

WebViewJavascriptBridge使用 「JS交互」



作者 LuisX (/u/6387bbe37e0a) ✓ 已关注

2017.03.10 16:04 字数 325 阅读 13 评论 0 喜欢 0

(/u/6387bbe37e0a)



WebViewJavascriptBridge

官方地址:<https://github.com/marcuswestin/WebViewJavascriptBridge>
(<https://github.com/marcuswestin/WebViewJavascriptBridge>)

用于在WKWebViews，UIWebViews和WebViews中的Obj-C和JavaScript之间发送消息的iOS / OSX桥接器。

一、安装

(1)使用CocoaPods安装

```
`pod 'WebViewJavascriptBridge', '~> 5.0'`
```

(2)手动安装

拖拽WebViewJavascriptBridge文件到工程之中
在出现的对话框中，取消选中“将项目复制到目标组的文件夹”，然后选择“为任何文件夹创建组”

二、使用

(1)导入头文件并声明属性

```
//导入头文件
#import "WebViewJavascriptBridge.h"
//声明属性
@property WebViewJavascriptBridge* bridge;
```

(2)为(WKWebView, UIWebView, WebView)实例化

WebViewJavascriptBridge

```
self.bridge = [WebViewJavascriptBridge bridgeForWebView:webView];
```

(3)在Objective-C中注册处理,并调用JS处理

```
/*
 *Objective-C
 */

//注册处理
[self.bridge registerHandler:@"ObjC Echo" handler:^(id data, WVJBResponseCallback
responseCallback) {
    NSLog(@"ObjC Echo called with: %@", data);
    responseCallback(data);
}];
//调用处理
[self.bridge callHandler:@"JS Echo" data:nil responseCallback:^(id responseData)
{
    NSLog(@"ObjC received response: %@", responseData);
}];
```

(4)复制并粘贴setupWebViewJavascriptBridge方法到JS中

```
/*
 *JS
 */
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 = 'https://__bridge_loaded__';
    document.documentElement.appendChild(WVJBIframe);
    setTimeout(function() { document.documentElement.removeChild(WVJBIframe) }, 0
)
}
```

(5)最后,调用setupWebViewJavascriptBridge方法,使用bridge注册处理,并调用Objective-C处理

```
/*
 *JS
 */
setupWebViewJavascriptBridge(function(bridge) {
    /* Initialize your app here */
    //注册处理
    bridge.registerHandler('JS Echo', function(data, responseCallback) {
        console.log("JS Echo called with:", data)
        responseCallback(data)
    })
    //调用处理
    bridge.callHandler('ObjC Echo', {'key':'value'}, function responseCallback(re
sponseData) {
        console.log("JS received response:", responseData)
    })
})
```

三、自动引用计数(ARC)

这个库依赖于ARC，所以如果你在你的项目中使用ARC，一切正常。但是如果你的项目没有ARC支持，一定要做下一步：

- 1) 在Xcode项目中打开项目设置 - > 'Build Phases'
 - 2) 展开 'Compile Sources'找到所有属于这个库的.m文件， 请注意此列表中每个源文件前面的 'Compiler Flags'
 - 3) 为每个文件添加 '-fobjc-arc'
- 现在所有的WVJB文件将被ARC支持编译

四、API参考

Objective-C API

```
//为web视图创建一个javascript bridge
[WebViewJavascriptBridge bridgeForWebView:(WKWebView/UIWebView/WebView*)webview

//注册一个名为"handlerName"的处理之后,javascript可以使用WebViewJavascriptBridge.callHandler调用这个处理
[bridge registerHandler:(NSString*)handlerName handler:(WVJBHandler)handler]

//调用名为handlerName的javascript处理
[bridge callHandler:(NSString*)handlerName data:(id)data]
//responseCallback的block,响应javascript的处理
[bridge callHandler:(NSString*)handlerName data:(id)data responseCallback:(WVJBResponseCallback)callback]

//（可选）如果需要响应Web视图的生命周期事件，请设置WKNavigationDelegate / UIWebViewDelegate
[bridge setWebViewDelegate:(id)webViewDelegate]

//不安全。通过禁用setTimeout安全检查来加快桥消息传递。只有在您不调用任何JavaScript弹出框功能（警报，确认和提示）时，才能安全地禁用此安全检查。如果您从桥接的JavaScript代码调用任何这些函数，该应用程序将挂起
[bridge disableJavascriptAlertBoxSafetyTimeout]
```

Javascript API

```
//注册一个名为"handlerName"的处理之后,Objective-C可以使用callHandler: data:或callHandler: data: responseCallback:调用这个处理
bridge.registerHandler("handlerName", function(responseData) { ... })

//调用名为handlerName的Objective-C处理
bridge.callHandler("handlerName", data)
//responseCallback的函数,响应Objective-C的处理
bridge.callHandler("handlerName", data, function responseCallback(responseData) { ... })

//调用bridge.disableJavascriptAlertBoxSafetyTimeout()方法和Objective-C中[bridge disableJavascriptAlertBoxSafetyTimeout]具有与调用相同的效果
bridge.disableJavascriptAlertBoxSafetyTimeout()
```

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！

赞赏支持

喜欢 | 0

更多分享

(http://cwb.assets.jianshu.io/notes/images/10002616



写下你的评论...

评论

智慧如你，不想发表一点想法咩~

被以下专题收入，发现更多相似内容

