我的论文

基于Android的APP终端数据交互

摘要：随着智能手机的大量普及，人类的吃穿住行，甚至娱乐，消费都离不开手机APP，在国内市场，Android操作系统远多于Ios操作系统，所以我们必须利用好Android平台来开发各类软件，为解决app终端与web服务端的数据交互问题，我们一般采用post和get请求方式通过解析服务端返回的Json数据流，实现数据传递。

1.引言

安卓（Android）是一种是由Google公司和开放手机联盟领导联合开发的自由开放的操作系统。目前大多用于移动设备，如智能手表，智能手机，平板电脑，电视等，Android操作系统分为五层，它们分别是内核层，硬件抽象层，Framework层，Runtime和公共库层，应用层，应用层是我们最熟悉的部分的，大部分软件开发者都集中在这一层，本文就是针对应用层数据交互举例来说明。

Json是一种轻量级的数据交互格式，易于阅读和编写，也易于机器的识别和读写，Json数据结构为 {key1：value1, key2：value2, ...} 的键值对结构。在面向对象的语言中，key 为对象的属性，value 为对应的值。键名可以使用整数和字符串来表示。值的类型可以是任意类型。本文主要阐述如何利用HTTP实现数据储存和Json数据流的解析。

2．Get方式请求Json数据及解析

在Android中，提供了标准的Java接口HttURLConnection,通过url.openConnection()方法获取，因为要访问网络，所以在android stdio项目的清单文件AndroidManifest.xml中添加访问网络的权限：＜uses-permission android:name=”android.permission.INTERNET”/＞本文以获取web端一个数据列表为例。.

Get请求代码：

public static String getresult(String urlStr){//创建获取数据流的函数，urlStr是web端服务器地址

try {

URL url=new URL(urlStr);

HttpURLConnection connect=(HttpURLConnection)url.openConnection();

InputStream input=connect.getInputStream();//获取文件字节数

BufferedReader in = new BufferedReader(new InputStreamReader(input));//构建Bufferedreader 变量用来存放获取的字节转化后的字符

String line = null

System.out.println(connect.getResponseCode());

StringBuffer sb = new StringBuffer();

while ((line = in.readLine()) != null) {//从Bufferedreader对象中读取内容。

sb.append(line);

}

return sb.toString();//返回字符流

} catch (Exception e) {

System.out.println(e.toString());

return null;//请求失败

}

}

数据解析代码：

ArrayList＜String＞ img = new ArrayList＜String＞();//将解析对象定义为一个列表

ArrayList＜String＞ title = new ArrayList＜String＞();

private void setinfo() {//构造解析数据函数

final String url = "";//请求数据的web端地址

new Thread() {//开启线程

@Override

public void run() {//在线程中执行解析操作

String text = interne.getresult(url);//获取返回的Json数据流，interne为创建的一个类

JSONObject result = null;

try {

result = new JSONObject(text);

final JSONArray list = result.getJSONArray("list");

for (int i = 0; i ＜ list.length(); i++) {//解析Json数据添加到列表

try {

JSONObject object =list.getJSONObject(i);

img.add(object.getInt("Ac\_imag\_0 "));//获取键值对的value值

title.add(object.getString("Ac\_title "));

} catch (JSONException e) {

e.printStackTrace();

}

}

} catch (JSONException e) {

e.printStackTrace();

}

}

}.start();

}

3.Post方式提交数据

Post请求代码Get请求代码差不多，只是传递参数的方式不一样，setRequestmessod方法传入Post,并且需要通过setRequestProperty方法设置请求属性，然后从connection中获取输出流，通过write方法将参数写进去，本文以实现登陆功能为例。

Post请求代码：

//构造post请求函数，

public static String login(String urlStr,String user\_phone,String user\_password){

String params = "{\"user\_phone\":" +user\_phone +",\"user\_password\":" + user\_password+ "}";//请求数据

try {

URL url=new URL(urlStr);//请求web端地址

HttpURLConnection connect=(HttpURLConnection)url.openConnection();

connect.setDoInput(true);

connect.setDoOutput(true);

connect.setRequestMethod("POST");//Post提交数据

connect.setUseCaches(false);

//设置请求头

connect.setRequestProperty("Content-Type", "application/json;charset=UTF-8");

OutputStream outputStream = connect.getOutputStream();//以数据流形式提交数据

outputStream.write(params.getBytes());

int response = connect.getResponseCode();

if (response== HttpURLConnection.HTTP\_OK)

{//获取返回数据流

System.out.println(response);

InputStream input=connect.getInputStream();获取返回数据流字节数

BufferedReader in = new BufferedReader(new InputStreamReader(input));//存储字节

String line = null;

StringBuffer sb = new StringBuffer();

while ((line = in.readLine()) != null) {//获取BufferedReader对象中的内容

sb.append(line);

}

return sb.toString();//返回数据流

}

else {

System.out.println(response);

return " ";//获取失败返回“”。

}

} catch (Exception e) {

return "";

}

}

获取参数及解析：

代码如下：

String urlstr="http//";//登陆请求地址

String phonenum=userphone.getText().toString();//获取参数电话号

String password=password.getText().toString();//获取密码

new Thread(){//开启线程

@Override

public void run() {执行线程操作

String result = interne.login(urlstr,phonenum,password);//引用post请求函数获取返回值，interne为一个java类，构造函数继承字该类

System.out.println(result);

try {/解析返回值

JSONObject obj = new JSONObject(result);

int code = obj.getInt("code");//定义参数获取返回值

if (code==1) {

Toast.makeText(getApplication(),"登录成功！",Toast.LENGTH\_SHORT).show();

}

if(code==2){

Toast.makeText(getApplication(),"未授权！",Toast.LENGTH\_SHORT).show();

}

if (code==3){

Toast.makeText(getApplaction().this,"账号密码错误！",Toast.LENGTH\_SHORT).show();

}

}

catch (JSONException e) {

e.printStackTrace();

System.out.println(e.toString());

}//数据解析失败执行数据

}

}.start();

4.总结

通过Get请求和Post请求及其数据解析两个例子让我们更清楚的认识到App终端的数据是如何与web端交互的。Get请求方式主要从web端获取数据，Post请求方式既可以向web端提交App终端数据，又可以从web端获取数据。Get请求和Post请求并无本质优劣，作为应用层的程序员，只有灵活应用Post和Get请求才可以在App开发过程中得心应手。

5.参考文献

[1]安辉 《Android App开发从入门到精通》。

[2] 张洪 巩曰成《基于 JSON 的 Android 移动终端与网络设备管理系统的数据通信》。