小程序开发文档02/21

# 一、页面导航

页面导航指的是页面之间的相互跳转。如浏览器中实现页面导航的方式有

<a>链接和location.herf

小程序中实现页面导航的两种方式：

1）声明式导航

在页面上声明一个<navigator>导航组件

通过点击<navigator>组件实现页面跳转

2）编程式导航

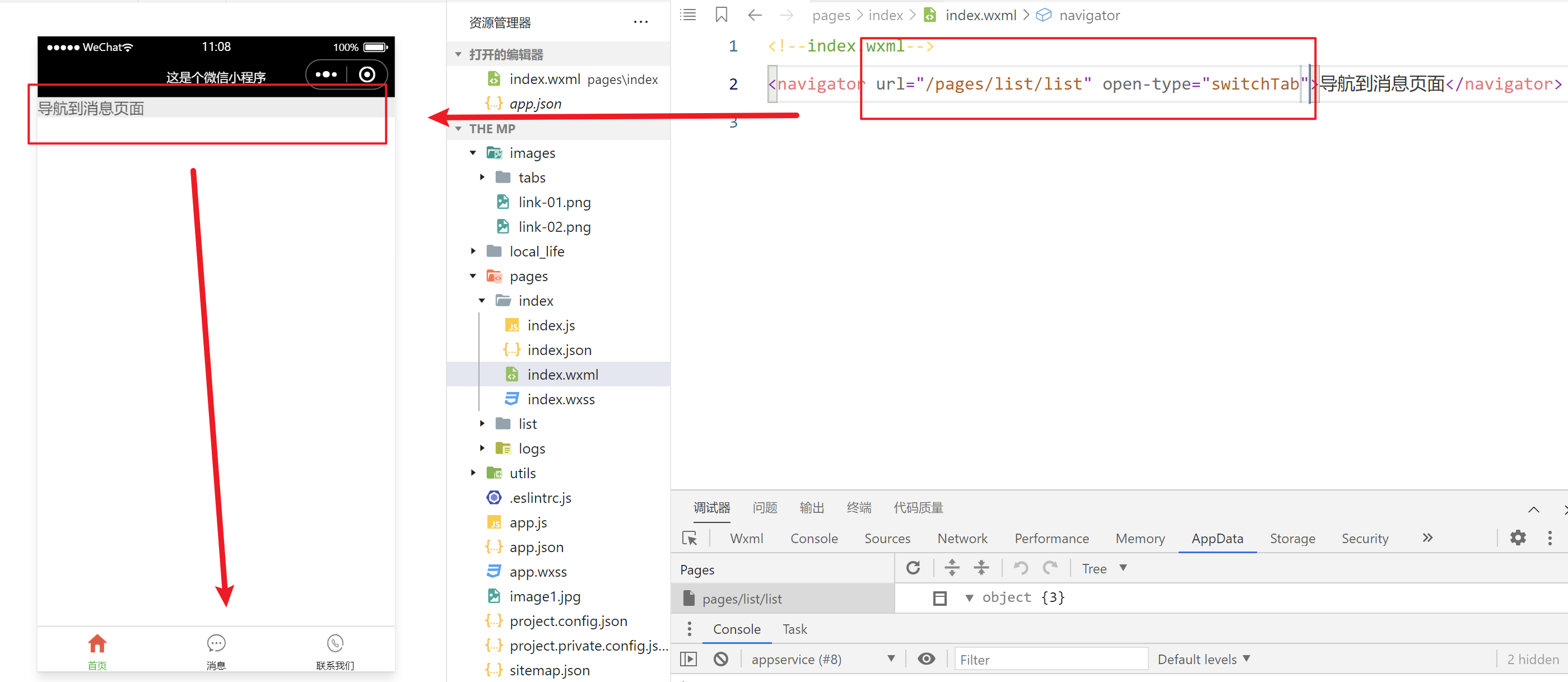
调用小程序的导航API，实现页面的跳转

## 1、声明式导航

### 1.1 导航到tabBar页面

在使用<navigator>组件跳转到指定的tabBar页面时，需指定url属性和open-type属性

1. url表示要跳转的页面的地址，必须以/开头
2. open-type表示跳转的方式，必须为switchTab

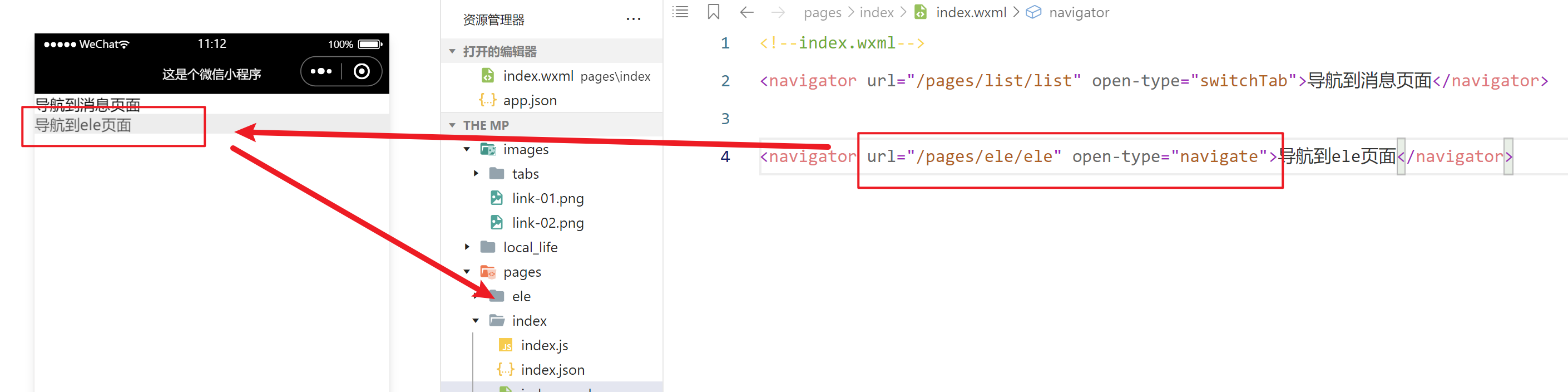


### 1.2 导航到非tabBar页面

在使用<navigator>组件跳转到指定的tabBar页面时，需指定url属性和open-type属性

1)url表示要跳转的页面的地址，必须以/开头

2)open-type表示跳转的方式，必须为navigate



注意：为了简便，在导航到非tabBar页面时，open-type=“naviagte”属性可以省略。

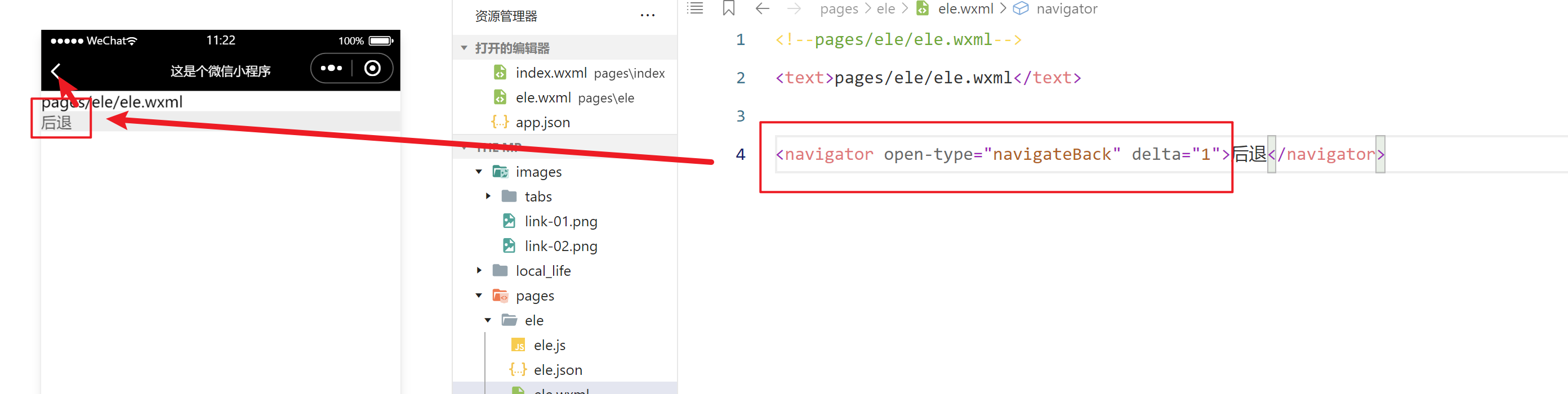
### 1.3 后退导航：

如果要后退到上一页或多级页面，则需要指定open-type属性和delta属性，其中：

1）open-type的值必须是navigateBack，表示要进行后退导航

2）delta的值必须是数字，表示要后退的层级

如:<naviagtor open-type="navigateBack" delta="1">返回上一页</naviagtor>



