

## 7.5.5 WebView缓存问题

分类 [Android 基础入门教程](#)

**Android 基础入门教程(Q群号 : 153836263)**

### 本节引言：

现在很多门户类信息网站，比如虎嗅，ifanr，钛媒体等等的APP，简单点说是信息阅读类的APP，很多都是直接嵌套一个WebView用来显示相关资讯的，这可能就涉及到了WebView的缓存了！

所谓的页面缓存 就是指：保存加载一个网页时所需的HTML，JS，CSS等页面相关的数据以及其他资源，当没网的时候或者网络状态较差的时候，加载本地保存好的相关数据！而实现这个缓存的方式有两种，一种是后台写一个下载的Service，将文章相关的数据按自己的需求下载到数据库或者保存到相应文件夹中，然后下次加载对应URL前先判断是否存在本地缓存，如果存在优先加载本地缓存，不存在则执行联网请求，同时缓存相关资源，典型的如旧版本的36Kr，在进去后会先离线文章，然后再显示！

当然，本节要讲解的不是这种自己写逻辑的方式，而是通过WebView本身自带的缓存功能来缓存页面，这种方式使用起来非常简单，我们只需为WebView设置开启相关功能，以及设置数据库的缓存路径即可完成缓存！具体的实现我们下面——道来~

### 1.缓存的分类：

首先要说的一点是缓存的分类，我们缓存的数据分为：**页面缓存和数据缓存**

**页面缓存**：加载一个网页时的html、JS、CSS等页面或者资源数据，这些缓存资源是由于浏览器的行为而产生，开发者只能通过配置HTTP响应头影响浏览器的行为才能间接地影响到这些缓存数据。而**缓存的索引**放在：`/data/data/<包名>/databases` **对应的文件**放在：`/data/data/package_name/cache/webviewCacheChron`下

**数据缓存**：分为AppCache和DOM Storage两种 我们开发者可以自行控制的就是这些缓存资源，

**AppCache**：我们能够有选择的缓冲web浏览器中所有的东西，从页面、图片到脚本、css等等。尤其在涉及到应用于网站的多个页面上的CSS和JavaScript文件的时候非常有用。其大

- 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

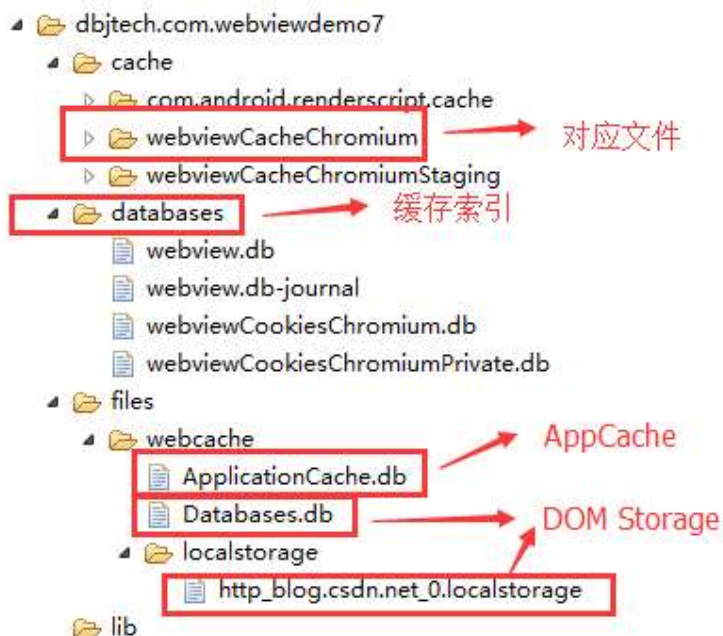
小目前通常是5M。在Android上需要手动开启（`setAppCacheEnabled`），并设置路径（`setAppCachePath`）和容量（`setAppCacheMaxSize`），而Android中使用**ApplicationCache.db**来保存AppCache数据！

**DOM Storage**：存储一些简单的用key/value对即可解决的数据，根据作用范围的不同，有Session Storage和Local Storage两种，分别用于会话级别的存储（页面关闭即消失）和本地化存储（除非主动删除，否则数据永远不会过期）在Android中可以手动开启DOM Storage（`setDomStorageEnabled`），设置存储路径（`setDatabasePath`）Android中Webkit会为DOMStorage产生两个文件（`my_path/localstorage/http_blog.csdn.net_0.localstorage`和`my_path/Databases.db`）

好吧，看完上面，是不是想说一句，卧槽，什么鬼，好复杂的样子



当然，不要去背，知道有这些东西就好了，实际开发用到再慢慢考究，而且我们一般只关心如何为WebView设置缓存以及如何删除缓存！我们可以看下我们下面写的demo运行后的文件结构，打开DDMS的File Explorer：



嘿嘿，一目了然了吧~，对了另外还要说下几种缓存的模式：

**LOAD\_CACHE\_ONLY**: 不使用网络，只读取本地缓存数据

**LOAD\_DEFAULT**: 根据cache-control决定是否从网络上取数据。

**LOAD\_CACHE\_NORMAL**: API level 17中已经废弃, 从API level 11开始作用同LOAD\_DEFAULT模式

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

2.4.7 ListView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 ViewPager(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 AsyncTask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

