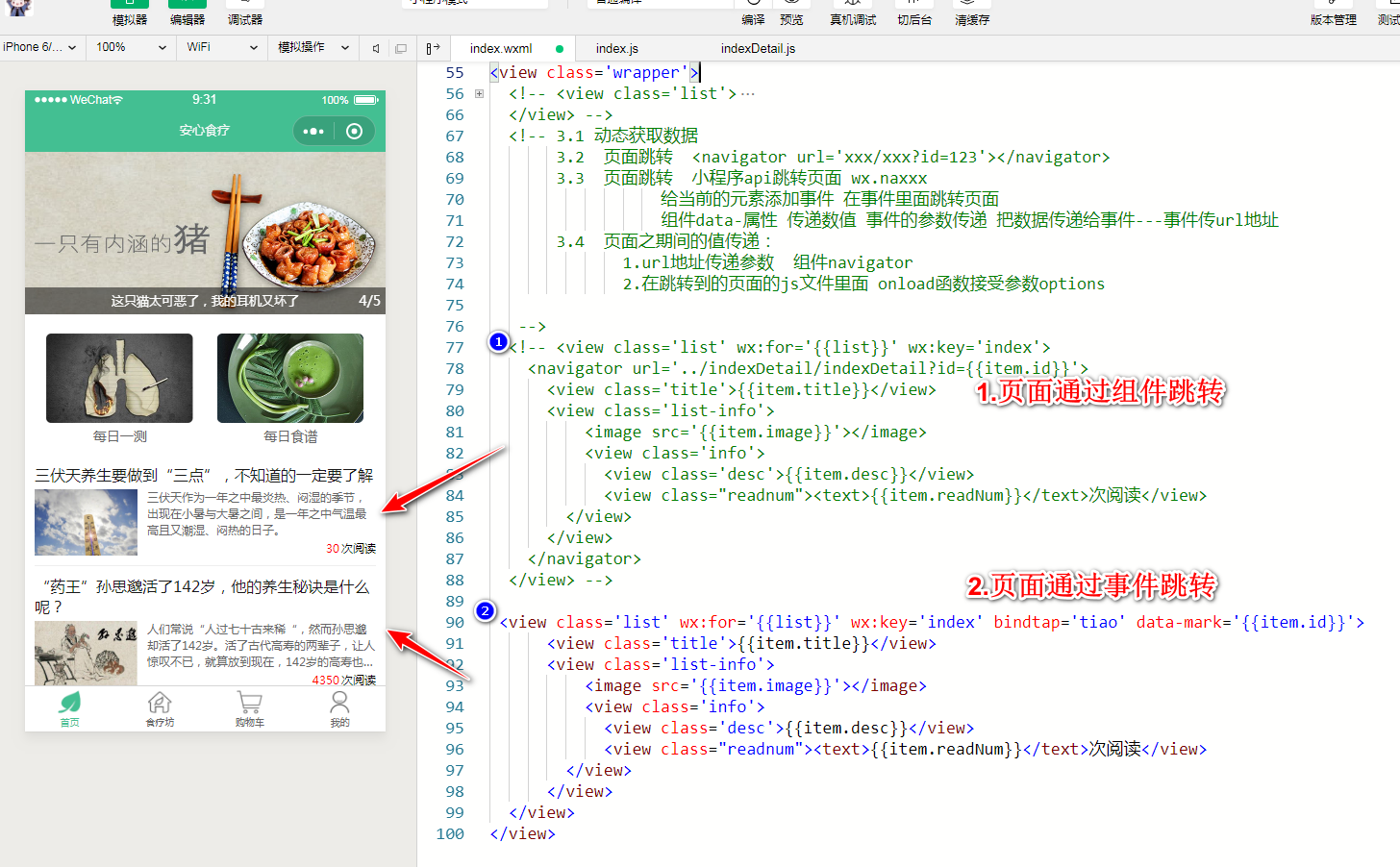
给当前的元素添加事件 在事件里面跳转页面

组件data-属性 传递数值 事件的参数传递 把数据传递给事件---事件传url地址

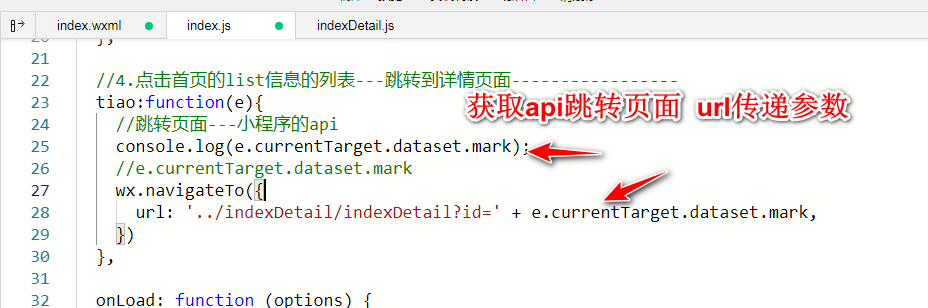
页面之期间的值传递：

1.url地址传递参数 组件navigator

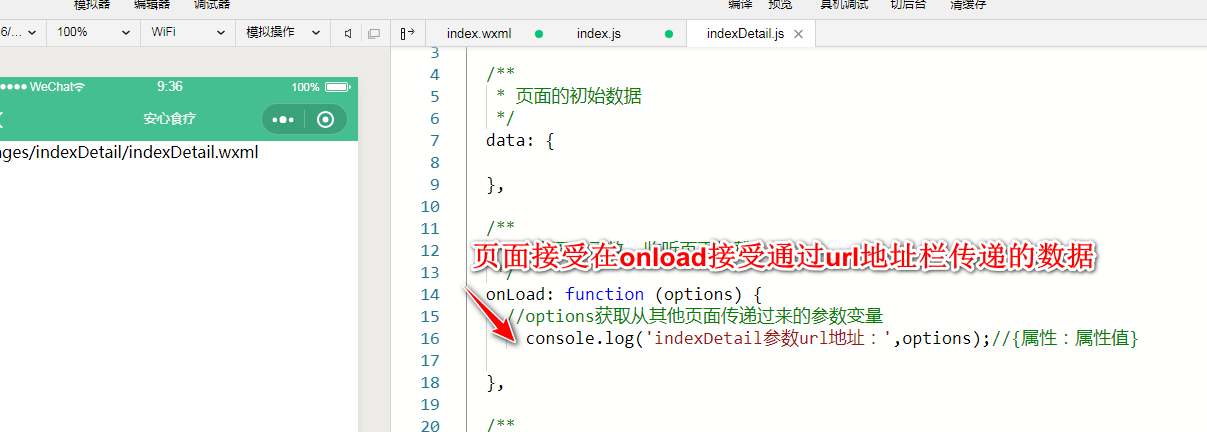
