

# ios下JS与OC互相调用（三） -- MessageHandler



Haley\_Wong (/u/a8cf6d63e889) [+ 关注](#)

2016.08.06 13:50\* 字数 855 阅读 12086 评论 48 喜欢 50 阅读 12086 评论 48 喜欢 50 (/u/a8cf6d63e889)

使用WKWebView的时候，如果想要实现JS调用OC方法，除了拦截URL之外，还有一种简单的方式。那就是利用WKWebView的新特性MessageHandler来实现JS调用原生方法。

## MessageHandler 是什么？

WKWebView 初始化时，有一个参数叫configuration，它是 WKWebViewConfiguration 类型的参数，而 WKWebViewConfiguration 有一个属性叫 userContentController，它又是 WKUserContentController 类型的参数。WKUserContentController 对象有一个方法 - addScriptMessageHandler:name:，我把这个功能简称为MessageHandler。

- addScriptMessageHandler:name: 有两个参数，第一个参数是userContentController的代理对象，第二个参数是JS里发送postMessage的对象。  
所以要使用MessageHandler功能，就必须要实现 WKScriptMessageHandler 协议。  
我们在该API的描述里可以看到在JS中的使用方法：

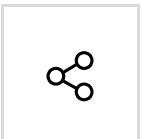
```
window.webkit.messageHandlers.<name>.postMessage(<messageBody>)
//其中<name>，就是上面方法里的第二个参数`name`。
//例如我们调用API的时候第二个参数填@"Share"，那么在JS里就是：
//window.webkit.messageHandlers.Share.postMessage(<messageBody>)
//<messageBody>是一个键值对，键是body，值可以有多种类型的参数。
// 在`WKScriptMessageHandler`协议中，我们可以看到message是`WKScriptMessage`类型，有一个属性叫body。
// 而注释里写明了body 的类型：
Allowed types are NSNumber, NSString, NSDate, NSArray, NSDictionary, and NSNull.
```

## 怎么使用MessageHandler？

1.创建 WKWebViewConfiguration 对象，配置各个API对应的 MessageHandler。

```
WKUserContentController 对象可以添加多个scriptMessageHandler。
```

看了示例代码，会很容易理解。示例代码如下：



```

// 这是创建configuration 的过程
WKWebViewConfiguration *configuration = [[WKWebViewConfiguration alloc] init]
;

WKPreferences *preferences = [WKPreferences new];
preferences.javaScriptCanOpenWindowsAutomatically = YES;
preferences.minimumFontSize = 40.0;
configuration.preferences = preferences;

- (void)viewWillAppear:(BOOL)animated
{
    [super viewWillAppear:animated];
    // addScriptMessageHandler 很容易导致循环引用
    // 控制器 强引用了WKWebView,WKWebView copy(强引用了) configuration, configuratio
n copy (强引用了) userContentController
    // userContentController 强引用了 self (控制器)
    [self.webView.configuration.userContentController addScriptMessageHandler:self
 name:@"ScanAction"];
    [self.webView.configuration.userContentController addScriptMessageHandler:self
 name:@"Location"];
    [self.webView.configuration.userContentController addScriptMessageHandler:self
 name:@"Share"];
    [self.webView.configuration.userContentController addScriptMessageHandler:self
 name:@"Color"];
    [self.webView.configuration.userContentController addScriptMessageHandler:self
 name:@"Pay"];
    [self.webView.configuration.userContentController addScriptMessageHandler:self
 name:@"Shake"];
    [self.webView.configuration.userContentController addScriptMessageHandler:self
 name:@"GoBack"];
    [self.webView.configuration.userContentController addScriptMessageHandler:self
 name:@"PlaySound"];
}

```

需要注意的是 addScriptMessageHandler 很容易引起循环引用，导致控制器无法被释放，所以需要加入以下这段：

```

- (void)viewWillDisappear:(BOOL)animated
{
    [super viewWillDisappear:animated];

    // 因此这里要记得移除handlers
    [self.webView.configuration.userContentController removeScriptMessageHandlerFo
rName:@"ScanAction"];
    [self.webView.configuration.userContentController removeScriptMessageHandlerFo
rName:@"Location"];
    [self.webView.configuration.userContentController removeScriptMessageHandlerFo
rName:@"Share"];
    [self.webView.configuration.userContentController removeScriptMessageHandlerFo
rName:@"Color"];
    [self.webView.configuration.userContentController removeScriptMessageHandlerFo
rName:@"Pay"];
    [self.webView.configuration.userContentController removeScriptMessageHandlerFo
rName:@"Shake"];
    [self.webView.configuration.userContentController removeScriptMessageHandlerFo
rName:@"GoBack"];
    [self.webView.configuration.userContentController removeScriptMessageHandlerFo
rName:@"PlaySound"];
}

