REPORT DATA TRANSFER FUNCTIONS

Project overview

This is describe simple client-server communicate between an Android app and a web service.

For this demo, I’ll sending and receiving data for some data transfer functions which you sent to me some days ago.

Edit \res\layout\activity\_main.xml

The interface is very simple, having two labels and two edit controls for latitude and longitude. It will also have three botton – one to GET, one to POST, and one to clear the controls.

<?xml version=*"1.0"* encoding=*"utf-8"*?>

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

android:id=*"@+id/tableLayout1"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

android:shrinkColumns=*"\*"*

android:stretchColumns=*"\*"* >

<TableRow

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"* >

<TextView

style=*""*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/latitude"* >

</TextView>

<EditText

android:id=*"@+id/latitude"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:inputType=*"textCapWords"*

android:layout\_span=*"2"* />

</TableRow>

<TableRow

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"* >

<TextView

style=*""*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/lagitude"* >

</TextView>

<EditText

android:id=*"@+id/lagitude"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:inputType=*"textCapWords"*

android:layout\_span=*"2"* />

</TableRow>

<TableRow

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"* >

<Button

android:id=*"@+id/bn\_retrieve"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:onClick=*"getSearchAddress"*

android:text=*"@string/get"* />

<Button

android:id=*"@+id/bn\_post"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:onClick=*"getNearestDriver"*

android:text=*"@string/post"* />

<Button

android:id=*"@+id/bn\_clear"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:onClick=*"clearControl"*

android:text=*"@string/clear"* />

</TableRow>

</TableLayout>

Take a look for the WebServiceTaskManager.java file

The AsyncTask class descendant allow a process to run in a separate thread. The communication with the web service occurs in WebServiceTaskManager is “doInBackground()” code.

**import** java.io.BufferedReader;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.InputStreamReader;

**import** java.util.ArrayList;

**import** org.apache.http.HttpResponse;

**import** org.apache.http.NameValuePair;

**import** org.apache.http.client.HttpClient;

**import** org.apache.http.client.entity.UrlEncodedFormEntity;

**import** org.apache.http.client.methods.HttpGet;

**import** org.apache.http.client.methods.HttpPost;

**import** org.apache.http.client.utils.URLEncodedUtils;

**import** org.apache.http.impl.client.DefaultHttpClient;

**import** org.apache.http.message.BasicNameValuePair;

**import** org.apache.http.params.BasicHttpParams;

**import** org.apache.http.params.HttpConnectionParams;

**import** org.apache.http.params.HttpParams;

**import** android.app.ProgressDialog;

**import** android.content.Context;

**import** android.os.AsyncTask;

**import** android.util.Log;

**public** **abstract** **class** WebServiceTaskManager **extends**

AsyncTask<String, Integer, String> {

**public** **static** **final** **int** *POST\_TASK* = 1;

**public** **static** **final** **int** *GET\_TASK* = 2;

**public** **static** **final** String *URL* = "http://104.42.107.187:82/api/";

**private** **static** **final** String *TAG* = "WebServiceTaskManager";

// connection timeout, in milliseconds (waiting to connect)

**private** **static** **final** **int** *CONN\_TIMEOUT* = 6000;

// socket timeout, in milliseconds (waiting for data)

**private** **static** **final** **int** *SOCKET\_TIMEOUT* = 10000;

**private** **int** taskType = *GET\_TASK*;

**private** Context mContext = **null**;

**private** String processMessage = "Processing...";

**private** ArrayList<NameValuePair> params = **new** ArrayList<NameValuePair>();

**private** ProgressDialog pDlg = **null**;

**private** String response;

**public** String getResponse() {

**return** response;

}

**public** **void** setResponse(String response) {

**this**.response = response;

}

**public** WebServiceTaskManager(**int** taskType, Context mContext,

String processMessage) {

**this**.taskType = taskType;

**this**.mContext = mContext;

**this**.processMessage = processMessage;

}

**public** WebServiceTaskManager(**int** taskType, Context mContext) {

**this**.taskType = taskType;

**this**.mContext = mContext;

}

**public** **void** addNameValuePair(String name, String value) {

params.add(**new** BasicNameValuePair(name, value));

}

@Override

**protected** String doInBackground(String... params) {

String url = params[0];

String result = "";

HttpResponse response = doResponse(url);

**if** (response == **null**) {

System.*out*.println(result + " null");

**return** result;

} **else** {

**try** {

result = inputStreamToString(response.getEntity().getContent());

} **catch** (IllegalStateException e) {

Log.*e*(WebServiceTaskManager.*TAG*, e.getLocalizedMessage(), e);

} **catch** (IOException e) {

Log.*e*(WebServiceTaskManager.*TAG*, e.getLocalizedMessage(), e);

}

}

**return** result;

}

@SuppressWarnings("deprecation")

