Android使用Google推荐的联网框架Volley,让连接网络更加简单

随着技术的进步，科技的发达，很少有应用是单机的了，大部分都需要联网访问服务器，以前我们都用

httpclient和httpurlconnection，感觉是不是非常麻烦，而Google特别为开发者考虑，推出了Volley，从此妈妈再也不担心我的联网问题了，哪里联不上联哪里，Volley让联网更容易和更快，并且

Volley请求会异步执行，不阻挡主线程

联网请求无非就是Get或者Post,我们很少用到DELETE这个请求方式，说了这么多废话，下面让我们看看Volley到底能干啥。它真的有这么神奇吗，芝麻开门，让我们一起来探索Volley 吧！

**一：Volley提供的功能**

1、封装了的异步的RESTful 请求API；

2、一个优雅和稳健的请求队列；

3、一个可扩展的架构，它使开发人员能够实现自定义的请求和响应处理机制；

4、能够使用外部HTTP Client库；

5、缓存策略；

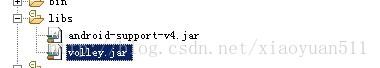
6、自定义的网络图像加载视图（NetworkImageView，ImageLoader等);

**二： 为什么使用异步HTTP请求？**

Android中要求HTTP请求异步执行，如果在主线程执行HTTP请求，可能会抛出android.os.NetworkOnMainThreadException  异常。阻塞主线程有一些严重的后果，它阻碍UI渲染，用户体验不流畅，它可能会导致可怕的ANR（Application Not Responding）。要避免这些陷阱，作为一个开发者，应该始终确保HTTP请求是在一个不同的线程。

**三：正文Volley的Get和Post方法**

首先我们要使用Volley要把 Volley.jar导入我们的项目中，如图：



这样我们就可以使了，让我们来看看Get和Post代码，如下：

**package** com.zqy.myvolley;

**import** java.util.HashMap;

**import** java.util.Map;

**import** com.android.volley.AuthFailureError;

**import** com.android.volley.Request;

**import** com.android.volley.RequestQueue;

**import** com.android.volley.Response;

**import** com.android.volley.VolleyError;

**import** com.android.volley.Request.Method;

**import** com.android.volley.Response.ErrorListener;

**import** com.android.volley.Response.Listener;

**import** com.android.volley.toolbox.StringRequest;

**import** com.android.volley.toolbox.Volley;

**import** android.os.Bundle;

**import** android.text.TextUtils;

**import** android.view.View;

**import** android.view.View.OnClickListener;

**import** android.widget.Button;

**import** android.widget.Toast;

**import** android.app.Activity;

**public** **class MainActivity extends Activity implements OnClickListener {**

**private** RequestQueue mQueue;

String url="http://www.baidu.com";

@Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mQueue=Volley.newRequestQueue(**this**); //创建一个RequestQueue对象。

initView();

}

**private** **void** initView(){

**final** Button btnGet=(Button) findViewById(R.id.btn\_get);

**final** Button btnPost=(Button) findViewById(R.id.btn\_post);

btnGet.setOnClickListener(**this**);

btnPost.setOnClickListener(**this**);

}

@Override

**public** **void** onClick(View v) {

**switch** (v.getId()) {

**case** R.id.btn\_get:

initGet();

**break**;

**case** R.id.btn\_post:

initPost();

**break**;

**default**:

**break**;

}

}

**private** **void** initGet(){*//Get请求 Method为GET*

mQueue.add(**new** StringRequest(Method.GET,url, **new** Listener<String>() {

@Override

**public** **void** onResponse(String arg0) {

Toast.makeText(MainActivity.**this**, "Get请求成功", 2).show();

*//成功，在这里写处理内容的代码*

}

}, **new** ErrorListener() {

@Override

**public** **void** onErrorResponse(VolleyError arg0) {

Toast.makeText(MainActivity.**this**, "Get请求失败", 2).show();

*//失败*

}

}));

}

**private** **void** initPost(){*//Post请求*

*//这写你自己的内部类PostResquest。。。。Method改成POST*

mQueue.add(**new** PostResquest(Method.POST,url, **new** Listener<String>() {

@Override

**public** **void** onResponse(String arg0) {

Toast.makeText(MainActivity.**this**, "Post请求成功", 2).show();

*//成功，在这里写处理内容的代码*

}

}, **new** ErrorListener() {

@Override

**public** **void** onErrorResponse(VolleyError arg0) {

Toast.makeText(MainActivity.**this**, "Post请求失败", 2).show();

*//失败*

}

}));

}

*//写个内部类，Post里面放一些服务器需要的参数*

**class PostResquest extends StringRequest {**

**public** PostResquest(**int** method, String url,

Listener<String> listener, ErrorListener errorListener) {

**super**(method, url, listener, errorListener);

}

**protected** Map<String, String> getParams() **throws** AuthFailureError {

Map<String, String> params = **new** HashMap<String, String>();

params.put("Name", "小源");*//参数*

params.put("Age",22+"");*//参数*

**return** params;

}

}

}

这就是联网的代码了，是不是特别简单，比 httpclient和httpurlconnection简单很多，并且还是 异步执行，根本无需Handler。让我们开发更便捷，更简单，更高效，在让我们看看XML里面的布局，里面我放了两个Button,一个执行Get方法，一个执行Post方法。如下：

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity" >

<Button

android:id="@+id/btn\_get"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="28dp"

android:text="Get请求" />

<Button

android:id="@+id/btn\_post"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/btn\_get"

android:layout\_below="@+id/btn\_get"

android:layout\_marginTop="47dp"

android:text="Post请求" />

</RelativeLayout>

OK.大功告成，是不是感觉很简单，看完这些代码，你是不是也打算跟 httpclient和httpurlconnection say goodbye了跟Volley say hello那。最后祝愿安卓系统越做越好，让我们Android必胜宝刀行走在科技的世界里吧，Yeah~~~~

让我们看看效果图，附Volley宣传图片一张！Go Go，回家，外面要下雨了！

对了，差点忘了，千万不要了加联网的权限。要不会报错的！

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



咦，Post怎么请求失败了，在手机上是成功的，我参数是瞎拼的！正好做个反面失败的案列，真是天助我也啊！

真走了！下雨了！