**LOAD\_NO\_CACHE:** 不使用缓存，只从网络获取数据。

**LOAD\_CACHE\_ELSE\_NETWORK**，只要本地有，无论是否过期，或者no-cache，都使用缓存中的数据。

**总结：**根据以上两种模式，建议缓存策略为，判断是否有网络，有的话，使用LOAD\_DEFAULT，无网络时，使用LOAD\_CACHE\_ELSE\_NETWORK。

接下来堆码时间！

## 2.为WebView开启缓存功能

下面我们就来为WebView开启缓存功能，先看下实现的效果图：

运行效果图：



**流程解析：**1.进入页面后默认加载url，然后随便点击一个链接跳到第二个页面，退出APP 2.关闭wifi以及移动网络，然后重新进入，发现无网络的情况下，页面还是加载了，打开第一个链接也可以加载，打开其他链接就发现找不到网页！ 3.点击清除缓存，把应用关闭，重新进入，发现页面已经打不开！

接下来是代码实现：**MainActivity.java**:

```
public class MainActivity extends AppCompatActivity {

    private WebView wView;
    private Button btn_clear_cache;
    private Button btn_refresh;
    private static final String APP_CACHE_DIRNAME = "/webcache"
; // web缓存目录
    private static final String URL = "http://blog.csdn.net/coder_pig";

    @Override
```

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

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    wView = (WebView) findViewById(R.id.wView);
    btn_clear_cache = (Button) findViewById(R.id.btn_clear_
cache);
    btn_refresh = (Button) findViewById(R.id.btn_refresh);
    wView.loadUrl(URL);
    wView.setWebViewClient(new WebViewClient() {
        //设置在webView点击打开的新网页在当前界面显示,而不跳转到
新的浏览器中
        @Override
        public boolean shouldOverrideUrlLoading(WebView vie
w, String url) {
            view.loadUrl(url);
            return true;
        }
    });
    WebSettings settings = wView.getSettings();
    settings.setJavaScriptEnabled(true);
    //设置缓存模式
    settings.setCacheMode(WebSettings.LOAD_CACHE_ELSE_NETWO
RK);
    // 开启DOM storage API 功能
    settings.setDomStorageEnabled(true);
    // 开启database storage API功能
    settings.setDatabaseEnabled(true);
    String cacheDirPath = getFilesDir().getAbsolutePath() +
APP_CACHE_DIRNAME;
    Log.i("cachePath", cacheDirPath);
    // 设置数据库缓存路径
    settings.setAppCachePath(cacheDirPath);
    settings.setAppCacheEnabled(true);
    Log.i("databasepath", settings.getDatabasePath());

    btn_clear_cache.setOnClickListener(new View.OnClickListener
ener() {
        @Override
        public void onClick(View v) {
            wView.clearCache(true);
        }
    });

    btn_refresh.setOnClickListener(new View.OnClickListener
() {
        @Override
        public void onClick(View v) {
            wView.reload();
        }
    });
}

//重写回退按钮的点击事件
@Override
public void onBackPressed() {

```

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...

```
        if (webView.canGoBack()) {  
            webView.goBack();  
        } else {  
            super.onBackPressed();  
        }  
    }  
}
```

代码很简单，我们做的仅仅是开启缓存的功能，以及设置缓存模式以及缓存的数据的路径而已！

### 3.删除WebView的缓存数据

上面的示例，我们通过调用WebView的clearCache(true)方法，已经实现了对缓存的删除！除了这种方法外，还有下述方法：

```
setting.setCacheMode(WebSettings.LOAD_NO_CACHE);
```

```
deleteDatabase("WebView.db");和  
deleteDatabase("WebViewCache.db");
```

```
webView.clearHistory();
```

```
webView.clearFormData();
```

```
getCacheDir().delete();
```

手动写delete方法，循环迭代删除缓存文件夹！

当然，前面也说，我们能这直接操作的只是数据部分，而页面缓存是由于浏览器的行为而产生，我们只能通过配置HTTP响应头影响浏览器的行为才能间接地影响到这些缓存数据。所以上述的方法仅仅是删除的数据部分的缓存！

### 4.示例代码下载：

WebViewDemo7.zip : [下载 WebViewDemo7.zip](#)

### 5.本节小结：

好的，本节关于WebView缓存问题就到这里，这里只是写了如何为WebView开启缓存，以及删除缓存，以后遇到再慢慢考究，这里有个映像先~嗯，就说这么多~谢谢





对了，差点忘了贴下本节的参考链接：

[Android webView 缓存 Cache + HTML5离线功能 解决](#)

[Android记录25-WebView实现离线缓存阅读](#)

[Android 清除WebView缓存](#)

← 7.5.4 WebView文件下载

7.5.6 WebView处理网页返回的错误码信息 →



**坚持1个月 看美剧不用字幕**  
每天45分钟 30天见证你的英语奇迹

立即行动 >

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

- 设置 SSH 通过密...
- 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...

站点信息

- 意见反馈
- 免责声明
- 关于我们
- 文章归档

关注微信



Copyright © 2013-2015 菜鸟教程  
runoob.com All Rights Reserved.  
备案号：闽ICP备15012807号-1