注意：如果只是后退到上一页，则可以省略delta属性，因为其默认值就是1

## 2、编程式导航

### 2.1导航到tabBar页面

调用wx.switchTab(Object object)方法，可以跳转到tabBar页面。其中Object参数对象的属性列表如下：



### 2.2导航到非tabBar页面

调用wx.navigateTo(Object object)方法，可以跳转到非tabBar的页面。其中Object参数对象的属性列表如下：



### 2.3后退导航

调用wx.navigateBack(Object object)方法，可以返回上一页面或多级页面。其中Object参数对象可选的属性如下：



# 二、导航传参

## 1、声明式导航传参

navigator组件的url属性用来指定将要跳转到的页面的路径。同时，路径的后面还可以携带参数：

1）参数与路径之间使用？分隔；

2）参数键与参数值用=相连

3）不同参数用&分隔

如：<naviagtor url="/pages/info/info?name=zs&age=20">跳转到info页面</naviagtor>

## 2、 编程式导航传参

调用wx.navigateTo(Object onject)方法跳转页面时，也可以携带参数，如：

//页面结构 .wxml

<button bindtap="gotoinfo2">跳转到info页面</button>

//通过编程式导航，跳转到info页面，并携带参数 .js

gotoinfo2(){

wx.navigateTo({

url:'/pages/info/info?name=ls&gender=男'

})

}

## 3、在onLoad中接收导航参数

通过声明式导航传参或编程式导航传参所携带的参数，可以直接在onLoad事件中直接获取到如：

//.js文件

onLoad:function(options){

console.log(options)

