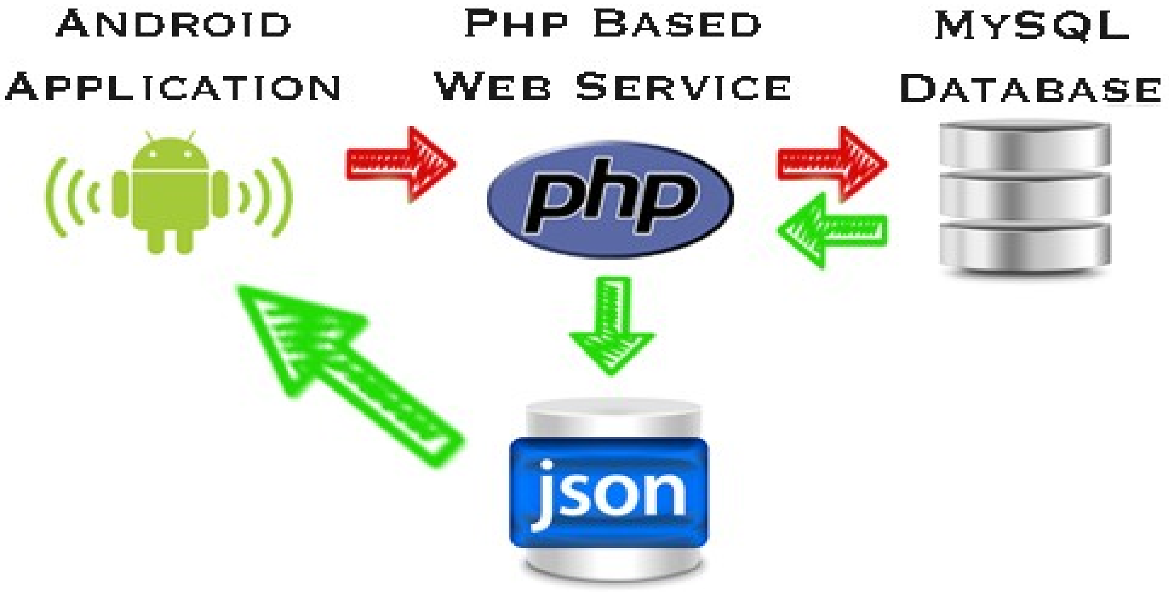
Connection Android à une base distante Mysql



I. Coté serveur

1.Base de données: contenant une table Client

2. Fichier PHP: envoie des requêtes sql à la base de données

récupère le résultat des requêtes, ensuite l'encode en JSON

grâce à l’appel de la fonction **json_encode()** :

Connexion_mysql.php

```
<?php
// on se connecte à notre base  pour recuperer les data

  $base = mysql_connect ('localhost', 'root', '') or die(mysql_error());;
mysql_select_db (‘Base', $base) or die(mysql_error()) ;
 $req=mysql_query("SELECT * FROM client WHERE ville= '".$_POST['ville']."%'");

          while ($row=mysql_fetch_array($req)) {   $output[]=$row;    }
//on encode en JSON
print(json_encode($output)); // JSON retourné au client
mysql_free_result ($req);   mysql_close();

<?php
```

```
$con = mysql_connect ("localhost", "root", "") or die("Error CONNECT") ;
mysql_select_db ("base", $con) or die("PROBLEME BASE")  ;
 $result=mysql_query("SELECT * FROM client ");
//$data = array();
$data["client"] = array();
 if ($result->num_rows > 0) {
        //Converting the results into an associative array
       while ($row = mysql_fetch_assoc($result)){
          $jsonArrayItem = array();
          $jsonArrayItem['nom'] = $row['nom'];
          $jsonArrayItem['ville'] = $row['ville'];
          $jsonArrayItem['solde'] = $row['solde'];
          //append the above created object into the main array.
        //  array_push($data, $jsonArrayItem);
        array_push($data["client"], $jsonArrayItem);
        }
 mysql_free_result ($result);  mysql_close();
print json_encode($data); } // afficher  la reponse JSON
?>
{" client":[{"nom":"nom1","solde":100},{"nom":"nom2","solde":200}] }
?>
```

II. Coté Client

Au niveau du Client android, on demande au serveur de renvoyer la liste des clients habitant une ville .

