# WebViewJavascriptBridge移动端IOS/Android与JS前端交互

## 一．交互

1. 苹果端：我们可以在OC中调用javascript方法，但是反过来不能在javascript中调用OC方法。所以WebViewJavascriptBridge的实现过程就是在OC环境和javascript环境各自保存一个相互调用的信息。每一个调用之间都有id和callbackid来找到两个环境对应的处理。

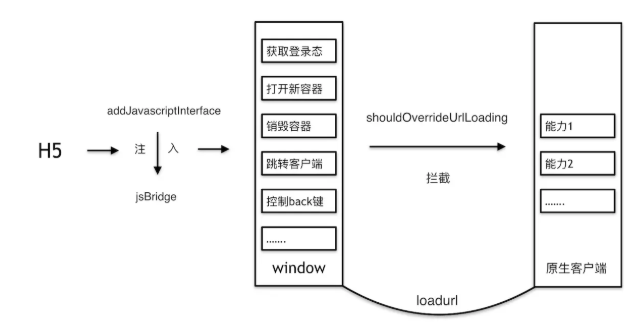
OC发送消息给WEB

WEB发消息给OC

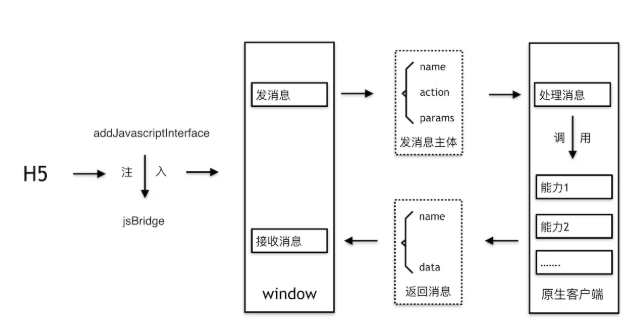
H5页面被webview承载，需要实现 js 和 native 的通信



通信流程



优化: 当js触发jsBridge对象触发某一个方法，其实都会走到客户端的拦截器里面然后native响应，这个过程就是在向客户端发消息。而客户端回调前端callback函数并且将数据返回其实就是在向H5页面发消息。那基于这种消息发送机制，可以将通信模型进一步优化



通信原理

刚才有提到jsBridge注入，客户端拦截器

1相关网址：<https://github.com/marcuswestin/WebViewJavascriptBridge>

1. Import the header file and declare an ivar property:

【导入头文件并声明一个ivar属性】

1. Instantiate WebViewJavascriptBridge with a WKWebView, UIWebView (iOS) or WebView (OSX):

【实例化webviewjavascriptbridge】

1. Register a handler in ObjC, and call a JS handler:

【在objc中注册一个处理程序，并调用JS处理程序：】

1. Copy and paste setupWebViewJavascriptBridge into your JS:

【复制并粘贴SetupWebViewJavaScriptBridge到您的JS中】

1. Finally, call setupWebViewJavascriptBridge and then use the bridge to register handlers and call ObjC handlers:

【最后，调用SetupWebViewJavaScriptBridge，然后使用桥注册处理程序并调用objc处理程序：】

## 二．匿名函数：

（立刻执行函数），因为是匿名的，所以不能被调用；因为立刻执行，所以在执行完之后就会被销毁。

1(function() {

alert(1);

})();

2(function() {

alert(2);

}());