

# YoYong

Offer The Valuable Development Notes



## android获取web服务器端session并验证登陆

22,269 views

Posted on **June 13, 2011** by **Charlie** Posted in **android** and tagged **android 用户验证登陆**, **android 获取php session id**, **android 采用session登录一致性**.

传统网页实现用户登陆一般采用session或cookie记录用户基本信息又或者两者结合起来使用。android也可以采用session实现用户登陆验证并记录用户登陆状态时的基本信息，session是在服务器端的;而类似cookie的记录方式，则可以在客户端采用xml文件记录用户基本信息,重要数据则可以加密存放客户端。android实现的session登陆功能与网页请求不同的是，网页形式的一次成功的登陆请求后，再点击其他页面时，session一直是存在的，在一定时间内是有效的；而采用android客户端请求的一次成功登陆后，再次发送新的请求，则会产生新的session，而不是原来的。这就需要记录session的id号，并在整个请求过程中都记录并传递这个id号，才能保证session的一致性。

以获取php session为例，主要思路实现分为客户端与服务器端3个步骤。

附件:[源码下载](#)

### 1.) 客户端 (ANDROID)

建立一个名为GetWebSession的android项目，编写GetWebSession.java,LoginSuccessActivity.java,GetUserInfoActivity.java三个activity类。

1. GetWebSession.java主要是实现布局界面以及发送用户名和密码到php服务器端验证，如果验证成功则跳转到LoginSuccessActivity.java类。GetWebSession.java主要涉及到与服务器端连接请求，对从服务器端返回的json数据(如用户id,session等)进行解析，并存入HashMap,传递到LoginSuccessActivity.java

代码如下:

[view plain](#) [copy to clipboard](#) [print](#) ?

```
package com.login.main;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
```

```

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.ClientProtocolException;
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 org.apache.http.protocol.HTTP;
import org.apache.http.util.EntityUtils;
import org.json.JSONException;
import org.json.JSONObject;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class GetWebSession extends Activity {
    /** Called when the activity is first created. */
    private EditText user;
    private EditText password;
    private Button loginBtn;
    private Button logoutBtn;
    //主要是记录用户会话过程中的一些用户的基本信息
    private HashMap<String, String> session =new HashMap<String, String>();
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        user=(EditText) findViewById(R.id.user);
        password=(EditText) findViewById(R.id.password);
        loginBtn=(Button) findViewById(R.id.loginBtn);
        loginBtn.setOnClickListener(loginClick);
        logoutBtn=(Button) findViewById(R.id.logoutBtn);
        logoutBtn.setOnClickListener(logoutClick);
    }
    OnClickListener loginClick=new OnClickListener() {
        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            if(checkUser()){
                Toast.makeText(v.getContext(), "用户登录成功!", Toast.LENGTH_SHORT).show();
                Context context = v.getContext();
                Intent intent = new Intent(context,
                    LoginSuccessActivity.class);
                //传递session参数,在用户登录成功后为session初始化赋值,即传递HashMap的值
                Bundle map = new Bundle();
                map.putSerializable("sessionid", session);
                intent.putExtra("session", map);
                context.startActivity(intent); // 跳转到成功页面
            }
            else
                Toast.makeText(v.getContext(), "用户验证失败!", Toast.LENGTH_SHORT).show();
        }
    };
    OnClickListener logoutClick=new OnClickListener() {
        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            System.exit(0);
        }
    };
    private boolean checkUser(){
        String username=user.getText().toString();
        String pass=password.getText().toString();
        DefaultHttpClient mHttpClient = new DefaultHttpClient();
        HttpPost mPost = new HttpPost("http://10.0.2.2/web/php/login.php");
        //传递用户名和密码相当于
        //http://10.0.2.2/web/php/login.php?username=''&password=''
        List<BasicNameValuePair> pairs = new ArrayList<BasicNameValuePair>();