Cela se fait en utilisant une classe principale qui va hériter de **ListActivity** dont le contenu est le suivant:

- Envoi des commandes *HttpClient* en indiquant le chemin du fichier PHP à interroger
- Conversion du flux reçu en chaine de caractères
- Récuperation des données codées sous forme d'objet JSON

- Affichage dans une liste ListView

```
package com.exa;

// bibliothèques natives d'Android
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.ArrayList;
import org.apache.http.HttpEntity;
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.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import android.app.ListActivity;
import android.net.ParseException;
import android.os.Bundle;
import android.util.Log;
import android.widget.ArrayAdapter;
import android.widget.Toast;

public class ConnexionSQLActivity extends ListActivity {
   /** Called when the activity is first created. */
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      //setContentView(R.layout.main);
     String result = null;

  // Paramètres de requête  POST

  ArrayList<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>();


      nameValuePairs.add(new BasicNameValuePair("ville",“Tana"));

 //  nameValuePairs.add(new BasicNameValuePair("nom", “koto"));



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

// Envoi d'une requête au serveur
```

```
    try{  //commandes httpClient
    HttpClient httpclient = new DefaultHttpClient();

  HttpPost httppost = new HttpPost("http://IP/connect.php");
   httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
```

// **Récupération d'une réponse**

```
      HttpResponse response = httpclient.execute(httppost);
       HttpEntity entity = response.getEntity();
       InputStream is = entity.getContent();
    }

catch(Exception e){
     Log.i("taghttppost",""+e.toString());
 Toast.makeText(getBaseContext(),e.toString(),Toast.LENGTH_LONG).show();
      }
```

//conversion de la réponse en chaine de caractère

```
       try {
        BufferedReader reader = new BufferedReader(new

         InputStreamReader(is,"UTF-8"));
        StringBuilder sb  = new StringBuilder();
         String line = null;
       while ((line = reader.readLine()) != null) {

       sb.append(line + "\n");}
        is.close();
        result = sb.toString();
       }
       catch(Exception e)  {
        Log.i("tagconvertstr",""+e.toString());
       }
```

//recuperation et parser des donnees json

[{"nom":"nom1","solde":100},{"nom":"nom2","solde":20}]

```
       try{
          JSONObject json_data=null;

         JSONArray jArray = new JSONArray(result);

            for(int i=0;i<jArray.length();i++){
                  json_data = jArray.getJSONObject(i);

                  String nom=json_data.getString("nom");

          int solde=Integer.parseInt(json_data.getString("solde"));
```

```
                donnees.add(nom);

                 System.outl.println( nom+"  "+solde);
              }
        setListAdapter(new ArrayAdapter<String>(this, android.R.layout.simple_expandable_list_item_1,
donnees));
            }
   catch(JSONException e){ Log.i("tagjsonexp",""+e.toString());
          }   catch (ParseException e)

              {Log.i("tagjsonpars",""+e.toString()) }
   }

}
```

```
{" client":[{"nom":"nom1","solde":100},{"nom":"nom2","solde":200}] }
```

On peut ranger les données json dans un tableau d'objets

```
 public static ArrayList<Client> getClients() {

ArrayList<Client> clients = new ArrayList<Client>();

        try {

            …………………………………………..

          // On récupère le JSON complet

          JSONObject jsonObject = new JSONObject(result);

          // On récupère le tableau d'objets qui nous concernent

          JSONArray array = new JSONArray(jsonObject.getString("client"));

          // Pour tous les objets on récupère les infos

          for (int i = 0; i < array.length(); i++) {

             // On récupère un objet JSON du tableau

             JSONObject obj = new JSONObject(array.getString(i));

             // On fait le lien Personne - Objet JSON

             Client client = new Client();

             client.setNom(obj.getString("nom"));

             client.setVille(obj.getString("ville"));

             // On ajoute la personne à la liste
            clients.add(personne);

          }

        } catch (Exception e) { e.printStackTrace();  }

        // On retourne la liste des personnes

        return personnes;

    }
```

