

Evan

Only let oneself become strong enough, good enough, can afford the life that you want to.

☰ 目录视图

☰ 摘要视图

🔒 订阅专栏

从创业到再就业，浅谈对程序员职业生​​涯的看法 征文 | 你会为 AI 转型么？ 赠书：7月大咖新书机器学习/Android/python

Android之Android WebView常见问题及解决方案汇总

标签： android webview 解决方案

2016-01-02 21:20 254人阅读 评论(0) 收藏 举报

☰ 分类： 移动开发 (38) ▾

目录(?) [-]

1. WebView cookies清理

2. 判断WebView是否已经滚动到页面底端

如有转载，请声明出处: 时之沙: <http://blog.csdn.net/t12x3456>

Android WebView常见问题解决方案汇总:

就目前而言，如何应对版本的频繁更新呢，又如何灵活多变地展示我们的界面呢,这又涉及到了web app与native app之间孰优孰劣的争论. 于是乎,一种混合型的app诞生了,灵活多变的部分，如淘宝商城首页的活动页面，一集凡客诚品中我们都可以见到web 页面与native页面的混合，既利用了web app的灵活易更新，也借助了native app本身的效率.

当然，就会用到webview这样的一个控件，这里，我把自己使用过程中遇到的一些问题整理下来.

首先上张图对WebView进行一个基本的回顾:



<http://blog.csdn.net/t12x3456>

以上思维导图原文件下载地址:
<http://download.csdn.net/detail/t12x3456/6509195>

然后看一下具体的问题及解决方案:

1.为WebView自定义错误显示界面:

覆写WebViewClient中的onReceivedError()方法:

```
[java]
01. <span style="font-size:14px;">    /**
02.  * 显示自定义错误提示页面，用一个View覆盖在WebView
03.  */
04. protected void showErrorPage() {
05.     LinearLayout webParentView = (LinearLayout)mWebView.getParent();
06.
07.     initErrorPage();
08.     while (webParentView.getChildCount() > 1) {
09.         webParentView.removeViewAt(0);
10.     }
11.     LinearLayout.LayoutParams lp = new LinearLayout.LayoutParams(LayoutParams.FILL_PARENT, LayoutParams.FILL_PARENT);
12.     webParentView.addView(mErrorView, 0, lp);
13.     mIsErrorPage = true;
14. }
15. protected void hideErrorPage() {
16.     LinearLayout webParentView = (LinearLayout)mWebView.getParent();
17.
18.     mIsErrorPage = false;
19.     while (webParentView.getChildCount() > 1) {
20.         webParentView.removeViewAt(0);
21.     }
22. }
23.
24.
25. protected void initErrorPage() {
26.     if (mErrorView == null) {
27.         mErrorView = View.inflate(this, R.layout.online_error, null);
28.         Button button = (Button)mErrorView.findViewById(R.id.online_error_btn_retry);
29.         button.setOnClickListener(new OnClickListener() {
30.             public void onClick(View v) {
31.                 mWebView.reload();
32.             }
33.         });
34.         mErrorView.setOnClickListener(null);
35.     }
36. }
37.
38.
39.
40.
41.
42. @Override
43. public void onReceivedError(WebView view, int errorCode, String description, String failingUrl) {
44.     <span style="white-space:pre">        </span>
45.     <span style="white-space:pre">        </span>mErrorView.setVisibility(View.VISIBLE);
46.     <span style="white-space:pre">        </span>super.onReceivedError(view, errorCode, description, failingUrl);
47. }
48.
49. </span>
```

2.WebView cookies清理 :

```
[java]
01. CookieSyncManager.createInstance(this);
02. CookieSyncManager.getInstance().startSync();
03. CookieManager.getInstance().removeSessionCookie();
```

3.清理cache 和历史记录:

```
[java]
```

```
01. <span style="font-size:14px;">    webView.clearCache(true);
02.    webView.clearHistory(); </span>
```

4.判断WebView是否已经滚动到页面底端:

[java]

```
01. <span style="font-size:14px;">    getScrollY()方法返回的是当前可见区域的顶端距整个页面顶端的距离,也就是当前内容滚动的距离.
02.    getHeight()或者getBottom()方法都返回当前WebView 这个容器的高度
03.    getContentHeight 返回的是整个html 的高度,但并不等同于当前整个页面的高度,因为WebView 有缩放功能,所以当前整个页面的高度实际上应该是原始
    html 的高度再乘上缩放比例. 因此,更正后的结果,准确的判断方法应该是:
04.    if(Webview.getContentHeight*WebView.getScale() == (webview.getHeight()+WebView.getScrollY())){ //已经处于底端 } </span>
```

5.URL拦截:

android WebView是拦截不到页面内的fragment跳转的。但是url跳转的话，又会引起页面刷新，H5页面的体验又下降了。只能给WebView注入js方法了。

