

ios下JS与OC互相调用（二） -- WKWebView 拦截URL



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

2016.08.06 13:47* 字数 938 阅读 8561 评论 12 喜欢 33 阅读 8561 评论 12 喜欢 33

(/u/a8cf6d63e889)

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交互。关于WKWebView与UIWebView的对比，大家请自动百度或者google。

| 打开百度网页前 | 打开百度网页后

----- | ----- | -----

UIWebView | 内存47M | 内存75.6M，最高峰83M

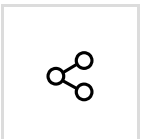
WKWebView | 内存47M | 内存51M

WKWebView 拦截URL

WKWebView 与 UIWebView 拦截URL 的处理方式基本一样。除了代理方法和WKWebView的使用不太一样，关于WKWebView更详尽的讲解和用法，还是自行搜索学习，本文重点还是讲解如何实现JS 与OC 互相调用。

提醒：WKWebView 是iOS 8 推出的WebKit.framework中的控件，只有app 不需要兼容iOS 7及以下的时候才可以使用。

先看动态效果图：



1.创建WKWebView，加载本地HTML。

WKWebView的创建有几点不同：

- 1.初始化多了个 configuration 参数，当然这个参数我们也可以不传，直接使用默认的设置就好。
- 2.WKWebView的代理有两个 navigationDelegate 和 UIDelegate 。我们要拦截URL，就要通过 navigationDelegate 的一个代理方法来实现。如果在HTML中要使用alert等弹窗，就必须得实现 UIDelegate 的相应代理方法。
- 3.在iOS 9之前，WKWebView加载本地HTML会有一些问题。（不能加载本地HTML，或者部分CSS/本地图片加载不了等）

我这里创建WKWebView的示例代码是这样的：

```
        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];
        NSURL *fileURL = [NSURL fileURLWithPath:urlStr];
        [self.webView loadFileURL:fileURL allowingReadAccessToURL:fileURL];

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

因为加载的本地HTML内容，跟上一篇UIWebView中介绍的HTML内容一样，所以关于HTML中的内容就不再讲解了。

2.拦截URL

使用WKNavigationDelegate中的代理方法，拦截自定义的URL来实现JS调用OC方法。

```
#pragma mark - WKNavigationDelegate
- (void)webView:(WKWebView *)webView decidePolicyForNavigationAction:(WKNavigationAction *)navigationAction decisionHandler:(void (^)(WKNavigationActionPolicy))decisionHandler
{
    NSURL *URL = navigationAction.request.URL;
    NSString *scheme = [URL scheme];
    if ([scheme isEqualToString:@"haleyaction"]) {
        [self handleCustomAction:URL];
        decisionHandler(WKNavigationActionPolicyCancel);
        return;
    }
    decisionHandler(WKNavigationActionPolicyAllow);
}
```

需要注意的是：

- 1.如果实现了这个代理方法，就必须得调用 decisionHandler 这个block，否则会导致app 崩溃。block参数是个枚举类型， WKNavigationActionPolicyCancel 代表取消加载，相当于UIWebView的代理方法return NO的情况； WKNavigationActionPolicyAllow 代表允许加载，相当于UIWebView的代理方法中 return YES的情况。

2.其他的关于为什么要统一设置scheme，在上一篇中讲过。

关于如何区分执行不同的OC 方法， 也与UIWebView的处理方式一样,通过URL 的host 来区分执行不同的方法：

```
#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];
    }
}
```

3.OC 调用 JS 方法

JS 调用OC 方法后，有的操作可能需要将结果返回给JS。这时候就是OC 调用JS 方法的场景。

WKWebView 提供了一个新的方法 evaluateJavaScript:completionHandler:，实现OC 调用JS 等场景。