<u>POUR LA METHODE GET</u>

```
String link = "http://10.0.2.2:81/test/test_url.php?ville=tana&nom=rabe";

HttpClient client = new DefaultHttpClient();

           HttpGet request = new HttpGet();

          request.setURI(new URI(link));

     // Récupération d'une réponse

       HttpResponse response = client.execute(request);

       HttpEntity entity = response.getEntity();

    InputStream is = entity.getContent();

………………….
```

Dans le manifeste, bien ajouter la ligne

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

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

autrement le programme ne sera pas autorisé à se connecter sur internet donc à contacter le PHP (java.net.SocketException: Permission denied)

<u>EXEMPLE 1</u> : envoyer une requête HTTP Post au serveur en utilisant l'objet HttpUrlConnection depuis une application Android

Introduction

Nous allons créer une connexion entre un client Android et le serveur à une certaine période puis envoyer ou recevoir la demande de données de l'application Android au serveur

Dans la classe **MainActivity.java du projet Android :**

- Création d'une méthode asynchrone appelée **SendPostRequest ()** qui va exécuter le processus en arrière-plan.

- Définir une URL dans la méthode Asynchronous et ajouter dans un objet JSONObject des donnes sous forme de clé-valeur
- Créer un JsonObject et un objet HttpURLConnection avec la méthode URL.openConnection() utilisé pour envoyer et récevoir les données à travers le web et définir aussi connection timeout, le type de la methode type et l'objet HttpURLConnectionn doit être configuré avec **setDoInput(true**).)
- Créer une méthode pour convertir (encoder) JSONObject en format de string de paramètres de requête (param1 & para2 & param3 ..) : **getPostDataString ().**
- _Retouner la reponse dans la méthode **onPostExecute()**

Premièrement , on encode la chaîne url de JSONObject. Cette URL envoie le serveur d'obtenir la reponse via getInputStream(). Et on lit la reponse avec l'objet StringBuffer et retourne la réponse sous forme de chaîne dans la méthode **onPostExcute()**

```
import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.OutputStream;

import java.io.OutputStreamWriter;

import java.net.HttpURLConnection;

import java.net.URL;

import java.net.URLEncoder;

import java.util.ArrayList;

import java.util.Iterator;

// import javax.net.ssl.HttpsURLConnection;

import org.apache.http.HttpEntity;

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.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import android.app.Activity;

import android.app.ListActivity;

import android.net.ParseException;

import android.os.AsyncTask;

import android.os.Bundle;

import android.util.Log;

import android.widget.ArrayAdapter;

import android.widget.Toast;

public class MainActivity extends  Activity {

    @Override

   protected void onCreate(Bundle savedInstanceState) {

      super.onCreate(savedInstanceState);

      setContentView(R.layout.activity_main);

      new SendPostRequest().execute();  // METHODE POST

   }

   public class SendPostRequest extends AsyncTask<String, Void, String> {

      protected void onPreExecute(){}

       protected String doInBackground(String... arg0) {

        try {


    URL url = new URL("http://IP/post.php"); // here is your URL path

    // URL url = new URL("https://IP/post.jsp");

   //  Définition des paramètres

         JSONObject postDataParams = new JSONObject();
```

```
          postDataParams.put("name", "abc");

          postDataParams.put("email", "abc@gmail.com");

         // Log.e("params",postDataParams.toString());

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

 //  on peut utiliser HttpsURLConnection  pour sécuriser la

// communication  avec HTTPS   (le trafic Web)

          conn.setReadTimeout(15000 /* milliseconds */);

          conn.setConnectTimeout(15000 /* milliseconds */);

          conn.setRequestMethod("POST");

          conn.setDoInput(true);

          conn.setDoOutput(true);// définir la sortie sur True : obligatoire pour POST pour pouvoir envoyer
les données

// créer   le flux de sortie du serveur  et l'envoyer  puis vider et fermer le flux lorsque on a terminé.

 OutputStream os = conn.getOutputStream();   //   envoi

             BufferedWriter writer = new BufferedWriter(

              new OutputStreamWriter(os, "UTF-8"));

             writer.write(getPostDataString(postDataParams));

              writer.flush();

             writer.close();

             os.close();

int responseCode=conn.getResponseCode();

 if (responseCode == HttpsURLConnection.HTTP_OK) {

   // pour lire les octets de InputStream et les décoder en caractères:

                BufferedReader in=new BufferedReader(new

              InputStreamReader( conn.getInputStream()));  //  reponse

                StringBuffer sb = new StringBuffer("");

                String line="";

                 while((line = in.readLine()) != null) {

                    sb.append(line);
                    break;

                }

                 in.close();

                return sb.toString();

              }               else { return new String("false : "+responseCode);   }

          }

   catch(Exception e){  return new String("Exception: " + e.getMessage());  }

       }

       @Override

      protected void onPostExecute(String result) {

        Toast.makeText(getApplicationContext(), result,

                Toast.LENGTH_LONG).show()   }

public String getPostDataString(JSONObject params) throws Exception {

       StringBuilder result = new StringBuilder();

      boolean first = true;

      Iterator<String> itr = params.keys();

      while(itr.hasNext()){


          String key= itr.next();

          Object value = params.get(key);


          if (first)

              first = false;

          else

              result.append("&");


          result.append(URLEncoder.encode(key, "UTF-8"));

          result.append("=");
          result.append(URLEncoder.encode(value.toString(), "UTF-8"));

      }

      return result.toString();

   }}
```

