

ios下JS与OC互相调用（六） -- WKWebView + WebViewJavascriptBridge



(https://github.com/marcuswestin/WebViewJavascriptBridge) + 关注

2016.08.29 13:47* 字数 1108 阅读 6761 评论 32 喜欢 26 赞赏 1 阅读 6761 评论 32 喜欢 26 赞赏 1

上一篇文章介绍了UIWebView 如何通过WebViewJavascriptBridge 来实现JS 与OC 的互相调用，这一篇来介绍一下WKWebView 又是如何通过WebViewJavascriptBridge 来实现JS 与OC 的互相调用的。WKWebView 下使用WebViewJavascriptBridge与UIWebView 大同小异。主要是示例化的类不一样，一些与webView 相关的API调用不一样罢了。



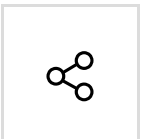
WKWebView 下使用 WebViewJavascriptBridge 来实现JS 与OC 的互相调用,也是通过拦截URL来实现的。

下面开始介绍WKWebView 如何通过WebViewJavascriptBridge 来实现JS 与OC 的互相调用。

关于下载WebViewJavascriptBridge (<https://link.jianshu.com?t=https://github.com/marcuswestin/WebViewJavascriptBridge>), 然后导入工程的部分就不再赘述了。

第一步，创建WKWebView。

这一步，唯一需要注意的地方，就是不用再设置 WKWebView 的 navigationDelegate ，下一步你就知道为什么了。



```
- (void)initWKWebView
{
    WKWebViewConfiguration *configuration = [[WKWebViewConfiguration alloc] init]
;
    configuration.userContentController = [WKUserContentController new];

    WKPreferences *preferences = [WKPreferences new];
    preferences.javaScriptCanOpenWindowsAutomatically = YES;
    preferences.minimumFontSize = 30.0;
    configuration.preferences = preferences;

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

    NSString *urlStr = [[NSBundle mainBundle] pathForResource:@"index.html" ofType:nil];
    NSString *localHtml = [NSString stringWithContentsOfFile:urlStr encoding:NSUTF8StringEncoding error:nil];
    NSURL *fileURL = [NSURL fileURLWithPath:urlStr];
    [self.webView loadHTMLString:localHtml baseURL:fileURL];

    self.webView.UIDelegate = self;
    [self.view addSubview:self.webView];
}
```

第二步，创建WebViewJavascriptBridge实例。

这里与上一篇文章有一些不同，WKWebView 使用的是 WKWebViewJavascriptBridge，而 UIWebView 使用的是 WebViewJavascriptBridge。

```
_webViewBridge = [WKWebViewJavascriptBridge bridgeForWebView:self.webView];
// 如果控制器里需要监听WKWebView 的`navigationDelegate`方法，就需要添加下面这行。
[_webViewBridge setWebViewDelegate:self];
```

上一步说了不用再设置 WKWebView 的 navigationDelegate，那是因为在 {-bridgeForWebView:} 内已经将 WKWebView 的 navigationDelegate 设置为 WKWebViewJavascriptBridge 的实例了。

```
+ (instancetype)bridgeForWebView:(WKWebView*)webView {
    WKWebViewJavascriptBridge* bridge = [[self alloc] init];
    [bridge _setupInstance:webView];
    [bridge reset];
    return bridge;
}

- (void) _setupInstance:(WKWebView*)webView {
    _webView = webView;
    _webView.navigationDelegate = self;
    _base = [[WebViewJavascriptBridgeBase alloc] init];
    _base.delegate = self;
}
```