```

```

pairs.add(new BasicNameValuePair("username", username));
pairs.add(new BasicNameValuePair("password", pass));
try {
    mPost.setEntity(new UrlEncodedFormEntity(pairs, HTTP.UTF_8));
} catch (UnsupportedEncodingException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
try {
    HttpResponse response = mHttpClient.execute(mPost);
    int res = response.getStatusLine().getStatusCode();
    if (res == 200) {
        HttpEntity entity = response.getEntity();
        if (entity != null) {
            String info = EntityUtils.toString(entity);
            System.out.println("info-----"+info);
            //以下主要是对服务器端返回的数据进行解析
            JSONObject jsonObject=null;
            //flag为登录成功与否的标记,从服务器端返回的数据
            String flag="";
            String name="";
            String userid="";
            String sessionid="";
            try {
                jsonObject = new JSONObject(info);
                flag = jsonObject.getString("flag");
                name = jsonObject.getString("name");
                userid = jsonObject.getString("userid");
                sessionid = jsonObject.getString("sessionid");
            } catch (JSONException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            //根据服务器端返回的标记,判断服务端验证是否成功
            if(flag.equals("success")){
                //为session传递相应的值,用于在session过程中记录相关用户信息
                session.put("s_userid", userid);
                session.put("s_username", name);
                session.put("s_sessionid", sessionid);
                return true;
            }
            else{
                return false;
            }
        }
        else{
            return false;
        }
    } catch (ClientProtocolException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return false;
}
}

```

2. LoginSuccessActivity.java主要获取php的session唯一的标识id以及用户的一些基本信息，session id则作为本次用户登录状态在服务器的唯一标识，即确定用户的唯一状态进行相关操作。

LoginSuccessActivity.java类的方法与GetWebSession.java类似。其主要功能是获取session id后再次发送session id到服务器进行验证，根据封装的session数据验证用户操作权限等。

代码如下：

view plain copy to clipboard print ?

```

package com.login.main;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.HashMap;

```

```

import java.util.List;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.ClientProtocolException;
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 org.apache.http.protocol.HTTP;
import org.apache.http.util.EntityUtils;
import org.json.JSONException;
import org.json.JSONObject;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
public class LoginSuccessActivity extends Activity{
    private HashMap<String, String>session;
    @SuppressWarnings("unchecked")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.login_success);
        //获取从登录成功后界面的传递的参数
        session = (HashMap<String, String>) this.getIntent().
            getBundleExtra("session").getSerializable("sessionid");
        //读取session的基本信息，并显示相应的控件
        String userid_info=session.get("s_userid");
        String username_info=session.get("s_username");
        String session_id=session.get("s_sessionid");
        //显示相应的内容到控件
        TextView userid_show=(TextView) findViewById(R.id.userid_show);
        userid_show.setText(userid_info);
        TextView username_show=(TextView) findViewById(R.id.username_show);
        username_show.setText(username_info);
        TextView sessionid_show=(TextView) findViewById(R.id.sessionid_show);
        sessionid_show.setText(session_id);
        //根据本次session再次获取用户信息
        Button getInfo=(Button) findViewById(R.id.getinfo);
        getInfo.setOnClickListener(getInfoClick);
    }
    OnClickListener getInfoClick=new OnClickListener() {
        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            if(getUserInfo()){
                Context context = v.getContext();
                Intent intent = new Intent(context,
                    GetUserInfoActivity.class);
                //传递session参数, 在用户登录成功后为session初始化赋值, 即传递HashMap的值
                Bundle map = new Bundle();
                map.putSerializable("sessionid", session);
                intent.putExtra("session", map);
                context.startActivity(intent); // 跳转到成功页面
            }else{
                Toast.makeText(v.getContext(), "数据为空!", Toast.LENGTH_SHORT).show();
            }
        }
    };
    private boolean getUserInfo(){
        String sess_username=session.get("s_username");
        String sess_userid=session.get("s_userid");
        String sess_id=session.get("s_sessionid");
        DefaultHttpClient mHttpClient = new DefaultHttpClient();
        HttpPost mPost = new HttpPost("http://10.0.2.2/web/php/getinfo.php");
        List<BasicNameValuePair> pairs = new ArrayList<BasicNameValuePair>();
        pairs.add(new BasicNameValuePair("sess_userid", sess_userid));
        pairs.add(new BasicNameValuePair("sess_username", sess_username));
        pairs.add(new BasicNameValuePair("sess_sessionid", sess_id));
        try {
            mPost.setEntity(new UrlEncodedFormEntity(pairs, HTTP.UTF_8));
        } catch (UnsupportedEncodingException e) {

