

iOS下JS与OC互相调用（一） -- UIWebView 拦截URL



Haley_Wong (/u/a8cf6d63e889) [+ 关注](#)

2016.08.06 13:44* 字数 1324 阅读 10333 评论 13 喜欢 55 阅读 10333 评论 13 喜欢 55

(/u/a8cf6d63e889)

最近准备把之前用UIWebView实现的JS与原生相互调用功能，用WKWebView来替换。
顺便搜索整理了一下JS 与OC 交互的方式，非常之多啊。目前我已知的JS 与 OC 交互的
处理方式：

- 1.在JS 中做一次URL跳转，然后在OC中拦截跳转。（这里分为UIWebView 和 WKWebView两种，去年因为还要兼容iOS 6，所以没办法只能采用UIWebView来做。）
- 2.利用WKWebView 的MessageHandler。
- 3.利用系统库JavaScriptCore，来做相互调用。（iOS 7推出的）
- 4.利用第三方库WebViewJavascriptBridge。
- 5.利用第三方cordova库，以前叫PhoneGap。（这是一个库平台的库）
- 6.当下盛行的React Native。

我去年也写过一个相互调用的总结：iOS下JS与原生OC互相调用(总结)
(<https://www.jianshu.com/p/d19689e0ed83>)。

写的比较粗糙，因此准备新开一个目录专题来记录JS 与原生交互的处理方式。只是记录JS与OC交互的多种方式，大家可以根据实际情况和场景选择适合自己的方式。

今天就详细的介绍一下使用UIWebView拦截URL 的方式来实现JS与OC 的交互。

为什么不使用第三方库或者RAC呢？

因为就相互调用的接口使用的非常少啊，就那么三两个，完全没必要使用牛刀啊。





UIWebView 拦截URL

我之前就使用的是UIWebView + 拦截URL 的方式实现的JS与OC 交互。
原因是因为要兼容iOS 6。

1.创建UIWebView，并加载本地HTML。

加载本地HTML的目的是便于自己写JS调用做测试，最终肯定还是加载网络HTML。

```
self.webView = [[UIWebView alloc] initWithFrame:self.view.frame];
self.webView.delegate = self;
NSURL *htmlURL = [[NSBundle mainBundle] URLForResource:@"index.html" withExtension:nil];
//    NSURL *htmlURL = [NSURL URLWithString:@"http://www.baidu.com"];
NSURLRequest *request = [NSURLRequest requestWithURL:htmlURL];

// 如果不要webView 的回弹效果
self.webView.scrollView.bounces = NO;
// UIWebView 滚动的比较慢，这里设置为正常速度
self.webView.scrollView.decelerationRate = UIScrollViewDecelerationRateNormal
;

[self.webView loadRequest:request];
[self.view addSubview:self.webView];
```

本地的HTML里，我定义了几个按钮，来触发调用原生的方法，然后再将执行结果回调到js 里。

```
<input type="button" value="扫一扫" onclick="scanClick()" />
<input type="button" value="获取定位" onclick="locationClick()" />
<input type="button" value="修改背景色" onclick="colorClick()" />
<input type="button" value="分享" onclick="shareClick()" />
<input type="button" value="支付" onclick="payClick()" />
<input type="button" value="摇一摇" onclick="shake()" />
<input type="button" value="返回" onclick="goBack()" />

// js 就列几个比较有代表性的functions:

function loadURL(url) {
    var iFrame;
    iFrame = document.createElement("iframe");
    iFrame.setAttribute("src", url);
    iFrame.setAttribute("style", "display:none;");
    iFrame.setAttribute("height", "0px");
    iFrame.setAttribute("width", "0px");
    iFrame.setAttribute("frameborder", "0");
    document.body.appendChild(iFrame);
    // 发起请求后这个iFrame就没用了，所以把它从dom上移除掉
    iFrame.parentNode.removeChild(iFrame);
    iFrame = null;
}

function asyncAlert(content) {
    setTimeout(function(){
        alert(content);
    },1);
}

function locationClick() {
    loadURL("haleyAction://getLocation");
}

function setLocation(location) {
    asyncAlert(location);
    document.getElementById("returnValue").value = location;
}
```

