eoe 特刊

第24期

eoe android 优亿开发

优亿市场 www.eoemarket.com

本期内容 wiki 地址

http://wiki.eoeandroid.com/%E7%AC%AC24%E6%9C%9F%EF%BC%9AAndroid_WebView

目录

前言

参与编写者简介

一、Android WebView 是什么

1.1 Android WebView 一些基本概念
1.2 Android WebView 组件的使用详解

二、Android WebView 入门

2.1 Android编写简单的WebView
2.2 WebView初探
2.3 WebView 再探

三、Android WebView 深入

3.1 Android 利用 WebView 实现在 js 中调用 android 代码
3.2 Android WebView 缓存
3.3 Android WebView 删除缓存
3.4 Android listview 中加入 WebView
3.5 Android 中 WebView 跟 JAVASCRIPT 中的交互
3.6 Android 的 WebView 与 ProgressDialog 结合
3.7 WebView 处理 404 错误

四、Android WebView CookBook

4.1 Android WebView 中点击打开默认的浏览器
4.2 追加一个图片到 Android WebView
4.3 WebView 中关于字符串编码问题
4.4 Android 的 WebView 研究
4.5 Android WebView 总结
4.6 Android 获取服务器中的 session 问题
4.7 WebView 设置实现两个手指缩放网页
4.8 WebView cookies 清理
4.9 Android WebView 使用 cmwap 无法联网解决办法
4.10 Android-判断 WebView 是否已经滚动到页面底端
4.11 Android 中 WebView 实现 Javascript 调用 Java 类方法

五、Android WebView SDK 参考

六、关于优亿开发者社区的一些新鲜玩意

前言

用千呼万唤始出来形容我们的特刊真是毫不为过，自打上期《寻找“冰淇淋三明治”的痕迹》之后已经阔别大家半年有余了，这次趁着六一儿童节到来之际，为亲爱的超龄儿童们送上一份特刊大礼！

在这半年中我们的行业出现了神奇的变化，各种操作系统一浪推一浪，但惟独我们的 Android 依然屹立于智能手机系统之林茁壮的生长着，傲视群雄！

在这半年里我们的优亿开发者社区也有了天翻地覆的变化，精彩内容层出不穷，技术牛人不断涌现，优亿开发者门户为开发者们量身定制的服务在喷薄而出！

在这半年里我们的用户也在不断的进步、发展着，昔日害羞的小菜鸟们现在已经化作雄鹰振翅在高高的天上，为着自己的梦想在不惧风雨、奋力前行着！

我们将继续努力，为各位开发者提供更全更好的服务，做中国最棒的 Android 开发者社区！

感谢大家对优亿开发者社区的支持！

By： 果子狸
2012 年 5 月 30 日

参与编写者简介

Kris 同学,现任社区超级版主,想必常在论坛里混的同学都不会陌生.本来应该他自己写这个简介的,但人家说了“人怕出名猪怕壮”...好吧,这页翻过去.Kris 同学参与了本次特刊的外文翻译部分,同时也对本期的校稿做出了贡献.两句话,感谢 Kris,向 Kris 学习!

猜猜看.这货在下面的照片里是哪一个? o(∩_∩)o

一、Android WebView 是什么

1.1 Android WebView 一些基本概念

在 Android 手机中内置了一款高性能 webkit 内核浏览器,在 SDK 中封装为一个叫做 WebView 组件.

什么是 webkit

WebKit 是 Mac OS X v10.3 及以上版本所包含的软件框架(对 v10.2.7 及以上版本也可通过软件更新获取). 同时,WebKit 也是 Mac OS X 的 Safari 网页浏览器的基础.WebKit 是一个开源项目,主要由 KDE 的 KHTML 修改而来并且包含了一些来自苹果公司的一些组件.

传统上,WebKit 包含一个网页引擎 WebCore 和一个脚本引擎 JavaScriptCore,它们分别对应的是 KDE 的 KHTML 和 KJS.不过, 随着 JavaScript 引擎的独立性越来越强,现在 WebKit 和 WebCore 已经基本上混用不分(例如 Google Chrome 和 Maxthon 3 采用 V8 引擎,却仍然宣称自己是 WebKit 内核).

这里我们初步体验一下在 android 是使用 WebView 浏览网页,在 SDK 的 Dev Guide 中有一个 WebView 的简单例子.

在开发过程中应该注意几点:

1.AndroidManifest.xml 中必须使用许可"android.permission.INTERNET",否则会出 Web page not available 错误

2.如果访问的页面中有 Javascript,则 WebView 必须设置支持 Javascript.

WebView.getSettings().setJavaScriptEnabled(true);

3.如果页面中链接,如果希望点击链接继续在当前 browser 中响应,而不是新开 Android 的系统 browser 中响应该链接,必须覆盖 WebView 的 WebViewClient 对象.

```
mWebView.setWebViewClient(new WebViewClient(){
public boolean shouldOverrideUrlLoading(WebView view, String url) {
view.loadUrl(url);
return true;
}
});
```

4.如果不做任何处理,浏览网页,点击系统“Back”键,整个 Browser 会调用 finish()而结束自身,如果希望浏览的网页回退而不是推出浏览器,需要在当前 Activity 中处理并消费掉该 Back 事件.

```
public boolean onKeyDown(int keyCode, KeyEvent event) {
if ((keyCode == KeyEvent.KEYCODE_BACK) && mWebView.canGoBack()) {
mWebView.goBack();
return true;
}
return super.onKeyDown(keyCode, event);
}
```

下一步让我们来了解一下 android 中 WebView 是如何支持 javascripte 自定义对象的,在 w3c 标准中 js 有 window,history,document 等标准对象,同样我们可以在开发浏览器时自己定义我们的对象调用手机系统功能来处理,这样使用 js 就可以为所欲为了.

看一个实例:

view plain copy to clipboard print ?

```
public class WebViewDemo extends Activity {
private WebView mWebView;
private Handler mHandler = new Handler();
public void onCreate(Bundle icicle) {
super.onCreate(icicle);
setContentView(R.layout.WebViewdemo);
mWebView = (WebView) findViewById(R.id.WebView);
WebSettings webSettings = mWebView.getSettings();
webSettings.setJavaScriptEnabled(true);
mWebView.addJavascriptInterface(new Object() {
public void clickOnAndroid() {
mHandler.post(new Runnable() {
public void run() {
mWebView.loadUrl("javascript：wave()");
}
});
}
}, "demo");
mWebView.loadUrl("file：///android_asset/demo.html");
}
}
```

我们看 addJavascriptInterface(Object obj,String interfaceName)这个方法,该方法将一个 java 对象绑定到一个 javascript 对象中,javascript 对象名就是 interfaceName(demo),作用域是 Global.这样初始化 WebView 后,在 WebView 加载的页面中就可以直接通过 javascript:window.demo 访问到绑定的 java 对象了.来看看在 html 中是怎样调用的.

```
<html>
<mce：script language="javascript"><！--
function wave() {
document.getElementById("droid").src="android_waving.png";
}
// --></mce：script>
<body>
<a onClick="window.demo.clickOnAndroid()">
<img id="droid" src="android_normal.png" mce_src="android_normal.png"/><br>
Click me！
</a>
</body>
</html>
```

这样在 javascript 中就可以调用 java 对象的 clickOnAndroid()方法了,同样我们可以在此对象中定义很多方法(比 如发短信,调用联系人列表等手机系统功能.),这里 wave()方法是 java 中调用 javascript 的例子.