METHODE POST

```
import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.OutputStream;

import java.io.OutputStreamWriter;

import java.net.HttpURLConnection;

import java.net.URL;

import java.net.URLEncoder;

import java.util.ArrayList;

import java.util.Iterator;

import javax.net.ssl.HttpsURLConnection;

import org.apache.http.HttpEntity;

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.impl.client.DefaultHttpClient;

import org.apache.http.message.BasicNameValuePair;
import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import android.app.Activity;

import android.app.ListActivity;

import android.net.ParseException;

import android.os.AsyncTask;

import android.os.Bundle;

import android.util.Log;

import android.widget.ArrayAdapter;

import android.widget.TextView;

import android.widget.Toast;

public class MainTest_UrlConnect_Post extends Activity {

   TextView textview;

   @Override

   protected void onCreate(Bundle savedInstanceState) {

      super.onCreate(savedInstanceState);

    //   setContentView(R.layout.activity_main);

      setContentView(R.layout.main_httpclient);

      textview= (TextView)findViewById(R.id.textView);

      new SendPostRequest().execute();

   }

public class SendPostRequest extends AsyncTask<String, Void, String> {

      protected void onPreExecute(){}

      protected String doInBackground(String... arg0) {

        try {

       URL url = new URL("http://10.0.2.2:81/test/test_url.php");

 HttpURLConnection conn = (HttpURLConnection) url.openConnection();
```

```
            conn.setReadTimeout(15000 /* milliseconds */);
            conn.setConnectTimeout(15000 /* milliseconds */);
            conn.setRequestMethod("POST");
            conn.setRequestProperty("Content-Type",      "application/x-www-form-urlencoded");
 //  Définition des paramètres
//String urlParameters = "name=" + URLEncoder.encode("rabe", "UTF-8") +
//"&email=" + URLEncoder.encode("rabe@yahoo.fr", "UTF-8");
//conn.setRequestProperty("Content-Length", "" +
//Integer.toString(urlParameters.getBytes().length));
StringBuilder urlParams=new StringBuilder("name=");
urlParameters.append(URLEncoder.encode("rabe","UTF-8"));
urlParameters.append("&email=");
urlParameters.append(URLEncoder.encode(" rabe@yahoo.fr ","UTF-8"));
conn.setDoInput(true);
            conn.setDoOutput(true);


             OutputStream os = conn.getOutputStream();
                BufferedWriter writer = new BufferedWriter(
                        new OutputStreamWriter(os, "UTF-8"));
               //  writer.write (urlParameters);
               writer.write (urlParams.toString());
                writer.flush();
                writer.close();
                os.close();
                int responseCode=conn.getResponseCode();
if (responseCode == HttpsURLConnection.HTTP_OK) {
                         StringBuffer sb = new StringBuffer("");
                                    BufferedReader in=new BufferedReader(new
                                 InputStreamReader(conn.getInputStream()));
                  //  StringBuffer sb = new StringBuffer("");
                   String line="";
 while((line = in.readLine()) != null) {
                       sb.append(line);
                       break; }
                  // Log.e("resultat",sb.toString());
                   in.close();
                   return sb.toString();
                }
                else {  return new String("false : "+responseCode);
                }
            }
            catch(Exception e){
                        Log.e("tagconnect"," ERREUR RESULTAT.... ");
                return new String("Exception....: " + e.getMessage());
            }
        }
@Override
        protected void onPostExecute(String result) {
           textview.setText(result);
          Toast.makeText(getApplicationContext(), result,
                   Toast.LENGTH_LONG).show();
        }
    }
```

