



作者 xshenpan (/users/5ffc0eacf99) 2016.05.18 20:26

写了4689字，被54人关注，获得了96个喜欢  
(/users/5ffc0eacf99)

+ 添加关注 (/sign\_in)

# OC与JS数据交互

字数1291 阅读1115 评论3 喜欢37

简  
(/)



登录 (/sign\_in) 注册 (/sign\_up)



(/collections)



(/apps)

- iOS7之后有了JavaScriptCore框架使得OC与JS的交互变得简单了起来.
- 在学习webView的时候，发现OC与JS做交互时用的是截取webView的响应来获得想要的数(不得不佩服，办法很高端)。感觉这种办法太蛋疼了，于是抱着试一试的心态到官方文档搜索JavaScript关键字，于是出来了下面一段

## Calling Objective-C Methods

QQ20160518-0.png

- 看到了这句话很是兴奋啊！于是又往下继续找

### A Sample Objective-C Class

Let's look at a sample class. In this case, we will create an Objective-C address book class and expose it to JavaScript. Let's start with the class definition:

```
1 @interface BasicAddressBook: NSObject {  
2 }  
3 + (BasicAddressBook *)addressBook;  
4 - (NSString *)nameAtIndex:(int) index;  
5 @end
```

Now we'll write the code to publish a BasicAddressBook instance to JavaScript:

```
1 BasicAddressBook *littleBlackBook = [BasicAddressBook addressBook];  
2  
3 id win = [webView windowScriptObject];  
4 [win setValue:littleBlackBook forKey:@"AddressBook"];
```

QQ20160518-1.png

- 竟然还有例子耶，来试试吧！
  - 结果可想而知，并没有什么卵用webView根本没有 windowScriptObject 方法
  - 回头一看还有一句 any WebKit-based OS X application 原来是OS X才能用
- 然后继续上网找资料啊

- 找到了这样一个博客

<http://blog.csdn.net/lwjok2007/article/details/47058101/>

(<http://blog.csdn.net/lwjok2007/article/details/47058101/>)

讲了在OC执行JS

<http://blog.csdn.net/lwjok2007/article/details/47058795>

(<http://blog.csdn.net/lwjok2007/article/details/47058795>)

讲了在JS调用OC方法

里面写了怎样使用JavaScriptCore框架让OC和JS的交互，我就不搬运了。里面的关于导入框架，介绍JavaScriptCore的使用等等一些东西写的都比较清楚了。

然后我把里面的东西拿出来验证了一下，也把JSExport用法写的更详细了一点，block的方式没有使用。上面的文章里面没有讲到 JSExportAs宏，所以JS函数名称比较恶心

- JS与OC的数据类型对应如下<JSValue.h>有描述

```
<pre>
@textblock
Objective-C type | JavaScript type
-----|-----
nil              | undefined
NSNull           | null
NSString         | string
NSNumber         | number, boolean
NSDictionary     | Object object
NSArray          | Array object
NSDate           | Date object
NSBlock (1)     | Function object (1)
id (2)           | Wrapper object (2)
Class (3)       | Constructor object (3)
@/textblock
</pre>
```

QQ20160518-2.png

- 要说的都在注释里了，OC代码如下(HTML在最下面):

```

#import "ViewController.h"
#import <JavaScriptCore/JavaScriptCore.h>

//自己定义一个协议遵守JSExport协议，然后想要导出这些方法的类再遵守这个协议就可以了
@protocol ObjcExportMethod <JSExport>

//JSExportAs宏将 "- showNumber1:andNumber2:"方法名 转换为showNumber(param1, param2)
//如果不使用JSExportAs转换，那么在js中调用的函数名为showNumber1AndNumber2(param1, param2);
//注意：JSExportAs宏只适合转换有参的方法，不能转换无参的方法
JSExportAs(showNumber,
- (void)showNumber1:(NSNumber *)n1 andNumber2:(NSNumber *)n2
);

JSExportAs(returnSum,
- (NSNumber *)returnSumWithNum1:(NSNumber *)n1 andNum2:(NSNumber *)n2
);

JSExportAs(dictTest,
- (void)dictTest:(NSDictionary *)dict
);

JSExportAs(arrayTest,
- (void)arrayTest:(NSArray *)array
);

- (void)show;
- (NSString *)returnString;
- (NSDictionary *)returnDict;
- (NSArray *)returnArray;
@end

/**
 * UIViewController
 */
@interface ViewController () <UIWebViewDelegate, ObjcExportMethod>
@property (weak, nonatomic) IBOutlet UIWebView *webView;
@end

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];

    //加载本地测试网页
    [self.webView loadRequest:[NSURLRequest requestWithURL:[NSBundle mainBundle] UR