这里还有几个知识点：

1) 为了让 WebView 从 apk 文件中加载 assets,Android SDK 提供了一个 schema,前缀为 "file:///android_asset/".WebView 遇到这样的 schema,就去当前包中的 assets 目录中找内容.如上面的 "file:///android_asset/demo.html"

2)addJavascriptInterface 方法中要绑定的 Java 对象及方法要运行另外的线程中,不能运行在构造他的线程中,这也是使用 Handler 的目的.

1.2 Android WebView 组件的使用详解

网络内容

1、LoadUrl 直接显示网页内容(单独显示网络图片)

2、LoadData 显示中文网页内容(含空格的处理)

APK 包内文件

1、LoadUrl 显示 APK 中 Html 和图片文件

2、LoadData(loadDataWithBaseURL)显示 APK 中图片和文字混合的 Html 内容

res/layout/main.xml

Xml 代码

```
< ？xml version="1.0" encoding="utf-8"？ >
  < LINEARLAYOUT android ： layout_height="fill_parent" android ： layout_width="fill_parent" android ：
orientation="vertical" xmlns：android="http：//schemas.android.com/apk/res/android">
   < WebView android ： layout_height="fill_parent" android ： layout_width="fill_parent" android ：
id="@+id/WebView" />
< /LINEARLAYOUT>
```

```
< ？xml version="1.0" encoding="utf-8"？ >
  < LINEARLAYOUT android ： layout_height="fill_parent" android ： layout_width="fill_parent" android ：
orientation="vertical" xmlns：android="http：//schemas.android.com/apk/res/android">
   < WebView android ： layout_height="fill_parent" android ： layout_width="fill_parent" android ：
id="@+id/WebView" />
< /LINEARLAYOUT>
```

Example_WebView.java

Java 代码

```
package cn.coolworks;
import java.net.URLEncoder;
import android.app.Activity;
import android.os.Bundle;
import android.webkit.WebView;
public class Example_WebView extends Activity {
WebView WebView;
final String mimeType = "text/html";
final String encoding = "utf-8";
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
WebView = (WebView) findViewById(R.id.WebView);
WebView.getSettings().setJavaScriptEnabled(true);
//
//webHtml();
```

```
//
//webImage();
//
//localHtmlZh();
//
//localHtmlBlankSpace();
//
//localHtml();
//
// localImage();
//
localHtmlImage();
}
/**
* 直接网页显示
*/
private void webHtml() {
try {
WebView.loadUrl("http：//www.google.com");
} catch (Exception ex) {
ex.printStackTrace();
}
}
/**
* 直接网络图片显示
*/
private void webImage() {
try {
WebView
.loadUrl("http：//www.gstatic.com/codesite/ph/images/code_small.png");
} catch (Exception ex) {
ex.printStackTrace();
}
}
/**
* 中文显示
*/
private void localHtmlZh() {
try {
String data = "测试含有 中文的 Html 数据";
// utf-8 编码处理(在 SDK1.5 模拟器和真实设备上都将出现乱码,SDK1.6 上能正常显示)
//WebView.loadData(data, mimeType, encoding);
// 对数据进行编码处理(SDK1.5 版本)
WebView.loadData(URLEncoder.encode(data, encoding), mimeType,
encoding);
} catch (Exception ex) {
ex.printStackTrace();
}
```

```
}
/**
* 中文显示(空格的处理)
*/
private void localHtmlBlankSpace() {
try {
String data = " 测试含有空格的 Html 数据 ";
// 不对空格做处理
WebView.loadData(URLEncoder.encode(data, encoding), mimeType,
encoding);
//WebView.loadData(data, mimeType, encoding);
// 对空格做处理(在 SDK1.5 版本中)
WebView.loadData(URLEncoder.encode(data, encoding).replaceAll(
"\+", " "), mimeType, encoding);
} catch (Exception ex) {
ex.printStackTrace();
}
}
/**
* 显示本地图片文件
*/
private void localImage() {
try {
// 本地文件处理(如果文件名中有空格需要用+来替代)
WebView.loadUrl("file：///android_asset/icon.png");
} catch (Exception ex) {
ex.printStackTrace();
}
}
/**
* 显示本地网页文件
*/
private void localHtml() {
try {
// 本地文件处理(如果文件名中有空格需要用+来替代)
WebView.loadUrl("file：///android_asset/test.html");
} catch (Exception ex) {
ex.printStackTrace();
}
}
/**
* 显示本地图片和文字混合的 Html 内容
*/
private void localHtmlImage() {
try {
String data = "测试本地图片和文字混合显示,这是 APK 里的图片";
// SDK1.5 本地文件处理(不能显示图片)
// WebView.loadData(URLEncoder.encode(data, encoding), mimeType,
```

```
// encoding);
// SDK1.6 及以后版本
// WebView.loadData(data, mimeType, encoding);
//  本地文件处理(能显示图片)
WebView.loadDataWithBaseURL("about：blank", data, mimeType,
encoding, "");
} catch (Exception ex) {
ex.printStackTrace();
}
}
}
```

二、Android WebView 入门

本章主要通过简单的例子演示如何使用 WebView.

2.1 Android 编写简单的 WebView

在 android 界面中如果使用 WebView,往往可以复用服务器端的内容.先写个简单的 WebView 实现.
非常简单,直接在 adt 默认项目上加的.加了个 WebView,访问我的博客首页.

首先,要记着在 AndroidManifest.xml 中加入访问 internet 的权限,否则页面无法访问.
<uses-permission android：name="android.permission.INTERNET"></uses-permission>
然后,布局文件加入：
<WebView android： id="@+id/WebView" android： layout_width="fill_parent" android： layout_height="0dip" android：layout_weight="1" />
最后,在代码中设置 WebView 的属性：

```
this.WebView=(WebView) this.findViewById(R.id.WebView);
this.WebView.getSettings().setSupportZoom(false);
this.WebView.getSettings().setJavaScriptCanOpenWindowsAutomatically(true);
this.WebView.loadUrl("http://marshal.easymorse.com/");
```

2.2 WebView 初探

Android SDK 的 Dev Guide 中有一个 WebView 的简单例子 ,寥寥几行代码就可以做一个自己的浏览器.
在实验时,有如下几个注意事项：
1)AndroidManifest.xml 中必须使用许可"android.permission.INTERNET",否则会出 Web page not available 错误;

2)如果访问的页面中有 Javascript,则 WebView 必须设置支持 Javascript:

```
WebView.getSettings().setJavaScriptEnabled(true);
```

否则显示空白页面.

3)如果页面中链接,如果希望点击链接继续在当前 browser 中响应,而不是新开 Android 的系统 browser 中响应该链接,必须覆盖 WebView 的 WebViewClient 对象：

```
mWebView.setWebViewClient(new WebViewClient(){
public boolean shouldOverrideUrlLoading(WebView view, String url) {
view.loadUrl(url);
return true;
}
});
```

上述方法告诉系统由我这个 WebViewClient 处理这个 Intent,我来加载 URL.点击一个链接的 Intent 是向上冒泡的,shouldOverrideUrlLoading 方法 return true 表示我加载后这个 Intent 就消费了,不再向上冒泡了.