METHODE GET

```
public class MainTest_UrlConnect_Get extends Activity {
             TextView textview;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main_httpclient);
        textview= (TextView)findViewById(R.id.textView);
        new SendGetRequest().execute();
    }

 public class SendGetRequest extends AsyncTask<String, Void, String> {
        protected void onPreExecute(){}
        protected String doInBackground(String... arg0) {
          try {
 URL url = new URL("http://10.0.2.2:81/test/test_url.php?name='rabe'&email=‘mon mail'");

/* StringBuilder stringBuilder = new StringBuilder("http://10.0.2.2:81/test/test_url.php");
            stringBuilder.append("?name=");
            stringBuilder.append(URLEncoder.encode("RIVO", "UTF-8"));
            stringBuilder.append("&email=");
          stringBuilder.append(URLEncoder.encode("MON EMAIL", "UTF-8"));
            URL url = new URL(stringBuilder.toString());
*/
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setReadTimeout(15000 /* milliseconds */);
            conn.setConnectTimeout(15000 /* milliseconds */);
            conn.setRequestMethod("GET");
int responseCode=conn.getResponseCode();
                if (responseCode == HttpsURLConnection.HTTP_OK) {
                        StringBuffer sb = new StringBuffer("");
```

```
  BufferedReader in=new BufferedReader(new  InputStreamReader(
                        conn.getInputStream()));
              StringBuffer sb = new StringBuffer("");
               String line="";
               while((line = in.readLine()) != null) {
                  sb.append(line);
                  break;
               }
              in.close();
              return sb.toString();
            }
            else {
               return new String("false : "+responseCode);
            }
         }
         catch(Exception e){
                  Log.e("tagconnect"," ERREUR RESULTAT.... ");
            return new String("Exception....: " + e.getMessage());
         }


      }
@Override
      protected void onPostExecute(String result) {
         textview.setText(result);
        Toast.makeText(getApplicationContext(), result,
               Toast.LENGTH_LONG).show();
      }
   }
```

Code Backend PHP

Ce code recupère les paramètres de la requête provenant de l’app Android et retourne la réponse

```
<?php

$email = $_POST['email']; //$email = $_GET['email'];

$name = $_POST['name']; //$name = $_GET['name'];

print_r(json_encode($_POST)); // $_GET;

?>
```

CODE JSP

```
<%

String  name= request.getParameter("name");

String  email = request.getParameter("email");

out.println(" BONJOUR "+ name + "   "+email);

%>
```

EXEMPLE 2

nous allons voir comment faire une simple demande HTTP GET à une Servlet en utilisant HttpURLConnection et afficher la réponse dans TextView.

Nous utilisons **android.os.AsyncTask** pour effectuer cette tâche.

