# **Android WebView基本使用配置**

### 基本配置

WebSettings webSettings = webView.getSettings(); //支持缩放，默认为true。

webSettings .setSupportZoom(false); //调整图片至适合webview的大小

webSettings .setUseWideViewPort(true); // 缩放至屏幕的大小

webSettings .setLoadWithOverviewMode(true); //设置默认编码

webSettings .setDefaultTextEncodingName("utf-8"); ////设置自动加载图片

webSettings .setLoadsImagesAutomatically(true);

### 高级配置

//多窗口

supportMultipleWindows(); //获取触摸焦点

webview.requestFocusFromTouch();//允许访问文件

setAllowFileAccess(true); //开启javascript

setJavaScriptEnabled(true);

//支持通过JS打开新窗口

setJavaScriptCanOpenWindowsAutomatically(true);//提高渲染的优先级

webSettings.setRenderPriority(RenderPriority.HIGH);

//支持内容重新布局

setLayoutAlgorithm(LayoutAlgorithm.SINGLE\_COLUMN);//关闭webview中缓存

setCacheMode(WebSettings.LOAD\_CACHE\_ELSE\_NETWORK);

### 访问网址

webView.loadUrl("https://www.google.com.hk/?gws\_rd=ssl");

### 访问应用内资源

新建assets文件夹，在此文件夹下新建HTML文件如：js\_demo.html。

webView.loadUrl("file:///android\_asset/js\_demo.html");

### 加载HTML片段或者富文本字符串

webView.loadData(data, "text/html; charset=UTF-8", null);  
这样写一般就不会乱码了，data为String字符串。

### 使用loadDataWithBaseURL方法加载内容

webView.loadDataWithBaseURL (String baseUrl, String data, String mimeType, String encoding, String historyUrl)

loadDataWithBaseURL()可以指定HTML代码片段中相关资源的相对根路径，也可以指定历史Url，baseUrl指定了data参数中数据的基准地址，data中的数据可能会有超链接或者是image元素使用了相对路径，如果没有baseUrl，webview将访问不到这些资源。

### 内存泄露问题

在 Activity 的onDestroy()生命周期，首先让 WebView 加载 null 内容，然后移除 WebView，再销毁 WebView，最后置空。代码如下：

@Override

protected void onDestroy() {

if (webView!= null) {

webView.loadDataWithBaseURL(null, "", "text/html", "utf-8", null);

webView.clearHistory();

((ViewGroup) webView.getParent()).removeView(mWebView);

webView.destroy();

webView= null;

}

super.onDestroy();

}