4)如果不做任何处理,在显示你的 Brower UI 时,点击系统“Back”键,整个 Browser 会作为一个整体“Back"到其他 Activity 中,而不是希望的在 Browser 的历史页面中 Back.如果希望实现在历史页面中 Back,需要在当前 Activity 中处理并消费掉该 Back 事件：

```
public boolean onKeyDown(int keyCode, KeyEvent event) {
if ((keyCode == KeyEvent.KEYCODE_BACK) && mWebView.canGoBack()) {
mWebView.goBack();
return true;
}
return super.onKeyDown(keyCode, event);
}
```

2.3 WebView 再探

从 WebView 初探 了解到 WebView 的强大,.听说 WebView 对 Javascript 的支持也很强,想从网上找些例子,还很难找,最终从 google 老家找了一个 Java 和 Javascript 互调的例子 ,当时看了,下巴“吮当”就掉在地上了,太强了！这样也行？

整个 Eclipse ADT 工程例子中都有,这里重点分析一下代码：

```
public class WebViewDemo extends Activity {
private WebView mWebView;
private Handler mHandler = new Handler();
public void onCreate(Bundle icicle) {
super.onCreate(icicle);
setContentView(R.layout.WebViewdemo);
mWebView = (WebView) findViewById(R.id.WebView);
WebSettings webSettings = mWebView.getSettings();
webSettings.setJavaScriptEnabled(true);
mWebView.addJavascriptInterface(new Object() {
public void clickOnAndroid() {
mHandler.post(new Runnable() {
public void run() {
mWebView.loadUrl("javascript:wave()");
}
});
```

```
}
}, "demo");
mWebView.loadUrl("file:///android_asset/demo.html");
}
}
```

这里的重点是 addJavascriptInterface(Object obj,String interfaceName)方法,该方法将一个 java 对象绑定到一个 javascript 对象中,javascript 对象名就是 interfaceName,作用域是 Global.这样初始化 WebView 后,在 WebView 加载的页面中就可以直接通过 javascript:window.demo 访问到绑定的 java 对象了.来看看在 html 中是怎样调用的：

```
<html>
<script language="javascript">
function wave() {
document.getElementById("droid").src="android_waving.png";
}
</script>
<body>
<a onClick="window.demo.clickOnAndroid()">
<img id="droid" src="android_normal.png"/><br>
Click me!
</a>
</body>
</html>
```

这样在 javascript 中就可以调用 java 对象的 clickOnAndroid()方法了,wave()方法是 java 中调用 javascript 的例子.

这里还有几个知识点：

1) 为了让 WebView 从 apk 文件中加载 assets,Android SDK 提供了一个 schema, 前缀为 "file:///android_asset/".WebView 遇到这样的 schema, 就去当前包中的 assets 目录中找内容.如上面的 "file:///android_asset/demo.html"

2)addJavascriptInterface 方法中要绑定的 Java 对象及方法要运行另外的线程中,不能运行在构造他的线程中,这也是使用 Handler 的目的.

三、Android WebView 深入

本章将讲解深入一些的 **WebView** 技术,比如和 **js** 交互,和本地文件,缓存,**WebView** 控件定制等～

3.1 Android 利用 WebView 实现在 js 中调用 android 代码

在 java 或 android 中,接口占有很大比重,做程序员当然要对接口了解了,哈哈！android 和 js 看似牛马不相及,但是由于 WebView 的连接在 js 中就能够调用 android 写的代码实现 android 的功能！android 和 js 就像 java 和才 c/c++靠 jni 连接似的,二者也是靠接口连接.不废话了,下面就以一个简单的案例讲述一下在 js 中调用 android 代码实现 android 功能！

1. 首先简述 WebView、WebViewClient、WebChromeClient 之间的区别：

在 WebView 的设计中,不是什么事都要 WebView 类干的,有些杂事是分给其他人的,这样 WebView 专心干好自己的解析、渲染工作就行了.WebViewClient 就是帮助 WebView 处理各种通知、请求事件等,WebChromeClient 是辅助 WebView 处理 Javascript 的对话框,网站图标,网站 title.

2. 功能实现：利用 android 中的 WebView 加载一个 html 网页,在 html 网页中定义一个按钮,点击按钮弹出一个 toast.

3. 实现步骤：

①定义一个接口类,将上下文对象传进去,在接口类中定义要在 js 中实现的方法.

②在 assets 资源包下定义一个 html 文件,在文件中定义一个 button.button 的点击事件定义为一个 js 函数.

③在 xml 中定义一个 WebView 组件,在活动类中获取 WebView 并对 WebView 参数进行设置,此处特别注意要设置 WebView 支持 js 且将定义的 js 接口类添加到 WebView 中去,此后在 js 中就可以利用该接口类中定义的函数了.即：

```
myWebView.getSettings().setJavaScriptEnabled(true);
myWebView.addJavascriptInterface(new JavaScriptinterface(this),"android");
```

④利用 WebView 加载本地 html 文件的方法是：myWebView.loadData(htmlText, "text/html", "utf-8");此处的 htmltext 是以字符串的方式读取 assets 报下 html 中的内容.具体代码见源码！哈哈.

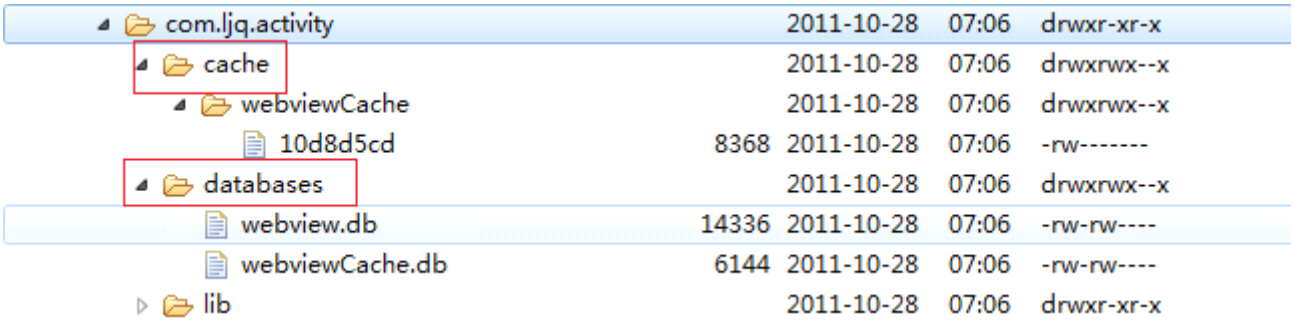
4. 实现利用返回键返回到上一页：

设置 WebView 的按键监听,监听到期返回键并判断网页是否能够返回,利用 WebView 的 goBack()返回到上一页.

本节源代码请前往 http://www.eoeandroid.com/thread-159155-1-1.html 下载

3.2 Android WebView 缓存

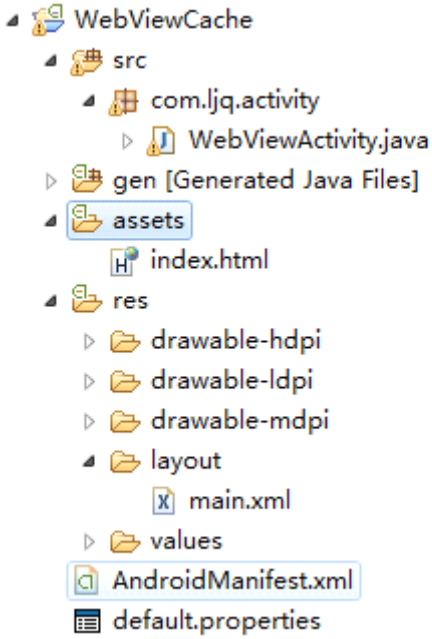
在项目中经常会使用到 WebView 控件,当加载 html 页面时,会在/data/data/应用 package 目录下生成 database 与 cache 两个文件夹如下图如示:



请求的 url 记录是保存在 WebViewCache.db,而 url 的内容是保存在 WebViewCache 文件夹下.

为了便于理解,接下来模拟一个案例,定义一个 html 文件,在里面显示一张图片,用 WebView 加载出来,然后再试着从缓存里把这张图片读取出来并显示.