虽然HTML 内容很少，但是还有不少学问：

1.为什么自定义一个 loadURL 方法，不直接使用 window.location.href？

答：因为如果当前网页正使用 window.location.href 加载网页的同时，调用 window.location.href 去调用OC原生方法，会导致加载网页的操作被取消掉。同样的，如果连续使用 window.location.href 执行两次OC原生调用，也有可能导致第一次的操作被取消掉。所以我们使用自定义的 loadURL，来避免这个问题。

loadURL 的实现来自关于UIWebView和PhoneGap的总结

(<https://link.jianshu.com?t=http://blog.devtang.com/blog/2012/03/24/talk-about-uiwebview-and-phonegap/>)一文。

2.为什么loadURL 中的链接，使用统一的scheme？

答:便于在OC 中做拦截处理，减少在JS中调用一些OC 没有实现的方法时，webView 做跳转。因为我在OC 中拦截URL 时，根据scheme (即 haleyAction)来区分是调用原生的方法还是正常的网页跳转。然后根据host（即//后的部分 getLocation）来区分执行什么操作。

3.为什么自定义一个 asyncAlert 方法？

答：因为有的JS调用是需要OC 返回结果到JS

的。 stringByEvaluatingJavaScriptFromString 是一个同步方法，会等待js 方法执行完成，而弹出的alert 也会阻塞界面等待用户响应，所以他们可能会造成死锁。导致alert 卡死界面。如果回调的JS 是一个耗时的操作，那么建议将耗时的操作也放入 setTimeout 的 function 中。

2.拦截 URL

UIWebView 有一个代理方法，可以拦截到每一个链接的Request。return YES,webView 就会加载这个链接；return NO,webView 就不会加载这个连接。我们就在这个拦截的代理方法中处理自己的URL。

这是我的示例代码：

```
#pragma mark - UIWebViewDelegate
- (BOOL)webView:(UIWebView *)webView shouldStartLoadWithRequest:(NSURLRequest *)request navigationType:(UIWebViewNavigationType)navigationType
{
    NSURL *URL = request.URL;
    NSString *scheme = [URL scheme];
    if ([scheme isEqualToString:@"haleyaction"]) {
        [self handleCustomAction:URL];
        return NO;
    }
    return YES;
}
```

这里通过scheme，来拦截掉自定义的URL 就非常容易了，如果不同的方法使用不同的scheme，那么判断起来就非常的麻烦。

然后，看看我的处理连接的方法：

```
#pragma mark - private method
- (void)handleCustomAction:(NSURL *)URL
{
    NSString *host = [URL host];
    if ([host isEqualToString:@"scanClick"]) {
        NSLog(@"扫一扫");
    } else if ([host isEqualToString:@"shareClick"]) {
        [self share:URL];
    } else if ([host isEqualToString:@"getLocation"]) {
        [self getLocation];
    } else if ([host isEqualToString:@"setColor"]) {
        [self changeBGColor:URL];
    } else if ([host isEqualToString:@"payAction"]) {
        [self payAction:URL];
    } else if ([host isEqualToString:@"shake"]) {
        [self shakeAction];
    } else if ([host isEqualToString:@"goBack"]) {
        [self goBack];
    }
}
```

顺便看一下如何将结果回调到JS中：

```
- (void)getLocation
{
    // 获取位置信息

    // 将结果返回给js
    NSString *jsStr = [NSString stringWithFormat:@"setLocation('%@')",@"广东省深圳市南山区学府路XXXX号"];
    [self.webView stringByEvaluatingJavaScriptFromString:jsStr];
}
```

当然，有时候我们在JS中调用OC 方法的时候，也需要传参数到OC 中，怎么传呢？
就像一个get 请求一样，把参数放在后面：

```
function shareClick() {
    loadURL("haleyAction://shareClick?title=测试分享的标题&content=测试分享的内容&url=http://www.baidu.com");
}
```

