搜索……

7.5.3 Android 4.4后WebView的一些注意事项

分类 **Android 基础入门教程**

本节引言：

本节参考原文：Android 4.4 中 WebView 使用注意事项.md

从Android 4.4开始，Android中的WebView不再是基于WebKit的，而是开始基于Chromium，这个改变 使得WebView的性能大幅提升，并且对HTML5，CSS，JavaScript有了更好的支持！

虽然chromium完全取代了以前的WebKit for Android，但Android WebView的API接口并没有变， 与老的版本完全兼容。这样带来的好处是基于WebView构建的APP，无需做任何修改， 就能享受chromium内核的高效与强大。

对于4.4后的WebView，我们需要注意下下面这些问题：

1.多线程

如果你在子线程中调用WebView的相关方法，而不在UI线程，则可能会出现无法预料的错误。 所以，当你的程序中需要用到多线程时候，也请使用runOnUiThread()方法来保证你关于 WebView的操作是在UI线程中进行的：

```
runOnUiThread(newRunnable(){
@Override
publicvoid run(){
   // Code for WebView goes here
   }
});
```

2.线程阻塞

永远不要阻塞UI线程，这是开发Android程序的一个真理。虽然是真理，我们却往往不自觉的 犯一些错误违背它，一个开发中常犯的错误就是：在UI线程中去等待JavaScript 的回调。 例如：

Android 基础入门教程(Q群号：153836263)

1.0 Android基础入门教程
1.0.1 2015年最新Android基…
1.1 背景相关与系统架构分析
1.2 开发环境搭建
1.2.1 使用Eclipse + ADT + S…
1.2.2 使用Android Studio开…
1.3 SDK更新不了问题解决
1.4 Genymotion模拟器安装
1.5.1 Git使用教程之本地仓…
1.5.2 Git之使用GitHub搭建…
1.6 .9(九妹)图片怎么玩
1.7 界面原型设计
1.8 工程相关解析(各种文件…
1.9 Android程序签名打包
1.11 反编译APK获取代码&…
2.1 View与ViewGroup的概念
2.2.1 LinearLayout(线性布局)
2.2.2 RelativeLayout(相对布…
2.2.3 TableLayout(表格布局)
2.2.4 FrameLayout(帧布局)
2.2.5 GridLayout(网格布局)
2.2.6 AbsoluteLayout(绝对…
2.3.1 TextView(文本框)详解
2.3.2 EditText(输入框)详解
2.3.3 Button(按钮)与ImageB…
2.3.4 ImageView(图像视图)
2.3.5.RadioButton(单选按钮…
2.3.6 开关按钮ToggleButton…
2.3.7 ProgressBar(进度条)
2.3.8 SeekBar(拖动条)
2.3.9 RatingBar(星级评分条)
2.4.1 ScrollView(滚动条)

反馈

```
// This code is BAD and will block the UI thread
webView.loadUrl("javascript:fn()");
while(result ==null) {
    Thread.sleep(100);
}
```

千万不要这样做，Android 4.4中，提供了新的Api来做这件事情。

evaluateJavascript() 就是专门来异步执行JavaScript代码的。

3.evaluateJavascript() 方法

专门用于异步调用JavaScript方法，并且能够得到一个回调结果。

示例：

```
mWebView.evaluateJavascript(script, new ValueCallback<String>()
 {
 @Override
 public void onReceiveValue(String value) {
      //TODO
 }
});
```

4.处理WebView中url的跳转

新版WebView对于自定义scheme的url跳转，新增了更为严格的限制条件。 当你实现了 shouldOverrideUrlLoading() 或 shouldInterceptRequest() 回调，WebView 也只会在跳转url是合法Url时才会跳转。 例如，如果你使用这样一个url ：

```
<a href="showProfile">Show Profile</a>
```

shouldOverrideUrlLoading() 将不会被调用。

正确的使用方式是：

```
<a href="example-app:showProfile">Show Profile</a>
```

对应的检测Url跳转的方式：

```
// The URL scheme should be non-hierarchical (no trailing slash
es)
 privatestaticfinalString APP_SCHEME ="example-app:";
 @Override
 publicboolean shouldOverrideUrlLoading(WebView view,String url
){
     if(url.startsWith(APP_SCHEME)){
         urlData =URLDecoder.decode(url.substring(APP_SCHEME.le
```