第一步:新建一个 Android 工程命名为 WebViewCache.目录结构如下:



第二步:在 assets 目录下新建一个 html 文件,命名为 index.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>WebViewCacheDemo</title>
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
<meta http-equiv="description" content="this is my page">
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
</head>
<body>
<img
src="http://img04.taobaocdn.com/imgextra/i4/608825099/T2nGXBXXpaXXXXXXX_!!608825099.jpg_310x310.jpg"/>
</body>
</html>
```

第三步:修改 main.xml 布局文件,一个 WebView 控件一个 Button(点击加载缓存图片用),代码如下:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:orientation="vertical"
android:layout_width="fill_parent" android:layout_height="fill_parent">
<WebView android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:id="@+id/WebView"/>
<Button android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center_horizontal"
android:text="从缓存读取图片"
android:id="@+id/button"/></LinearLayout>
```

第四步:修改主核心程序 WebViewCacheDemo.java,这里我只加载了 index.html 文件,按钮事件暂时没写,代码如下:

```
package com.ljq.activity;
import java.io.File;
import java.io.FileInputStream;
import android.app.Activity;
import android.app.Dialog;
import android.app.AlertDialog.Builder;
import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebView;
import android.widget.Button;
import android.widget.ImageView;
public class WebViewActivity extends Activity {
private WebView WebView;
private static final String url="file:///android_asset/index.html";
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
WebView=(WebView)findViewById(R.id.WebView);
WebView.loadUrl(url);
}
}
```

第五步:在 AndroidMainifest.xml 文件中加访问网络的权限:

```
<uses-permission android:name="android.permission.INTERNET" />
```

运行效果如下：

此时我们在 WebViewCache.db 里的 cache.table 里多了一条记录如下图所示:

_id	url	filepath	lastmodify
1	http://img04.taobaocdn.com/imgextra/i4/608825099/T2nGXBXXpaXXXXXXXX_!!608825099.jpg_310x310.jpg	10d8d5cd	Thu, 10 Feb 2011 01:41:03 GMT

在 cache/WebViewCache/目录下多了一个 10d8d5cd 文件,刚好和 cache.table 里的 filepath,我们可以断定这个文件就是我们从网上拽下来的图片:

为了验证猜想,我给 Button 增加事件响应,就是弹出 Dialog,里面加载缓存的图片,完整代码如下:

```
package com.ljq.activity;import java.io.File;
import java.io.FileInputStream;
import android.app.Activity;
import android.app.Dialog;
import android.app.AlertDialog.Builder;
import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;import android.os.Bundle;
import android.view.View;
```

```
import android.webkit.WebView;
import android.widget.Button;
import android.widget.ImageView;
public class WebViewActivity extends Activity {
private WebView WebView;
private static final String url="file:///android_asset/index.html";
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
WebView=(WebView)findViewById(R.id.WebView);
WebView.loadUrl(url);
//点击按钮时弹出对话框
Button button=(Button)findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
public void onClick(View v) {
ImageView imageView=new ImageView(WebViewActivity.this);
imageView.setImageBitmap(getPictureFromCache());
Builder builder=new android.app.AlertDialog.Builder(WebViewActivity.this);
//设置对话框的图标
builder.setTitle("从缓存查看图片");
builder.setView(imageView);
//退出按钮
builder.setPositiveButton("退　出", new OnClickListener(){
public void onClick(DialogInterface dialog, int which) {
//关闭 alert 对话框架
dialog.cancel();
}
});
builder.create().show();
}
});
}
/**
*  从缓存获取图片
*
* @return
*/
private Bitmap getPictureFromCache(){
Bitmap bitmap=null;
try {
//这里写死,在实际开发项目中要灵活使用
File file=new File(getCacheDir()+"/WebViewCache/10d8d5cd");
FileInputStream inStream=new FileInputStream(file);
bitmap=BitmapFactory.decodeStream(inStream);
} catch (Exception e) {
e.printStackTrace();
}
```

```
return bitmap;
}
}
```

第六步:再次运行工程,点击 button 按钮,效果如下图所示:

3.3 Android WebView 删除缓存

删除保存于手机上的缓存:

```
// clear the cache before time numDays
private int clearCacheFolder(File dir, long numDays) {
int deletedFiles = 0;
if (dir!= null && dir.isDirectory()) {
try {
for (File child:dir.listFiles()) {
if (child.isDirectory()) {
deletedFiles += clearCacheFolder(child, numDays);
}
if (child.lastModified() < numDays) {
if (child.delete()) {
```

```
deletedFiles++;
}
}
}
} catch(Exception e) {
e.printStackTrace();
}
}
return deletedFiles;
}
```

打开关闭使用缓存

```
//优先使用缓存：
WebView.getSettings().setCacheMode(WebSettings.LOAD_CACHE_ELSE_NETWORK);
//不使用缓存：
WebView.getSettings().setCacheMode(WebSettings.LOAD_NO_CACHE);
```

在退出应用的时候加上如下代码

```
File file = CacheManager.getCacheFileBaseDir();
if (file != null && file.exists() && file.isDirectory()) {
for (File item : file.listFiles()) {
item.delete();
}
file.delete();
}
context.deleteDatabase("WebView.db");
context.deleteDatabase("WebViewCache.db");
```

3.4 Android listview 中加入 WebView

想在 listview 的 item 中加入一个 WebView 页面,只要一个 item 和 WebView 就可以,不知道怎么在 item 中加入 WebView,需要的就是类似上图的效果,在 listview 中加入 WebView,实现下拉刷新 WebView 的效果.知道的朋友说下,谢谢.

解决办法如下.

```
List list = new ArrayList();
list.add("http://www.....");
ItemAdapter ia = new ItemAdapter(getApplicationContext(), list);
setListAdapter(ia);
.....
public class ItemAdapter extends BaseAdapter{
private Handler handler;
private List list;
private Context context;
private ViewHolder vh = new ViewHolder();
public ItemAdapter(Context con,List list) {
// TODO Auto-generated constructor stub
this.list = list;
this.context = con;
}
private class ViewHolder {
WebView web;
}
.....
@Override
public View getView(int position, View convertView, ViewGroup parent) {
// TODO Auto-generated method stub
View view = LayoutInflater.from(context).inflate(
R.layout.webpage, null);//写一个 webpage 的布局 xml,WebView id 为 WebView
vh.web = (WebView) view.findViewById(R.id.WebView);
setWebpageView(list.get(position).toString(),vh.web);
return view;
}
public void setWebpageView(final String url,WebView WebView){
loadurl(WebView, url);
handler = new Handler(){
public void handleMessage(Message msg){//定义一个 Handler,用于处理线程与 UI 间通讯
if (!Thread.currentThread().isInterrupted()){
switch (msg.what){
case 0:
pd.show();//显示进度对话框
break;
case 1:
pd.hide();
break;
}
}
super.handleMessage(msg);
```

```
}
};
}
public void loadurl(final WebView view,final String url){
new Thread(){
public void run(){
handler.sendEmptyMessage(0);
view.loadUrl(url);//载入网页
}
}.start();
}
}
```

本文来自：http://www.eoeandroid.com/thread-163566-1-1.html

3.5 Android 中 WebView 跟 JAVASCRIPT 中的交互

在 android 的应用程序中,可以直接调用 WebView 中的 javascript 代码,而 WebView 中的 javascript 代码,也可以去调用 ANDROID 应用程序(也就是 JAVA 部分的代码).下面举例说明之:

1、JAVASCRIPT 脚本调用 android 程序
要在 WebView 中,调用 addJavascriptInterface(OBJ,interfacename)
其中,obj 为和 javascript 通信的应用程序,interfacename 为提供给 JAVASCRIPT 调用的名称,设置如下:

```
WebView WebView = new WebView(this);
WebView.getSettings().setJavaScriptEnabled(true);
WebView.loadUrl(getIntent().getCharSequenceExtra("url").toString());
//设定 JavaScript 脚本代码的界面名称是”android”
WebView.addJavascriptInterface(this, "android");
```

其中 WebView 调用的 HTML 页中,JS 如下:

```
<script type="text/javascript">
function ok() {
android.js(document.forms[0].elements[0].value, document.forms[0].elements[1].value);
 }
```

而这个 android.js 在哪呢?那是在应用程序中的

```
//JavaScript 脚本代码可以调用的函数 js()处理
public void js(String action, String uri) {
...../
}
```

这个 JS 中就是处理 JAVASCRIPT 发送过来的请求了.