```

```
}

#pragma mark - <ObjcExportMethod>

/**
 * 无参导出方法
 */
- (void)show
{
    NSLog(@"%s", __func__);
}

/**
 * 返回一个OC字符串到JavaScript
 */
- (NSString *)returnString
{
    return @"从OC返回成功";
}

/**
 * 返回一个OC数组到JavaScript
 */
- (NSArray *)returnArray
{
    return @[@(1), @"hello"];
}

/**
 * 返回一个OC字典到JavaScript
 */
- (NSDictionary *)returnDict
{
    return @{
        @"name" : @"jack",
        @"date" : @[@(1), @"hello"]
    };
}

/**
 * 从JavaScript接收两个NSNumber参数，计算他们的和并返回计算结果到JS
 */
- (NSNumber *)returnSumWithNum1:(NSNumber *)n1 andNum2:(NSNumber *)n2
{
    return @(n1.intValue + n2.intValue);
}

/**
 * 从JavaScript接收两个NSNumber类型的值，打印出来
 */
- (void)showNumber1:(NSNumber *)n1 andNumber2:(NSNumber *)n2
{
    NSLog(@"number1 = %@, number2 = %@", n1, n2);
}
```

```

}
/**
 * 从JavaScript接收一个字典
 */
- (void)dictTest:(NSDictionary *)dict
{
    NSLog(@"%@", dict);
}
/**
 * 从JavaScript接收一个数组
 */
- (void)arrayTest:(NSArray *)array
{
    NSLog(@"%@", array);
}

#pragma mark - <UIWebViewDelegate>

/**
 * 完成加载
 */
- (void)webViewDidFinishLoad:(UIWebView *)webView
{
    //得到当前页面的JSContext, 如果不在这里获得当前网页的context可能有问题
    JSContext *js = [webView valueForKeyPath:@"documentView.webView.mainFrame.javaSc
    //设置 js 代码中的method对象为控制器
    js[@"method"] = self;

    //在OC中执行js并取得js返回的数据
    //方式1 使用JSContext的对象方法执行JS脚本
    NSString *jsScript = @"document.title"; //获取网页标题
    //evaluateScript方法返回的是JSValue类型的对象, 需要将类型转换为OC对象
    NSString *title = [js evaluateScript:jsScript].toString;
    self.navigationItem.title = title;
    NSLog(@"evaluateScript ----> %@", title);

    //方式2 使用webView的对象方法执行js, 该方法返回的数据类型始终是NSString, 不管JS实际传递过来的
    //如JS传递过来的是10 那么该方法的返回值任然是字符串
    title = [webView stringByEvaluatingJavaScriptFromString:jsScript];
    NSLog(@"stringByEvaluating ----> %@", title);
}

@end

```

- 测试用HTML代码如下:

```
<html>
  <head>
    <meta charset="UTF-8"/>
    <title>测试网页</title>
    <script>
      //遍历数组
      function ArrayFun(array)
      {
        for(var i = 0; i < array.length; i++)
        {
          alert(array[i]);
        }
      }
      // 遍历字典
      function DictionaryFun(dict)
      {
        for (var key in dict)
        {
          alert(key + " : " + dict[key]);
        }
      }
    </script>
  </head>

  <body>
    <h5 style="color:blue" onclick="method.show();">调用OC无参无返回值方法</h5>
    <h5 style="color:blue" onclick="alert(method.returnString());">从OC返回字符串</h5>
    <h5 style="color:blue" onclick="method.showNumber(1,2);">传递2个整数到OC</h5>
    <h5 style="color:blue" onclick="alert(method.returnSum(10,20));">传递2个整数到OC</h5>
    <h5 style="color:blue" onclick="method.arrayTest([1,2,'hello oc', {'name':'jack'}]);">传递数组到OC</h5>
    <h5 style="color:blue" onclick="method.dictTest({'name':'jack', 'data':[1, 'data']});">传递字典到OC</h5>
    <h5 style="color:blue" onclick="DictionaryFun(method.returnDict());">从OC返回字典</h5>
    <h5 style="color:blue" onclick="ArrayFun(method.returnArray());">从OC返回数组</h5>
  </body>
</html>
```

➕ 推荐拓展阅读 (/sign\_in)

© 著作权归作者所有

如果觉得我的文章对您有用，请随意打赏。您的支持将鼓励我继续创作！

[¥ 打赏支持](#)[♡ 喜欢 | 37](#)[👤 分享到微博](#) [👤 分享到微信](#)  
更多分享 ▼3条评论 ( [按时间正序](#) · [按时间倒序](#) · [按喜欢排序](#) )[✎ 添加新评论 \(/sign\\_in\)](#)[一点墨香 \(/users/99e37a598093\)](/users/99e37a598093)[2楼 2016.05.28 07:39 \(/p/9da1c0ceb5fa/comments/2458572#comment-2458572\)](/users/99e37a598093)

很好唉

[♡ 喜欢\(0\)](#)[回复](#)[择择lee \(/users/81f69e7cd638\)](/users/81f69e7cd638)[3楼 2016.08.03 11:45 \(/p/9da1c0ceb5fa/comments/3426670#comment-3426670\)](/users/81f69e7cd638)

很棒

[♡ 喜欢\(0\)](#)[回复](#)

xshenpan (/users/5ffca0eacf99): @择择lee (/users/81f69e7cd638) 谢谢  
2016.08.03 12:18 (/p/9da1c0ceb5fa/comments/3427273#comment-3427273)

[回复](#)[✎ 添加新回复](#)[登录后发表评论 \(/sign\\_in\)](#)

被以下专题收入，发现更多相似内容：

[首页投稿 \(/collection/bDHhpK\)](/collection/bDHhpK)

[玩转简书的第一步，从这个专题开始。想上首页热门榜么？好内容想被更多人看到](/collection/bDHhpK)  
[2楼 来投稿吧！如果被拒也不要灰心哦~入选文章会进一个队...](/collection/bDHhpK)

[+ | 添加关注 \(/sign\\_in\)](#)

104267篇文章 (/collection/bDHhpK) · 131553人关注



**iOS Developer (/collection/3233d1a249ca)**

分享 iOS 开发的知识，解决大家遇到的问题，讨论iOS开发的前沿，欢迎大家投稿

+ 添加关注 (/sign\_in)

(/collection/3233d1a249ca)

13514篇文章 (/collection/3233d1a249ca) · 26550人关注



**iOS 开发 (/collection/2ffaa203eb6a)**

7975篇文章 (/collection/2ffaa203eb6a) · 6860人关注

+ 添加关注 (/sign\_in)

(/collection/2ffaa203eb6a)

A



(/sign\_in)