Etape 1 : créer une servlet et la déployer dans un serveur TOMCAT

```
package com.exa.servlets;

 import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;


public class HelloWorldServlet extends HttpServlet {
    public HelloWorldServlet() {

        super();

    }

     protected void doGet(HttpServletRequest request,

               HttpServletResponse response) throws ServletException, IOException {

        PrintWriter out = response.getWriter();

        out.println("Hello Android !!!!");

    }

}
```

Etape2 : Créer un Projet Android “SimpleHttpGetServlet“.

res/values/string.xml

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

<resources>

    <string name="hello">Accessing Servlet from Android</string>

    <string name="app_name">SimpleHttpGetServlet</string>

    <string name="button">Invoke Servlet</string>

</resources>
```

Fichier layout XML : main

```
<?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" >


    <TextView

        android:layout_width="fill_parent"

        android:layout_height="wrap_content"
        android:text="@string/hello" />


    <Button
        android:id="@+id/button"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/button" />


    <TextView
        android:id="@+id/outputTxt"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textColor="#CC0033"
        android:textSize="16sp"
        android:textStyle="bold" />


</LinearLayout>
```

Activity

```
package com.exa.android;
 import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLConnection;
import android.app.Activity;
import android.os.AsyncTask;
import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.TextView;

 public class HttpGetServletActivity3 extends Activity implements OnClickListener {

    Button button;

    TextView outputText;

    public static final String

URL =     http://10.0.2.2:8080/HttpGetServlet/HelloWorldServlet;

    /** Called when the activity is first created. */

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.main);

        findViewsById(); button.setOnClickListener(this);

    }

     private void findViewsById() {

        button = (Button) findViewById(R.id.button);

        outputText = (TextView) findViewById(R.id.outputTxt);

    }


    public void onClick(View view) {

        GetXMLTask task = new GetXMLTask();

        task.execute(new String[] { URL });

    }

 private class GetTask extends AsyncTask<String, Void, String> {

        @Override
        protected String doInBackground(String... urls) {
            String output = null;
            for (String url : urls) {
                output = getOutputFromUrl(url);
            }
            return output;
        }

        private String getOutputFromUrl(String url) {
            StringBuffer output = new StringBuffer("");
            try {
                InputStream stream = getHttpConnection(url);
                BufferedReader buffer = new BufferedReader(
                        new InputStreamReader(stream));
                String s = "";
                while ((s = buffer.readLine()) != null)
                    output.append(s);
            } catch (IOException e1) {
                e1.printStackTrace();
            }
            return output.toString();
        }

// Makes HttpURLConnection and returns InputStream
    private InputStream getHttpConnection(String urlString) throws IOException {
            InputStream stream = null;
            URL url = new URL(urlString);
            URLConnection connection = url.openConnection();
```

```
      try {
        HttpURLConnection httpConnection = (HttpURLConnection) connection;
        httpConnection.setRequestMethod("GET");
        httpConnection.connect();
      if (httpConnection.getResponseCode() == HttpURLConnection.HTTP_OK) {
          stream = httpConnection.getInputStream();
        }
      } catch (Exception ex) {
        ex.printStackTrace();
      }
      return stream;
    }
     @Override
    protected void onPostExecute(String output) {
      outputText.setText(output);
    }
  }
}
```

Manifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.theopentutorials.android"
  android:versionCode="1"
  android:versionName="1.0" >
   <uses-sdk android:minSdkVersion="15" />
   <uses-permission android:name="android.permission.INTERNET" />
 <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

```
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".HttpGetServletActivity"
            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>
```

Gérer l'état du réseau

Pour vérifier la connexion réseau, utilisez les classes suivantes:

- **ConnectivityManager** répondant aux requêtes concernant l'état de la connectivité réseau. Il notifie également les applications lorsque la connectivité réseau change.
- **NetworkInfo** décrivant l'état d'une interface réseau d'un type donné (actuellement mobile ou Wi-Fi).

L'extrait de code suivant teste si le Wi-Fi et le mobile sont connectés.

Dans le code:

- La méthode g**etSystemService** obtient une instance de ConnectivityManager .
- La méthode **getNetworkInfo** obtient le statut de la connexion Wi-Fi du périphérique, puis sa connexion mobile. La méthode getNetworkInfo renvoie un objet NetworkInfo , qui contient des informations sur l'état de connexion du réseau donné (qu'il soit inactif, en cours de connexion, etc.).
- La méthode **networkInfo.isConnected()** renvoie **true** si le réseau donné est connecté. Si le réseau est connecté, il peut être utilisé pour établir des sockets et transmettre des données.