},

# 三、下拉刷新

下拉刷新是移动端的专有名词，指的是通过手指在屏幕上的下拉滑动操作，从而重新加载页面数据的行为。

## 1、启动下拉刷新有两种方式：

1）全局开启下拉刷新（不推荐）

在app.json的window节点中，将enablePullDownRefresh设置为true；

2）局部开启下拉刷新

在页面的.json配置文件中，将enablePullDownRefresh设置为true

## 2、配置下拉刷新窗口的样式：

在全局或页面的.json配置文件中，通过backgroundColor和backgroundTextStyle来配置下拉刷新窗口的样式，其中：

1）backgroundColor用来配置下拉刷新窗口的背景颜色，仅支持16进制的颜色值

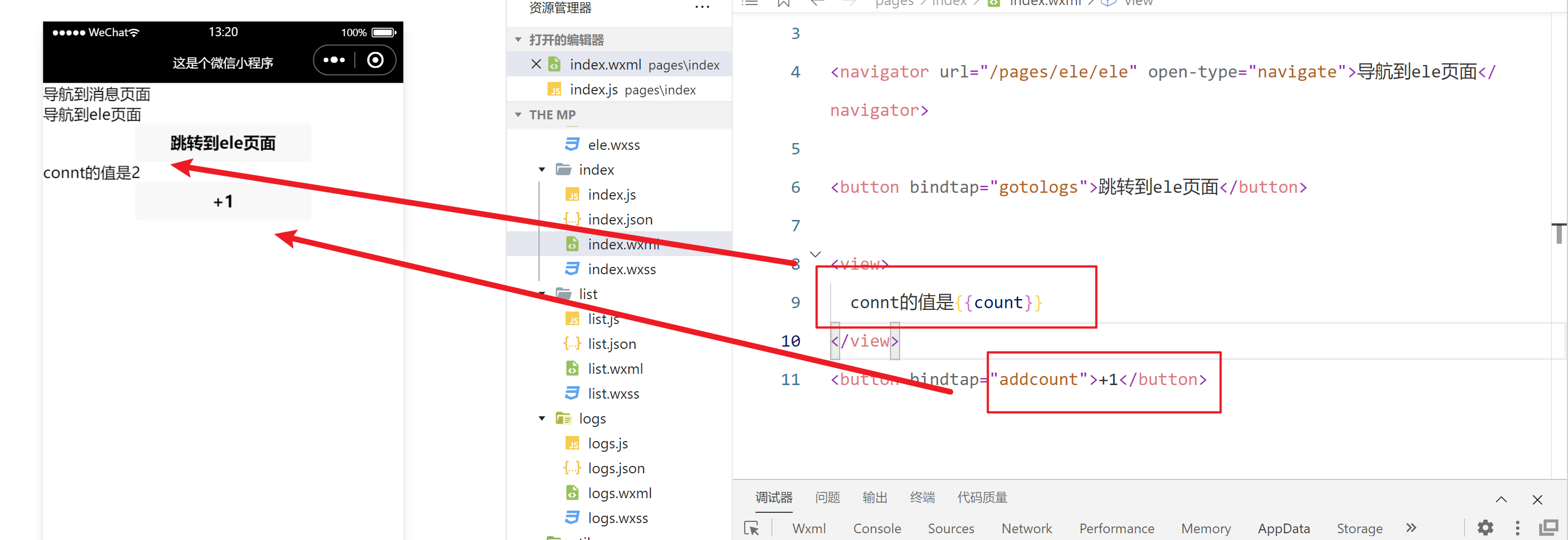
2）backgroundTextStyle用来配置下拉刷新loading的样式，仅支持dark和light

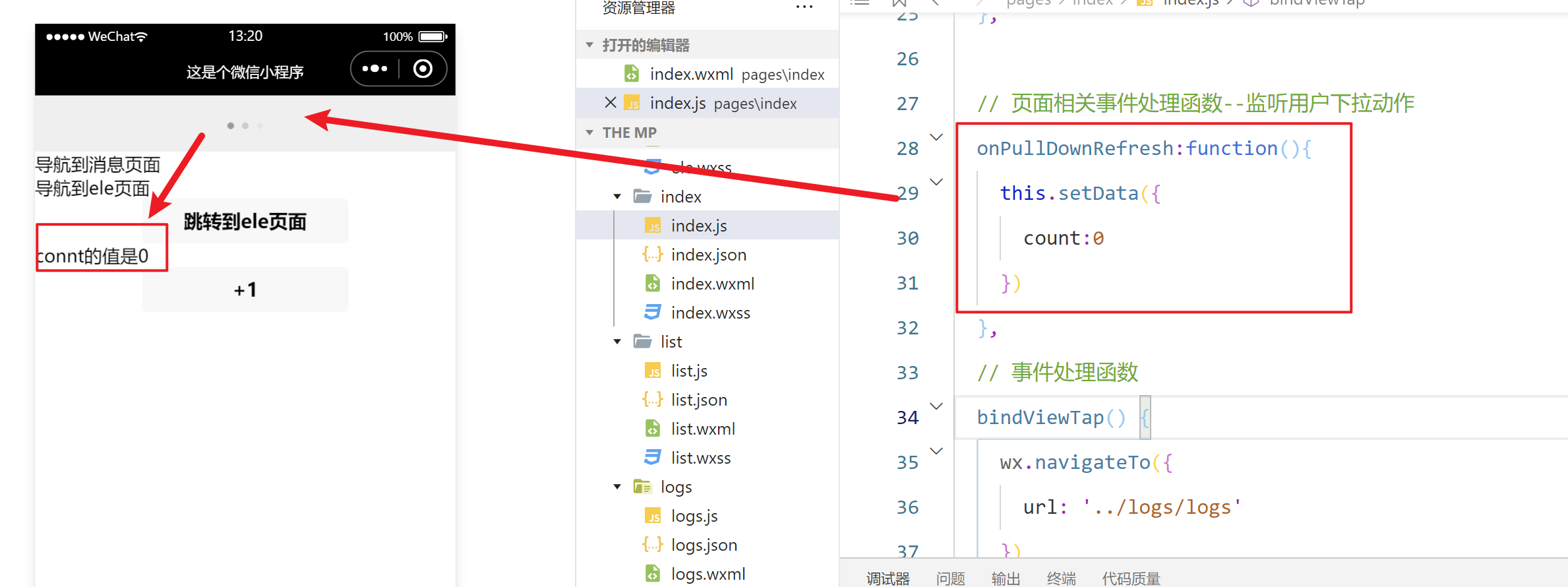
## 3、停止下拉刷新的效果：

当处理完下拉刷新后，下拉刷新的loading效果会一直显示，不会主动消失，所以需要主动隐藏loading效果。此时，调用wx.stopPullDownRefresh()可以停止当前页面的下拉刷新。

