Android与H5交互

Android实现H5和Native交互的两种方法：一种是拦截Url，另一种是JavaScript注入

**WebView**

1. 获取到webview之后，设置webview属性  
   private void settingProperty(){

myJavascriptInterface = new MyJavascriptInterface();

WebSettings webSettings = mWebView.getSettings();

//支持js

webSettings.setJavaScriptEnabled(true);

//支持通过JS打开新窗口

webSettings.setJavaScriptCanOpenWindowsAutomatically(true);

//缩放至屏幕的大小

webSettings.setLoadWithOverviewMode(true);

//将图片调整到适合webview的大小

webSettings.setUseWideViewPort(false);

//设置缓存方式

webSettings.setCacheMode(WebSettings.LOAD\_DEFAULT);

//开启DOM storage API功能

webSettings.setDomStorageEnabled(true);

//开启 database storage API 功能

webSettings.setDatabaseEnabled(true);

String cacheDirPath = getCacheDir().getAbsolutePath()+ "/webViewCache";

//设置数据库缓存路径

webSettings.setDatabasePath(cacheDirPath);

//开启Application H5 Caches 功能

webSettings.setAppCacheEnabled(true);

//设置Application Caches 缓存目录

webSettings.setAppCachePath(cacheDirPath);

/\*\*对系统API在19以上的版本作了兼容。因为4.4以上系统在onPageFinished时再恢复图片加载时, 如果存在多张图片引用的是相同的src时，会只有一个image标签得到加载，因而对于这样的系统我们就先直接加载 \*/

if (Build.VERSION.SDK\_INT >= 19){

webSettings.setLoadsImagesAutomatically(true);

} else {

webSettings.setLoadsImagesAutomatically(false);

}

}

2. 重写WebViewClient处理webview的各种通知与事件

3. 重写WebChromeClient，辅助webview处理javascript对话框，网页的icon和title，以及加载进度条等。

4. 提供接口供js调用(@JavascriptInterface注解标注该方法是供js调用)

WebView.addJavascriptInterface(new MyJavascriptInterface(), “android”); // “android”是为js接口取的名字

public class MyJavascriptInterface{

@JavascriptInterface

public String h5NativtCallback(){

LogUtils.i(TAG, "h5NativeCallback");

return "js 调用 android 的回调 "+(num\_android++);

}

}

5. android调用js方法

mWebView.loadUrl("javascript:callbackFromNative('the data is from android! ‘)"

loadUrl方法中的字符串参数构成：”javascript:”+js方法，其中js方法中的参数是传递给js方法的参数

H5

H5调用android native方法：window.android.h5NativtCallback();

即window.js接口名.加方法名