```
private static final String DEBUG_TAG = "NetworkStatusExample";

ConnectivityManager connMgr = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);

NetworkInfo networkInfo = connMgr.getNetworkInfo(ConnectivityManager.TYPE_WIFI);
```

```
boolean isWifiConn = networkInfo.isConnected();

networkInfo = connMgr.getNetworkInfo(ConnectivityManager.TYPE_MOBILE);

boolean isMobileConn = networkInfo.isConnected();

Log.d(DEBUG_TAG, "Wifi connected: " + isWifiConn);

Log.d(DEBUG_TAG, "Mobile connected: " + isMobileConn);
```

EXEMPLE 2 : récuperer le role d'un utilisateur reconnu par son login et mot de passe dans une base de données Mysql

```
mport android.app.Activity;

import android.os.Bundle;

import android.view.View;

import android.widget.EditText;

import android.widget.TextView;

public class LoginActivity extends Activity {

  private EditText usernameField,passwordField;

  private TextView status,role,method;

  @Override

  protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_login);

    usernameField = (EditText)findViewById(R.id.editText1);

    passwordField = (EditText)findViewById(R.id.editText2);

    status = (TextView)findViewById(R.id.textView6);

    role = (TextView)findViewById(R.id.textView7);

    method = (TextView)findViewById(R.id.textView9);

  }


public boolean isNetworkAvailable() {

      ConnectivityManager cm = (ConnectivityManager)

        getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = cm.getActiveNetworkInfo();
        // si aucun réseau n'est disponible, networkInfo sera null
        // sinon, vérifier si nous sommes connectés
        if (networkInfo != null && networkInfo.isConnected()) {
            return true;    }
        return false;
    }
 public void login(View view){
      String username = usernameField.getText().toString();
      String password = passwordField.getText().toString();
      method.setText("Get Method");
if(isNetworkAvailable()){
   new SigninActivity(this,status,role,0).execute(username,password);}
   }
   public void loginPost(View view){
      String username = usernameField.getText().toString();
      String password = passwordField.getText().toString();
      method.setText("Post Method");
   if(isNetworkAvailable()){
 new SigninActivity(this,status,role,1).execute(username,password); }  }
}
```

La classe SigninActivity

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URI;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLEncoder;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;
import android.content.Context;
import android.os.AsyncTask;
import android.util.Log;
import android.widget.TextView;
public class SigninActivity  extends AsyncTask<String, String, String>{
  private TextView statusField,roleField;
  private Context context;
  private int byGetOrPost = 0;
  //flag 0 means get and 1 means post.(By default it is get.)
  public SigninActivity(Context context,TextView statusField,TextView roleField,int flag) {
    this.context = context;
    this.statusField = statusField;
    this.roleField = roleField;
    byGetOrPost = flag;
  }
  protected void onPreExecute(){
  }
protected String doInBackground(String... arg0) {
 if(byGetOrPost == 0){ //means by Get Method
      try{
        String username = (String)arg0[0];
        String password = (String)arg0[1];
String link = "http://10.0.2.2:81/test/login_get.php?username="+username+"&password="+password;
         URL url = new URL(link);
    /*    StringBuilder stringBuilder = new StringBuilder("http://10.0.2.2:81/test/test_url.php");
         stringBuilder.append("?username=");
         stringBuilder.append(URLEncoder.encode(username, "UTF-8"));
         stringBuilder.append("&password =");
         stringBuilder.append(URLEncoder.encode(password , "UTF-8"));
         URL url = new URL(stringBuilder.toString());*/
}
HttpURLConnection conn = (HttpURLConnection) url.openConnection();
         conn.setReadTimeout(15000 /* milliseconds */);
         conn.setConnectTimeout(15000 /* milliseconds */);
         conn.setRequestMethod("GET");
        BufferedReader in=new BufferedReader(new
             InputStreamReader(conn.getInputStream()));


        StringBuffer sb = new StringBuffer("");
        String line="";
        while ((line = in.readLine()) != null) {
          sb.append(line);
          break;
        }
        in.close();
        return sb.toString();
      } catch(Exception e){
        Log.e("TAGERROR"," GET ERROR");
        return new String("Exception: " + e.getMessage());
      }
    }
else
//  Post Method
try{
        String username = (String)arg0[0];
        String password = (String)arg0[1];
        String link="http://10.0.2.2:81/test/login_post.php";
          URL url = new URL(link);
        URLConnection conn = url.openConnection();
  conn.setRequestProperty("Content-Type",
       "application/x-www-form-urlencoded");
String urlParameters = "username=" + URLEncoder.encode(username, "UTF-8") +
"&password=" + URLEncoder.encode(password, "UTF-8");
         conn.setDoInput(true);
        conn.setDoOutput(true);
        OutputStreamWriter wr = new OutputStreamWriter(conn.getOutputStream());
        wr.write( urlParameters );
        wr.flush();
BufferedReader reader = new BufferedReader(new
          InputStreamReader(conn.getInputStream()));
        StringBuilder sb = new StringBuilder();
        String line = null;
       // Read Server Response
        while((line = reader.readLine()) != null) {
          sb.append(line);
          break;
        }
```

```
            return sb.toString();

        } catch(Exception e){

        Log.e("TAGERROR"," POST ERROR");

           return new String("Exception: " + e.getMessage());

        }

     }

  }
@Override

  protected void onPostExecute(String result){

     this.statusField.setText("Login Successful");

     this.roleField.setText(result);  }

}
```