2.4.2 Date & Time组件(上)
2.4.3 Date & Time组件(下)
2.4.4 Adapter基础讲解
2.4.5 ListView简单实用
2.4.6 BaseAdapter优化
2.4.7ListView的焦点问题
2.4.8 ListView之checkbox错…
2.4.9 ListView的数据更新问题
2.5.0 构建一个可复用的自定…
2.5.1 ListView Item多布局的…
2.5.2 GridView(网格视图)的…
2.5.3 Spinner(列表选项框)…
2.5.4 AutoCompleteTextVie…
2.5.5 ExpandableListView(…
2.5.6 ViewFlipper(翻转视图)…
2.5.7 Toast(吐司)的基本使用
2.5.8 Notification(状态栏通…
2.5.9 AlertDialog(对话框)详解
2.6.0 其他几种常用对话框基…
2.6.1 PopupWindow(悬浮框…
2.6.2 菜单(Menu)
2.6.3 ViewPager的简单使用
2.6.4 DrawerLayout(官方侧…
3.1.1 基于监听的事件处理机制
3.2 基于回调的事件处理机制
3.3 Handler消息传递机制浅析
3.4 TouchListener PK OnTo…
3.5 监听EditText的内容变化
3.6 响应系统设置的事件(Co…
3.7 AnsyncTask异步任务
3.8 Gestures(手势)
4.1.1 Activity初学乍练
4.1.2 Activity初窥门径
4.1.3 Activity登堂入室
4.2.1 Service初涉
4.2.2 Service进阶
4.2.3 Service精通
4.3.1 BroadcastReceiver牛…
4.3.2 BroadcastReceiver庖…
4.4.1 ContentProvider初探

```
ngth()),"UTF-8");
            respondToData(urlData);
            returntrue;
        }
        returnfalse;
}
```

当然，也可以这样使用：

```
webView.loadDataWithBaseURL("example-app://example.co.uk/", HTM
L_DATA,null,"UTF-8",null);
```

5.UserAgent变化

如果你的App对应的服务端程序，会根据客户端传来的UserAgent来做不同的事情，那么你需要注意 的是，新版本的WebView中，UserAgent有了些微妙的改变：

```
Mozilla/5.0 (Linux; Android 4.4; Nexus 4 Build/KRT16H)
AppleWebKit/537.36(KHTML, like Gecko) Version/4.0 Chrome/30.0.0
.0
Mobile Safari/537.36
```

使用**getDefaultUserAgent()**方法可以获取默认的UserAgent，也可以通过：

```
mWebView.getSettings().setUserAgentString(ua);
mWebView.getSettings().getUserAgentString();
```

来设置和获取自定义的UserAgent。

6.使用addJavascriptInterface()的注意事项

从Android4.2开始。 只有添加 @JavascriptInterface 声明的Java方法才可以被JavaScript调用， 例如：

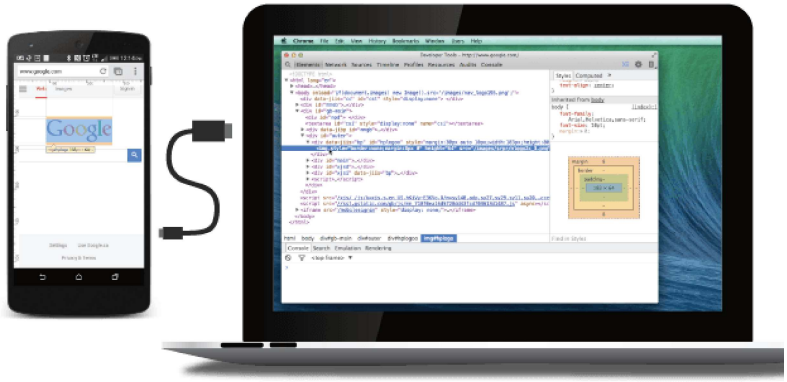
```
class JsObject {
    @JavascriptInterface
    public String toString() { return "injectedObject"; }
}

webView.addJavascriptInterface(new JsObject(), "injectedObject"
);
webView.loadData("", "text/html", null);
webView.loadUrl("javascript:alert(injectedObject.toString())");
```