6.处理WebView中的非超链接请求(如Ajax请求):

有时候需要加上请求头，但是非超链接的请求，没有办法再shouldOverrinding中拦截并用webView.loadUrl(String url,HashMap headers)方法添加请求头

目前用了一个临时的办法解决：

首先需要在url中加特殊标记/协议, 如在onWebViewResource方法中拦截对应的请求，然后将要添加的请求头，以get形式拼接到url末尾
在shouldInterceptRequest()方法中,可以拦截到所有的网页中资源请求，比如加载JS，图片以及Ajax请求等等

Ex:

[java]

```
01. <span style="font-size:14px;">    @SuppressWarnings("NewApi")
02.    @Override
03.    public WebResourceResponse shouldInterceptRequest(Webview view,String url) {
04.        // 非超链接(如Ajax)请求无法直接添加请求头, 现拼接到url末尾,这里拼接一个imei作为示例
05.
06.        String ajaxUrl = url;
07.        // 如标识:req=ajax
08.        if (url.contains("req=ajax")) {
09.            ajaxUrl += "&imei=" + imei;
10.        }
11.
12.        return super.shouldInterceptRequest(view, ajaxUrl);
13.
14.    } </span>
```

7.在页面中先显示图片:

[java]

```
01. <span style="font-size:14px;">    @Override
02.    public void onLoadResource(Webview view, String url) {
03.        mListener.onWebViewEvent(CustomWebView.this, OnWebViewEventListener.EVENT_ON_LOAD_RESOURCE, url);
```

```
04.         if (url.indexOf(".jpg") > 0) {
05.             hideProgress(); //请求图片时即显示页面
06.             mListener.onWebViewEvent(CustomWebView.this, OnWebViewEventListener.EVENT_ON_HIDE_PROGRESS, view.getUrl());
07.         }
08.         super.onLoadResource(view, url);
09.     } </span>
```

8.屏蔽掉长按事件 因为webview长按时将会调用系统的复制控件:

[java]

```
01. <span style="font-size:14px;"> mWebView.setOnLongClickListener(new OnLongClickListener() {
02.
03.     @Override
04.     public boolean onLongClick(View v) {
05.         return true;
06.     }
07. }); </span>
```

9.在WebView加入 flash支持:

[java]

```
01. <span style="font-size:14px;"> String temp = "<html><body bgcolor=\"" + "black"
02.     + "\"> <br/><embed src=\"" + url + "\" width=\"" + "100%"
03.     + "\" height=\"" + "90%" + "\" scale=\"" + "noscale"
04.     + "\" type=\"" + "application/x-shockwave-flash"
05.     + "\"> </embed></body></html>";
06. String mimeType = "text/html";
07. String encoding = "utf-8";
08. web.loadDataWithBaseURL("null", temp, mimeType, encoding, ""); </span>
```

10.WebView保留缩放功能但隐藏缩放控件:

[java]

```
01. <span style="font-size:14px;"> mWebView.getSettings().setSupportZoom(true);
02. mWebView.getSettings().setBuiltInZoomControls(true);
03. if (DeviceUtils.hasHoneycomb())
04.     mWebView.getSettings().setDisplayZoomControls(false); </span>
```

注意：setDisplayZoomControls是在Android 3.0中新增的API.

这些是目前我整理出来的一些注意事项和问题解决方案,也欢迎大家多提一些关于webview的问题,如果有合适的解决方案,我会直接更新到这篇文章.

8月份更新:

11.WebView 在Android4.4的手机上onPageFinished()回调会多调用一次(具体原因待追查)

需要尽量避免在onPageFinished()中做业务操作，否则会导致重复调用，还有可能会引起逻辑上的错误。

12.需要通过获取Web页中的title用来设置自己界面中的title及相关问题:

需要给WebView设置 WebChromeClient,并在onReceiveTitle()回调中获取

```
[java]
01. <span style="font-size:14px;"> WebChromeClient webChromeClient = new WebChromeClient() {
02.     @Override
03.     public void onReceivedTitle(WebView view, String title) {
04.         super.onReceivedTitle(view, title);
05.
06.         txtTitle.setText(title);
07.     }
08.
09. }; </span>
```

但是发现在小米3的手机上，当通过webview.goBack()回退的时候，并没有触发onReceiveTitle()，这样会导致标题仍然是之前子页面的标题，没有切换回来。

这里可以分两种情况去处理：

(1) 可以确定webview中子页面只有二级页面，没有更深的层次，这里只需要判断当前页面是否为初始的主页面，可以goBack的话，只要将标题设置回来即可。

(2)webview中可能有多级页面或者以后可能增加多级页面,这种情况处理起来要复杂一些:

因为正常顺序加载的情况onReceiveTitle是一定会触发的，所以需要自己来维护webview loading的一个url栈及url与title的映射关系

那么就需要一个ArrayList来保持加载过的url,一个HashMap保存url及对应的title.