第三步，注册 js 要调用的Native 功能

为了便于维护，我将所有js 要调用Native 功能放在了一个方法里添加，然后每个功能再单独处理。

示例代码如下：

```
#pragma mark - private method
- (void)registerNativeFunctions
{
    [self registScanFunction];

    [self registShareFunction];

    [self registLocationFunction];

    [self regitstBGColorFunction];

    [self registPayFunction];

    [self registShakeFunction];
}

// 注册的获取位置信息的Native 功能
- (void)registLocationFunction
{
    [_webViewBridge registerHandler:@"locationClick" handler:^(id data, WVJBResponseCallback responseCallback) {
        // 获取位置信息

        NSString *location = @"广东省深圳市南山区学府路XXXX号";
        // 将结果返回给js
        responseCallback(location);
    }];
}
```

关于 `- (void)registerHandler:(NSString *)handlerName handler:(WVJBHandler)handler` , 我们可以这样理解, 后面的block 参数是js 要调用的Native 实现, 前面的handlerName 是这个Native 实现的别名。然后js 里调用handlerName 这个别名, `WebViewJavascriptBridge` 最终会执行block 里的Native 实现。

第四步，在HTML添加关键的js

HMTL 里在调用Native 功能之前, 要先添加一个js 方法, 然后主动调用一次该方法。要添加的方法是：

```
function setupWebViewJavascriptBridge(callback) {
    if (window.WebViewJavascriptBridge) { return callback(WebViewJavascriptBridge); }
    if (window.WVJBCallbacks) { return window.WVJBCallbacks.push(callback); }
    window.WVJBCallbacks = [callback];
    var WVJBIframe = document.createElement('iframe');
    WVJBIframe.style.display = 'none';
    WVJBIframe.src = 'wjbscheme://__BRIDGE_LOADED__';
    document.documentElement.appendChild(WVJBIframe);
    setTimeout(function() { document.documentElement.removeChild(WVJBIframe) }, 0 )
}
```

如果你看过iOS下JS与OC互相调用（一）--UIWebView 拦截URL (<https://www.jianshu.com/p/7151987f012d>),你就会发现这个方法与 `loadURL` 很像。然后在js 中要主动调用一次上述的 `setupWebViewJavascriptBridge` 。

```
setupWebViewJavascriptBridge(function(bridge) {

    // 这里注册Native 要调用的js 功能。
    bridge.registerHandler('testJSFunction', function(data, responseCallback) {
        alert('JS方法被调用:'+data);
        responseCallback('js执行过了');
    })
    // 如果要有其他Native 调用的js 功能, 在这里按照上面的格式添加。
})
```

主动调用 `setupWebViewJavascriptBridge` 有两个目的：

- 1、执行一次 `wjbscheme://__BRIDGE_LOADED__` 请求。
- 2、注册Native 要调用的js 功能。

执行 wvjbscheme://__BRIDGE_LOADED__，然后在WKWebView 的 navigationDelegate 方法中拦截该URL，然后往HMTL中注入js。以下源码都摘自 WebViewJavascriptBridge：

```
- (void)webView:(WKWebView *)webView
decidePolicyForNavigationAction:(WKNavigationAction *)navigationAction
decisionHandler:(void (^)(WKNavigationActionPolicy))decisionHandler {
    if (webView != _webView) { return; }
    NSURL *url = navigationAction.request.URL;
    __strong typeof(_webViewDelegate) strongDelegate = _webViewDelegate;

    if ([_base isCorrectProcotocolScheme:url]) {
        // 在这里拦截wvjbscheme://__BRIDGE_LOADED__
        if ([_base isBridgeLoadedURL:url]) {
            // 这里会注入js
            [_base injectJavascriptFile];
        } else if ([_base isQueueMessageURL:url]) {
            [self WKFlushMessageQueue];
        } else {
            [_base logUnkownMessage:url];
        }
        decisionHandler(WKNavigationActionPolicyCancel);
    }

    if (strongDelegate && [strongDelegate respondsToSelector:@selector(webView:decidePolicyForNavigationAction:decisionHandler:)]) {
        [_webViewDelegate webView:webView decidePolicyForNavigationAction:navigationAction decisionHandler:decisionHandler];
    } else {
        decisionHandler(WKNavigationActionPolicyAllow);
    }
}

- (void)injectJavascriptFile {
    //读取js 内容
    NSString *js = WebViewJavascriptBridge_js();
    // 执行Native 的API, 实现将js 注入 到HMTL中。
    [self _evaluateJavascript:js];
    if (self.startupMessageQueue) {
        NSArray* queue = self.startupMessageQueue;
        self.startupMessageQueue = nil;
        for (id queuedMessage in queue) {
            [self _dispatchMessage:queuedMessage];
        }
    }
}
```