那么如果获取到这些参数呢？
所有的参数都在URL的query中，先通过 & 将字符串拆分，在通过 = 把参数拆分成key 和实际的值。下面有示例代码：

```
- (void)share:(NSURL *)URL
{
    NSArray *params =[URL.query componentsSeparatedByString:@"&"];

    NSMutableDictionary *tempDic = [NSMutableDictionary dictionary];
    for (NSString *paramStr in params) {
        NSArray *dicArray = [paramStr componentsSeparatedByString:@"="];
        if (dicArray.count > 1) {
            NSString *decodeValue = [dicArray[1] stringByReplacingPercentEscapesUsingEncoding:NSUTF8StringEncoding];
            [tempDic setObject:decodeValue forKey:dicArray[0]];
        }
    }

    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 stringByEvaluatingJavaScriptFromString:jsStr];
}
```

3. OC调用JS方法

关于将OC 执行结果返回给JS 需要注意的是：

如果回调执行的JS 方法带参数，而参数不是字符串时，不要加 单引号 ,否则可能导致调用JS 方法失败。比如我这样的：

```
NSData *jsonData = [NSJSONSerialization dataWithJSONObject:userProfile options:NSJSONWritingPrettyPrinted error:nil];
NSString *jsonStr = [[NSString alloc] initWithData:jsonData encoding:NSUTF8StringEncoding];
NSString *jsStr = [NSString stringWithFormat:@"loginResult('%@','%@')",type, jsonStr];
[_webView stringByEvaluatingJavaScriptFromString:jsStr];
```

如果第二个参数用单引号包起来，就会导致JS端的loginResult不会调用。

另外，利用 [webView stringByEvaluatingJavaScriptFromString:@"var arr = [3, 4, 'abc'];"]; ,可以往HMTL的JS环境中插入全局变量、JS方法等。

示例工程地址： JS_OC_URL (https://link.jianshu.com?t=https://github.com/Haley-Wong/JS_OC/tree/master/JS_OC_URL)

小礼物走一走，来简书关注我

赞赏支持

 JS与OC交互 (/nb/5513092)

[举报文章](#)

© 著作权归作者所有



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)

55



更多分享

(<http://cwb.assets.jianshu.io/notes/images/5123977/>)



下载简书 App ▶

随时随地发现和创作内容



(/apps/download?utm_source=nbc)




后发表评论

(/sign_in?utm_source=desktop&utm_medium=not-signed-in-comment-form)

13条评论

只看作者

[按喜欢排序](#) [按时间正序](#) [按时间倒序](#)



态度哥 (/u/046e3fe2beea)

2楼 · 2016.08.08 15:31


(/u/046e3fe2beea)

如何拦截 请求中的 css 地址呢 ...


 赞

 回复

Haley_Wong (/u/a8cf6d63e889): @魏同学 (/users/046e3fe2beea) css 你可以在UIWebView的代理方法中测试一下能否触发加载css文件，不行的话，只能用JavaScriptCore了。


2016.08.08 17:28  回复


Misaki_yuyi (/u/abb55beb0e35): @Haley_Wong (/users/a8cf6d63e889) 不知道层主有没有解决问题，现在也遇到了这样的需求，不知道怎么办？

2016.09.07 09:21  回复

态度哥 (/u/046e3fe2beea): @Misaki_yuyi (/users/abb55beb0e35) 可以用NSURLProtocol 这是我写的简书地址

<http://www.jianshu.com/p/2ed92d399b4b>
(<http://www.jianshu.com/p/2ed92d399b4b>)

2016.09.07 09:53  回复

 添加新评论



小小聪明屋123 (/u/f9752836b891)

3楼 · 2016.10.18 11:10


(/u/f9752836b891)

你好，想问一下，网页加载完成，如何实现自动调用这个方法，而不是通过点击事件；我们后台那边回调我们iOS的方法时，无法实现自动调用，只能通过点击才能回调，安卓那边是可以自动调用的，请问这个问题该怎样解决？