2、下面的例子,当 WebView 网页中输入后,点提交按钮,会跟 ANDROID 的应用程序进行交互

```
WebView WebView = new WebView(this);
WebView.getSettings().setJavaScriptEnabled(true);
WebView.setWebChromeClient(new MyWebChromeClient());
WebView.loadUrl(getIntent().getCharSequenceExtra("url").toString());
//onJsAlert()函数接收到来自 HTML 网页的 alert()警告信息
public boolean onJsAlert(WebView view, String url, String message, JsResult result) {
if (message.length() != 0) {
AlertDialog.Builder builder = new AlertDialog.Builder(JExample02.this);
builder.setTitle("From JavaScript").setMessage(message).show();
result.cancel();
return true;
}
return false;
}
```

而 HTML 页中的 JS 事件为:

```
<input type="button" value="alert" onclick="alert(document.forms[0].elements[0].value)">
```

特别提示下,在自定义的 MyWebChromeClient()) 中,除了可以重写 onJSAlert 外,还可以重写 onJsPrompt,onJsConfirm 等,可以参考.

http://618119.com/archives/2010/12/20/199.html

3、下面这个例子,先显示第一张图片,点一点后,再显示第 2 张图片

HTML JS 中:

```
<script language="javascript">
function changeImage02() {
document.getElementById("image").src="navy02.jpg";
}
function changeImage01() {
document.getElementById("image").src="navy01.jpg";
}
</script>
</head>
<body>
<a onClick="window.demo.onClick()">
<img id="image" src="navy01.jpg"/></a>
</body>
```

当点<a onClick="window.demo.onClick()">后,调用 ANDROID 应用程序中的处理部分,看程序:

```
WebView.addJavascriptInterface(new JSInterface(),"demo");
public final class JSInterface {
//JavaScript 脚本代码可以调用的函数 onClick()处理
public void onClick() {
handler.post(new Runnable() {
public void run() {
if (flag == 0) {
```

```
flag = 1;
WebView.loadUrl("javascript:changeImage02()");
} else {
flag = 0;
WebView.loadUrl("javascript:changeImage01()");
}
}
});
}
}
```

可以看到,ANDROID 中,通过 WebView.loadUrl 去调用 HTML 页面中的 JS

3.6 Android 的 WebView 与 ProgressDialog 结合

WebView 组件支持直接加载网页,可以将其视为一个浏览器,要实现该功能,具体步骤如下：

WebView.xml

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent">
<WebView android:id="@+id/WebView"
android:layout_width="fill_parent"
android:layout_height="fill_parent"/>
</LinearLayout>
```

WebViewActivity.java

```
public class WebViewActivity extends Activity{
private WebView WebView;
private AlertDialog alertDialog;
private ProgressDialog progressBar;
jQuery datatables 使用
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.WebView);
//加载 WebView
initWebView();
}
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
if(keyCode == KeyEvent.KEYCODE_BACK && WebView.canGoBack()){
WebView.goBack();
return true;
```

```
}
return super.onKeyDown(keyCode, event);
}
class MyWebViewClient extends WebViewClient{
@Override
public boolean shouldOverrideUrlLoading(WebView view, String url) {
view.loadUrl(url);
return true;
}
@Override
public void onPageFinished(WebView view, String url) {
if(progressBar.isShowing()){
progressBar.dismiss();
}
}
@Override
public void onReceivedError(WebView view, int errorCode,
String description, String failingUrl) {
Toast.makeText(WebViewActivity.this, "网页加载出错！", Toast.LENGTH_LONG);
alertDialog.setTitle("ERROR");
alertDialog.setMessage(description);
alertDialog.setButton("OK", new DialogInterface.OnClickListener(){
@Override
public void onClick(DialogInterface dialog, int which) {
// TODO Auto-generated method stub
}
});
alertDialog.show();
}
}
protected void initWebView(){
//设计进度条
progressBar = ProgressDialog.show(WebViewActivity.this, null, "正在进入网页,请稍后…");
//获得 WebView 组件
WebView = (WebView) this.findViewById(R.id.WebView);
WebView.getSettings().setJavaScriptEnabled(true);
WebView.loadUrl("http://www.baidu.com");
alertDialog = new AlertDialog.Builder(this).create();
//设置视图客户端
WebView.setWebViewClient(new MyWebViewClient());
}
}
```

最后,需要在**Manifest.xml 中添加访问互联网的权限,否则不能显示：

```
<uses-permission android:name="android.permission.INTERNET"/>
```

3.7 WebView 处理 404 错误

```
import java.io.IOException;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpHead;
import org.apache.http.impl.client.DefaultHttpClient;
import android.app.Activity;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.view.KeyEvent;
import android.webkit.WebView;
import android.webkit.WebViewClient;
public class WebView_404 extends Activity {
private final String HOMEPAGE = "http://10.0.0.95/index.html";//请求的网站的主页
private WebView web;
private Handler handler = new Handler() {
@Override
public void handleMessage(Message msg) {
if(msg.what==404) {//主页不存在
//载入本地 assets 文件夹下面的错误提示页面 404.html
web.loadUrl("file:///android_asset/404.html");
} else {
web.loadUrl(HOMEPAGE);
}
}
};
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.WebView_404);
web = (WebView) findViewById(R.id.WebView01);
web.getSettings().setJavaScriptEnabled(true);
web.setWebViewClient(new WebViewClient() {
@Override
public boolean shouldOverrideUrlLoading(WebView view, String url) {
if(url.startsWith("http://") && getRespStatus(url)==404) {
view.stopLoading();
//载入本地 assets 文件夹下面的错误提示页面 404.html
view.loadUrl("file:///android_asset/404.html");
} else {
view.loadUrl(url);
}
return true;
}
});
```

```
new Thread(new Runnable() {
@Override
public void run() {
Message msg = new Message();
//此处判断主页是否存在,因为主页是通过 loadUrl 加载的,
//此时不会执行 shouldOverrideUrlLoading 进行页面是否存在的判断
//进入主页后,点主页里面的链接,链接到其他页面就一定会执行 shouldOverrideUrlLoading 方法了
if(getRespStatus(HOMEPAGE)==404) {
msg.what = 404;
}
handler.sendMessage(msg);
}
}).start();
}
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
if(keyCode==KeyEvent.KEYCODE_BACK && web.canGoBack()) {
web.goBack();
return false;
}
return super.onKeyDown(keyCode, event);
}
private int getRespStatus(String url) {
int status = -1;
try {
HttpHead head = new HttpHead(url);
HttpClient client = new DefaultHttpClient();
HttpResponse resp = client.execute(head);
status = resp.getStatusLine().getStatusCode();
} catch (IOException e) {}
return status;
}
}
```

四、Android WebView CookBook

这里收集一些很实用的小技巧～

4.1 Android WebView 中点击打开默认的浏览器

论坛帖子：http://www.eoeandroid.com/thread-174926-1-1.html

我遇到一个简单的问题,我在 WebView 中加载了一个外部的 url ,现在需要的是在页面加载后,用户点击里面的连接能像默认的浏览器一样在我的 WebView 中打开连接 ,但问题是现在它会打开 android 的默认浏览器去加载页面？？？我已经启用了 javascript ,但是没用,是否我忘记了什么？

一楼：

如果你不想要开 android 默认的浏览器,那么我就得自己去截取在你 WebView 中的点击事件 .

你可以通过 WebView 的 WebViewClient 来监控事件.你现在需要的是 shouldOverrideUrlLoading()这个方法,它会允许在一个特定的 url 被选中的时候来执行你自己的一个动作.

在 sdk 的 samples 中有一个示例.代码如下：