2.在跳转到的页面的js文件里面 onload函数接受参数options



index.js代码：



indexDetail页面：

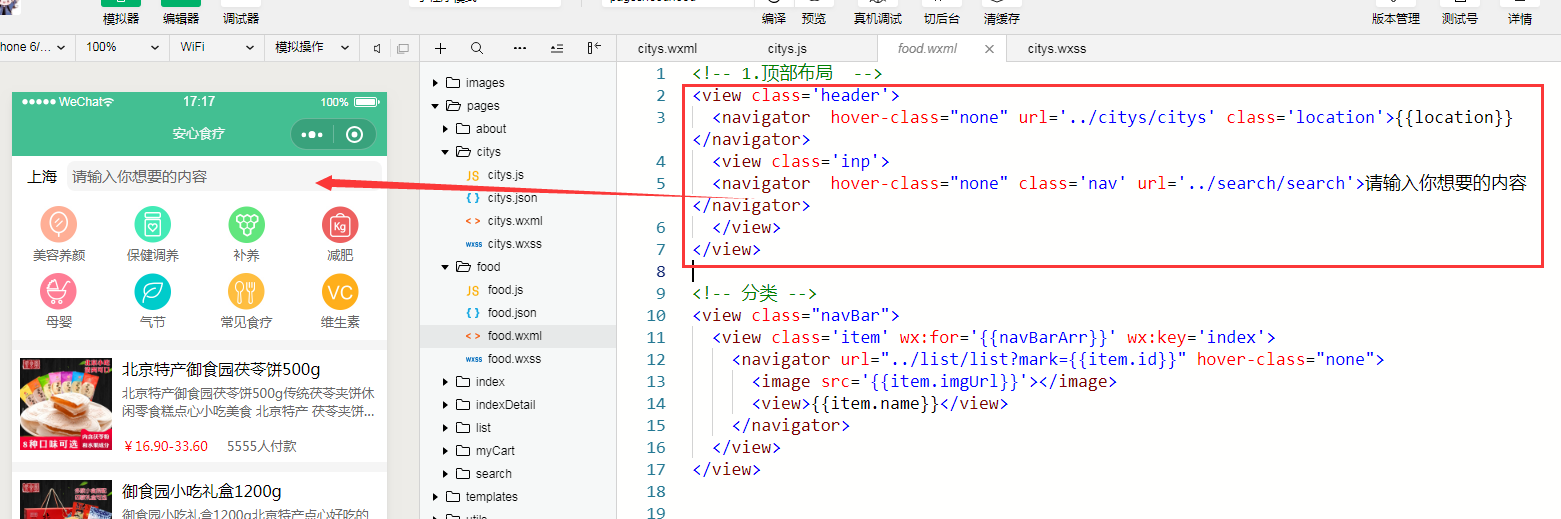


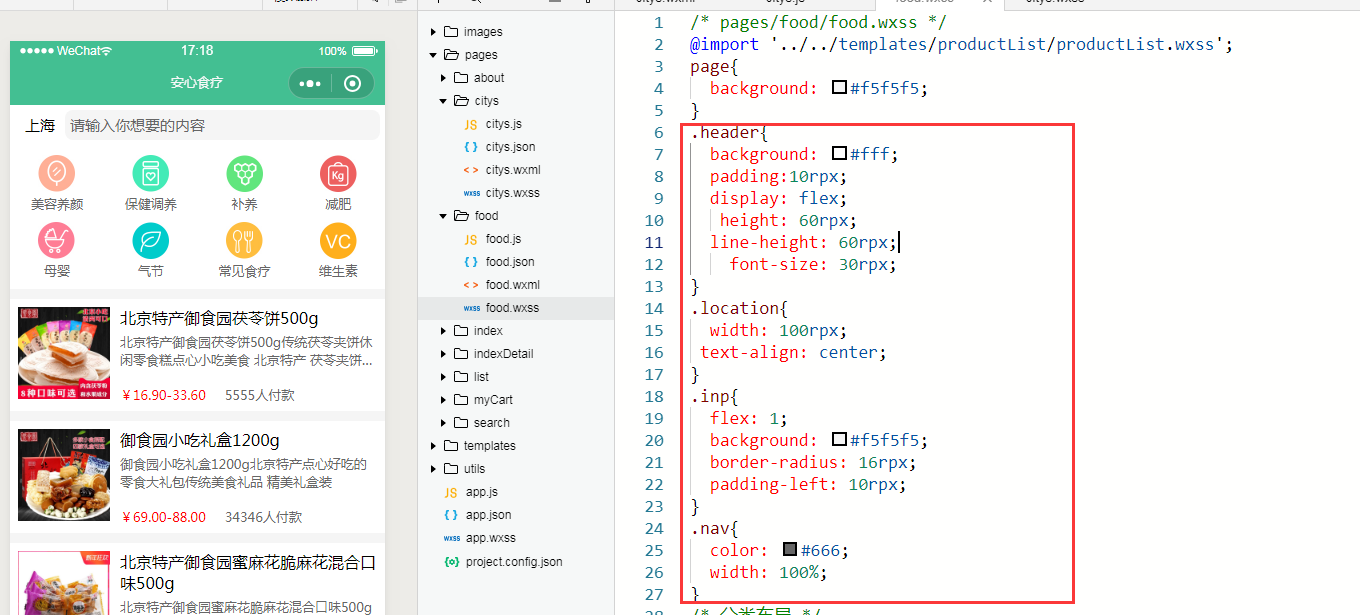
## 食疗坊顶部

顶部布局：左侧是默认选中的城市 右侧是搜索框，点击分别跳转到对应的页面



