

## dodo607: 微信小程序开发日记: 重要的 `var that=this`

### 什么是微信小程序

关于什么是微信小程序在网络上已经有很多文章介绍，这里就不再阐述了。简单来说微信小程序应该是嵌套在微信里面的

应用，这个应用体现“用完即走”的理念，用户无需安装过多 APP，应用无需下载，用完就销毁。

这给我们带来很多的思考，关于微信小程序未来会给我们带来什么样的体验以及更多的思考笔者以后会在其他文章进行阐述，这里就不过多的描述。

---

### 四个交互接口

新版本的微信小程序最新版本号是 **0.10.102800**，这次更新增加了 四个交互 API:

- `wx.showToast`
- `wx.showModal`
- `wx.hideModal`
- `wx.showActionSheet`  
同时停用
- `<toast/>`
- `<loading/>`
- `<action-sheet/>`

- <modal/>

这四个组件。

所有微信提供的 API 接口都是以 wx. 开头，并且一般情况下都接收一个对象作为参数。接收的对象中可以指定 success, fail, complete 这三个函数来接收接口调用结果。

参数名	类型	必填	说明
success	Function	否	接口调用成功的回调函数
fail	Function	否	接口调用失败的回调函数
complete	Function	否	接口调用结束的回调函数（调用成功、失败都会执行）

### 回调函数

一般情况下。我们调用某个接口后都会在 success 回调函数里面处理一下成功之后的业务逻辑，比如发送个提示什么的。

这里我举例使用 wx.showActionSheet 接口，截图看程序

视图页面

< 返回

Actionsheet演示

选择国家

中国

美国

日本

英国

取消

视图页面

```
<view class="btn">  
    <button type="primary" size="default" bindtap="showactionsheet">选择国家</button>  
</view>
```

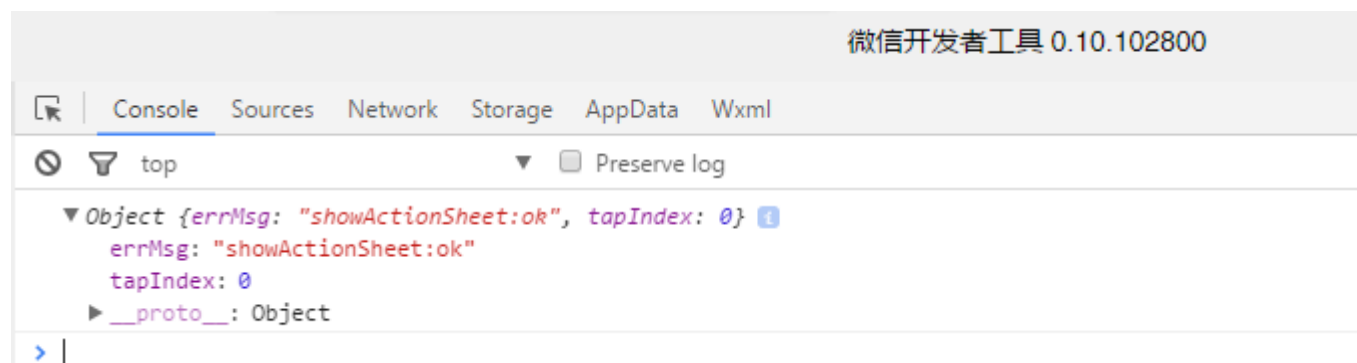
## 文件页面

```
Page({  
    data:{  
        itemLists: ['中国', '美国', '日本', '英国']  
    },  
    showactionsheet: function () {  
        wx.showActionSheet({  
            itemList: this.data.itemLists,  
            itemColor: "red",  
            success: function (res) {  
                if (!res.cancel) {  
                    console.log(res);  
                }  
            }  
        })  
    }  
})
```

```

        }
    }
})
}
})

```



调试窗口

从调试窗口可以看到接口调用成功后 `success` 函数回调返回的参数是一个对象，其中只有 `tapIndex` 代表点击中的项目索引（从上往下 `index` 从 0 开始），如果用户点击的是取消,则返回一个布尔类型的 `cancel`，并且 `cancel="true"`！这时候，如果我们想打印点击的项目的文本，就只能通过 `tapIndex` 索引去数组里面显示，聪明的你一定会想到