```
- (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 evaluateJavaScript:jsStr completionHandler:^(id _Nullable result, NSError * _Nullable error) {
        NSLog(@"%@----%@",result, error);
    }];
}
```

evaluateJavaScript:completionHandler: 没有返回值， JS 执行成功还是失败会在 completionHandler 中返回。所以使用这个API 就可以避免执行耗时的JS，或者alert 导致界面卡住的问题。

4.WKWebView中使用弹窗

在上面提到，如果在WKWebView中使用alert、confirm 等弹窗，就得实现WKWebView 的 WKUIDelegate 中相应的代理方法。

例如，我在JS中要显示alert 弹窗，就必须实现如下代理方法，否则alert 并不会弹出。

```
#pragma mark - WKUIDelegate
- (void)webView:(WKWebView *)webView runJavaScriptAlertPanelWithMessage:(NSString *)message initiatedByFrame:(WKFrameInfo *)frame completionHandler:(void (^)(void))completionHandler
{
    UIAlertController *alert = [UIAlertController alertControllerWithTitle:@"提醒" message:message preferredStyle:UIAlertControllerStyleAlert];
    [alert addAction:[UIAlertAction actionWithTitle:@"知道了" style:UIAlertActionStyleCancel handler:^(UIAlertAction * _Nonnull action) {
        completionHandler();
    }]];

    [self presentViewController:alert animated:YES completion:nil];
}
```

其中 completionHandler 这个block 一定得调用，至于在哪里调用，倒是无所谓，我们也可以写在方法实现的第一行，或者最后一行。

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)





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) | 33



更多分享

(<http://cwb.assets.jianshu.io/notes/images/5125549>)




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文本 (/c/c7c4a8437d02?utm_source=desktop&utm_medium=notes-included-collection)
-  ios专题 (/c/9f09a5caa1f6?utm_source=desktop&utm_medium=notes-included-collection)
-  使用WKWeb... (/c/0257907a7b13?utm_source=desktop&utm_medium=notes-included-collection)

展开更多

iOS下JS与OC互相调用（一）--UIWebView 拦截URL (/p/7151987f012d?...

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，

 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)

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/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)

(/p/21f4b9020d18?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)
(转) WKWebview 处理 (/p/21f4b9020d18?utm_campaign=maleskine...

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

 CoderLF (/u/fce901c26622?
utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/157d14cb2095?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)
蛋炒饭——我的成人礼 (/p/157d14cb2095?utm_campaign=maleskine&...

对我而言，成年的好处真是太多了。好处之一，比如说，我可以像现在这样什么也不用考虑就开始写东西，不用考虑问题、字数，出卷老师心理，尽管写出的东西够幼稚，甚至还带有应试的烙印，但还是很开心，能

 zyppppp (/u/bf6e43479a60?
utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/15ffcbdc9c10?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)
行走的丧尸 (/p/15ffcbdc9c10?utm_campaign=maleskine&utm_content...

最近朋友圈被《釜山行》这部电影刷屏了，身边的朋友也都强烈推荐我看一看，一来恐怖电影很和我的口味，二来也不想错过这么大热的电影，晚上便看了看。但是看完心里着实不舒服，电影不恐怖，却极其残

 爱夏天 (/u/b8eed70207ea?
utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

JavaScript基础语法学习笔记 (/p/794ce90f090f?utm_campaign=males...

1. JavaScript用法 html中的javascript脚本必须放在 之间. 脚本可以被放置在html页面的 和 部分中. 浏览器会解释并执行位于 和 之间的 JavaScript 代码 .以前需要在 标签中使用type="text/javascript"声...

 Dev_hell03W (/u/ea059360a6f6?
utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

你和成功距离有多远？ (/p/cb77879cba77?utm_campaign=maleskine&u...

在创业的朋友有时候很迷茫，为何我这么努力还没有取得成功？我和成功还有多远距离？也有很多创业者因为不知道自己什么时候能取得成功，也对未来感到困惑而放弃了继续努力。今天我们一起来看看你和成功的

 巅峰之旅 (/u/b9a523323f1f?
utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)