## 4、监听页面的下拉刷新事件

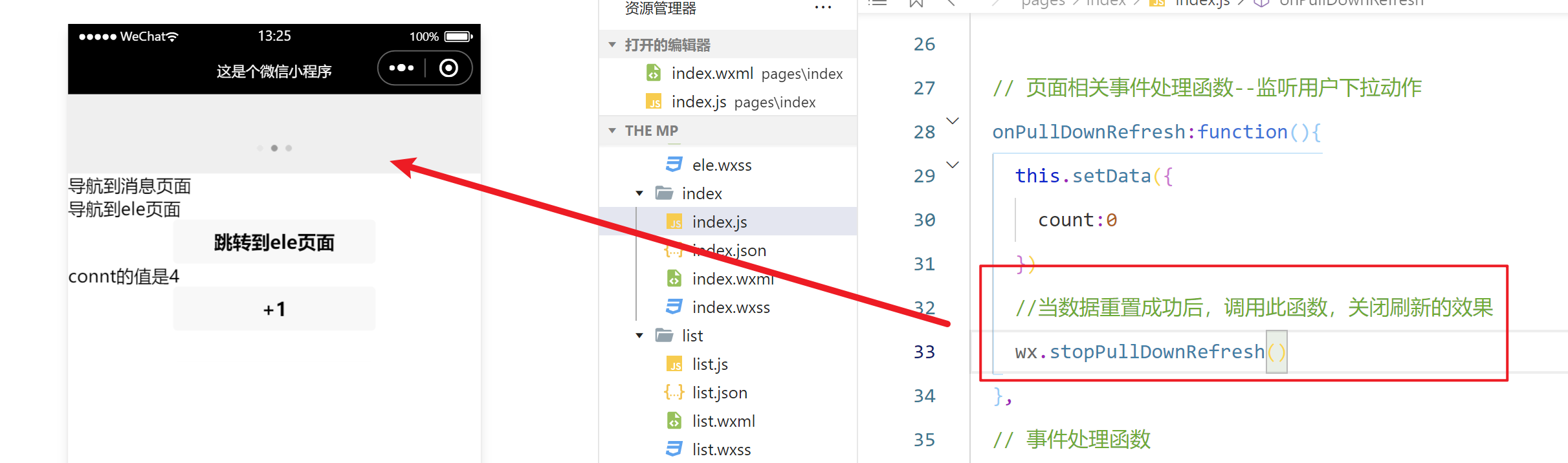
在触发页面的下拉刷新事件的时候，如果要把count的值重置为0：





## 5、停止下拉刷新的效果

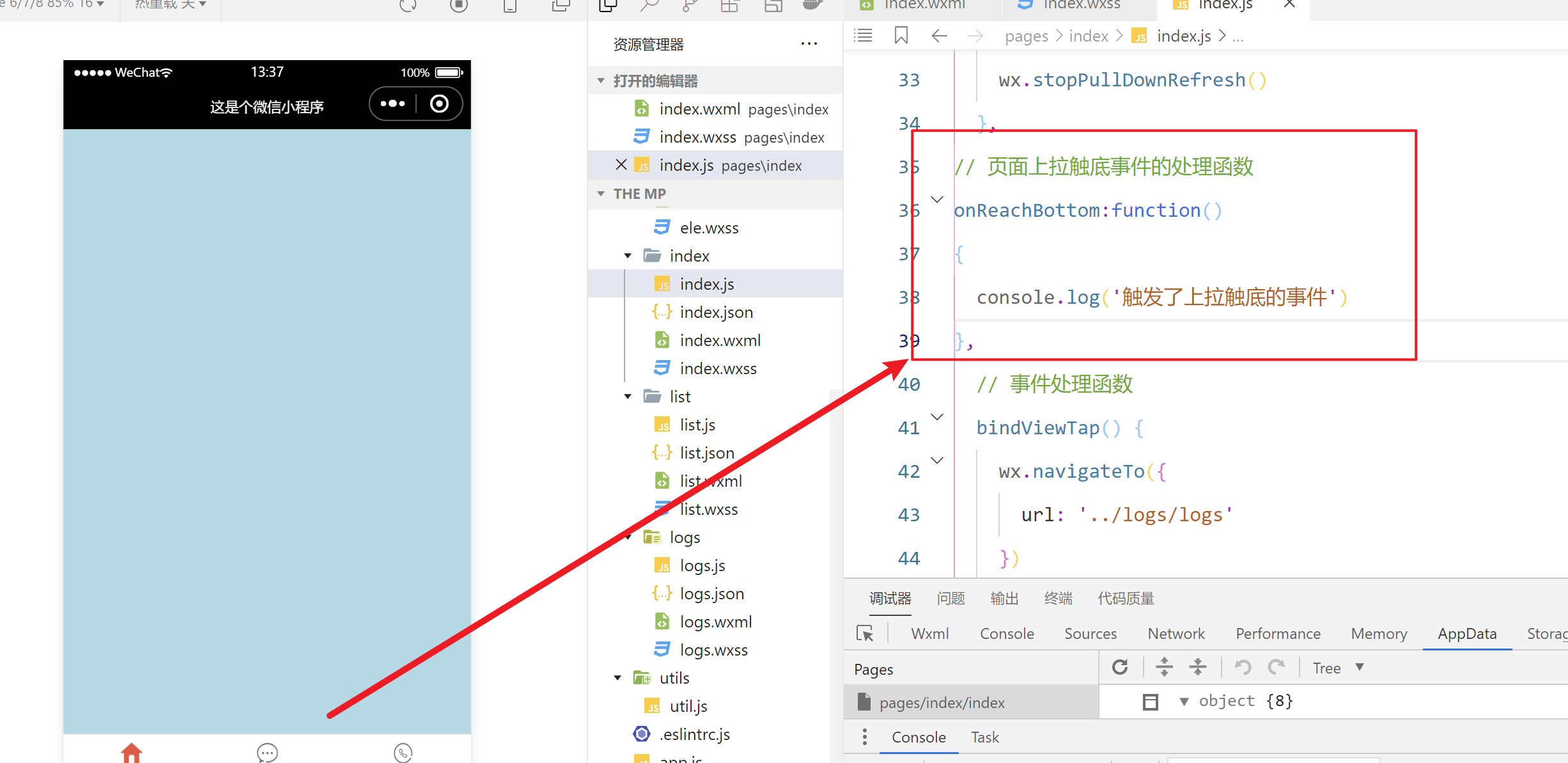
当处理完下拉刷新后，下拉刷新的loading的效果会一直显示，不会主动消息，需要调用 wx.stopPullDownRefresh()可以停止当前页面的下拉刷新：



