H5和Native交互---说明文档 (H5端)

*注意：H5仅可以调用本手册提供的Native方法，不支持的功能后期扩展。*

*注意：H5调用Native方法时，约定必须传入所有参数，若没有值，传入空值。*

*(必须传入的参数参考Tower【美着呢wiki】：H5&Native交互规范文档)*

*注意：native所调用的js函数的命名方式( bri\_openModal()除外 )：调用的Native方法名 + ”\_” + 调用函数的参数名；*

如：调用bri\_showAler()时，参数confirmMethod用于，当alert框点击【确定】后Native会调用的js函数；

综上所述：参数confirmMethod传入的值就是 ’bri\_showAlert\_confirmMethod’，等用户点击【确定】后，native就会调用js的声明函数：bri\_showAlert\_confirmMethod() ；

*处理native返回值的函数的命名方式：调用的Native方法名+”\_”+ handle\_result；*

如：调用bri\_capture()时，Native会返回status和image值，处理返回值的函数名为：

bri\_capture\_handle\_result；

*注意：当页面存在多次(n>1)调用同一个native方法时，调用时在末尾添加一个参数，用于标识，方便对返回值做不同处理；*

如：jsBridgeNative.bri\_showActionSheet('BookList ',['JavaScript ','Html5 '],'cancel ','markBook');jsBridgeNative.bri\_showActionSheet('北京区域',['海淀区','朝阳区',’取消’ ,'markAddress'); 可以通过判断'markBook', 'markAddress'在bri\_showActionSheet\_handle\_result()做不同处理；

*注意：当页面调用bri\_openModal()方法时，在对返回值做处理时，通过modalName和mark的值，进行不同处理；（mark标识多次调用，modalName标识模态窗口名）*

如：通过判断’mark1’和 'native\_city\_choose'在bri\_openModal\_handle\_result()做不同处理

1. 什么情况下使用？

当APP内嵌H5页面，在H5页面内需要调用native原生的功能时。

1. 如何使用--- H5调用Native方法？

首先，页面引入js文件：<script src=”jsBridgeNative.js”></script>；

然后， 页面引入js文件：<script src=” jsBridgeNativeHandle.js”></script>；

jsBridgeNativeHandle.js主要用于声明函数供native调用和处理native返回值；

最后，根据页面需要功能调用Native方法；

本次供H5调用Native的功能集合，如下：

* 1. Alert()方法，解决现在H5页面的alert和Native的alert体验不一致的情况

举例——弹出native的Alert框：

* + 1. 调用方式： jsBridgeNative.bri\_showAlert(2,'content','title','',true,'bri\_showAlert\_confirmMethod',’mark1’)
    2. 声明函数：

(在jsBridgeNativeHandle.js声明bri\_showAlert\_confirmMethod函数，在Alert框点击【确定】时Native会调用此函数)

* + 1. 处理函数：

(在jsBridgeNativeHandle.js声明bri\_showAlert\_handle\_result函数处理native返回值)

* 1. getConfig ()方法，打通H5获取Natvie相应数据的通道

举例——获取用户ID：

* + 1. 调用方式：

jsBridgeNative.bri\_getConfig('userID',true)

* + 1. 处理函数：

(在jsBridgeNativeHandle.js声明bri\_getConfig\_handle\_result函数处理native返回值)

* 1. onLoaded()方法，为H5定制Webview的展现样式提供可能

举例——页面onLoad完成后，定义native顶部栏：

* + 1. 调用方式：

jsBridgeNative.bri\_onLoaded('我是标题','mdj://close',, '左上角文案', '右上角文案', ‘’,'bri\_onLoaded\_operateMethod', true, true, true, true,);

* + 1. 声明函数：

(在jsBridgeNativeHandle.js声明bri\_onLoaded\_operateMethod函数，在点击【右上角文案】时会调用此函数)

* + 1. 处理函数：

(在jsBridgeNativeHandle.js声明bri\_ onLoaded \_handle\_result函数处理native返回值)

* 1. onFullscreenLoaded ()方法
     1. 调用方式：

jsBridgeNative. bri\_onFullscreenLoaded (true,true,true,true);

* + 1. 处理函数：

(在jsBridgeNativeHandle.js声明bri\_onFullscreenLoaded\_handle\_result函数处理native返回值)

* 1. callMethod()方法，Native提供一些无需返回参数的方法供H5使用

举例——更新用户信息：

* + 1. 调用方式：

jsBridgeNative.bri\_callMethod('update\_user\_info',{},true,’mark1’)

* + 1. 处理函数

(在jsBridgeNativeHandle.js声明bri\_callMethod\_handle\_result函数处理native返回值)

* 1. openModal()方法，将一些通用的模态窗口抽象出来，共享为H5的能力；

举例——打开native登录窗口：

* + 1. 调用方式：

jsBridgeNative.bri\_openModal(‘native\_login’, ‘ ’, true, ‘native\_login \_callbackMethod,’mark1’);

* + 1. 声明函数：

(在jsBridgeNativeHandle.js声明 native\_login \_callbackMethod函数，登录成功后，native会调用该js函数

* + 1. 处理函数：

(在jsBridgeNativeHandle.js声明bri\_openModal\_handle\_result函数处理native返回值)

* 1. fetchFile ()方法，将Native的强大的拍照能力和选图功能共享到H5；

举例——拍摄原图：

* + 1. 调用方式：

jsBridgeNative. bri\_fetchFile (1,1, true, false, 0.8, true,’markCapture’);

举例——选取方图：

* + 1. 调用方式：

jsBridgeNative. bri\_fetchFile (2,2, false,false, 0.8, false,’markSelectImage’)

* + 1. 处理函数

(在jsBridgeNativeHandle.js声明bri\_fetchImage\_handle\_result函数处理native返回值)

* 1. showActionSheet ()方法，将Native的弹起菜单选项能力共享到H5；

举例—— 选择北京区域

* + 1. 调用方式：

jsBridgeNative.bri\_showActionSheet('北京区域',['海淀区','朝阳区'],'cancel','markAddress')

* + 1. 处理函数

(在jsBridgeNativeHandle.js声明bri\_showActionSheet\_handle\_result函数处理native返回值)

* 1. redirect ()方法，此接口沿用了消息中心跳转的type和参数，使得H5和Native跳转更加灵活；

举例—— 跳转到妆容详情页

* + 1. 调用方式：

jsBridgeNative.bri\_redirect(10,{'id':'570dea180cf2d7830b5c0675'},false)

* + 1. 处理函数

(在jsBridgeNativeHandle.js声明bri\_redirect\_handle\_result函数处理native返回值)

1. 如何使用--- Native调用H5方法