 赞  回复

Haley_Wong (/u/a8cf6d63e889): @小小聪明屋123 (/users/f9752836b891)

我测试过自动调用是没问题的，我在js 中调用改变背景色的方法都是正常的。你让你们的前端改下调用的时机。


2016.10.18 13:39  回复


LVLVZhiYu (/u/1a51b25c4e57): 这个问题解决了没 我也遇到了这样的问题 很急呀

2016.12.08 15:38  回复

我只是前端 (/u/937f2cd2bd28): @Haley_Wong (/users/a8cf6d63e889) 你

好，能看下js那块是怎么写的吗

2017.08.18 15:34  回复

 添加新评论



zcz (/u/3944b67b2dcc)


4楼 · 2017.03.06 16:29

(/u/3944b67b2dcc)


为什么点击js中的返回按钮，有进入goBack方法，但是就是返回不了呢


 赞  回复

Haley_Wong (/u/a8cf6d63e889): 看下线程是不是在主线程？

2017.03.06 17:21  回复

zcz (/u/3944b67b2dcc): @Haley_Wong (/users/a8cf6d63e889) 已经解决，3Q

2017.03.06 17:44  回复

 添加新评论



sock (/u/0781fa865c11)

5楼 · 2017.08.02 19:47

(/u/0781fa865c11)

//我这样 调不动前端得方法

```
- (void)webViewDidFinishLoad:(UIWebView *)webView
{
```

```
NSString* jsStr = [NSString stringWithFormat:@"setLocation('%@')",@"11111"];
[self.webView stringByEvaluatingJavaScriptFromString:jsStr];
```

```
}
```

//前端实现的方法

```
function setLocation(location) {
alert(location)
asyncAlert(location);
}
```

我用dome实验可以调

sock (/u/0781fa865c11): 这就很尴尬😓

2017.08.02 19:48 💬 回复

✍️ 添加新评论

被以下专题收入，发现更多相似内容

- 

iOS富文本 (/c/835b45134617?utm_source=desktop&utm_medium=notes-included-collection)
- 

iOS Dev... (/c/e84a7722d673?utm_source=desktop&utm_medium=notes-included-collection)
- 

iOS Dev... (/c/c3b8ecefe225?utm_source=desktop&utm_medium=notes-included-collection)
- 

[0010]O... (/c/331f76ad0d36?utm_source=desktop&utm_medium=notes-included-collection)
- 

iOS开发 (/c/bd4548805c58?utm_source=desktop&utm_medium=notes-included-collection)
- 

iOS文本 (/c/c7c4a8437d02?utm_source=desktop&utm_medium=notes-included-collection)
- 

技术 (/c/f089e5440a71?utm_source=desktop&utm_medium=notes-included-collection)

展开更多 ▾

推荐阅读

更多精彩内容 > (/)

iOS 知识小集（Status Bar变换） (/p/c2cb6f9b189e?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

背景 iOS 中经常会有需要在某个界面改变状态栏颜色或者某个界面隐藏状态栏的需求。而改变状态栏颜色和控制状态栏显示和隐藏的API，在iOS 的不同版本中也发生了很多变化。iOS 7以前 在iOS 7之前，状态栏是

Haley_Wong (/u/a8cf6d63e889?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

iOS下JS与OC互相调用（二）--WKWebView 拦截URL (/p/99c3af6894f... (/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=pc_all_hots&utm_source=recommendation)

月薪5千的北漂，晒出的账单都像月薪2万 (/p/dbfba5e43f3e?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

最近流行晒年度账单，很多人得到的关键词，都是一个大大的“穷”字。一笔笔账单下来，就是一个个暴击。账单里写着超过全国80%、甚至90%的用户，突然

谈心社 (/u/50e0da62c77d?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

2017年读了一年书，我最喜欢这4本小说 (/p/8779a0ff699b?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