4.4.2 ContentProvider再探…
4.5.1 Intent的基本使用
4.5.2 Intent之复杂数据的传递
5.1 Fragment基本概述
5.2.1 Fragment实例精讲—…
5.2.2 Fragment实例精讲—…
5.2.3 Fragment实例精讲—…
5.2.4 Fragment实例精讲—…
5.2.5 Fragment实例精讲—…
6.1 数据存储与访问之——文…
6.2 数据存储与访问之——S…
6.3.1 数据存储与访问之——…
6.3.2 数据存储与访问之——…
7.1.1 Android网络编程要学…
7.1.2 Android Http请求头与…
7.1.3 Android HTTP请求方…
7.1.4 Android HTTP请求方…
7.2.1 Android XML数据解析
7.2.2 Android JSON数据解析
7.3.1 Android 文件上传
7.3.2 Android 文件下载（1）
7.3.3 Android 文件下载（2）
7.4 Android 调用 WebService
7.5.1 WebView(网页视图)基…
7.5.2 WebView和JavaScrip…
7.5.3 Android 4.4后WebVie…
7.5.4 WebView文件下载
7.5.5 WebView缓存问题
7.5.6 WebView处理网页返…
7.6.1 Socket学习网络基础准备
7.6.2 基于TCP协议的Socket…
7.6.3 基于TCP协议的Socket…
7.6.4 基于UDP协议的Socke…
8.1.1 Android中的13种Draw…
8.1.2 Android中的13种Draw…
8.1.3 Android中的13种Draw…
8.2.1 Bitmap(位图)全解析 P…
8.2.2 Bitmap引起的OOM问题
8.3.1 三个绘图工具类详解
8.3.2 绘图类实战示例

7.Remote Debugging

新版的WebView还提供了一个很厉害的功能：使用Chrome来调试你运行在WebView中的程序 具体可以看：remote-debugging PS：需要梯子~你也可以直接百度remote-debugging了解相关信息，以及如何使用！



上一节中N5读取联系人的问题解决：

嘿嘿，看完上面的，我们知道在Android4.2后，只有添加 @JavascriptInterface 声明的Java方法才可以被JavaScript调用，于是乎我们为之前的两个方法加上@JavascriptInterface

```
public class SharpJS {
    @JavascriptInterface
    public void contactlist() {
        try {
            System.out.println("contactlist()方法执行了！");
            String json = buildJson(getContacts());
            wView.loadUrl("javascript:show('" + json + "')");
        } catch (Exception e) {
            System.out.println("设置数据失败" + e);
        }
    }
    @JavascriptInterface
    public void call(String phone) {
        System.out.println("call()方法执行了！");
        Intent it = new Intent(Intent.ACTION_CALL, Uri.parse("tel:" + phone));
        startActivity(it);
    }
}
```

但是，加完以后，并没有和我们的预想一样，出现我们想要的联系人列表，这是为什么呢？ 我们通过查看Log发现下面这样一段信息：

```
W/WebView:  java.lang.Throwable: A WebView method was called on thread 'JavaBridge'. All WebView methods must be called on the same thread. (Expected Looper Loop
            at android.webkit.WebView.checkThread(WebView.java:2194)
            at android.webkit.WebView.loadUrl(WebView.java:851)
            at com.jay.webviewdemo5.MainActivity$SharpJS.contactlist(MainActivity.java:49)
            at org.chromium.base.SystemMessageHandler.nativeDoRunLoopOnce(Native Method)
            at org.chromium.base.SystemMessageHandler.handleMessage(SystemMessageHandler.java:53)
            at android.os.Handler.dispatchMessage(Handler.java:102)
            at android.os.Looper.loop(Looper.java:135)
            at android.os.HandlerThread.run(HandlerThread.java:61)
09-12 11:51:43.128    2670-2748/? I/System.out:  设置数据失败java.lang.RuntimeException: java.lang.Throwable: A WebView method was called on thread 'JavaBridge'.
09-12 11:51:43.128     759-6547/? W/ActivityManager:  getTasks: caller 10202 does not hold GET_TASKS; limiting output
```