WKWebView 执行js 的API 与 UIWebView 有些不同，WKWebView 用的是 {-evaluateJavaScript: completionHandler:}，这个API 不会立刻返回执行结果，js 的执行结果会在block 中返回。

第五步，在js 中调用 Native 功能。

讲完过程，终于到了 js 调用Native 的用法了。其实非常的简单，例如我想要利用Native 获取定位信息，那么在HTML中添加一个按钮，onclick事件是locationClick()，按照如下实现即可。

```
function locationClick() {
    WebViewJavascriptBridge.callHandler('locationClick',null,function(response) {
        alert(response);
        document.getElementById("returnValue").value = response;
    });
}
```

Native 执行完代码，将获取到的定位信息，通过callHandler 的第三方参数，回调返回到js 中。
response 可以是单个值，也可以是数组、键值对等。
当然如果我们调用Native 的时候，没有参数或者不需要Native 返回信息到js 中。我们还可以这样写：

```
// 没有参数，有回调可以这样写
function locationClick() {
    WebViewJavascriptBridge.callHandler('locationClick',function(response) {
        alert(response);
        document.getElementById("returnValue").value = response;
    });
}

// 没有参数，又不需要回调可以这样写
function shake() {
    WebViewJavascriptBridge.callHandler('shakeClick');
}
```

至此，JS 通过 WebViewJavascriptBridge 调用Native 的功能就完成了。

第六步，Native 调用 JS 功能。

Native 调用js 功能与 js 调用Native 的原理和流程一样。

- 1、现在js 中注册，Native 要调用的功能。
- 2、Native 调用注册时，该功能的别名，就可以完成调用。

在js 中注册 Native 要调用的功能，同样需要为该功能设置一个别名HandlerName。

其实这个步骤在前面介绍过，代码如下：

```
setupWebViewJavascriptBridge(function(bridge) {

    // 这里注册Native 要调用的js 功能。
    bridge.registerHandler('testJSFunction', function(data, responseCallback) {
        alert('JS方法被调用:'+data);
        responseCallback('js执行过了');
    })
    // 如果要有其他Native 调用的js 功能，在这里按照上面的格式添加。
})
```

上述代码，是在JS 中注册了一个别名叫 testJSFunction 的js功能，第二个参数是一个function。function里的data ，就是Native 调用该功能时传过来的参数，responseCallback是执行完js 代码后，通过responseCallback将必要的信息返回到Native中。

Native 调用js 里注册的功能，示例代码：

```
- (void)rightClick
{
    //      // 如果不需要参数，不需要回调，使用这个
    //      [_webViewBridge callHandler:@"testJSFunction"];
    //      // 如果需要参数，不需要回调，使用这个
    //      [_webViewBridge callHandler:@"testJSFunction" data:@"一个字符串"];
    // 如果既需要参数，又需要回调，使用这个
    [_webViewBridge callHandler:@"testJSFunction" data:@"一个字符串" responseCallba
ck:^(id responseData) {
        NSLog(@"调用完JS后的回调：%@", responseData);
    }];
}
```

WKWebView 通过 WebViewJavascriptBridge 实现js 与Native 的交互，到这里就已经完成了。

这是工程示例中的效果图：

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

Have Fun!