```

```
// TODO Auto-generated catch block
e.printStackTrace();
}
try {
    HttpResponse response = mHttpClient.execute(mPost);
    int res = response.getStatusLine().getStatusCode();
    if (res == 200) {
        HttpEntity entity = response.getEntity();
        if (entity != null) {
            String info = EntityUtils.toString(entity);
            System.out.println("info-----"+info);
            //以下主要是对服务器端返回的数据进行解析
            JSONObject jsonObject=null;
            //flag为登录成功与否的标记,从服务器端返回的数据
            String flag="";
            String userinfo="";
            String level="";
            String sessionid="";
            try {
                jsonObject = new JSONObject(info);
                flag = jsonObject.getString("flag");
                userinfo = jsonObject.getString("info");
                level = jsonObject.getString("level");
                sessionid = jsonObject.getString("sessionid");
            } catch (JSONException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            //根据服务器端返回的标记,判断服务端验证是否成功
            if(flag.equals("notempty")){
                //为session传递相应的值,用于在session过程中记录相关用户信息
                session.put("info_userinfo", userinfo);
                session.put("info_level", level);
                session.put("info_sessionid", sessionid);
                return true;
            }
            else{
                return false;
            }
        }
        else{
            return false;
        }
    } catch (ClientProtocolException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return false;
}
}
```

3.GetUserInfoActivity.java类是根据用户登录后产生唯一session 标识进行操作获取用户详细信息的类。

代码如下:

```
view plain copy to clipboard print ?

package com.login.main;
import java.util.HashMap;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
public class GetUserInfoActivity extends Activity{
    private HashMap<String, String>session;
    @SuppressWarnings("unchecked")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.get_info);
        //获取从登录成功后界面的再次传递的参数
```

```

session = (HashMap<String, String>) this.getIntent().
getBundleExtra("session").getSerializable("sessionid");
//读取session的基本信息，并显示相应的控件
String session_info=session.get("info_userinfo");
String session_level=session.get("info_level");
String session_id=session.get("info_sessionid");
//显示相应的内容到控件
System.out.println("session_info-----"+session_info);
TextView get_info=(TextView)findViewById(R.id.get_info);
get_info.setText(session_info);
TextView get_level=(TextView)findViewById(R.id.get_level);
get_level.setText(session_level);
TextView get_sessionid=(TextView)findViewById(R.id.get_sessionid);
get_sessionid.setText(session_id);
}
}

```

#### 4.三个布局的xml文件

##### (1.)main.xml

view plain copy to clipboard print ?

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <TextView android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="用户"></TextView>
    <EditText android:layout_height="wrap_content"
        android:text="" android:layout_width="fill_parent"
        android:singleLine="true" android:id="@+id/user" ></EditText>
    <TextView android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="密码"></TextView>
    <EditText android:id="@+id/password"
        android:layout_height="wrap_content"
        android:text="" android:layout_width="fill_parent"
        android:password="true" android:singleLine="true"></EditText>
    <LinearLayout android:layout_height="wrap_content"
        android:layout_width="fill_parent"
        android:orientation="horizontal"
        android:paddingLeft="0dip">

    <TableRow android:layout_width="fill_parent"
        android:layout_height="wrap_content">
        <Button android:layout_height="fill_parent"
            android:layout_width="fill_parent" android:text="登录"
            android:id="@+id/loginBtn"
            android:layout_weight="1"></Button>
        <Button android:layout_height="fill_parent"
            android:layout_width="fill_parent"
            android:text="退出"
            android:id="@+id/logoutBtn"
            android:layout_weight="1"></Button>

    </TableRow> </LinearLayout> </LinearLayout>

```

##### (2.)login\_success.xml

view plain copy to clipboard print ?

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    android:orientation="vertical">
    <LinearLayout android:layout_height="wrap_content"
        android:layout_width="fill_parent"
        android:orientation="horizontal"
        android:paddingLeft="0dip">
    <TextView

