# **[Android总结之WebView与Javascript交互](http://www.cnblogs.com/whoislcj/p/5980240.html)**

### **前言：**

   最近公司的App为了加快开发效率选择了一部分功能采用H5开发，从目前市面的大部分App来讲，大致分成Native App、Web App、Hybrid App三种方式，个人觉得目前以Hybrid App居多，单纯的数据展示我们直接采用WebView来渲染就可以了，但是有时候可能会用到两者之间传递参数的情况，今天就来总结一下两者之间如何互相调用。本篇主要介绍WebView与Javascript交互数据，关于如何将H5网页呈现在WebView上可以参考这篇博客：[Android总结之WebView使用总结](http://www.cnblogs.com/whoislcj/p/5645025.html)。

### **WebView与Javascript交互：**

   WebView与Javascript交互是双向的数据传递，1.H5网页的JS函数调用Native函数 2.Native函数调用JS函数，具体实现以下面例子为主：

#### **1.）mainfest.xml中加入网络权限**

<uses-permission android:name="android.permission.INTERNET"/>

#### **2.）WebView开启支持JavaScript**

mWebView.getSettings().setJavaScriptEnabled(true);

#### **3.）简单的H5网页实现，主要实现actionFromNative()、actionFromNativeWithParam(String str)，放在assets文件下**

[IMG_256](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

<html><head>

<meta http-equiv="Content-Type" content="text/html;charset=utf-8">

<script type="text/javascript">

function actionFromNative(){

document.getElementById("log\_msg").innerHTML +=

"<br\>Native调用了js函数";

}

function actionFromNativeWithParam(arg){

document.getElementById("log\_msg").innerHTML +=

("<br\>Native调用了js函数并传递参数："+arg);

}

</script></head><body><p>WebView与Javascript交互</p><div>

<button onClick="window.wx.actionFromJs()">点击调用Native代码</button></div><br/><div>

<button onClick="window.wx.actionFromJsWithParam('come from Js')">点击调用Native代码并传递参数</button></div><br/><div id="log\_msg">调用打印信息</div></body></html>

[IMG_257](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

#### **4.）Native实现与JS交互函数：actionFromJs()、actionFromJsWithParam()**

[IMG_258](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

public class MainActivity extends Activity {

private WebView mWebView;

private TextView logTextView;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

mWebView = (WebView) findViewById(R.id.webview);

// 启用javascript

mWebView.getSettings().setJavaScriptEnabled(true);

// 从assets目录下面的加载html

mWebView.loadUrl("file:///android\_asset/wx.html");

mWebView.addJavascriptInterface(this, "wx");

logTextView = (TextView) findViewById(R.id.text);

Button button = (Button) findViewById(R.id.button);

button.setOnClickListener(new Button.OnClickListener() {

public void onClick(View v) {

// 无参数调用

mWebView.loadUrl("javascript:actionFromNative()");

// 传递参数调用

mWebView.loadUrl("javascript:actionFromNativeWithParam(" + "'come from Native'" + ")");

}

});

}

@android.webkit.JavascriptInterface

public void actionFromJs() {

runOnUiThread(new Runnable() {

@Override

public void run() {

Toast.makeText(MainActivity.this, "js调用了Native函数", Toast.LENGTH\_SHORT).show();

String text = logTextView.getText() + "\njs调用了Native函数";

logTextView.setText(text);

}

});

}

@android.webkit.JavascriptInterface

public void actionFromJsWithParam(final String str) {

runOnUiThread(new Runnable() {

@Override

public void run() {

Toast.makeText(MainActivity.this, "js调用了Native函数传递参数：" + str, Toast.LENGTH\_SHORT).show();

String text = logTextView.getText() + "\njs调用了Native函数传递参数：" + str;

logTextView.setText(text);

}

});

}

}

[IMG_259](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

mWebView.addJavascriptInterface(this, "wx");相当于添加一个js回调接口，然后给这个起一个别名，我这里起的名字wx（微信哈哈）。@android.webkit.JavascriptInterface为了解决addJavascriptInterface漏洞的，在4.2以后才有的。

#### **5.）布局文件实现**

[IMG_260](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

<?xml version="1.0" encoding="utf-8"?><LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<WebView

android:id="@+id/webview"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_weight="1"/>

<TextView android:id="@+id/text"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text=""/>

<Button

android:id="@+id/button"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Native调用js函数"/>

</LinearLayout>

[IMG_261](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

#### **5.）代码简单解说**

（1.）js（HTML）访问Android（Java）端代码是通过jsObj对象实现的，调用jsObj对象中的函数，如： window.jsObj.actionFromJs()，这里的jsObj就是Native中添加接口的别名

（2.）Android（Java）访问js（HTML）端代码是通过loadUrl函数实现的，访问格式如：mWebView.loadUrl("javascript:actionFromNative()");

demo运行截图：



### **总结：**

   这里简单的实现了Js与Native的交互，后期会抽空看下WebViewJavascriptBridge这个开源框架。