文/怀左同学 今年年初的时候，我给自己定了一个写一百万字的目标，但很遗憾，最后写了一年，也只完成了一半，兜兜转转，五十万字。好在除写作之

怀左同学 (/u/62478ec15b74?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

贾乃亮道歉：爱得卑微的人，到底错在哪里？ (/p/76367b241763?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

01 关于李小璐和PG one的“夜宿风波”，已经过去了一周了，但网络上的议论热度丝毫没有减退。PG one一夜之间从云端跌入谷底，新华网、紫光阁、人民日报

衷曲无闻 (/u/deeea9e09cbc?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

iOS下JS与OC互相调用（五）--UIWebView + WebViewJavascriptBridge...

WebViewJavascriptBridge是一个有点年代的JS与OC交互的库，使用该库的著名应用还挺多的，目前这个库有7000+star。我去翻看了它的第一版本已经是4年前了，在版本V4.1.4以及之前，该库只有一个类和一个

 Haley_Wong (/u/a8cf6d63e889?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

剖析OC与JS交互 (/p/63a5d1678d6a?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

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)

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/e09305d90fba?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)



iOS与JS(一)：使用URL拦截的方式进行JS与OC互相调用 (/p/e09305d90f...)

参考：iOS下JS与OC互相调用（二）--WKWebView 拦截URLiOS下JS与原生OC互相调用(总结) 关于WKWebView与UIWebView之间的区别，本文就不在说明，本文只使用WKWebView。效果图如下： html代

 Andy_Ron (/u/efce1a2a95ab?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

[iOS]JS与原生OC互相调用(总结) (/p/3728275e4284?utm_campaign=m...)

iOS开发免不了要与UIWebView打交道，然后就要涉及到JS与原生OC交互，今天总结一下JS与原生OC交互的两种方式。JS调用原生OC篇 方式一 第一种方式是用JS发起一个假的URL请求，然后利用UIWebView的

 德山_ (/u/46898049a33f?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/1c68555723c1?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)



木偶娃娃 (/p/1c68555723c1?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

希望有大佬提点一下上色

 公子语_ (/u/48a087342426?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/c5ef9abb8737?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

保险等待期发生重大疾病如何理赔？ (/p/c5ef9abb8737?utm_campaign=...

保险合同“等待期”条款中规定，从合同生效之日起90日内被保险人“因导致重大疾病的相关疾病就诊”，其不承担保险责任。保险公司能否据此免责?保险等待期发生重大疾病如何理赔？下面请看详细案例介绍。 【案

 Kevin_he (/u/1f6266fcdc4e?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/cdb6d7f6cbd7?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

王者荣耀摇摇卡之少女鲁班 (/p/cdb6d7f6cbd7?utm_campaign=maleski...

鲁：快把那个大耳朵正太给做出来！ 狄：没我的命令，谁敢做！ 作者：哈哈哈哈做不做一切取决于本小姐！！ 孙：谁在叫本小姐？ 众英雄直指作者。下一秒，作者卒。

 凉音丶 (/u/6c4e4ceff4f1?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/3dc23eee094b?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

笑晕在厕所0727！这是个公橙子，不信你看 (/p/3dc23eee094b?utm_ca...

[1]很唯美的 [2]摄影师 Daisuke Tashiro在日本拍到一只打哈欠的猴子，这手一看就知道保养得很好。 [3]怎么可以这么乖！！ [4]少年，你这样很容易被打 [5]一只非常奔放的鸭 [6]法鲨Michael Fassbender跟肥伦玩”鸡

 iamtinker (/u/fedddcebc98f?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

2017-11-23 (/p/eb377f392e9b?utm_campaign=maleskine&utm_conte...

珺珺的复盘仪式第七天 17.11.22 珺珺的决定性瞬间 今天是坚持复盘仪式第七天，坚持的感觉真好呀！很有成就感。加入三人复盘小组的感觉就更好咯，我们三姐妹互相温暖对方，并且帮助和鉴证咱们的成长。博士大

 珺珺927 (/u/96062d82f887?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)