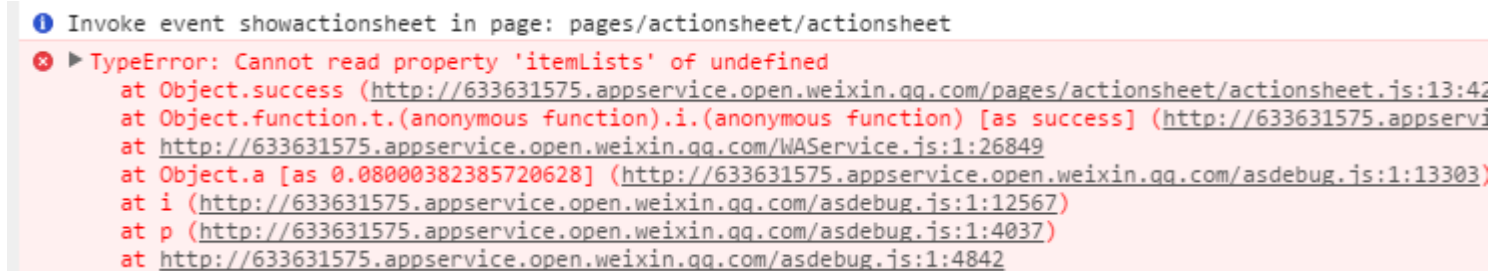
```

function (res) {
    if (!res.cancel) {

```

```
    console.log(this.data.itemLists[res.tapIndex]);  
  
  }  
  
}
```

## 这是时候我们发现报错了



```
Invoke event showactionsheet in page: pages/actionsheet/actionsheet  
✖ ▶ TypeError: Cannot read property 'itemLists' of undefined  
    at Object.success (http://633631575.appservice.open.weixin.qq.com/pages/actionsheet/actionsheet.js:13:42)  
    at Object.function.t.(anonymous function).i.(anonymous function) [as success] (http://633631575.appservice.open.weixin.qq.com/WAService.js:1:26849)  
    at http://633631575.appservice.open.weixin.qq.com/WAService.js:1:26849  
    at Object.a [as 0.08000382385720628] (http://633631575.appservice.open.weixin.qq.com/asdebug.js:1:13303)  
    at i (http://633631575.appservice.open.weixin.qq.com/asdebug.js:1:12567)  
    at p (http://633631575.appservice.open.weixin.qq.com/asdebug.js:1:4037)  
    at http://633631575.appservice.open.weixin.qq.com/asdebug.js:1:4842
```

报错 itemLists 为 undefiend

原因是说 itemLists 未定义。我们尝试把对象 this.data 也打印出来。同样发现是 undefined.

