**1. 简介**

一个基于webkit引擎、展现web页面的控件

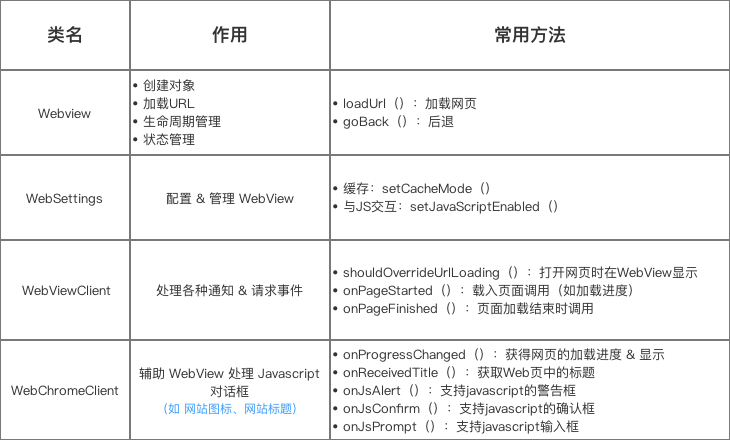
a. Android 4.4前：Android Webview在低版本 & 高版本采用了不同的webkit版本的内核   
b. Android 4.4后：直接使用了Chrome内核

**2. 作用**

* 在 Android 客户端上加载h5页面
* 在本地 与 h5页面实现交互 & 调用
* 其他：对 url 请求、页面加载、渲染、对话框 进行额外处理。

**3. 具体使用**

* Webview的使用主要包括：Webview类 及其 工具类（WebSettings类、WebViewClient类、WebChromeClient类）

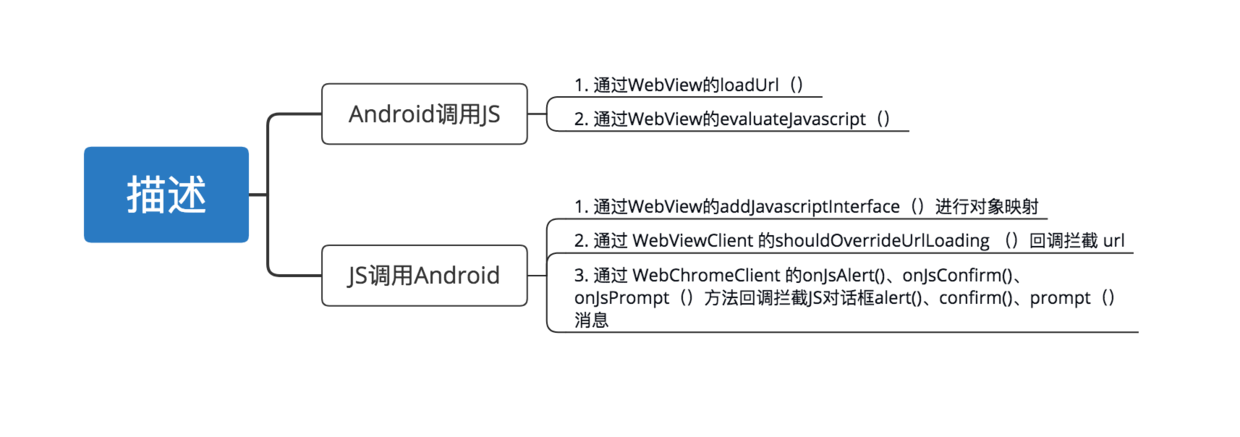


* 下面我将详细介绍上述4个使用类 & 使用方法
* 具体请看文章：[Android开发：最全面、最易懂的Webview详解](https://blog.csdn.net/carson_ho/article/details/52693322)

**4. WebView与 JS 的交互方式**

* 在Android WebView的使用中，与前端h5页面交互的需求十分常见
* Android 与 JS 通过WebView互相调用方法，实际上是：Android 去调用JS的代码 + JS去调用Android的代码