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)

26



更多分享

(<http://cwb.assets.jianshu.io/notes/images/5510032>)



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网络 (/c/f2a8efbe36c0?utm_source=desktop&utm_medium=notes-included-collection)
-  ios专题 (/c/9f09a5caa1f6?utm_source=desktop&utm_medium=notes-included-collection)
-  iOS学霸笔记 (/c/c7c627650a35?utm_source=desktop&utm_medium=notes-included-collection)
-  H5和Nati... (/c/2c939545ddd0?utm_source=desktop&utm_medium=notes-included-collection)
-  JS与OC交互 (/c/4633b70f44e4?utm_source=desktop&utm_medium=notes-included-collection)
-  iOS备忘录 (/c/536499c136f6?utm_source=desktop&utm_medium=notes-included-collection)
-  ios学习目录 (/c/e5a12890ff4e?utm_source=desktop&utm_medium=notes-included-collection)


展开更多 ∨

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)

(/p/ac45d99cf912?



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 iOS开发中，用来显示一个html页、H5页，经常会用的一个控件是WebView。说到WebView，你知道多少呢？是简单的展示，还是要和OC交互实现比较复杂的功能呢？本文将为您介绍iOS中的WebView，并且

 Dark_Angel (/u/9f088cde1295?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)


(/p/6f34903be630?

utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation) [iOS 开发] WebViewJavascriptBridge 从原理到实战 (/p/6f34903be630...

前言：iOS 开发中，h5 和原生实现通信有多种方式，JSBridge 就是最常用的一种，各 JSBridge 类库的实现原理大同小异，这篇文章主要是针对当前使用最为广泛的 WebViewJavascriptBridge（v6.0.2），从功能

 祥龙Shannon (/u/4ef5e287fc91?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/52668d5b2e68?



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 在前面的文章中，我们介绍了UIWebView、WKWebView一些使用，与JS的交互和一些坑，相信看过的小伙伴们，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最后一篇，主要为小伙伴们分

 Dark_Angel (/u/9f088cde1295?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

厚积薄发让我三次逆袭人生 (/p/f8f2ad1c4443?utm_campaign=maleskin...

临近毕业，你是不是在为没有门路，没有关系背景而不知道怎么找工作发愁？ 已经工作的你，发现工作清闲的很，却感觉人生没目标，闲的蛋疼？ 虽然身为单位边缘人，却有着一颗想打入核心团队的心？ 其实上上说

 鱼清风 (/u/96f16d14f719?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

第一次银行卡被吞 (/p/2f1e41ed7c44?utm_campaign=maleskine&utm_c...

今天醒来的第一意识是，我该起床去学校报到，还是告诉李主任我不能那儿上班了呢？ 其实答案早就明了了，只是，我还多此一举地问了问自己。拿起手机编辑了两条歉意满满的短信，点击发送，然后放下手机继

 春天里的一只熊 (/u/5eb8f38ddf11?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/bc226db0c02b?



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

一次21天的行动带给我的改变 (/p/bc226db0c02b?utm_campaign=male...

月初参加了一个21天的PPT训练营，其实自己本身PPT水平还算说得过去，应付目前的生活绰绰有余。然而每当看到大神们的作品，我内心的求知欲和好奇心便被激起，然而每次都是"三天打鱼两天晒网"，如此反

 DazeWander (/u/cd98ec3464db?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/cdeb741b936e?



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

五篇文章 (/p/cdeb741b936e?utm_campaign=maleskine&utm_content...

 一只自私的小婊哥 (/u/823fdd1d51c4?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

不知道拥挤在回忆里会不会是美好的？ (/p/bf8fa8287058?utm_campaig...

以为大学就不用抢饭了，但是也许是我想多啦，下课晚呀还是那么挤，最喜欢的就是遇着那种中午最后一节课急着赶校车的老师，因为他们总是会提前十几分钟下课，我们可是开心死啦。因为如果要是中午准点下课



东L鱼Y儿E (/u/99ee034460d5?



东L鱼Y儿E (/u/99ee034460d5?

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