正常顺序加载时，将url和对应的title保存起来，webview回退时，移除当前url并取出将要回退到的web 页的url,找到对应的title进行设置即可。

这里还要说一点，当加载出错的时候，比如无网络，这时onReceiveTitle中获取的标题为 找不到该网页,因此建议当触发onReceiveError时，不要使用获取到的title.

13.WebView因addJavaScriptInterface()引起的安全问题.

这个问题主要是因为会有恶意的js代码注入,尤其是在已经获取root权限的手机上，一些恶意程序可能会利用该漏洞安装或者卸载应用.

关于详细的情况可以参考下面这篇文章：

<http://blog.csdn.net/leehong2005/article/details/11808557>

还有一个开源项目可以参考: <https://github.com/pedant/safe-java-js-webview-bridge>, 该项目利用onJsPrompt() 替代了addJavaScriptInterface(), (解决方案类似上述参考的博客)同时增加了异步回调,

很好地解决了webview js注入的安全问题.

10月份更新:

14.WebView页面中播放了音频,退出Activity后音频仍然在播放

需要在Activity的onDestory()中调用

```
01. webView.destroy();
```

但是直接调用可能会引起如下错误:

```
[java]
01. <span style="font-size:14px;"> 10-10 15:01:11.402: E/ViewRootImpl(7502): sendUserActionEvent() mView == null
02. 10-10 15:01:26.818: E/webview(7502): java.lang.Throwable: Error: WebView.destroy() called while still attached!
03. 10-10 15:01:26.818: E/webview(7502): at android.webkit.WebViewClassic.destroy(WebViewClassic.java:4142)
04. 10-10 15:01:26.818: E/webview(7502): at android.webkit.WebView.destroy(WebView.java:707)
05. 10-
10 15:01:26.818: E/webview(7502): at com.didi.taxi.ui.webview.OperatingWebViewActivity.onDestroy(OperatingWebViewActivity.java:23)
06. 10-10 15:01:26.818: E/webview(7502): at android.app.Activity.performDestroy(Activity.java:5543)
07. 10-10 15:01:26.818: E/webview(7502): at android.app.Instrumentation.callActivityOnDestroy(Instrumentation.java:1134)
08. 10-10 15:01:26.818: E/webview(7502): at android.app.ActivityThread.performDestroyActivity(ActivityThread.java:3619)
09. 10-10 15:01:26.818: E/webview(7502): at android.app.ActivityThread.handleDestroyActivity(ActivityThread.java:3654)
10. 10-10 15:01:26.818: E/webview(7502): at android.app.ActivityThread.access$1300(ActivityThread.java:159)
11. 10-10 15:01:26.818: E/webview(7502): at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1369)
12. 10-10 15:01:26.818: E/webview(7502): at android.os.Handler.dispatchMessage(Handler.java:99)
13. 10-10 15:01:26.818: E/webview(7502): at android.os.Looper.loop(Looper.java:137)
14. 10-10 15:01:26.818: E/webview(7502): at android.app.ActivityThread.main(ActivityThread.java:5419)
15. 10-10 15:01:26.818: E/webview(7502): at java.lang.reflect.Method.invokeNative(Native Method)
16. 10-10 15:01:26.818: E/webview(7502): at java.lang.reflect.Method.invoke(Method.java:525)
17. 10-10 15:01:26.818: E/webview(7502): at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:1187)
18. 10-10 15:01:26.818: E/webview(7502): at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:1003)
19. 10-10 15:01:26.818: E/webview(7502): at dalvik.system.NativeStart.main(Native Method) </span>
```

如上所示，webview调用destory时,webview仍绑定在Activity上.这是由于自定义webview构建时传入了该Activity的context对象,因此需要先从父容器中移除webview,然后再销毁webview:

```
[java]
01. <span style="font-size:14px;">rootLayout.removeView(webView); </span>
```

from:<http://www.cnblogs.com/lee0oo0/p/4026774.html>

顶

0

踩

0

- [上一篇](#) Intent详解
- [下一篇](#) Android实现推送方式解决方案

相关文章推荐

- Android WebView常见问题及解决方案汇总
 - Android Launcher开发之自动添加桌面快捷方式...
 - (转)Android WebView常见问题及解决方案汇总
 - Android WebView Touch事件及相关问题处理
 - Android之Android WebView常见问题及解决方...
- 双机热备常见问题及解决方案（摘自网络）
 - QQ浏览器X5内核问题汇总
 - Android WebView常见问题解决方案汇总
 - 电脑公司特别版论坛常见问题及解决方案
 - Android WebView常见问题及解决方案汇总 .