

[首页](#) [ANDROID](#) [互联网](#) [杂乱无章](#) [科技资讯](#) [程序员人生](#) [程序员笑话](#) [编程技术](#) [网址导航](#)

7.1.4 Android HTTP请求方式:HttpClient

分类 [Android 基础入门教程](#)

Android 基础入门教程(Q群号 : 153836263)

本节引言：

在上一节中我们对URLConnection进行了学习，本节到第二种方式：HttpClient，尽管被Google 弃用了，但是我们平时也可以拿HttpClient来抓下包，配合Jsoup解析网页效果更佳！HttpClient用于接收/发送Http请求/响应，但不缓存服务器响应，不执行HTML页面潜入的JS代码，不会对页面内容 进行任何解析，处理！开始本节内容！

1.HttpClient使用流程

基本流程：

```
①创建HttpClient对象:HttpClient client = new DefaultHttpClient();
②发送Get请求的话,创建HttpGet对象;发送Post请求的话创建HttpPost对象
③设置请求参数:两者都可以用setParams(HttpParams);Post还可用setEntity(HttpEntity)方法
④调用httpClient对象的execute()方法发送请求,该方法会返回一个HttpResponse
⑤对返回的HttpResponse调用:getAllHeaders(),getHeaders(name)等方法可以获得服务器的响应头
调用getEntity()方法可以取得HttpEntity对象,该对象包装了服务器的响应内容
```

2.HttpClient使用示例

1) 使用HttpClient发送GET请求

直接贴下简单的发送Get请求的代码：

```
public class MainActivity extends Activity implements OnClickListener {

    private Button btnGet;
    private WebView wView;
    public static final int SHOW_DATA = 0x123;
    private String detail = "";

    private Handler handler = new Handler() {
        public void handleMessage(Message msg) {
            if(msg.what == SHOW_DATA)
            {
```

- 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)

```

        webView.loadDataWithBaseURL("", detail, "text/html", "UTF-8", "");
    }

};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    initView();
    setView();
}

private void initView() {
    btnGet = (Button) findViewById(R.id.btnGet);
    webView = (WebView) findViewById(R.id.wView);
}

private void setView() {
    btnGet.setOnClickListener(this);
    webView.getSettings().setDomStorageEnabled(true);
}

@Override
public void onClick(View v) {
    if (v.getId() == R.id.btnGet) {
        GetByHttpClient();
    }
}

private void GetByHttpClient() {
    new Thread()
    {
        public void run()
        {
            try {
                HttpClient httpClient = new DefaultHttpClient();

                HttpGet httpGet = new HttpGet("http://www.w3cschool.cc/python/python-tutorial.html");
                HttpResponse httpResponse = httpClient.execute(httpGet);

                if (httpResponse.getStatusLine().getStatusCode() == 200) {
                    HttpEntity entity = httpResponse.getEntity();

                    detail = EntityUtils.toString(entity, "utf-8");

                    handler.sendMessage(SHOW_DATA);
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }.start();
}

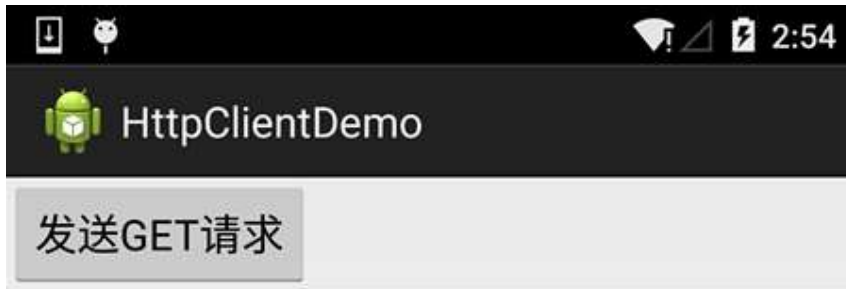
```

[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](#)

}

}

运行截图



[Search w3cschool.cn](#) [HTML](#) [CSS](#) [JavaScript](#) [jQuery](#) [jQueryUI](#) [Bootstrap](#) [jQueryMobile](#) [EasyUI](#) [XML](#) [SQL](#) [MySQL](#) [SQLite](#) [PHP](#) [C#](#) [Python](#) [Java](#) [JSP](#) [Ruby](#) [前端工具](#) [Bug 反馈](#) [更多.....](#)