**private** **void** showProgressDialog() {

pDlg = **new** ProgressDialog(mContext);

pDlg.setMessage(processMessage);

pDlg.setProgressDrawable(mContext.~~getWallpaper~~());

pDlg.setProgressStyle(ProgressDialog.*STYLE\_SPINNER*);

pDlg.setCancelable(**false**);

pDlg.show();

}

@Override

**protected** **void** onPreExecute() {

showProgressDialog();

}

@Override

**protected** **void** onPostExecute(String response) {

pDlg.dismiss();

handleResponse(response);

}

// Establish connection and socket (data retrieval) timeouts

**private** HttpParams getHttpParams() {

HttpParams htpp = **new** BasicHttpParams();

HttpConnectionParams.*setConnectionTimeout*(htpp, *CONN\_TIMEOUT*);

HttpConnectionParams.*setSoTimeout*(htpp, *SOCKET\_TIMEOUT*);

**return** htpp;

}

**private** HttpResponse doResponse(String url) {

// Use our connection and data timeouts as parameters for our

// DefaultHttpClient

HttpClient httpclient = **new** DefaultHttpClient(getHttpParams());

HttpResponse response = **null**;

**try** {

**switch** (taskType) {

**case** *POST\_TASK*:

HttpPost httppost = **new** HttpPost(url);

// Add parameters

httppost.setEntity(**new** UrlEncodedFormEntity(params));

response = httpclient.execute(httppost);

**break**;

**case** *GET\_TASK*:

**if** (!params.isEmpty()) {

String paramString = URLEncodedUtils.*format*(params, "utf-8");

url += "?" + paramString;

}

HttpGet httpget = **new** HttpGet(url);

response = httpclient.execute(httpget);

**break**;

}

} **catch** (Exception e) {

Log.*e*(*TAG*, e.getLocalizedMessage(), e);

}

**return** response;

}

**private** String inputStreamToString(InputStream is) {

String line = "";

StringBuilder total = **new** StringBuilder();

// Wrap a BufferedReader around the InputStream

BufferedReader rd = **new** BufferedReader(**new** InputStreamReader(is));

**try** {

// Read response until the end

**while** ((line = rd.readLine()) != **null**) {

total.append(line);

}

} **catch** (IOException e) {

Log.*e*(WebServiceTaskManager.*TAG*, e.getLocalizedMessage(), e);

}

// Return full string

**return** total.toString();

}

**public** **abstract** **void** handleResponse(String response);

}

Edit the MainActivity.java file

Have some functions transfer data here.

**import** flexjson.JSONSerializer;

**import** android.app.Activity;

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.EditText;

**import** android.widget.Toast;

**public** **class** MainActivity **extends** Activity {

**private** EditText edLagitude;

**private** EditText edLatitude;

@Override

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

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

edLagitude = (EditText) findViewById(R.id.*lagitude*);;

edLatitude = (EditText) findViewById(R.id.*latitude*);

}

**public** **void** getNearestDriver(View vw) {

String url = WebServiceTaskManager.*URL* + "NearestDriver";

WebServiceTaskManager wst = **new** WebServiceTaskManager(WebServiceTaskManager.*POST\_TASK*, **this**, "Retrieving the nearest driver ...") {

@Override

**public** **void** handleResponse(String response) {

showResponse(response);

}

};

Location location = **new** Location();

String latitude = edLatitude.getText().toString();

String lagitude = edLagitude.getText().toString();

**if**(lagitude.length() == 0 || latitude.length() == 0) {

lagitude = "0";

latitude = "0";

}

location.setLat(Double.*parseDouble*(latitude));

location.setLgn(Double.*parseDouble*(lagitude));

String json = **new** JSONSerializer().exclude("\*.class").serialize(

location);

wst.addNameValuePair("", json);

wst.execute(**new** String[] { url });

}

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

edLagitude.setText("");

edLatitude.setText("");

}

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

String url = WebServiceTaskManager.*URL* + "NearbyPlaces";

WebServiceTaskManager wst = **new** WebServiceTaskManager(WebServiceTaskManager.*GET\_TASK*, **this**, "Retrieving the nearly places ...") {

@Override

**public** **void** handleResponse(String response) {

showResponse(response);

}

};

wst.addNameValuePair("prefix", "se137jj");

wst.execute(**new** String[] { url });

}

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

String url = WebServiceTaskManager.*URL* + "SearchAddress";

WebServiceTaskManager wst = **new** WebServiceTaskManager(WebServiceTaskManager.*GET\_TASK*, **this**, "Searching address ...") {

@Override

**public** **void** handleResponse(String response) {

showResponse(response);

}

};

wst.addNameValuePair("prefix", "se137jj");

wst.execute(**new** String[] { url });

}

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

String url = WebServiceTaskManager.*URL* + "Quotation";

WebServiceTaskManager wst = **new** WebServiceTaskManager(WebServiceTaskManager.*POST\_TASK*, **this**, "Searching address ...") {

@Override

**public** **void** handleResponse(String response) {

showResponse(response);