```

```

android:layout_height="fill_parent"
android:layout_width="wrap_content"
android:text="用户ID: " >
</TextView>
<TextView android:layout_height="fill_parent"
android:layout_width="fill_parent"
android:text=""
android:id="@+id/userid_show" ></TextView>
</LinearLayout>
<LinearLayout android:layout_height="wrap_content"
android:layout_width="fill_parent"
android:orientation="horizontal"
android:paddingLeft="0dip">
<TextView android:layout_height="fill_parent"
android:layout_width="wrap_content"
android:text="用户名: " ></TextView>
<TextView android:layout_height="fill_parent"
android:layout_width="fill_parent"
android:text=""
android:id="@+id/username_show" ></TextView>
</LinearLayout>
<LinearLayout android:layout_height="wrap_content"
android:layout_width="fill_parent"
android:orientation="horizontal"
android:paddingLeft="0dip">
<TextView android:layout_height="fill_parent"
android:layout_width="wrap_content"
android:text="本次会话: " ></TextView>
<TextView android:layout_height="fill_parent"
android:layout_width="fill_parent"
android:text=""
android:id="@+id/sessionid_show" ></TextView>
</LinearLayout>
<LinearLayout android:layout_height="wrap_content"
android:layout_width="fill_parent"
android:orientation="horizontal"
android:paddingLeft="0dip">
<Button android:layout_height="fill_parent"
android:layout_width="wrap_content"
android:id="@+id/getinfo"
android:text="根据本次会话再次获取用户信息"
></Button>
</LinearLayout>
</LinearLayout>

```

### (3.)get\_info.xml

view plain copy to clipboard print ?

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical">
<LinearLayout android:layout_height="wrap_content"
android:layout_width="fill_parent"
android:orientation="horizontal"
android:paddingLeft="0dip">
<TextView android:layout_height="fill_parent"
android:layout_width="wrap_content"
android:text="用户信息: " >
</TextView>
<TextView android:layout_height="fill_parent"
android:layout_width="fill_parent"
android:text=""
android:id="@+id/get_info" ></TextView>
</LinearLayout>
<LinearLayout android:layout_height="wrap_content"
android:layout_width="fill_parent"
android:orientation="horizontal"
android:paddingLeft="0dip">
<TextView android:layout_height="fill_parent"
android:layout_width="wrap_content"
android:text="用户级别: " ></TextView>

```



```

<TextView android:layout_height="fill_parent"
android:layout_width="fill_parent"
android:text=""
android:id="@+id/get_level" ></TextView>
</LinearLayout>

<LinearLayout android:layout_height="wrap_content"
android:layout_width="fill_parent"
android:orientation="horizontal"
android:paddingLeft="0dip">
<TextView android:layout_height="fill_parent"
android:layout_width="wrap_content"
android:text="本次会话: " ></TextView>
<TextView android:layout_height="fill_parent"
android:layout_width="fill_parent" android:text=""
android:id="@+id/get_sessionid" ></TextView>
</LinearLayout>
<LinearLayout android:layout_height="wrap_content"
android:layout_width="fill_parent"
android:orientation="horizontal"
android:paddingLeft="0dip"> </LinearLayout> </LinearLayout>

```

## 2.) 服务器端 (PHP)

php服务器端主要有三个文件,conn.php,login.php和getinfo.php。

1. conn.php是连接数据库的配置文件。

2. login.php主要是用来验证android客户端发送的请求,请求成功则返回flag='success'的状态标识,采用数组记录用户基本信息,存储用户数据到session,并且记录本次产生的session id。用户基本数据及本次session产生的id均封装成json格式(json\_encode(\$arr)),发送android客户端。产生本次session id的方法

```
$sessionid=session_id();//注意没有参数
```

具体代码如下：

view plain copy to clipboard print ?

```

<?php
header("Content-Type: text/html; charset=utf-8") ;
//包含数据库连接文件
include('conn.php');
session_start();
$username = htmlspecialchars($_POST["username"]);
$password=$_POST["password"];
mysql_query("set names utf8");
//检测用户名及密码是否正确
$check_query = mysql_query("select id ,name from user where name='$username'
password='$password' limit 1");
$arr=array();//空的数组,该数组主要是格式化数据并封装成json格式发送到客户端
if($result = mysql_fetch_array($check_query)){
//登录成功
$_SESSION['username'] = $result['name'];
$_SESSION['userid'] = $result['id'];
//获取当前session id
$sessionid=session_id();
$_SESSION['$sessionid'] = $sessionid;
$arr = array(
'flag'=>'success',
'name'=>$result['name'],
'userid'=>$result['id'],
'sessionid'=>$sessionid
);
//封装json,如果php版本低于5.2,则不支持json_encode()方法,
//可以参考本文件夹中php_json_encode.php中php_json_encode()方法代替
json_encode();
echo json_encode($arr);
} else {
$arr = array(

```