【代码演示】





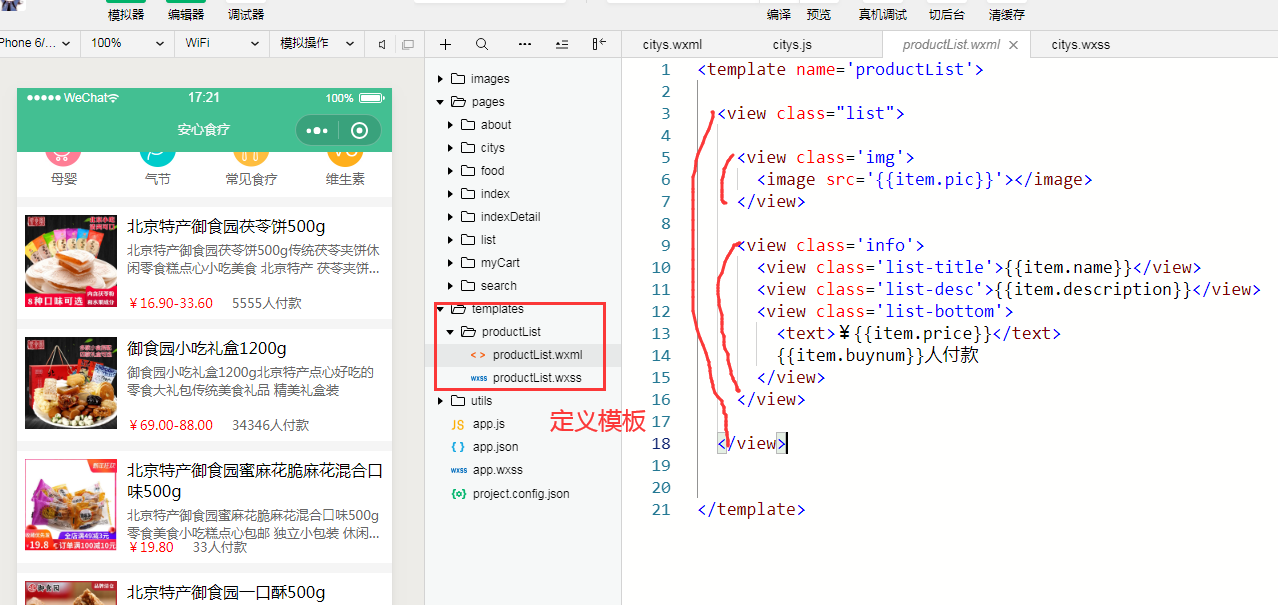
## 列表信息

商品列表信息 多个页面都是用，使用模板

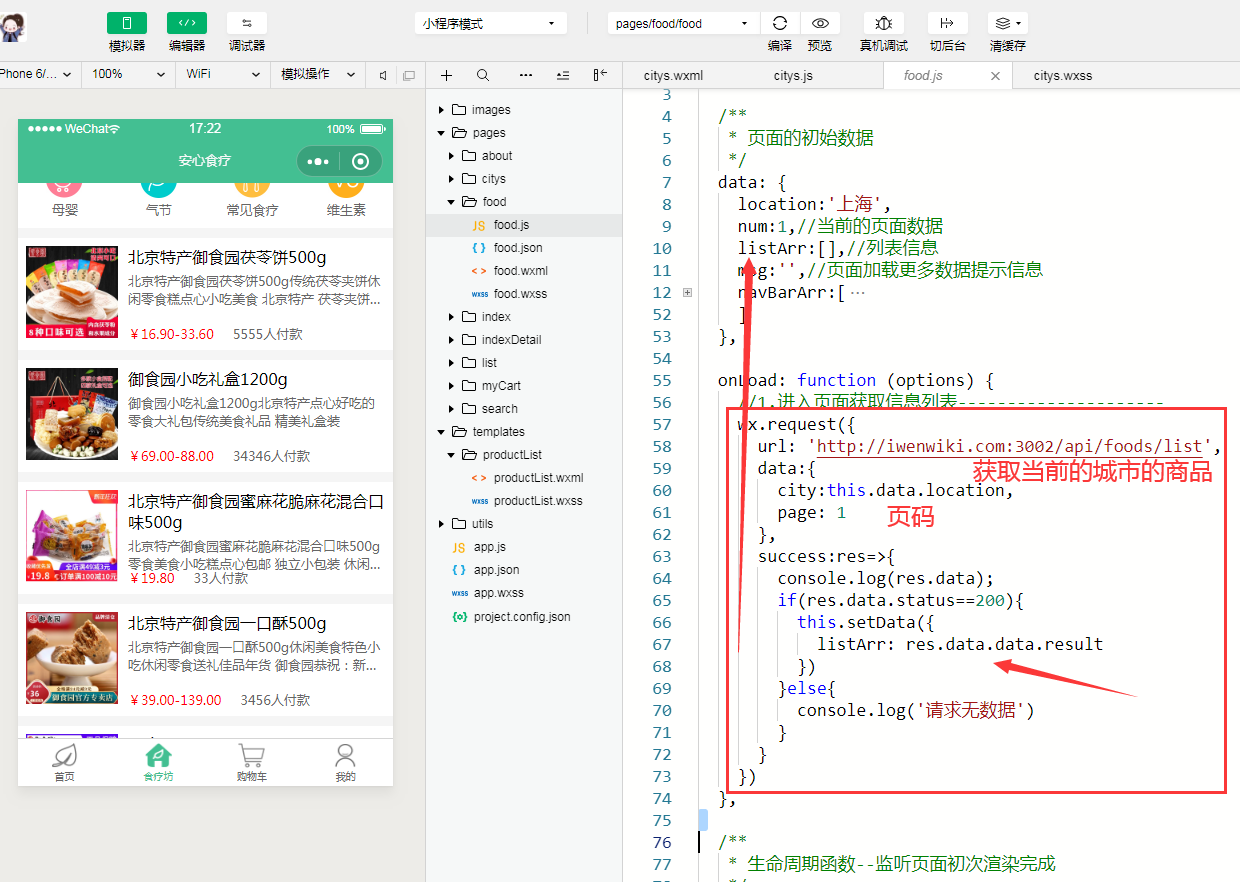


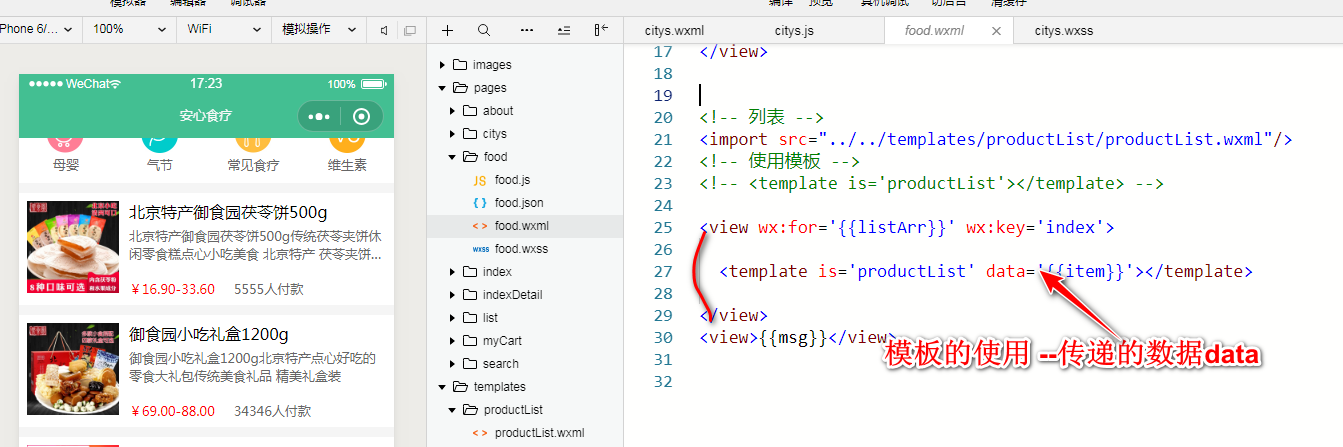
步骤：

1. 定义模板
2. 引入模板 并使用 同时引入模板的wxss