## 重要的 var that=this

原来我们忘记了 **javascript** 语言中 this 关键字的用法了。在 **javascript** 语言中，this 代表着当前的对象，它在程序中

随着执行的上下文随时会变化。在本例中回调函数对象相对于 showactionsheet 点击事件函数对象已经发生了变化。所以

已经不是原来的页面对象了。自然就没有了 data 属性，也没有了 data.itemLists 属性了。解决的办法就是复制一份当前

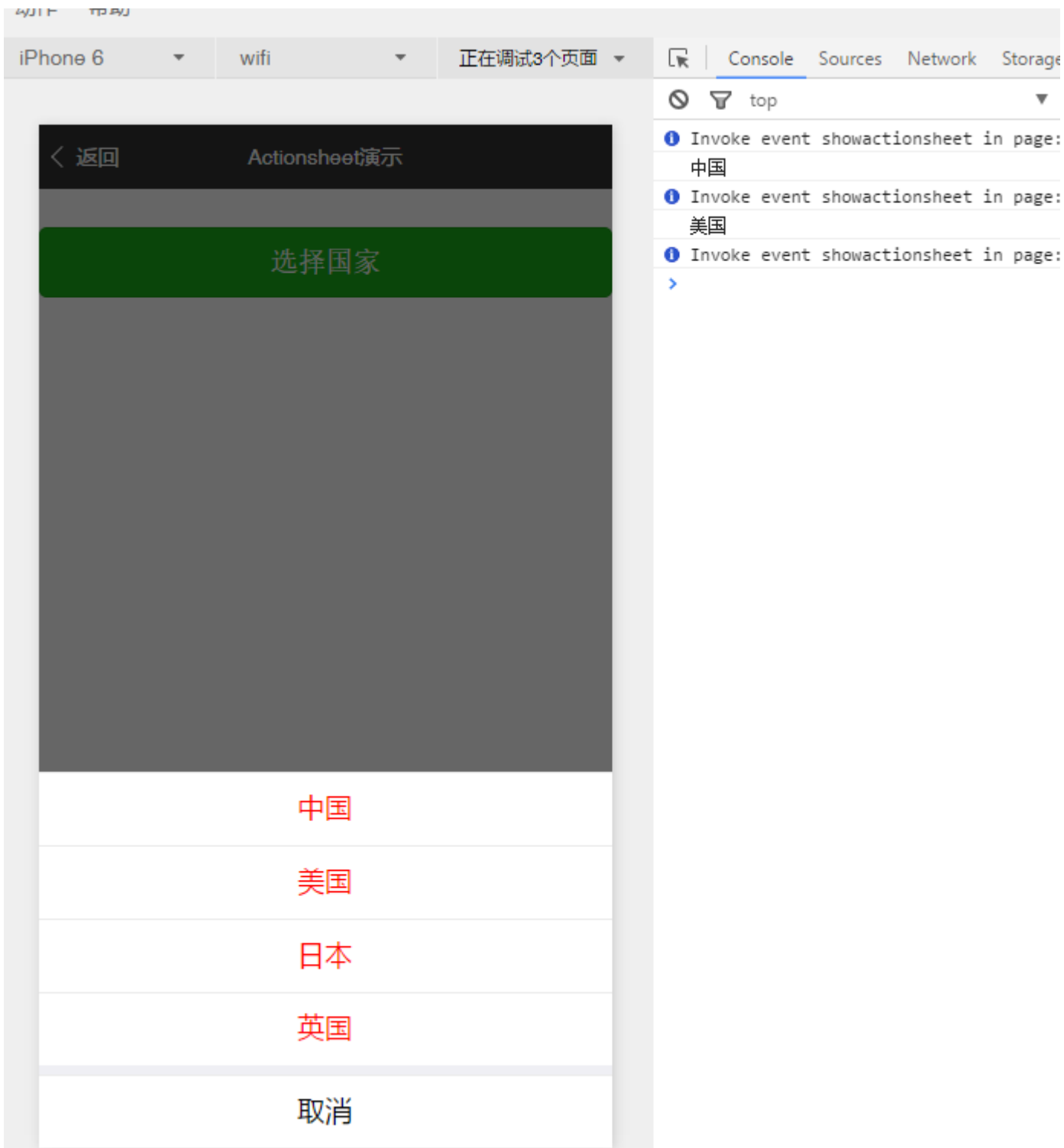
的对象。所以我们有了这个重要的语句：

```
var that=this;//把 this 对象复制到临时变量 that.
```

这时候我们使用 that 就不会找不到原来的对象了。

```
console.log(that.data.itemLists[res.tapIndex]);//使用 that.data 属性
```





正确的结果

## 其它接口也一样

这里虽然只是举例了一个接口。但是鉴于微信小程序几乎所有的 API 接口都会有 `success, fail, complete` 三个回调函数来处理业务的后续逻辑，很多时候我们要获取当前页面对象来对视图进行渲染。当我们想要获取页面的初始数据 `data` 时候，在回调函数里面就不能使用 `this.data` 来获取，同样的就不能使用 `this.setData()` 函数来渲染视图。我们需要做的就是把我们想要的 `this` 对象复制复制到 `that`。因此就有了本文的解说 **重要的 `var that=this`**。希望能够给正在学习微信小程序开发的初学者帮助。

关于 **javascript** `this` 的用法大家还是要通过实例去好好理解。这里不再详细的解释。这里附送一个传送门，可以看看这篇写得很好的 [关于深入理解 `this` 的文章](#):

[深入理解 JavaScript 中的 `this` 关键字](#)