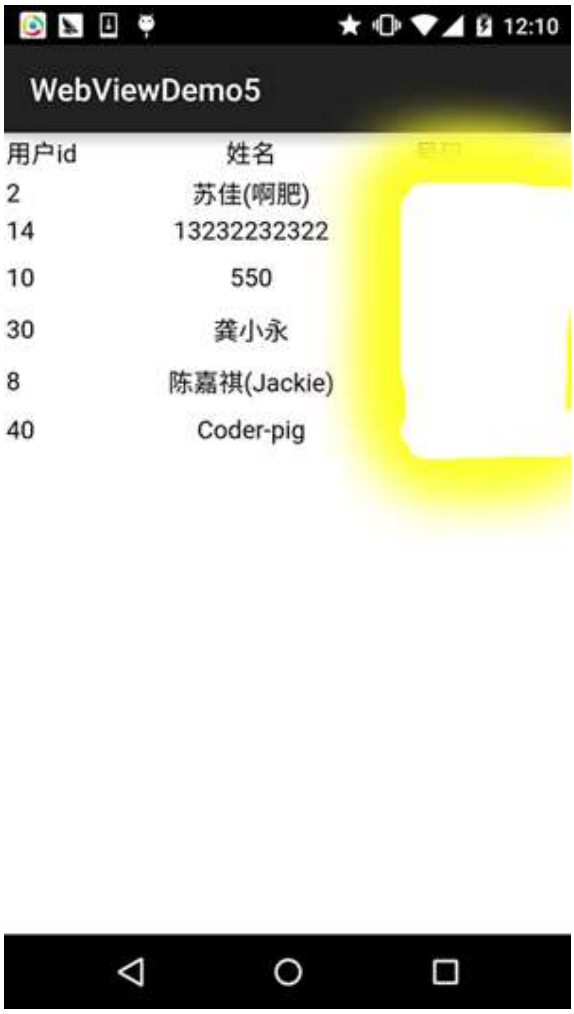
大概的意思就是：所有的WebView方法都应该在同一个线程程中调用，而这里的contactlist方法却在 JavaBridge线程中被调用了！所以我们要要把contactlist里的东东写到同一个线程中，比如一种解决 方法，就是下面这种：

8.3.3 Paint API之—— Mask…
8.3.4 Paint API之—— Xferm…
8.3.5 Paint API之—— Xferm…
8.3.6 Paint API之—— Xferm…
8.3.7 Paint API之—— Xferm…
8.3.8 Paint API之—— Xferm…
8.3.9 Paint API之—— Color…
8.3.10 Paint API之—— Colo…
8.3.11 Paint API之—— Colo…
8.3.12 Paint API之—— Path…
8.3.13 Paint API之—— Sha…
8.3.14 Paint几个枚举/常量值…
8.3.15 Paint API之——Type…
8.3.16 Canvas API详解(Part 1)
8.3.17 Canvas API详解(Part…
8.3.18 Canvas API详解(Part…
8.4.1 Android动画合集之帧…
8.4.2 Android动画合集之补…
8.4.3 Android动画合集之属…
8.4.4 Android动画合集之属…
9.1 使用SoundPool播放音…
9.2 MediaPlayer播放音频与…
9.3 使用Camera拍照
9.4 使用MediaRecord录音
10.1 TelephonyManager(电…
10.2 SmsManager(短信管理…
10.3 AudioManager(音频管…
10.4 Vibrator(振动器)
10.5 AlarmManager(闹钟服务)
10.6 PowerManager(电源服…
10.7 WindowManager(窗口…
10.8 LayoutInflater(布局服务)
10.9 WallpaperManager(壁…
10.10 传感器专题(1)——相…
10.11 传感器专题(2)——方…
10.12 传感器专题(3)——加…
10.12 传感器专题(4)——其…
10.14 Android GPS初涉
11.0《2015最新Android基…

```
@JavascriptInterface
public void contactlist() {
    wView.post(new Runnable() {
        @Override
        public void run() {
            try {
                System.out.println("contactlist()方法执行了！");
                String json = buildJson(getContacts());
                wView.loadUrl("javascript:show('" + json + "')");
            } catch (Exception e) {
                System.out.println("设置数据失败" + e);
            }
        }
    });

}
```

嘿嘿，接下来运行下程序，神奇的发现，我们N5的手机联系人可以读取到了~



同理，之前第一个示例也可以这样解决~

本节小结：

本节跟大家走了一趟Android 4.4后WebView要注意的事项，以及一些对上一节中N5问题 的解决~相信会给大家在实际开发中使用WebView带来便利~谢谢

← 7.5.2 WebView和JavaScrip交互基础 7.5.4 WebView文件下载 →

在线实例

- HTML 实例
- CSS 实例
- JavaScript 实例
- Ajax 实例
- jQuery 实例
- XML 实例
- Java 实例

字符集&工具

- HTML 字符集设置
- HTML ASCII 字符集
- HTML ISO-8859-1
- HTML 实体符号
- HTML 拾色器
- JSON 格式化工具

最新更新

- CSS all 属性
- Px、Em 换算工具
- px,pt,em换算表
- px、em、rem区别...
- Viewport 模板
- 移动WEB前端开发...
- C语言– 打...

站点信息

- 意见反馈
- 免责声明
- 关于我们
- 文章归档

关注微信