```
private class HelloWebViewClient extends WebViewClient {
@Override
public boolean shouldOverrideUrlLoading(WebView view, String url) {
view.loadUrl(url);
return true;
}
}
```

二楼：

在某些情况下,你或许也要生写 onLoadResource.比如说重定向时,因为它不会触发你的加载方法.我这种情况下,我尝试如下：

```
@Override
public void onLoadResource(WebView view, String url)
{
if (url.equals("http://redirectexample.com"))
{
//do your own thing here
}
else
{
super.onLoadResource(view, url);
}
}
```

4.2 追加一个图片到 Android WebView

我正在尝试用 javascript 来向 WebView 中追加一个图片.我自己写的 html,并且包含了自己方法.
代码如下：

```
<script type="text/javascript">
function image(src) {
var img = document.createElement("IMG");
img.src = src;
document.getElementById('image').appendChild(img);
}
</script>
```

但是我发现我不能通过 myview.loadurl(javascript:image('line1.png'));来实现.所以我想试试别的方法.比如说能不能通过 loadurl()方法来实现,但是会报错.不过我也听说可以使用 addJavascriptInterface 来实现,但是好像比较复杂,但是也不确定行不行.不知道你们有没有好的解决办法 ？

解决办法：

```
import android.app.Activity;
import android.os.Bundle;
import android.os.Handler;
import android.webkit.WebView;
public class StackOverFlowActivity extends Activity {
private Handler mHandler = new Handler();
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
WebView view=(WebView)findViewById(R.id.WebView1);
view.getSettings().setJavaScriptEnabled(true);
view.loadUrl("file:///android_asset/index.html");
view.addJavascriptInterface(new MyJavaScriptInterface(), "Android");
}
final class MyJavaScriptInterface
{
public void ProcessJavaScript(final String scriptname, final String args)
{
mHandler.post(new Runnable()
{
public void run()
{
String url="file:///android_asset/img.jpg";
WebView.loadUrl("javascript:image(\""+url+"\")");
}
});
}
}
```

```
}
index.html:
<script type="text/javascript">
function getimage()
{
Android.ProcessJavaScript("image",null);
}
function image(src) {
var img = document.createElement("IMG");
img.src = src;
document.getElementById('image').appendChild(img);
}
</script>
and add the following,
<body onload="getimage()" >
```

4.3 WebView 中关于字符串编码问题

我从 web service 中获取到的响应如下：

```
<html><head>
<style type="text/css">
body{
color: #FFFFFF;
background-color: #000000;
}
</style>
</head>
<body><center><div>
Ricardo Viana Vargas ?
```

09-14 Ricardo Viana Vargas é especialista em gerenciamento de projetos, portfólio e riscos. Foi, nos últimos 15 anos, responsável por mais de 80 projetos de grande porte em diversos países, nas áreas de petróleo, energia, infraestrutura, telecomunica??es, informática e finan?as, com um portfólio de investimentos gerenciado superior a 18 bilh?es de dólares. Foi o primeiro voluntário latino-americano a ser eleito para exercer a fun??o de presidente do conselho diretor (Chairman) do Project Management Institute (PMI), maior organiza??o do mundo voltada para a administra??o de projetos, com cerca de 500 mil membros e profissionais certificados em 175 países. Ricardo Vargas escreveu dez livros sobre gerenciamento de projetos, publicados em português e inglês, com mais de 240 mil exemplares vendidos mundialmente. Recebeu em 2005 o prêmio PMI Distinguished Award pela sua contribui??o.. (PRMIA).

```
</div> </center>
</body>
</html>
```

怎么才能让 WebView 显示出正确的字符串呢？我在 WebView 中的代码如下：

```
System.out.println("strContent is :: " + strContent);
WebView wv = (WebView) findViewById(R.id.WebView_portugage);
```