```

## 2.创建WKWebView。

这里没什么好说的，直接看示例代码吧：

```

self.webView = [[WKWebView alloc] initWithFrame:self.view.frame configuration
:configuration];

NSString *urlStr = [[NSBundle mainBundle] pathForResource:@"index.html" ofType:
nil];
NSURL *fileURL = [NSURL fileURLWithPath:urlStr];
[self.webView loadFileURL:fileURL allowingReadAccessToURL:fileURL];

self.webView.navigationDelegate = self;
self.webView.UIDelegate = self;
[self.view addSubview:self.webView];

```

## 3.实现协议方法。

我这里实现了两个协议 <WKUIDelegate,WKScriptMessageHandler> , WKUIDelegate 是因为我在JS中弹出了alert 。 WKScriptMessageHandler 是因为我们要处理JS调用OC方法的请求。先看实现协议方法的示例代码：

```
#pragma mark - WKScriptMessageHandler
- (void)userContentController:(WKUserContentController *)userContentController didReceiveScriptMessage:(WKScriptMessage *)message
{
    // message.body -- Allowed types are NSNumber, NSString, NSDate, NSArray, NSDictionary, and NSNull.
    if ([message.name isEqualToString:@"ScanAction"]) {
        NSLog(@"扫一扫");
    } else if ([message.name isEqualToString:@"Location"]) {
        [self getLocation];
    } else if ([message.name isEqualToString:@"Share"]) {
        [self shareWithParams:message.body];
    } else if ([message.name isEqualToString:@"Color"]) {
        [self changeBGColor:message.body];
    } else if ([message.name isEqualToString:@"Pay"]) {
        [self payWithParams:message.body];
    } else if ([message.name isEqualToString:@"Shake"]) {
        [self shakeAction];
    } else if ([message.name isEqualToString:@"GoBack"]) {
        [self goBack];
    } else if ([message.name isEqualToString:@"PlaySound"]) {
        [self playSound:message.body];
    }
}
```

WKScriptMessage 有两个关键属性 name 和 body 。  
因为我们给每一个OC 方法取了一个name, 那么我们就可以根据name 来区分执行不同的方法。body 中存着JS 要给OC 传的参数。  
关于参数body 的解析, 我就举一个body中放字典的例子, 其他的稍后可以看demo。  
解析JS 调用OC 实现分享的参数：

```
- (void)shareWithParams:(NSDictionary *)tempDic
{
    if (![tempDic isKindOfClass:[NSDictionary class]]) {
        return;
    }

    NSString *title = [tempDic objectForKey:@"title"];
    NSString *content = [tempDic objectForKey:@"content"];
    NSString *url = [tempDic objectForKey:@"url"];
    // 在这里执行分享的操作

    // 将分享结果返回给js
    NSString *jsStr = [NSString stringWithFormat:@"shareResult('%@','%@','%@')",title,content,url];
    [self.webView evaluateJavaScript:jsStr completionHandler:^(id _Nullable result, NSError * _Nullable error) {
        NSLog(@"%@----%@",result, error);
    }];
}
```

message.boby 就是JS 里传过来的参数。我们不同的方法先做一下容错性判断。然后正常取值就可以了。

### 4.处理HTML中JS调用。

HMTL的源码跟之前的HTML内容差不多, 只有JS的调用部分改变了。

```
// 传null
function scanClick() {
    window.webkit.messageHandlers.ScanAction.postMessage(null);
}
// 传字典
function shareClick() {
    window.webkit.messageHandlers.Share.postMessage({title: '测试分享的标题', content: '测试分享的内容', url: 'http://www.baidu.com'});
}
// 传字符串
function playSound() {
    window.webkit.messageHandlers.PlaySound.postMessage('shake_sound_male.wav');
}
// 传数组
function colorClick() {
    window.webkit.messageHandlers.Color.postMessage([67,205,128,0.5]);
}
```