# 四、上拉触底事件

## 1、上拉触底：

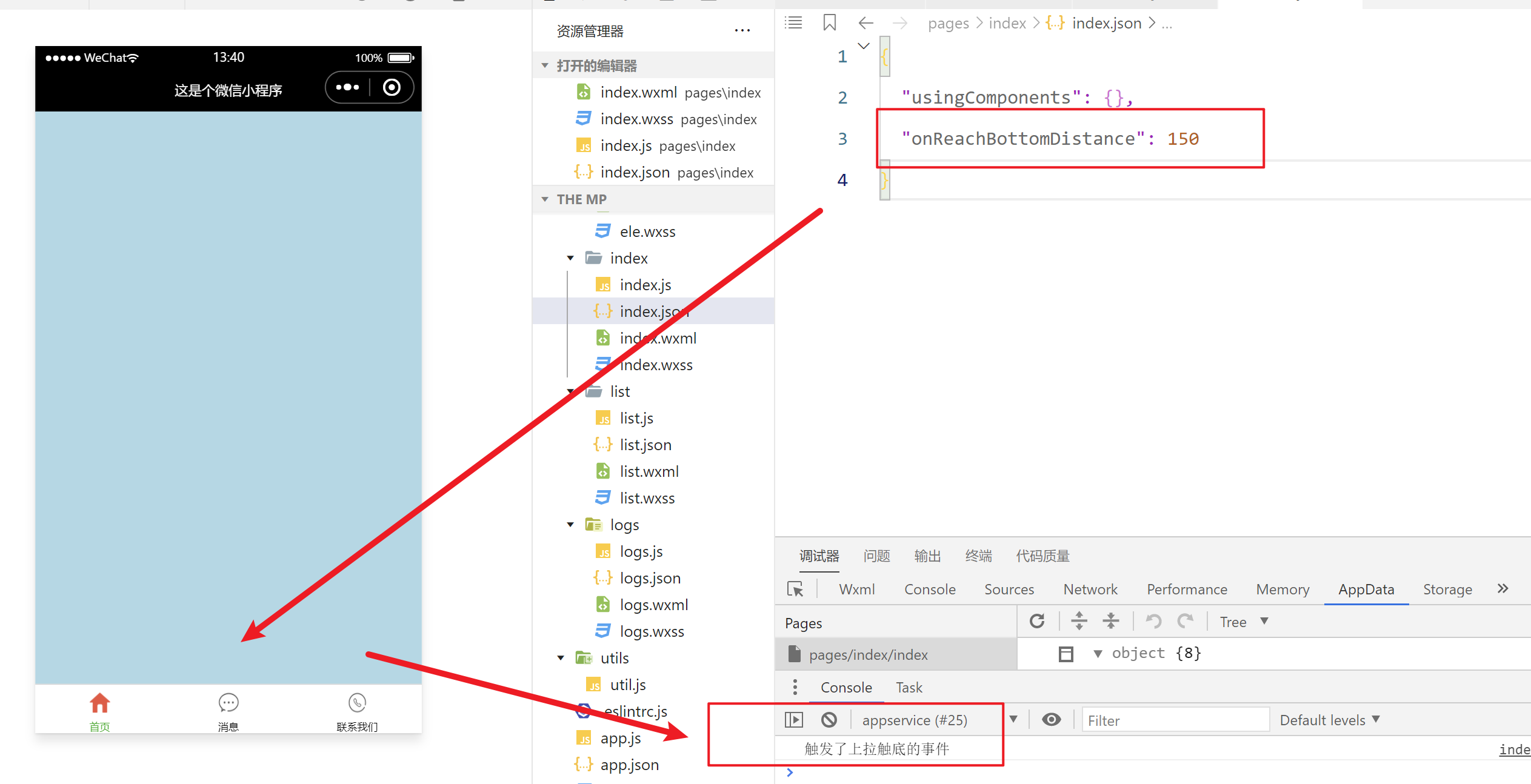
通过手指在屏幕上的上拉滑动作，从而加载更多数据的行为。



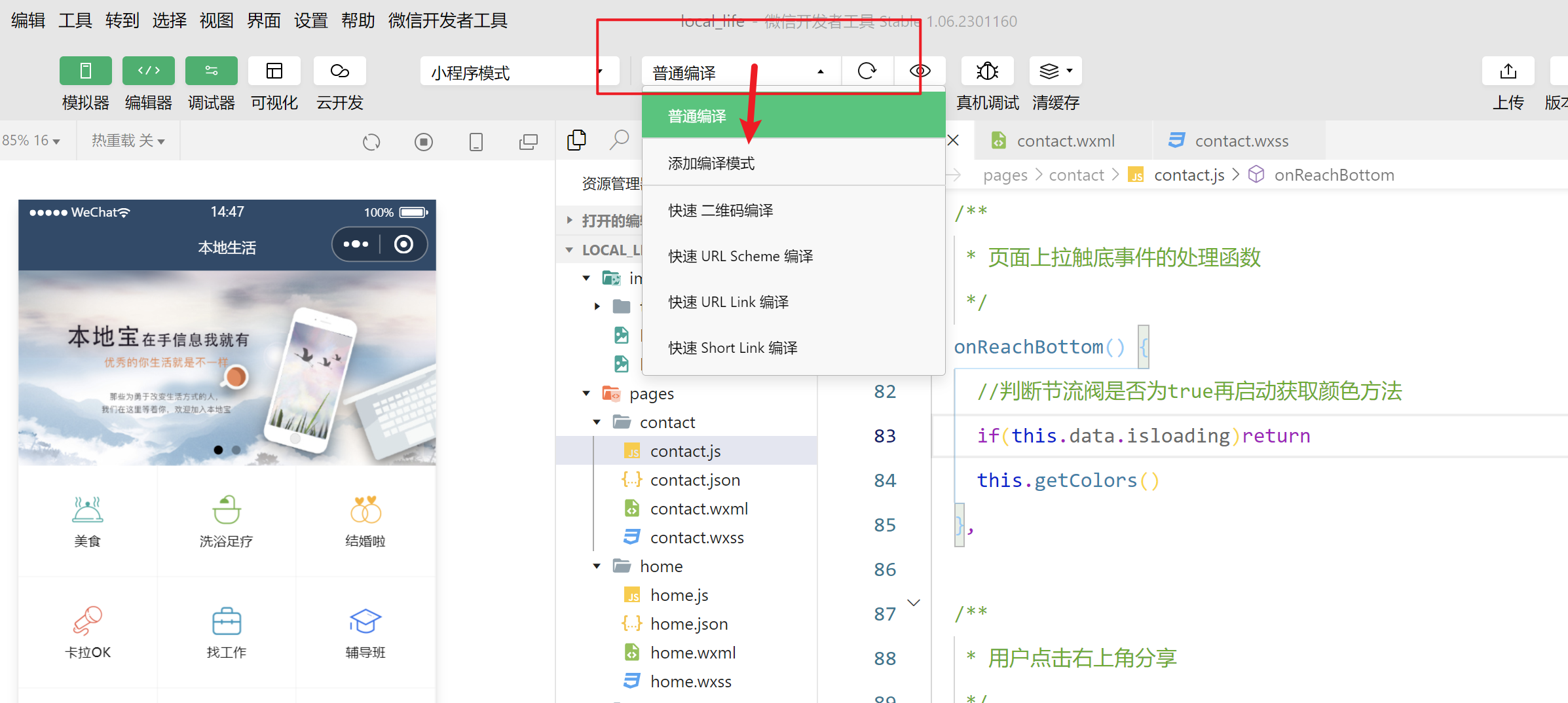
## 2、配置上拉触底距离

上拉触底距离指的是触发上拉触底事件时，滚动条距离页面底部的距离。

可以在全局或页面的.json配置文件中，通过onReachBottomDistance属性来配置上拉触底的距离（默认触底距离50px）



# 五、自定义编译模式





# 六、生命周期

生命周期是指一个对象从创建->运行->销毁->的整个阶段，强调的是一个时间段。

## 1、生命周期的分类：

1）应用生命周期：

特指小程序从启动->运行->销毁的过程

2）页面生命周期：

特指小程序中，每个页面的加载->渲染->销毁的过程

生命周期函数：由小程序框架提供的内置函数，会伴随着生命周期，自动按次序执行。

生命周期函数的作用：允许程序员在特定的时间点，执行某些特定的操作。如页面刚加载时，可在onLoad生命周期函数中初始化页面的数据。

## 2、生命周期函数分类：

1）应用生命周期函数：

特指小程序从启动->运行->销毁期间依次调用的那些函数

2）页面生命周期函数：

特指小程序中，每个页面的加载->渲染->销毁期间依次调用的那些函数

## 3、应用的生命周期函数

小程序的应用生命周期函数需要在app.js中进行声明，如：

//app.js文件

APP({

//小程序初始化完成时，执行此函数，全局只会触发一次。可以做一些初始化的工作。

onLaunch:function(options){},

//小程序启动，或从后台进入前台显示时触发。

onShow:function(options){},

//小程序从前台进入后台时触发

onHide:function(){}

})

## 4、页面的生命周期函数

小程序的页面生命周期函数需要在页面的.js文件中进行声明如：  
//页面的.js文件

Pages({

onLoad:function(options){}, //监听页面加载，一个页面只能调用1次

onShow:function(){}, //监听页面显示

onReady:function(){}, //监听页面初次渲染完成，一个页面只调用1次

onHide:function(){}, //监听页面隐藏

onUnload:function(){}, //监听页面卸载，一个页面只调用1次

})