```
wv.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);
wv.loadData(strContent, "text/html", "UTF-8");
```

1 楼：

(1)用 WebView.loadDataWithBaseURL,它的行为与 WebView.loadData 不同(应该说是更好)

(2)试试把 UTF-8 替换成其它的编码,如 US-ASCII.最好是确认你显示的字符中用的什么编码.

用户回馈：

1,用第一种办法,是可以行.

2. 我用了 wv.loadData(object1.getString("post_content").toString(), "text/html", "US-ASCII");,还是一样的问题.(此方法无效)

2 楼：

我是在 WebView.loadData 中设置 MIME-type 为"text/html; charset=utf-8"来解决的.

这种方法并不是对所有的设备都是有效的,猜测可能与浏览器的内核版本(webkit)有关.但是 loadDataWithBaseURL 在所有的设备(已测试)都是有效的.

4.4 Android 的 WebView 研究

最近做的项目大量用到了 WebView,用网页来布局. Android 的 WebView 是基于 webkit 内核,不过他的运行效果和 firefox 上一模一样,所以写的时候都是先用 firefox 测试,测试 OK 了再放到程序里面看效果,基本上不会有什么问题.其实 android 的 WebView 跟 iphone 的 WebView 差不多, iphone 上的 WebView 比 android 上的强大多了.

谈一下研究 WebView 的一些成果：

1、加载资源的速度不慢,但是资源多了,就很慢.图片、 css 、 js 、 html 这些资源每个大概需要 10-200ms,一般都是 30ms 就 ok 了.如果一个页面上的资源很多,就很浪费时间.

2、Js 和 css 的执行速度.开始的时候,我的页面都是用 js 生成 DOM,添加样式等也用 js 添加.后来发现,加载一个页面居然要 5-6 秒.然后我就怀疑是不是 js 的执行效率不高,然后就把能用 css 的地方都用 css,能直接写到 html 上的就不用 js 动态生成.结果,速度并没有多大的提升,最多提升了 1 秒.看来, Js 的执行速度虽然比不上 css,但是还不至于慢到那种程度.那会是什么原因使得页面加载速度这么慢？经过仔细的排查,最终发现,是因为我用了 jQuery 框架.

WebView 加载页面的顺序是这样的：先加载 html,然后从里面解析出 css 、 js 文件和页面上写死的图片资源进行加载,如果 webkit 的缓存里面有,就不加载.加载完这些资源之后,就进行 css 的渲染和 js 的执行. Css 的渲染一般不需要很长时间,几十毫秒就 ok.关键是 js 的执行,如果用了 jQuery,则执行起来需要 5-6 秒.而在这段时间,如果不在 WebView 里设置背景,网页部分是白色的,很难看.这是一个很糟糕的用户体验.所以如果用网页布局程序,最好别用很大的 js 框架.

3、网页和 Java 之间的互调.这个功能是 iphone 里面就有的,网上也有很多资料,可以告诉我们怎么做,这些都是很简单、很基本的.我研究了一段时间,总结一下：

①Java 调用 js 里面的函数,速度并不令人满意,大概一次一两百毫秒吧,如果要做交互性很强的事情,这种速度会让人疯掉的.而反过来就不一样了, js 去调 java 的方法,速度很快,基本上 40-50 毫秒一次.所以尽量用 js 调用 java 方法,而不是 java 去调用 js 函数.

②Java 调用 js 的函数,没有返回值,而 Js 调用 java 方法,可以有返回值.返回值可以是字符串,也可以是对象.如果是字符串,有个很讨厌的问题,第 3 点我会讲的.如果是对象,这个对象会被转换为 js 的对象,直接可以访问里面的方法.但是我不推荐 java 返回给 js 的是对象,除非是必须.因为 js 收到 java 返回的对象,会产生一些交换对象,而如果这些对象的数量增加到了 500 或 600 以上,程序就会出问题.所以尽量返回基本数据类型或者字符串.

③Js 调用 java 的方法,返回值如果是字符串,你会发现这个字符串是 native 的,不能对它进行一些修改操

作,比如想对它 substr ,取不到.怎么解决呢？转成 locale 的.使用 toLocaleString() 函数就可以了.不过这个函数的速度并不快,转化的字符串如果很多,将会很耗费时间.

4、网页上拖动元素.网页上有一个 div ,想要拖动它到另外一个地方,怎么做？如果用 PC 上的网页做法,监听 onmousedown 、 onmousemove 和 onmouseup 就可以了.但是在手机上,事件模型就不一样了.在网页上点击,拖动,然后释放,手离开屏幕的时候, WebView 才会触发 onmousedown 、 onmousemove 、 onmouseup 事件.所以,要想拖动,不能这么做.这个问题困扰我很长时间,后来发现 iphone 上的做法,才解决了. Iphone 上的 WebView 有专为触摸屏设计的事件 ontouchstart 、 ontouchmove 、 ontouchend ,这几个事件的响应是实时的,就能解决拖动的问题了.

5、一些小问题. WebView 里面的网页,如果有 input ,需要输入,但是点上去却没反应,输入法不出来.这种情况是因为 WebView 没有获取焦点.需要在 java 里面给 WebView 设置一下 requestFocus() 就行了.

6、Android 上的 WebView 和 iphone 的 WebView 区别.目前为止,我发现的区别有这么几个：

①Android 上, WebView 不支持多点触控,没有 ongesture 系列事件,而 iphone 上有.

②Android 上的 WebView 不支持透明, iphone 上可以

4.5 Android WebView 总结

1、添加权限：AndroidManifest.xml 中必须使用许可"android.permission.INTERNET",否则会出 Web page not available 错误.

2、在要 Activity 中生成一个 WebView 组件：WebView WebView = new WebView(this);

3、设置 WebView 基本信息：

如果访问的页面中有 Javascript,则 WebView 必须设置支持 Javascript.

WebView.getSettings().setJavaScriptEnabled(true);

触摸焦点起作用

r equestFocus();

取消滚动条

this.setScrollBarStyle(SCROLLBARS_OUTSIDE_OVERLAY);

4、设置 WevView 要显示的网页：

互联网用：WebView.loadUrl("http://www.google.com");

本地文件用：WebView.loadUrl("file:///android_asset/XX.html"); 本地文件存放在：assets 文件中

5、如果希望点击链接由自己处理,而不是新开 Android 的系统 browser 中响应该链接.

给 WebView 添加一个事件监听对象(WebViewClient)

并重写其中的一些方法

shouldOverrideUrlLoading：对网页中超链接按钮的响应.

当按下某个连接时 WebViewClient 会调用这个方法,并传递参数：按下的 url

onLoadResource

onPageStart

onPageFinish

onReceiveError

onReceivedHttpAuthRequest

6、如果用 WebView 点链接看了很多页以后,如果不做任何处理,点击系统“Back”键,整个浏览器会调用 finish() 而结束自身,如果希望浏览的网页回退而不是退出浏览器,需要在当前 Activity 中处理并消费掉该 Back 事件.

覆盖 Activity 类的 onKeyDown(int keyCoder,KeyEvent event)方法.

```
public boolean onKeyDown(int keyCoder,KeyEvent event){
if(WebView.canGoBack() && keyCoder == KeyEvent.KEYCODE_BACK){
WebView.goBack();     //goBack()表示返回 WebView 的上一页面
return true;
}
return false;
}
```

4.6 Android 获取服务器中的 session 问题

1、Android 中的 WebView 如何获取服务器页面的 jsessionid 的值

2、Android 的 WebView 又是如何把得到的 jsessionid 的值在 set 到服务器中,一致达到他们在同一个 jsessionid 的回话中.

解决如下：

```
CookieManager cm = CookieManager.getInstance();
cm.removeAllCookie();
cm.getCookie(url);
cm.setCookie(url, cookie);
```

另外还有个 CookieSyncManager,没搞清干嘛使的,但是我按以下顺序调用,设置 Cookie 没问题

```
CookieSyncManager csm = CookieSyncManager.createInstance(this);
CookieManager cm = CookieManager.getInstance();
cm.removeAllCookie();
csm.sync();
cm.setCookie(url, cookie);
```

4.7 WebView 设置实现两个手指缩放网页

```
mWebView.getSettings().setSupportZoom(true);
mWebView.getSettings().setBuiltInZoomControls(true);
```

代码是这样,但是首先是你的硬件支持多点触控

4.8 WebView cookies 清理

在使用新浪微博账户登录应用时,WebView 会自动登录上次的微博帐号！(因为 WebView 记录了微博帐号和密码的 cookies)

所以,需要清除 SessionCookie：

```
CookieSyncManager.createInstance(this);
CookieSyncManager.getInstance().startSync();
CookieManager.getInstance().removeSessionCookie();
```

另外,清理 cache 和历史记录 的方法：

```
WebView.clearCache(true);
WebView.clearHistory();
```

4.9 Android WebView 使用 cmwap 无法联网解决办法

1、某些 Rom 在 wifi 环境下取代理依然会取到 cmwap 设置的代理值,所以取代理时判断一下手机网络环境,如果是 gprs 上网,则不取代理.

```
ConnectivityManager connectivityManager = (ConnectivityManager) context
.getSystemService(Context.CONNECTIVITY_SERVICE);
NetworkInfo activeNetworkInfo = connectivityManager
.getActiveNetworkInfo();
if (activeNetworkInfo != null) {
int type = activeNetworkInfo.getType();
// TODO 防止 wifi 下读取移动的代理
if (type == ConnectivityManager.TYPE_MOBILE) {
//取代理
} else {
//不取代理
}
}
```

2. 某些 rom 的 webkit 在 cmwap 下,webkit 连接不上或是时断时连,需要在 onResume()添加 WebView.enablePlatformNotifications();,onStop 添加 WebView.disablePlatformNotifications();.但是此时如果你需要刚进入这个 activity 就 load 页面,第一次进入时 load 的第一个页面是 load 不出来的,会停留很久并且不出错误页面,不调用 WebViewClient.onReceivedError.个人试过 setHttpAuthUsernamePassword 也不好使.

个人解决方案：

```
mWebPage.post(new Runnable(){
@Override
public void run() {
mWebPage.loadUrl(url);
}
});
```

此时第一此 load 页面会很快调用 WebViewClient.onReceivedError,你再在 WebViewClient.onReceivedError 里面设置标志,重新载入第一个页面即可.

4.10 Android 判断 WebView 是否已经滚动到页面底端

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.

getHeight()或者 getBottom()方法都返回当前 WebView 这个容器的高度

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：