```

'flag'=>'error',
'name'=>'',
'userid'=>'',
'sessionid'=>'')
); //封装json,如果php版本低于5.2,则不支持json_encode()方法,
//可以参考本文件夹中php_json_encode.php中php_json_encode()方法代替
json_encode();
echo json_encode($arr);
}
?>

```

3. getinfo.php文件主要是用户再次查询信息验证session，而不是重新产生session，以记录用户状态。通过验证flag是否为empty判断数据是否显示。最后封装成json发送到客户端

获取本次session的方法：

```
$sessionid=$_POST["sess_sessionid");//获取android客户端的sessionid
```

```
session_id($sessionid);//有参数
```

```
session_start();//启动session
```

具体代码如下：

view plain copy to clipboard print ?

```

<?php
header("Content-Type: text/html; charset=utf-8") ;
include('conn.php');
//获取从客户端LoginSuccessActivity类传递的参数
$userid=$_POST["sess_userid"];
$username=$_POST["sess_username"];
//获取客户端传递的session标识
$sessionid=$_POST["sess_sessionid"];
session_id($sessionid);
//将会根据session id获得原来的session
session_start();
//获取服务器端原来session记录的username,并且根据客户端传过来的username比较进行验证
操作
$sess_username=$_SESSION['username'];
if($username==$sess_username){
mysql_query("set names utf8");
//查询用户基本信息
$check_query = mysql_query("select userinfo,level from info where userid='
$arr=array();//空的数组
if($result = mysql_fetch_array($check_query)){
$arr = array(
'flag'=>'notempty',
'info'=>$result['userinfo'],
'level'=>$result['level'],
'sessionid'=>$sessionid
);
echo json_encode($arr);
}
} else {
$arr = array(
'flag'=>'empty',
'name'=>'',
'userid'=>'',
'sessionid'=>$sessionid
);
echo json_encode($arr);
}
?>

```

### 3.) 数据库端 (MYSQL)

采用mysql建立数据库，建立两个简单的数据表：user和info。

view plain copy to clipboard print ?

```

/*
MySQL Data Transfer
Source Host: localhost
Source Database: login
Target Host: localhost
Target Database: login
Date: 2011-6-14 11:10:46
*/

SET FOREIGN_KEY_CHECKS=0;
-----
-- Table structure for info
-----

CREATE TABLE `info` (
  `id` int(12) NOT NULL AUTO_INCREMENT,
  `userid` int(12) DEFAULT NULL,
  `userinfo` varchar(100) DEFAULT NULL,
  `level` int(2) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `userid` (`userid`),
  CONSTRAINT `userid` FOREIGN KEY (`userid`) REFERENCES `user` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;

-----
-- Table structure for user
-----

CREATE TABLE `user` (
  `id` int(12) NOT NULL AUTO_INCREMENT,
  `name` varchar(20) DEFAULT NULL,
  `password` varchar(20) DEFAULT NULL,
  `status` int(2) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;