}

};

Quotation quotation = **new** Quotation();

quotation.setCustid(0);

quotation.setPickLat(0.0);

quotation.setPickLong(0.0);

quotation.setDoffLat(0.0);

quotation.setDoffLong(0.0);

quotation.setBookingdate("0001-01-01T00:00:00");

quotation.setPaq(0);

quotation.setBags(0);

String json = **new** JSONSerializer().exclude("\*.class").serialize(

quotation);

wst.addNameValuePair("", json);

wst.execute(**new** String[] { url });

}

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

String url = WebServiceTaskManager.*URL* + "SignIn";

WebServiceTaskManager wst = **new** WebServiceTaskManager(WebServiceTaskManager.*POST\_TASK*, **this**, "Searching address ...") {

@Override

**public** **void** handleResponse(String response) {

showResponse(response);

}

};

SignIn signIn = **new** SignIn();

String json = **new** JSONSerializer().exclude("\*.class").serialize(

signIn);

wst.addNameValuePair("", json);

wst.execute(**new** String[] { url });

}

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

String url = WebServiceTaskManager.*URL* + "SignUp";

WebServiceTaskManager wst = **new** WebServiceTaskManager(WebServiceTaskManager.*POST\_TASK*, **this**, "Searching address ...") {

@Override

**public** **void** handleResponse(String response) {

showResponse(response);

}

};

SignUp signUp = **new** SignUp();

String json = **new** JSONSerializer().exclude("\*.class").serialize(

signUp);

wst.addNameValuePair("", json);

wst.execute(**new** String[] { url });

}

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

String url = WebServiceTaskManager.*URL* + "ElectronicPayment";

WebServiceTaskManager wst = **new** WebServiceTaskManager(WebServiceTaskManager.*POST\_TASK*, **this**, "Searching address ...") {

@Override

**public** **void** handleResponse(String response) {

showResponse(response);

}

};

ElectronicPayment electronicPayment = **new** ElectronicPayment();

electronicPayment.setCustID(0);

electronicPayment.setAmount(0.0);

String json = **new** JSONSerializer().exclude("\*.class").serialize(

electronicPayment);

wst.addNameValuePair("", json);

wst.execute(**new** String[] { url });

}

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

String url = WebServiceTaskManager.*URL* + "SaveBooking";

WebServiceTaskManager wst = **new** WebServiceTaskManager(WebServiceTaskManager.*POST\_TASK*, **this**, "Searching address ...") {

@Override

**public** **void** handleResponse(String response) {

showResponse(response);

}

};

SaveBooking saveBooking = **new** SaveBooking();

saveBooking.setCusid(0);

saveBooking.setRoutedistance(0.0);

saveBooking.setVehTypeID(0);

saveBooking.setTravelTime(0.0);

saveBooking.setTotalfare(0.0);

saveBooking.setFare(0.0);

saveBooking.setPkLat(0.0);

saveBooking.setPkLong(0.0);

saveBooking.setBookingdate("0001-01-01T00:00:00");

saveBooking.setPaq(0);

saveBooking.setBags(0);

saveBooking.setDoLat(0.0);

saveBooking.setDoLong(0.0);

String json = **new** JSONSerializer().exclude("\*.class").serialize(

saveBooking);

wst.addNameValuePair("", json);

wst.execute(**new** String[] { url });

}

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

String url = WebServiceTaskManager.*URL* + "CurrentBookings";

WebServiceTaskManager wst = **new** WebServiceTaskManager(WebServiceTaskManager.*GET\_TASK*, **this**, "Searching address ...") {

@Override

**public** **void** handleResponse(String response) {

showResponse(response);

}

};

wst.addNameValuePair("CustID", "0");

wst.addNameValuePair("DeviceID", **null**);

wst.execute(**new** String[] { url });

}

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

String url = WebServiceTaskManager.*URL* + "JourneyHistory";

WebServiceTaskManager wst = **new** WebServiceTaskManager(WebServiceTaskManager.*GET\_TASK*, **this**, "Searching address ...") {

@Override

**public** **void** handleResponse(String response) {

showResponse(response);

}

};

wst.addNameValuePair("Phone", **null**);

wst.execute(**new** String[] { url });

}

**public** **void** showResponse(String response) {

Toast.*makeText*(**this**, response, Toast.*LENGTH\_SHORT*).show();

}

}

Update the AndroidManifest.xml

<?xml version=*"1.0"* encoding=*"utf-8"*?>

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

package=*"com.example.taxi"*

android:versionCode=*"1"*

android:versionName=*"1.0"* >

<uses-sdk

android:minSdkVersion=*"16"*

android:targetSdkVersion=*"19"* />

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

<application

android:allowBackup=*"true"*

android:icon=*"@drawable/ic\_launcher"*

android:label=*"@string/app\_name"*

android:theme=*"@style/AppTheme"* >

<activity

android:name=*".MainActivity"*

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>

Demo

  