3. 获取食疗坊的动态列表数据 ---遍历渲染列表



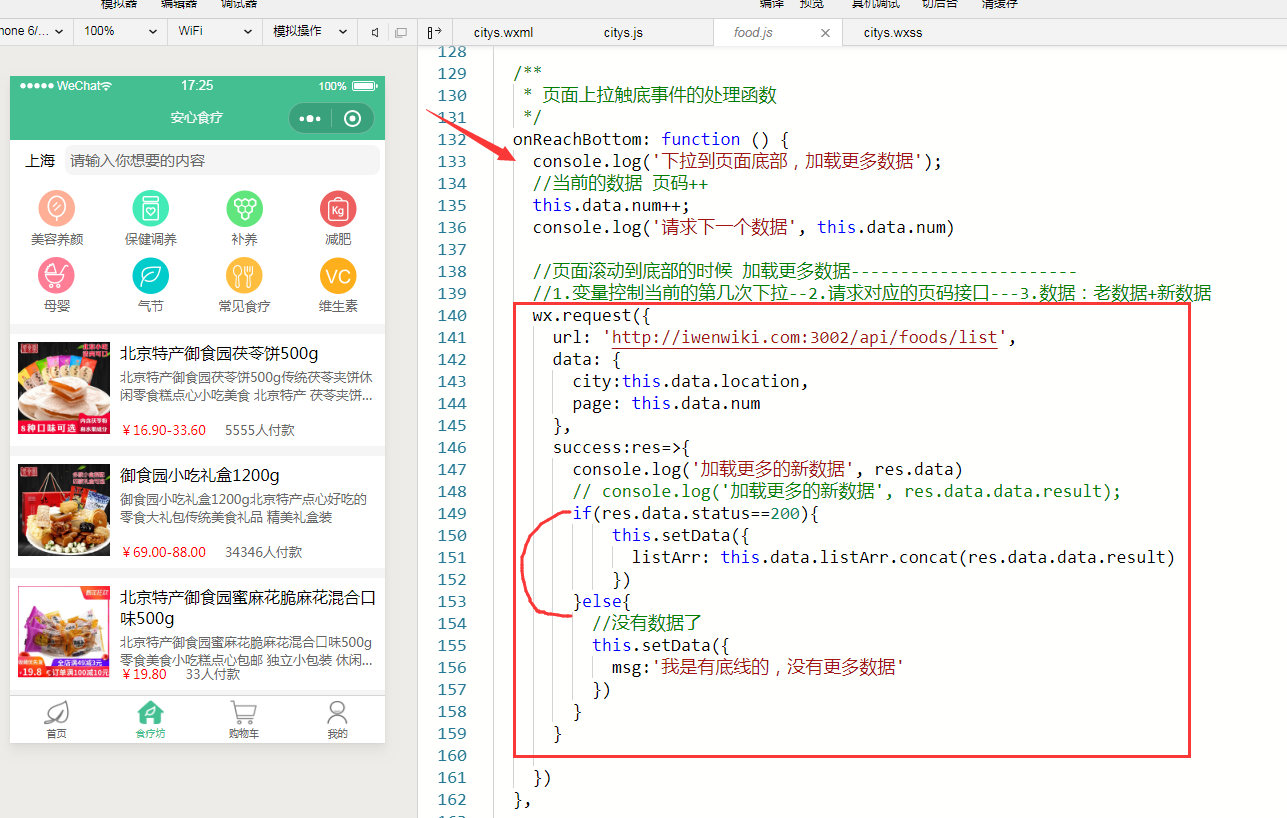
【js代码】





## 5.下拉刷新 下拉底部加载更多数据





## 6.定位

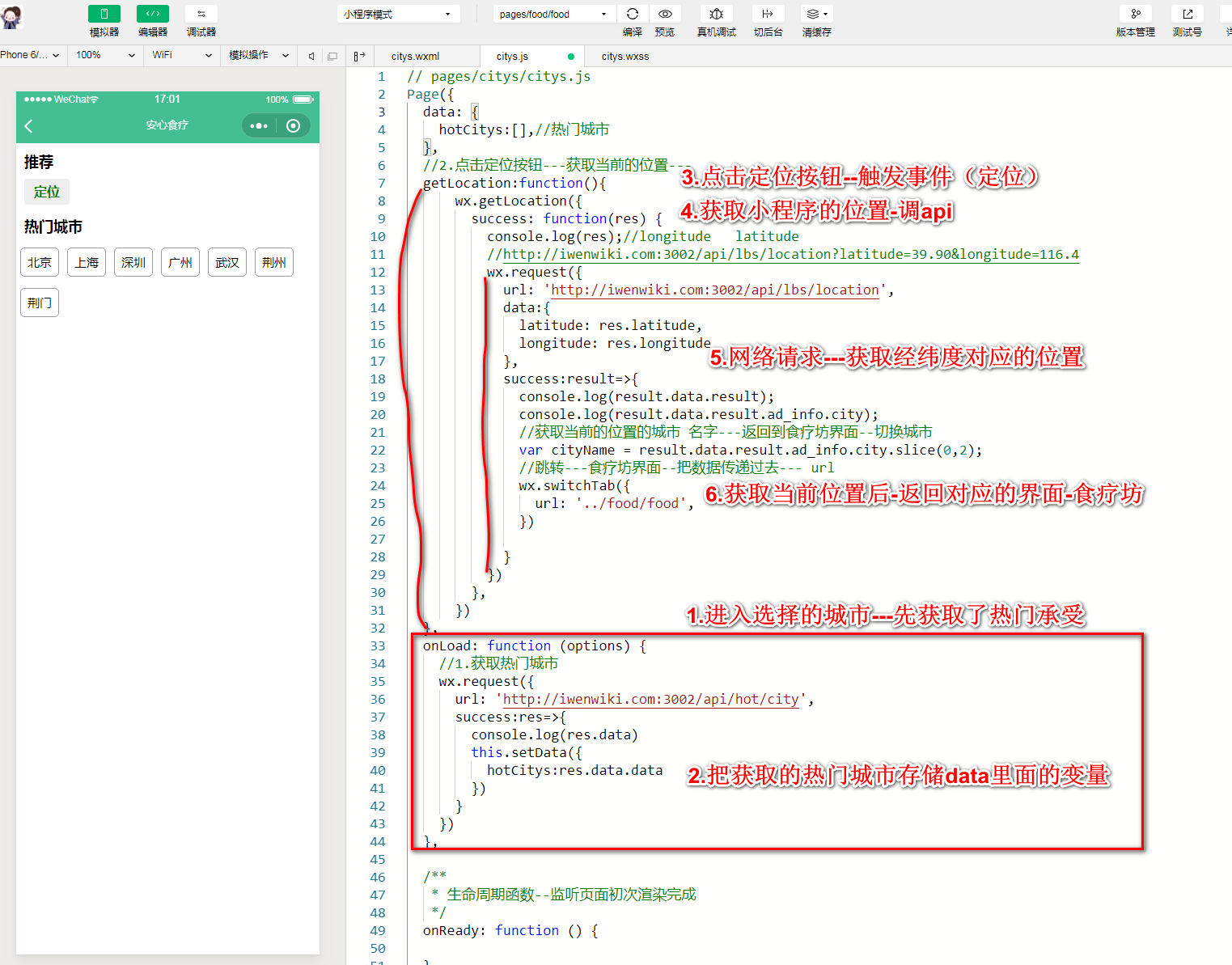
点击进入定位的界面 ---进入对应的承受选择界面



思路：