# 七、wxs脚本

## 1、介绍

Wxs是小程序独有的一套脚本语言，结合WXML，可以构建出页面的结构。

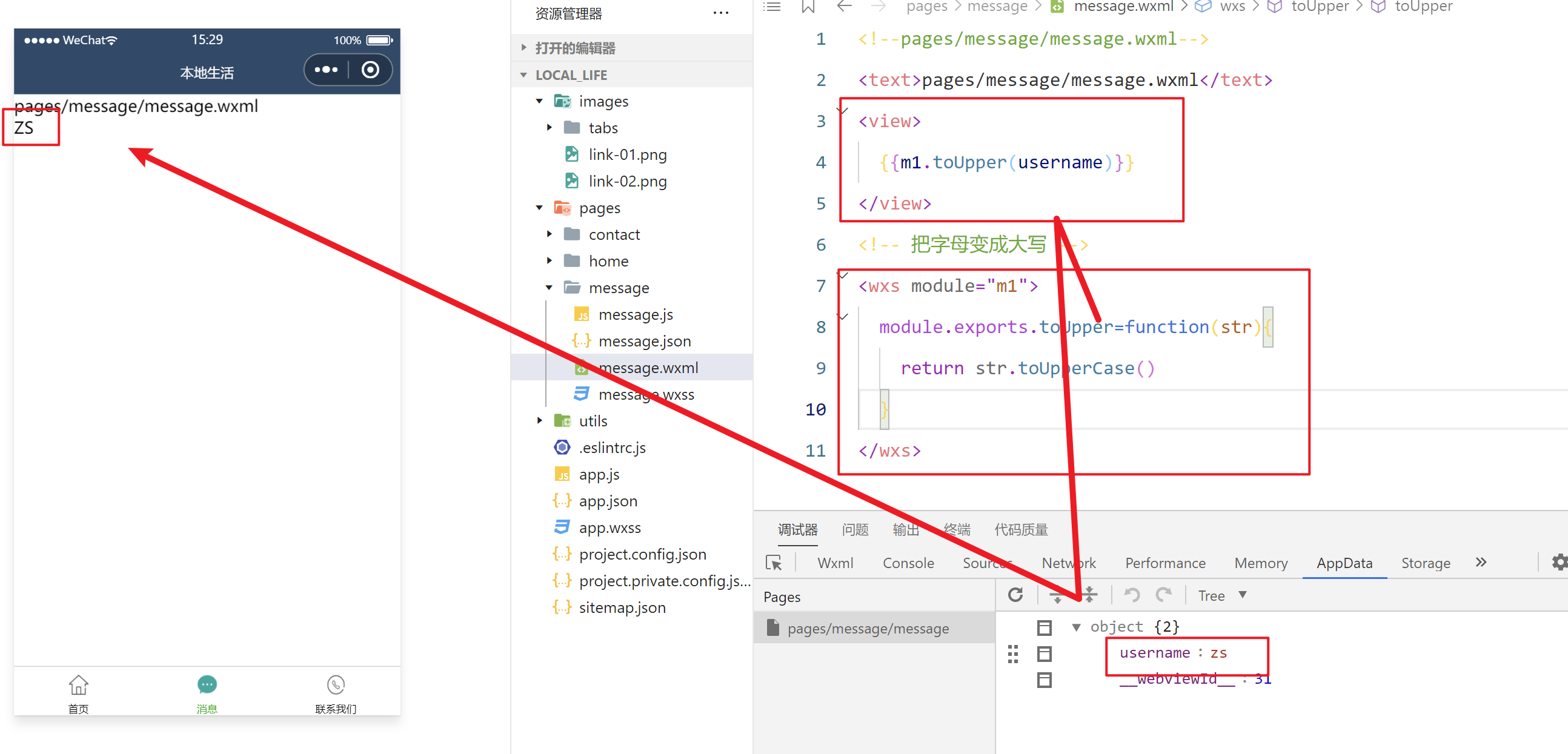
## 2、应用场景

Wxml中无法调用在页面的.js中定义的函数，但是，在wxml中可以调用wxs中定义的函数。因此，在小程序中wxs的典型应用场景就是“过滤器。

## 3、内嵌wxs脚本

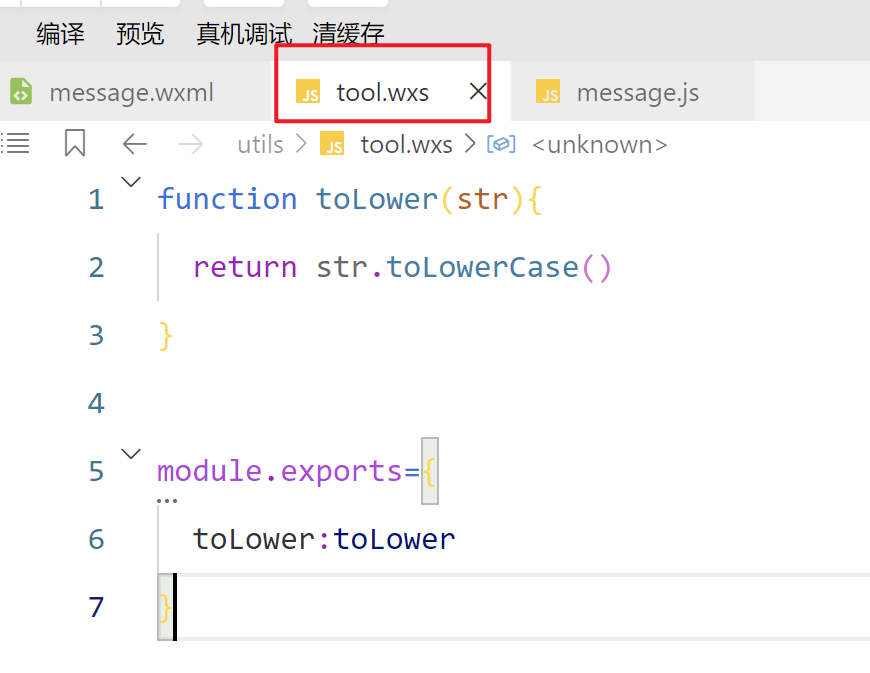
Wxs代码可以编写在wxml文件中的<wxs>标签中，就像是Javascript代码有编写在html的<script>标签内一样。