```
if(WebView.getContentHeight*WebView.getScale() -(webvi ew.getHeight()+WebView.getScrollY())){
//已经处于底端
}
```

4.11 Android 中 WebView 实现 Javascript 调用 Java 类方法

为了方便网页和 Android 应用的交互,Android 系统提供了 WebView 中 JavaScript 网页脚本调用 Java 类方法的机制.只要调用 addJavascriptInterface 方法即可映射一个 Java 对象到 JavaScript 对象上.

1、映射 Java 对象到 JavaScript 对象上

```
mWebView = (WebView) findViewById(R.id.wv_content);
mWebView.setVerticalScrollbarOverlay(true);
final WebSettings settings = mWebView.getSettings();
settings.setSupportZoom(true);
//WebView 启用 Javascript 脚本执行
settings.setJavaScriptEnabled(true);
settings.setJavaScriptCanOpenWindowsAutomatically(true);
//映射 Java 对象到一个名为”js2java“的 Javascript 对象上
//JavaScript 中可以通过"window.js2java"来调用 Java 对象的方法
mWebView.addJavascriptInterface(new JSInvokeClass(), "HTMLOUT");
```

2、JavaScript 调用 Java 对象示例

这里才是最关键的地方所在了,下面是通过 javascript 来调用 java 中的方法,也就是 JSInvokeClass 中的 back 方法 window.HTMLOUT.back();

但是上面的方法放在哪里呢？如何去执行喃？请看：

```
mWebView.loadUrl("javascript:window.HTMLOUT.back();");
```

当然如果想传参数怎么办？不要急,首先在 JSInvokeClass.back 方法处,申明一个代参数的方法就行了：

```
/**网页 Javascript 调用接口**/
class JSInvokeClass {
public void back() {
activity.finish();
}
public void showHtml(String html)
{
Log.e("Html:"+html);
```

```
}
}
```

然后通过 javascript 调用就行了.

```
mWebView.loadUrl("javascript:window.HTMLOUT.showHtml(document.documentElement.innerHTML);");
```

更多技巧,请访问这里：
http://search.eoeandroid.com/f/search?q=WebView&sId=7379687&ts=1335235840&mySign=4f90227f&menu=1&rfh=1&qs=txt.form.a

五、Android WebView SDK 参考

这里是 **google** 官方的 **WebView sdk** 的介绍～

http://developer.android.com/reference/android/webkit/WebView.html

android.webkit

Provides tools for browsing the web.
The only classes or interfaces in this package intended for use by SDK developers are WebView, BroswerCallbackAdapter, BrowserCallback, and CookieManager.

Interfaces

DownloadListener	
GeolocationPermissions.Callback	Callback interface used by the browser to report a Geolocation permission state set by the user in response to a permissions prompt.
PluginStub	This interface is used to implement plugins in a WebView.
ValueCallback<T>	A callback interface used to returns values asynchronously
WebChromeClient.CustomViewCallback	A callback interface used by the host application to notify the current page that its custom view has been dismissed.
WebIconDatabase.IconListener	Interface for receiving icons from the database.
WebStorage.QuotaUpdater	Encapsulates a callback function to be executed when a new quota is made available.
WebView.PictureListener	*This interface is deprecated. This interface is now obsolete.*

Classes

CacheManager	*This class is deprecated. Access to the HTTP cache will be removed in a future release.*
CacheManager.CacheResult	*This class is deprecated. Access to the HTTP cache will be removed in a future release.*
ConsoleMessage	Public class representing a JavaScript console message from WebCore.
CookieManager	CookieManager manages cookies according to RFC2109 spec.

CookieSyncManager	The CookieSyncManager is used to synchronize the browser cookie store between RAM and permanent storage.
DateSorter	Sorts dates into the following groups: Today Yesterday seven days ago one month ago older than a month ago
GeolocationPermissions	This class is used to get Geolocation permissions from, and set them on the WebView.
HttpAuthHandler	HTTP authentication request that must be handled by the user interface.
JsPromptResult	Public class for handling javascript prompt requests.
JsResult	
MimeTypeMap	Two-way map that maps MIME-types to file extensions and vice versa.
SslErrorHandler	SslErrorHandler: class responsible for handling SSL errors.
URLUtil	
WebBackForwardList	This class contains the back/forward list for a WebView.
WebChromeClient	
WebHistoryItem	A convenience class for accessing fields in an entry in the back/forward list of a WebView.
WebIconDatabase	Functions for manipulating the icon database used by WebView.
WebResourceResponse	A WebResourceResponse is return by shouldInterceptRequest(WebView, String) and contains the response information for a particular resource.
WebSettings	Manages settings state for a WebView.
WebStorage	Functionality for manipulating the webstorage databases.
WebStorage.Origin	Class containing the HTML5 database quota and usage for an origin.
WebView	A View that displays web pages.
WebView.HitTestResult	
WebView.WebViewTransport	Transportation object for returning WebView across thread boundaries.
WebViewClient	
WebViewDatabase	
WebViewFragment	A fragment that displays a WebView.

Enums

ConsoleMessage.MessageLevel	
WebSettings.LayoutAlgorithm	Enum for controlling the layout of html.
WebSettings.PluginState	The plugin state effects how plugins are treated on a page.
WebSettings.RenderPriority	
WebSettings.TextSize	*This enum is deprecated. Use setTextZoom(int) and getTextZoom() instead.*
WebSettings.ZoomDensity	Enum for specifying the WebView's desired density.

六、关于优亿开发者社区的一些新鲜玩意

6.1 关于优亿特刊的订阅

各位亲，往常的时候，大家获得特刊只能通过在论坛下载的方式，又繁琐又不及时。现在这问题解决了，我们推出了订阅的方式，您只需要用电子邮箱订阅一下，便可以一劳永逸的第一时间获得最新发布的特刊内容。

亲，订阅非常简单哦。

详细页面，请您来这里看：

http://www.eoeandroid.com/eoemagazine/

6.2 关于论坛史上最强开发内容索引

这可是好东西哦，此贴汇集了论坛近 20 名网友耗时 1 个多月倾力打造！集中了优亿开发者社区自创建以来的精华内容，并根据内容做了最为详细的分类，号称优亿开发者社区史上最强内容整理，没有之一！

帖子好不好，您一看便知：

http://www.eoeandroid.com/thread-168008-1-1.html

6.3 关于全新打造的优亿移动人才招聘服务

亲们，是我们在六一儿童节全新推出的，旨在为大家提供最专业最对口最便捷的招聘服务平台。

在这里大家能够看到业内的名企，能够找到高薪的岗位，能够迅速的推销自己。

如果您对我们这项新服务感兴趣，那么请点击下面的链接：

http://zhaopin.eoe.cn/

6.4 关于优亿移动开放日沙龙活动

常有网友问，我们什么时候能够组织线下的聚会啊，什么时候能够见见技术大拿。

其实我们的优亿移动开放日就是为大家准备的。该沙龙活动每月 2 期，北京——也就是优亿的大本营一期，其他城市一期 —— 基本都是开发者比较聚集的地方，例如上海深圳广州成都等地。

2012 年 6 月 16 日将来到天府之国——成都，在附近的亲们不要错过了哦！

报名参加，请猛击链接：

http://www.eoeandroid.com/eoesalon/2012/0522/20.html

6.5 关于其他为开发者准备的贴心服务

目前已有应用统计和应用云测试两项服务为大家准备就绪，亲们只需使用社区账号登录便可享受到方便快捷的服务。

真心希望我们的服务，能够为大家解决问题。O(∩_∩)o

http://sso.eoeandroid.com/

优亿开发者社区

责任编辑：果子狸

美术支持：大丸子　　技术支持：郝留有

中国最大的Android 开发者社区：www.eoeandroid.com

中国本土的Android 软件下载平台：www.eoemarket.com