

## Android 网络与通信基本功能使用

作者：梦宇

作者博客：<http://plter.com>

### URLConnection 使用

1.打开 URL 并且连得到输入流数据的操作如下：

```
URL u=new URL("http://baidu.com");
u.openStream();
```

2.然后可以使用BufferedReader来读取缓冲数据，操作如下：

```
BufferedReader br=new BufferedReader(new
InputStreamReader(u.openStream()));
```

3.接下来就可以用while循环来读取缓冲数据了，操作如下：

```
String line="";
String content="";
while((line=br.readLine())!=null){
    content+=line;
}
System.out.println(content);
```

4.在上面程序执行完成后，所得到的content就是加载下载的数据。所有参考代码如下所示：

```
package com.bupt3g.ldbfb;
```

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.MalformedURLException;
import java.net.URL;
```

```
import android.app.Activity;
import android.os.Bundle;
```

```
public class L001_LoadDataFromBaiduActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        try {
            URL u=new URL("http://baidu.com");

            BufferedReader br=new BufferedReader(new
```

```

InputStreamReader(u.openStream())));
    String line="";
    String content="";

    while((line=br.readLine())!=null){
        content+=line;
    }

    System.out.println(content);

} catch (MalformedURLException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}
}

```

注意：该程序使用了网络相关的功能，请申请相关的权限，具体操作是在AndroidManifest.xml文件中加入如下一行代码：

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

则AndroidManifest.xml文件的参考代码如下所示：

```

<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.bupt3g.ldfb"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="8" />
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".L001_LoadDataFromBaiduActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>

```

```
</activity>
</application>
```

```
</manifest>
```

HttpClient 的使用

下面例子中均需要申请权限

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

GET 方式与 POST 方式的区别

- 1.GET 方式可以将参数写在地址中，POST 方式不能
- 2.POST 方式可以用来传递大数据（100M 以内，实际开发中应该保持在 10M 以内），而 GET 方式不能

使用 HttpClient 以 GET 方法与服务器通信

- 1.创建 HttpClient 对象的方式如下所示：

```
HttpClient hc = new DefaultHttpClient();
```

- 2.接下来我们需要创建一个 HttpGet 对象用于与服务器进行 GET 方式的通信，操作如下：

```
HttpGet hg=new HttpGet("http://192.168.1.100/add.php?a=9&b=7");
```

- 3.下面我们需要用 HttpClient 对象来执行 HttpGet 对象，操作如下：

```
HttpResponse hr = hc.execute(hg);
```

在这个过程中，execute 方法会阻塞当前线程，直到数据加载完成，所以更合理的编程习惯是将通信过程放在 AsyncTask 中来完成，关于 AsyncTask 的用法可以参考 <http://plter.sinaapp.com/?p=173> 、  
<http://developer.android.com/reference/android/os/AsyncTask.html>

在数据加载完成后，execute 方法会返回 HttpResponse 对象，服务器的返回数据就是被封装在此对象中，此时可以开始读取数据。

- 4.读取数据的过程与 URLConnection 中所使用的方式一样，首先取得到 InputStrem 对象，方式如下：

```
hr.getEntity().getContent();
```

然后继续使用 BufferedReader 来封装数据的读取，操作如下：

```
BufferedReader br =new BufferedReader(new  
InputStreamReader(hr.getEntity().getContent()));
```

- 5.接下来用 while 循环读取数据，操作如下：

```
String content="";  
String line="";  
while((line=br.readLine())!=null){  
    content+=line;  
}
```

在循环执行完毕后所得到的`content`就是全部服务器的返回值。

6.完整的示例代码如下：

```
package com.bupt3g.usinghttpclient;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

import org.apache.http.HttpResponse;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;

import android.app.Activity;
import android.os.Bundle;

public class L002_UsingHttpClientActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        HttpClient hc = new DefaultHttpClient();
        HttpGet hg=new
HttpGet("http://192.168.1.100/add.php?a=9&b=7");

        try {
            HttpResponse hr = hc.execute(hg);
            BufferedReader br =new BufferedReader(new
InputStreamReader(hr.getEntity().getContent()));

            String content="";
            String line="";
            while((line=br.readLine())!=null){
                content+=line;
            }

            System.out.println(content);
        } catch (ClientProtocolException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```

    }
}
}

```

在通信过程中所使用的php脚本如下：

```

<?php
if (!empty($_GET['a'])&&
    !empty($_GET['b'])){

    echo $_GET['a']+$_GET['b'];
}else{
    echo 'Arguments Error';
}

```

使用 HttpClient 以 POST 方法与服务器通信

1.建立 HttpClient 对象，如下所示：

```
HttpClient hc=new DefaultHttpClient();
```

2.构建 HttpPost 对象，如下所示：

```
HttpPost hp=new HttpPost("http://192.168.1.100/add-post.php");
```

3.配置上传的 POST 参数，如下所示：

```

List<NameValuePair> args=new ArrayList<NameValuePair>();
args.add(new BasicNameValuePair("a", "18"));
args.add(new BasicNameValuePair("b", "10"));

```

4.将参数进行URL编码，操作如下所示：

```
UrlEncodedFormEntity en=new UrlEncodedFormEntity(args);
```

5.设置POST方法要上传的参数，操作如下所示：

```
hp.setEntity(en);
```

6.接下来执行通信操作，代码如下所示：

```
HttpResponse hr = hc.execute(hp);
```

7.下面就是读取返回数据的过程，与GET方式的返回数据读取是一样的，如下所示：

```

BufferedReader br=new BufferedReader(new
InputStreamReader(hr.getEntity().getContent()));

```

```

String line=null;
String content="";

```

```
while((line=br.readLine())!=null){
    content+=line;
}
```

数据读取完成后所得到的content对象就是所有的服务器返回数据。

8.完整的代码如下所示:

```
package com.bupt3g.pats;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;

import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;

import android.app.Activity;
import android.os.Bundle;

public class L003_PostArgsToServerActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        HttpClient hc=new DefaultHttpClient();

        HttpPost hp=new HttpPost("http://192.168.1.100/add-
post.php");

        try {

            List<NameValuePair> args=new ArrayList<NameValuePair>();
            args.add(new BasicNameValuePair("a", "18"));
```

```

        args.add(new BasicNameValuePair("b", "10"));

        UrlEncodedFormEntity en=new UrlEncodedFormEntity(args);

        hp.setEntity(en);

        HttpResponse hr = hc.execute(hp);

        BufferedReader br=new BufferedReader(new
InputStreamReader(hr.getEntity().getContent()));

        String line=null;
        String content="";

        while((line=br.readLine())!=null){
            content+=line;
        }

        System.out.println(content);

    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

9.此示例所使用的php脚本如下所示:

```

<?php
if (!empty($_POST['a'])&&
    !empty($_POST['b'])){

    echo $_POST['a']+$_POST['b'];
}else{
    echo 'Arguments Error';
}

```