## 5.OC调用JS

这里使用WKWebView 实现OC 调用JS方法跟上一篇是一样的，还是利用

– evaluateJavaScript:completionHandler: 。像下面这样使用：

```
// 将分享结果返回给js
NSString *jsStr = [NSString stringWithFormat:@"shareResult('%@','%@','%@')",title,content,url];
[self.webView evaluateJavaScript:jsStr completionHandler:^(id _Nullable result, NSError * _Nullable error) {
    NSLog(@"%@----%@",result, error);
}];
```

## 使用MessageHandler 的好处

- 1.在JS中写起来简单，不用再用创建URL的方式那么麻烦了。
- 2.JS传递参数更方便。使用拦截URL的方式传递参数，只能把参数拼接在后面，如果遇到要传递的参数中有特殊字符，如&、=、？等，必须得转换，否则参数解析肯定会出错。

例如传递的url是这样的：

http://www.baidu.com/share/openShare.htm?share\_uuid=shdfxdfdsfsdf&name=1234556

使用拦截URL 的JS调用方式

```
loadURL("firstClick://shareClick?title=分享的标题&content=分享的内容&url=链接地址&imagePath=图片地址"); }
```

将上面的url 放入链接地址这里后，根本无法区分share\_uuid是其他参数，还是url里附带的参数。

但是使用MessageHandler 就可以避免特殊字符引起的问题。

## 效果图

更详细的使用步骤还是去工程中查看吧。地址：JS\_OC\_MessageHandler  
(https://link.jianshu.com?t=https://github.com/Haley-Wong/JS\_OC/tree/master/JS\_OC\_MessageHandler)



Haley\_Wong (/u/a8cf6d63e889)

写了 79307 字，被 2076 人关注，获得了 1488 个喜欢  
(/u/a8cf6d63e889) 写了 79307 字，被 2076 人关注，获得了 1488 个喜欢

+ 关注

Brighter! Brighter! Brighter!

喜欢 (/sign\_in?utm\_source=desktop&utm\_medium=not-signed-in-like-button) | 50







更多分享

(http://cwb.assets.jianshu.io/notes/images/5141270/)






下载简书 App ▶

随时随地发现和创作内容



(/apps/download?utm\_source=nbc)

被以下专题收入，发现更多相似内容

-  iOS进阶 (/c/4c5ec59fad9a?utm\_source=desktop&utm\_medium=notes-included-collection)
-  iOS富文本 (/c/835b45134617?utm\_source=desktop&utm\_medium=notes-included-collection)
-  iOS Dev... (/c/c3b8ecef225?utm\_source=desktop&utm\_medium=notes-included-collection)
-  iOS Dev... (/c/e84a7722d673?utm\_source=desktop&utm\_medium=notes-included-collection)
-  [0010]O... (/c/331f76ad0d36?utm\_source=desktop&utm\_medium=notes-included-collection)
-  iOS-swift (/c/31622a1d4959?utm\_source=desktop&utm\_medium=notes-included-collection)
-  关注 (/c/6505b3ff2a72?utm\_source=desktop&utm\_medium=notes-included-collection)

展开更多

### 剖析OC与JS交互 (/p/63a5d1678d6a?utm\_campaign=maleskine&utm\_...

随着H5技术的兴起，在iOS开发过程中，难免会遇到原生应用需要和H5页面交互的问题。其中会涉及方法调用及参数传值等场景。iOS原生应用和web页面的交互大致上有这几种方法：1) URL拦截协议；（兼容iOS6

 Chris\_js (/u/f8c509fc37d0?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/4db513ed2c1a?

Objective-C	JavaScript	Bridge Convert	Bridge Constructor
nil	undefined		valueWithUndefinedInContext
NSNumber	null		valueWithNullInContext
NSString	string	toString	
NSNumber	number, boolean	toNumber, toBoolean	valueWithBooleanInContext, valueWithDoubleInContext, valueWithIntegerInContext, valueWithIntegerInContext, valueWithIntegerInContext
NSDictionary	Object object	toDictionary	valueWithNewObjectInContext
NSArray	Array object	toArray	valueWithNewArrayInContext
NSDate	Date object	toDate	
NSString	Function object		
id	Wrapper object	toObject, toObjectClass	valueWithObjectInContext
Class	Constructor object		

utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

### iOS下JS与OC互相调用（四）--JavaScriptCore (/p/4db513ed2c1a?utm...

前面讲完拦截URL的方式实现JS与OC互相调用，终于到JavaScriptCore了。它是从iOS7开始加入的，用Objective-C 把 WebKit 的 JavaScript 引擎封装了一下，提供了简单快捷的方式与JavaScript交互。关于

 Haley\_Wong (/u/a8cf6d63e889?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/870dba42ec15?