二者沟通的桥梁是WebView



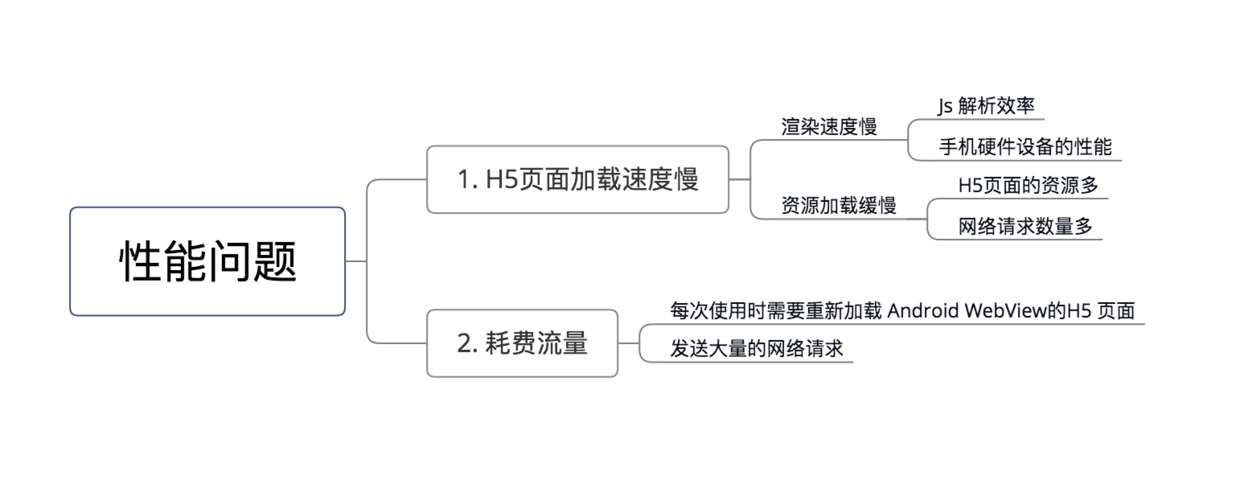
* 具体介绍请看文章：[Android：你要的WebView与 JS 交互方式 都在这里了](https://blog.csdn.net/carson_ho/article/details/64904691)

**5. 使用漏洞**

* WebView 使用过程中存在许多漏洞，容易造成用户数据泄露等等危险，而很多人往往会忽视这个问题
* WebView中，主要漏洞有3类：**任意代码执行漏洞、密码明文存储漏洞、域控制不严格漏洞**
* 漏洞具体介绍 & 修复方式请看文章：[Android：你不知道的 WebView 使用漏洞](https://blog.csdn.net/carson_ho/article/details/64904635)

**6. 缓存机制构建**

* Android WebView由于前端h5本身的原因，存在加载效率慢 & 流量耗费的性能问题，具体介绍如下：



* 本文通过 **H5缓存机制 + 资源预加载 + 资源拦截**的方式 构建了一套WebView缓存机制，从而解决Android WebView的性能问题，最终提高用户使用体验
* 具体缓存机制的讲解请看文章：[手把手教你构建 Android WebView 的缓存机制 & 资源预加载方案](https://blog.csdn.net/carson_ho/article/details/71402764)

至此，关于Android WebView的所有知识讲解完毕。

**7. 总结**

* 本文全面讲解了 WebView的相关知识，含具体介绍、使用教程、与前端JS交互、缓存机制构建等等，相信你对Android WebView的使用已经非常熟悉了。
* 关于WebView的系列文章希望对你有所帮助   
  [Android开发：最全面、最易懂的Webview详解](https://blog.csdn.net/carson_ho/article/details/52693322)   
  [Android：你要的WebView与 JS 交互方式 都在这里了](https://blog.csdn.net/carson_ho/article/details/64904691)   
  [手把手教你构建 Android WebView 的缓存机制 & 资源预加载方案](https://blog.csdn.net/carson_ho/article/details/71402764)   
  [Android：你不知道的 WebView 使用漏洞](https://blog.csdn.net/carson_ho/article/details/64904635)
* 接下来我会继续讲解其他安卓开发的知识，有兴趣可以继续关注[Carson\_Ho的安卓开发笔记](https://blog.csdn.net/carson_ho)！！！！