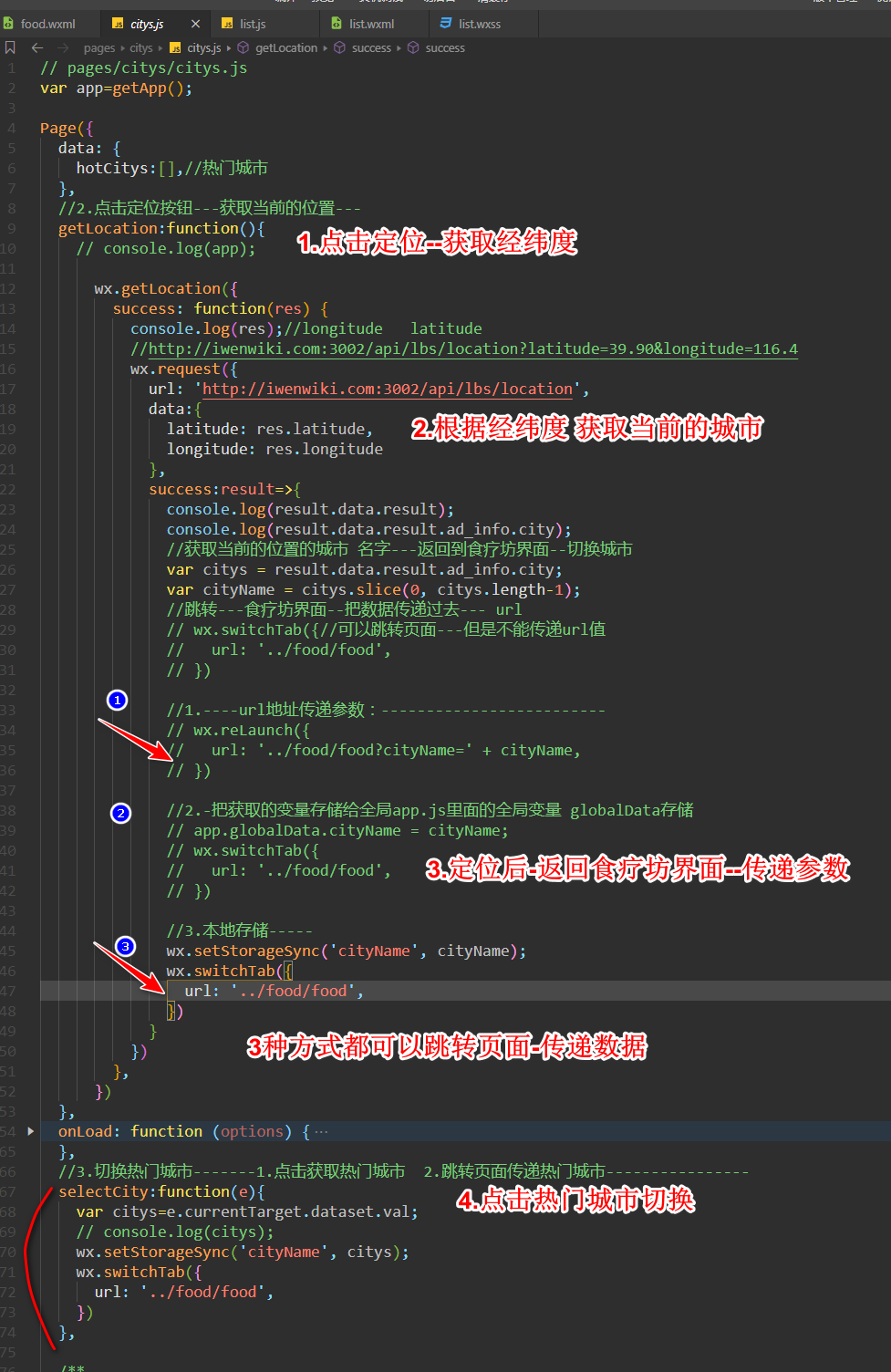
1. 先获取热门城市 进入js后 获取网络请求获取热门城市
2. 获取的数据渲染页面显示热门城市
3. 点击定位按钮--触发事件--获取小程序api--获取当前的位置经纬度---根据经纬度获取当前的城市
4. 获取到定位城市后--把当前的承受传递页面food页面，并更新食疗坊页面的列表商品信息





## 页面跳转之间的参数传递

1. url地址传递数据 组件：<navigator url=''> 事件+路由的url地址 对应的页面onload接受
2. 全局变量 app.js globalData实现页面数据共享
3. 本地存储



## 数据存储

小程序也封装了api 本地存储数据：

设置存储语法：wx.setStorageSync() wx.setStorage()

说明： 后缀带sync的是同步的存储 不带sync的是异步存储。

**语法：**

设置:wx.setStorageSync('key',value);

wx.setStorage({

key:"key",

data:"value"

})

删除： wx.removeStorageSync('key')

获取：wx.getStorageSync(string key)

清空：wx.clearStorageSync()

分析：获取到了定位的城市，页面传递数据，把当前的获取的城市--存储到本地。返回到食疗坊界面的时候，再次获取本地的数据，如果有数据--获取当前的区域对应的商品，否则获取默认的位置的商品。

## 列表详情页面

详情页面又是模板，多个地方反复使用， 跳转页面。当前的当前的商品，进入对应的商品的详情页。

进入对应的页面，页面是同一个页面，只是当前的页面获取的数据不同，获取对应的数据。

跳转页面，并且传递参数。



方法1：事件触发 点击当前的商品，触发事件---路由跳转---传递参数data-