Wxml文件中的每个<wxs></wxs>标签，必须提供module属性，用来指定当前wxs的模块名称，方便在wxml中访问模块中的成员：



## 4、定义外联的wxs脚本

Wxs代码还可以编写在以.wxs为后缀名的文件内，就像JavaScript代码可以编写在以.js为后缀名的文件中一样。

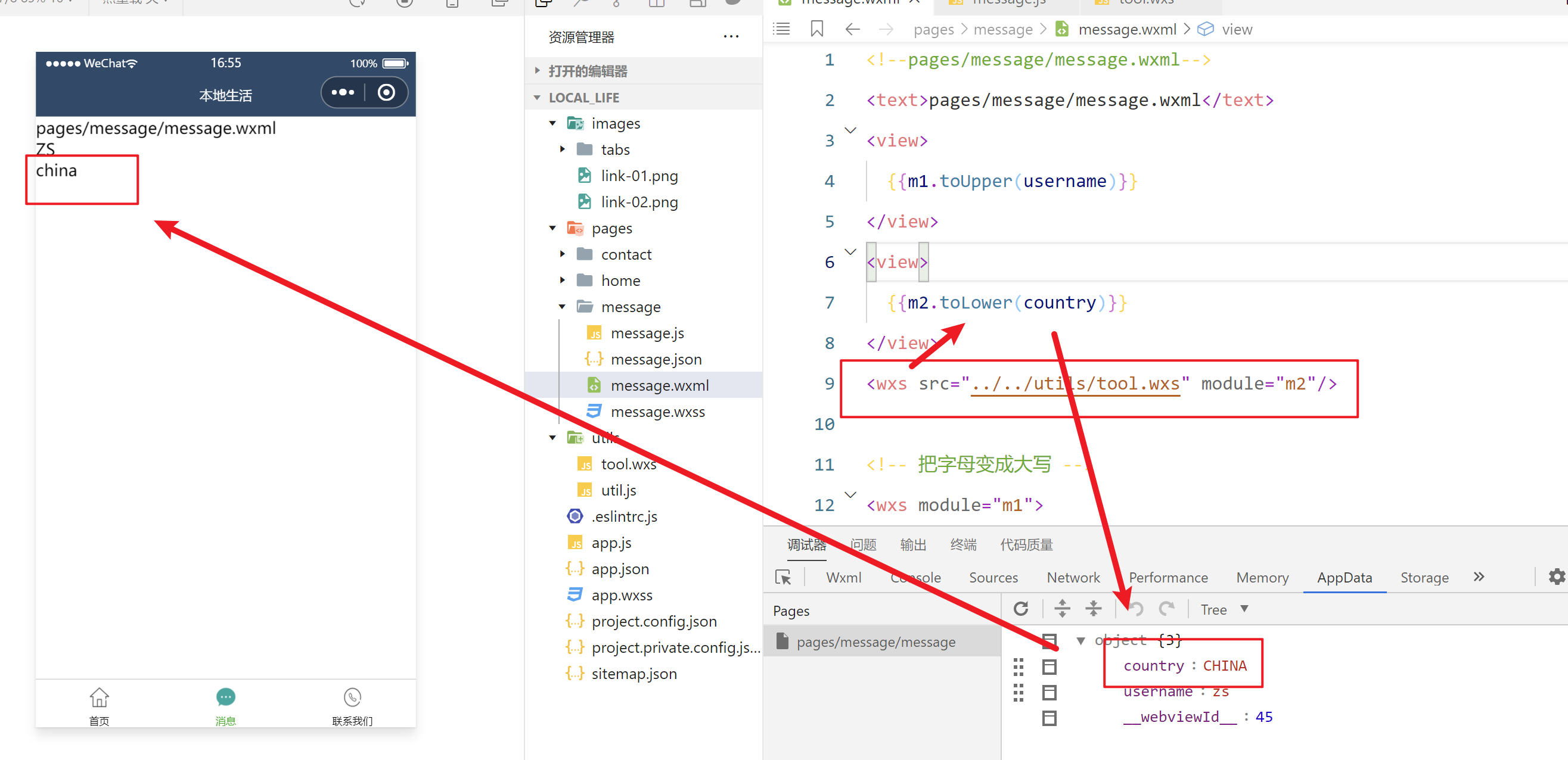


## 5、使用外联的wxs脚本

在wxml中引入外联的wxs脚本时，必须为<wxs>标签添加module和src属性，其中：

Module用来指定模块名称

Src用来指定要引入的脚本的路径，且必须是相对路径



## 6、wxs的特点

1）Wxs借鉴了JavaScript的语法，但本质上是两种完全不同的语言。

2）不能作为组件的事件回调：

Wxs典型应用场景就是过滤器，经常配合Mustache语法进行使用

3）隔离性

指不能调用js中定义的函数和小程序提供的API

4）性能好

在IOS上wxs会比JavaScript快，但android设备上无差异。