Python 基础教程

[Python 基础教程](#) [Python 简介](#) [Python 环境搭建](#) [Python 中文编码](#) [Python 基础语法](#) [Python 变量类型](#) [Python 运算符](#) [Python 条件语句](#) [Python 循环语句](#) [Python While循环语句](#) [Python for 循环语句](#) [Python 循环嵌套](#) [Python break 语句](#) [Python continue 语句](#) [Python pass 语句](#) [Python 数字](#) [Python 字符串](#) [Python 列表\(Lists\)](#) [Python 元组](#) [Python 字典\(Dictionary\)](#) [Python 日期和时间](#) [Python 函数](#) [Python 模块](#) [Python 文件I/O](#) [Python 异常处理](#)



另外，如果是带有参数的GET请求的话，我们可以将参数放到一个List集合中，再对参数进行URL编码，最后和URL拼接下就好了：

```
List<BasicNameValuePair> params = new LinkedList<BasicNameValuePair>();  
params.add(new BasicNameValuePair("user", "猪小弟"));  
params.add(new BasicNameValuePair("pwd", "123"));  
String param = URLEncodedUtils.format(params, "UTF-8");
```

[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JavaScip...](#)[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](#)

```
HttpGet httpGet = new HttpGet("http://www.baidu.com"+"?" + param)
;
```

2) 使用HttpClient发送POST请求

POST请求比GET稍微复杂一点，创建完HttpPost对象后，通过NameValuePair集合来存储等待提交的参数，并将参数传递到UrlEncodedFormEntity中，最后调用setEntity(entity)完成，HttpClient.execute(HttpPost)即可；这里就不写例子了，暂时没找到Post的网站，又不想自己写个Servlet，So,直接贴核心代码吧~

核心代码:

```
private void PostByHttpClient(final String url)
{
    new Thread()
    {
        public void run()
        {
            try{
                HttpClient httpClient = new DefaultHttpClient()
;
                HttpPost httpPost = new HttpPost(url);
                List<NameValuePair> params = new ArrayList<Name
ValuePair>();
                params.add(new BasicNameValuePair("user", "猪大
哥"));
                params.add(new BasicNameValuePair("pwd", "123"
));
                UrlEncodedFormEntity entity = new UrlEncodedFor
mEntity(params,"UTF-8");
                httpPost.setEntity(entity);
                HttpResponse httpResponse = httpClient.execute
(httpPost);
                if (httpResponse.getStatusLine().getStatusCode(
) == 200) {
                    HttpEntity entity2 = httpResponse.getEntity
();
                    detail = EntityUtils.toString(entity2, "utf
-8");
                    handler.sendMessage(SHOW_DATA);
                }
            } catch (Exception e) {e.printStackTrace();}
        }
    }.start();
}
```

3.HttpClient抓数据示例(教务系统数据抓取)

其实关于HttpClient的例子有很多，比如笔者曾经用它来抓学校教务

- 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...

系统上学生的课程表: 这就涉及到Cookie, 模拟登陆的东西, 说到抓数据(爬虫), 一般我们是搭配着JSoup来解析 抓到数据的, 有兴趣可以自己查阅相关资料, 这里贴下笔者毕设app里获取网页部分的关键代码! 大家可以体会下:

HttpClient可以通过下述代码获取与设置Cookie: **HttpResponse**

loginResponse = new DefaultHttpClient().execute(getLogin); 获得

Cookie:**cookie = loginResponse.getFirstHeader("Set-**

Cookie").getValue(); 请求时带上

Cookie:**httpPost.setHeader("Cookie", cookie);**

//获得链接,模拟登录的实现:

```
public int getConnect(String user, String key) throws Exception
{
    // 先发送get请求 获取cookie值和__ViewState值
    HttpGet getLogin = new HttpGet(true_url);
    // 第一步:主要的HTML:
    String loginhtml = "";
    HttpResponse loginResponse = new DefaultHttpClient().execute(getLogin);
    if (loginResponse.getStatusLine().getStatusCode() == 200) {
        HttpEntity entity = loginResponse.getEntity();
        loginhtml = EntityUtils.toString(entity);
        // 获取响应的cookie值
        cookie = loginResponse.getFirstHeader("Set-Cookie").getValue();
        System.out.println("cookie= " + cookie);
    }

    // 第二步:模拟登录
    // 发送Post请求,禁止重定向
    HttpPost httpPost = new HttpPost(true_url);
    httpPost.getParams().setParameter(ClientPNames.HANDLE_REDIRECTS, false);

    // 设置Post提交的头信息的参数
    httpPost.setHeader("User-Agent",
        "Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; rv:11.0) like Gecko");
    httpPost.setHeader("Referer", true_url);
    httpPost.setHeader("Cookie", cookie);

    // 设置请求数据
    List<NameValuePair> params = new ArrayList<NameValuePair>();

    params.add(new BasicNameValuePair("__VIEWSTATE",
        getViewState(loginhtml))); // __VIEWSTATE参数, 如果变化可以动态抓取获取
    params.add(new BasicNameValuePair("Button1", ""));
    params.add(new BasicNameValuePair("hidPdrs", ""));
```



```

params.add(new BasicNameValuePair("hidsc", ""));
params.add(new BasicNameValuePair("lbLanguage", ""));
params.add(new BasicNameValuePair("RadioButtonList1", "%D1%A7%C9%FA"));
params.add(new BasicNameValuePair("txtUserName", user));
params.add(new BasicNameValuePair("TextBox2", key));
params.add(new BasicNameValuePair("txtSecretCode", "")); //
(ノ口ノ)逗比正方,竟然不需要验证码

// 设置编码方式,响应请求,获取响应状态码:
httpPost.setEntity(new UrlEncodedFormEntity(params, "gb2312"));
HttpResponse response = new DefaultHttpClient().execute(httpPost);
int Status = response.getStatusLine().getStatusCode();
if(Status == 200) return Status;
System.out.println("Status= " + Status);

// 重定向状态码为302
if (Status == 302 || Status == 301) {
    // 获取头部信息中Location的值
    location = response.getFirstHeader("Location").getValue();

    System.out.println(location);
    // 第三步:获取管理信息的主页面
    // Get请求
    HttpGet httpGet = new HttpGet(ip_url + location); // 带上location地址访问
    httpGet.setHeader("Referer", true_url);
    httpGet.setHeader("Cookie", cookie);

    // 主页的html
    mainhtml = "";
    HttpResponse httpResponseget = new DefaultHttpClient().execute(httpGet);
    if (httpResponseget.getStatusLine().getStatusCode() == 200) {
        HttpEntity entity = httpResponseget.getEntity();
        mainhtml = EntityUtils.toString(entity);
    }
}
return Status;
}

```

4.使用HttpPut发送Put请求

示例代码如下：

```

public static int PutActCode(String actCode, String licPlate, Context mContext) {
    int resp = 0;

```

```
String cookie = (String) SPUtils.get(mContext, "session",
");
HttpPut httpPut = new HttpPut(PUTACKCODE_URL);
httpPut.setHeader("Cookie", cookie);
try {

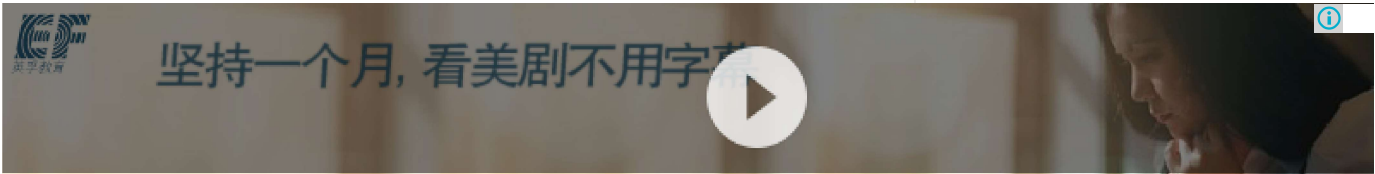
    List<NameValuePair> params = new ArrayList<NameValuePair>();
    params.add(new BasicNameValuePair("activation_code", activationCode));
    params.add(new BasicNameValuePair("license_plate", licensePlate));
    httpPut.setEntity(new UrlEncodedFormEntity(params, "UTF-8"));
    HttpResponse course_response = new DefaultHttpClient().execute(httpPut);
    if (course_response.getStatusLine().getStatusCode() == 200) {
        HttpEntity entity2 = course_response.getEntity();
        JSONObject jsonObject = new JSONObject(EntityUtils.toString(entity2));
        resp = Integer.parseInt(jsonObject.getString("status_code"));
        return resp;
    }
} catch (Exception e) {
    e.printStackTrace();
}
return resp;
}
```

^

反馈

本节小结：

好的，本节关于Android HTTP的第二种请求方式:HttpClient就到这里，下节开始我们来学习XML以及Json的解析，本节就到这里，谢谢~



在线实例	字符集&工具	最新更新	站点信息
· HTML 实例	· HTML 字符集设置	· C语言- 打...	· 意见反馈
· CSS 实例		· Win7和Linux	· 免责声明

- JavaScript 实例
- Ajax 实例
- jQuery 实例
- XML 实例
- Java 实例

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

- 下的...
- Linux下 Nginx+To...
- linux安装svn 服...
- PHP 换行符 PHP_EOL
- PHP 7 新特性
- Linux Crontab ...

- 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