-----
-- Records
-----

INSERT INTO `info` VALUES ('1', '1', 'charlie is a developer.', '1');
INSERT INTO `info` VALUES ('2', '2', 'william is a boss.', '20');
INSERT INTO `user` VALUES ('1', 'charlie', 'password', '1');
INSERT INTO `user` VALUES ('2', 'william', 'mypassword', '2');

```

运行效果如图：

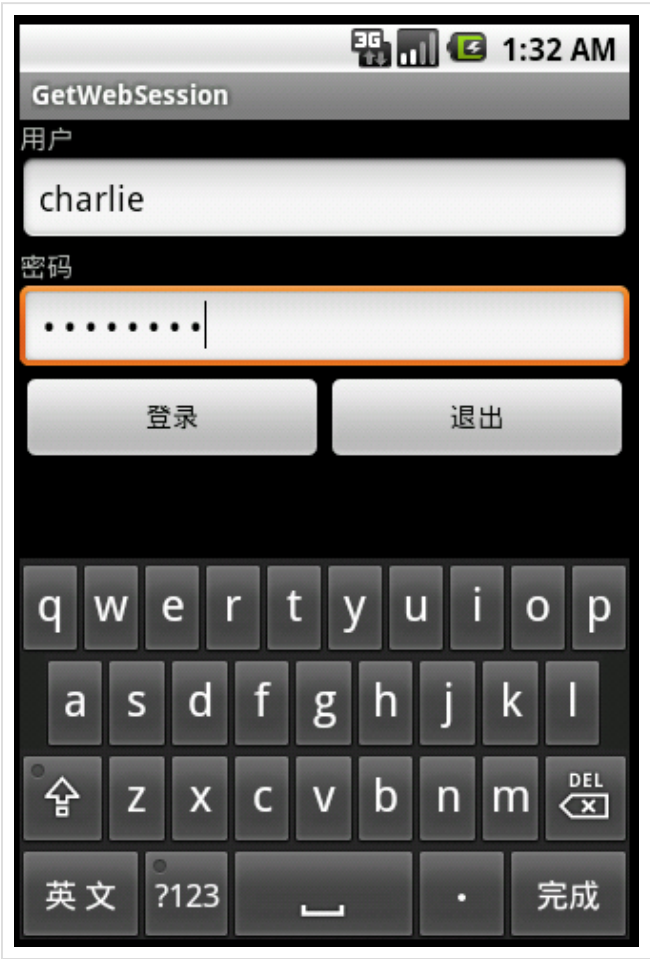


图 -1 GetWebSession.java类的布局



图 -2 LoginSuccessActivity.java类获取的session id以及用户基本信息



图 -3 GetWebSession.java获取用户详细信息及本次session的一致性


This entry was posted in [android](#) and tagged [android 用户验证登陆](#), [android 获取php session id](#), [android 采用session登录一致性](#) by [Charlie](#). Bookmark the [permalink](#) [<http://www.yoyong.com/archives/178>].

**歌倫比亞學英語 每天只要45分鐘** 

宅在家裡，1天花45分鐘，就能學好英文，前3個月只要990，再送3千

〇 〇

11 THOUGHTS ON “ANDROID获取WEB服务器端SESSION并验证登陆”



YueFengQing  
on **June 21, 2011 at 3:16 pm** said:

this is perfect | 佩服 呵呵



liaoweibin

on **October 25, 2011 at 2:41 pm** said:

做得很好，佩服。。。我把你的代码下载去运行时，出现错误，a jsonobject text must begin with '{',不知道为什么，



Charlie

on **October 25, 2011 at 3:07 pm** said:

查看打印的内容(System.out.println("info——"+info);)，应该是返回的数据格式出错了，无法json解析。



liaoweibin

on **November 15, 2011 at 3:47 pm** said:

确实是返回数据格式错了，但是使用你的源码运行的，不知道为什么出错，可否留下个邮箱或者qq号，方便请教问题谢谢



liaoweibin

on **November 15, 2011 at 3:52 pm** said:

确实是返回数据格式错误，可否留个邮箱，或者QQ，方便请教问题，谢谢！



Charlie

on **November 15, 2011 at 9:50 pm** said:

email:zhuguocan@gmail.com



Sean.Hwang

on **November 30, 2011 at 9:12 am** said:

找了好久这个,下了源码试试,谢谢分享



zcsln

on **February 16, 2012 at 2:43 pm** said:

怎么回事啊，我B页输出服务器端保存的SESSION一直为空。求教了



Charlie

on **February 16, 2012 at 5:04 pm** said:

回传过来的sessionid标识(字符串)要正确，第二个页面获取session的顺序要正确，  
session\_id(\$sessionid);不要带其他符号，如不能session\_id('\$sessionid');



zcsln

on **February 18, 2012 at 9:29 am** said:

还是不行啊，B页的SESSION数组还是为空。我详细核对过了真不清楚哪儿出了问题。



Tony

on **March 2, 2012 at 10:10 am** said:

非常感谢，正是需要的~

linezing