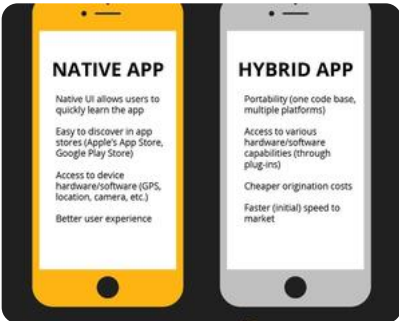
utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

### iOS中UIWebView与WKWebView、JavaScript与OC交互、Cookie管理看...

前言 关于UIWebView的介绍，相信看过上文的小伙伴们，已经大概清楚了吧，如果有问题，欢迎提问。本文是本系列文章的第二篇，主要为小伙伴们分享下WKWebView相关的内容：iOS中UIWebView与

 Dark\_Angel (/u/9f088cde1295?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/a5512a4a6817?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)



## iOS 与 JS 交互开发知识总结 (/p/a5512a4a6817?utm\_campaign=maleskine...

前言 Web 页面中的 JS 与 iOS Native 如何交互是每个 iOS 猿必须掌握的技能。而说到 Native 与 JS 交互，就不得不提一嘴 Hybrid。 Hybrid 的翻译结果并不是很文明（擦汗，不知道为啥很多翻译软件会译为“杂

 幽城88 (/u/3c2f66e712e9?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

## iOS下JS与OC互相调用（二）--WKWebView 拦截URL (/p/99c3af6894f...

在上篇文章中讲述了使用UIWebView拦截URL的方式来处理JS与OC交互。由于UIWebView比较耗内存，性能上不太好，而苹果在iOS 8中推出了WKWebView。同样的用WKWebView也可以拦截URL，做JS 与OC交

 Haley\_Wong (/u/a8cf6d63e889?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/e89d13ba0869?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)  
**华润紫云府 70 现代风格 (/p/e89d13ba0869?utm\_campaign=maleskine...**

此房子设计的颇具个性居室，多以原木色家居为蓝调，配以淡暖色配饰，简约，大方。本案是一室格局，但是居住要求是两室格局，所以本案将客厅一分为二，实用感较强。整个空间结构有较大的改动，动线清晰，

 0660fcfd2d40 (/u/0660fcfd2d40?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/e969ef722cc9?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)  
**周六 (/p/e969ef722cc9?utm\_campaign=maleskine&utm\_content=not...**

早上睡到自然醒，很无聊，本来身体不太舒服想躺着，可是心里着急，躺不住了。然后就起床，开始彻底打扫宿舍，打扫完了，我已经浑身出汗了，换好衣服，梳起头发，穿好鞋，出门！先出去在伊能逛了逛，没啥

 二苗\_5c0f (/u/690ee9aaae0f?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

## 2017-07-06 (/p/777dd375868c?utm\_campaign=maleskine&utm\_cont...

成熟并不是丧失了纯真和热情的老于世故、左右逢源，而是智慧和知识的增长带来的理性、持重和责任感，读书越多，责任越大～

 简单点xixi (/u/a2ef6b5bd4e1?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

## 3-3今天，驱动你早上起床的到底是什么？ (/p/607f64fe8ec4?utm\_camp...

1、理解概念的三个视角是：显微镜，平面镜和望远镜。 2、生涯把职业分成三个阶段：最主要的区别是我们对于职业需求的改变。生存期大约3-5年；发展期的特征，是越过收入看增速、越过赚钱看本钱；发展期两

 高平中学生职业规划咨询师 (/u/937666af121e?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/80b0eddd63c0?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

## 钢造尤物，声来狂野——BGVP DM5双圈双铁耳塞评测 (/p/80b0eddd63c...

圈铁类耳塞是目前耳机市场中颇受欢迎的产品，各色各样的圈铁耳塞层出不穷。市面上的圈铁类耳塞多以一圈一铁和一圈多铁为主，而以多圈多铁为卖点的耳塞则少之又少，一直以来多单元耳塞都存在衔接的问题，



翰逗逗 (/u/9002cb1fd9c9?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)