Layout : activity_loginxml

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

<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" >

  <EditText

     android:id="@+id/editText2"

     android:layout_width="wrap_content"

     android:layout_height="wrap_content"

     android:layout_alignRight="@+id/editText1"

     android:layout_below="@+id/editText1"

     android:layout_marginTop="25dp"

     android:ems="10"

     android:inputType="textPassword" >
  </EditText>
  <EditText
     android:id="@+id/editText1"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:layout_alignParentRight="true"
     android:layout_alignParentTop="true"
     android:layout_marginTop="44dp"
     android:ems="10" >
  <requestFocus android:layout_width="wrap_content" />
  </EditText>

<TextView
     android:id="@+id/textView1"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:layout_alignBottom="@+id/editText1"
     android:layout_alignParentLeft="true"
     android:text="@string/Username" />
  <TextView
     android:id="@+id/textView3"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:layout_alignParentTop="true"
     android:layout_centerHorizontal="true"
     android:text="@string/App"
     android:textAppearance="?android:attr/textAppearanceLarge" />
  <TextView
```

```
    android:id="@+id/textView7"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_alignBottom="@+id/textView5"

    android:layout_alignLeft="@+id/textView6"

    android:text="@string/Role"

    android:textAppearance="?android:attr/textAppearanceMedium"

    android:textSize="10sp" />

<TextView

    android:id="@+id/textView5"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_below="@+id/textView6"

    android:layout_marginTop="27dp"

    android:layout_toLeftOf="@+id/editText1"

    android:text="@string/LoginRole" />

  <TextView

    android:id="@+id/textView8"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_above="@+id/textView6"

    android:layout_alignLeft="@+id/textView5"

    android:layout_marginBottom="27dp"

    android:text="@string/method" />

  <TextView

    android:id="@+id/textView4"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"
```

```
        android:layout_alignLeft="@+id/textView8"
        android:layout_below="@+id/button1"
        android:layout_marginTop="86dp"
        android:text="@string/LoginStatus" />
    <TextView
        android:id="@+id/textView6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/textView4"
        android:layout_centerHorizontal="true"
        android:text="@string/Status"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textSize="10sp" />
<TextView
        android:id="@+id/textView9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/textView8"
        android:layout_alignLeft="@+id/textView6"
        android:text="@string/Choose"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textSize="10sp" />
    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:layout_toRightOf="@+id/textView6"
        android:onClick="loginPost"
        android:text="@string/LoginPost" />
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/button2"
        android:layout_alignBottom="@+id/button2"
        android:layout_alignLeft="@+id/textView2"
        android:onClick="login"
        android:text="@string/LoginGet" />
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/editText2"
        android:layout_alignBottom="@+id/editText2"
        android:layout_alignParentLeft="true"
        android:text="@string/Password" />
</RelativeLayout>
```

Login_get.php

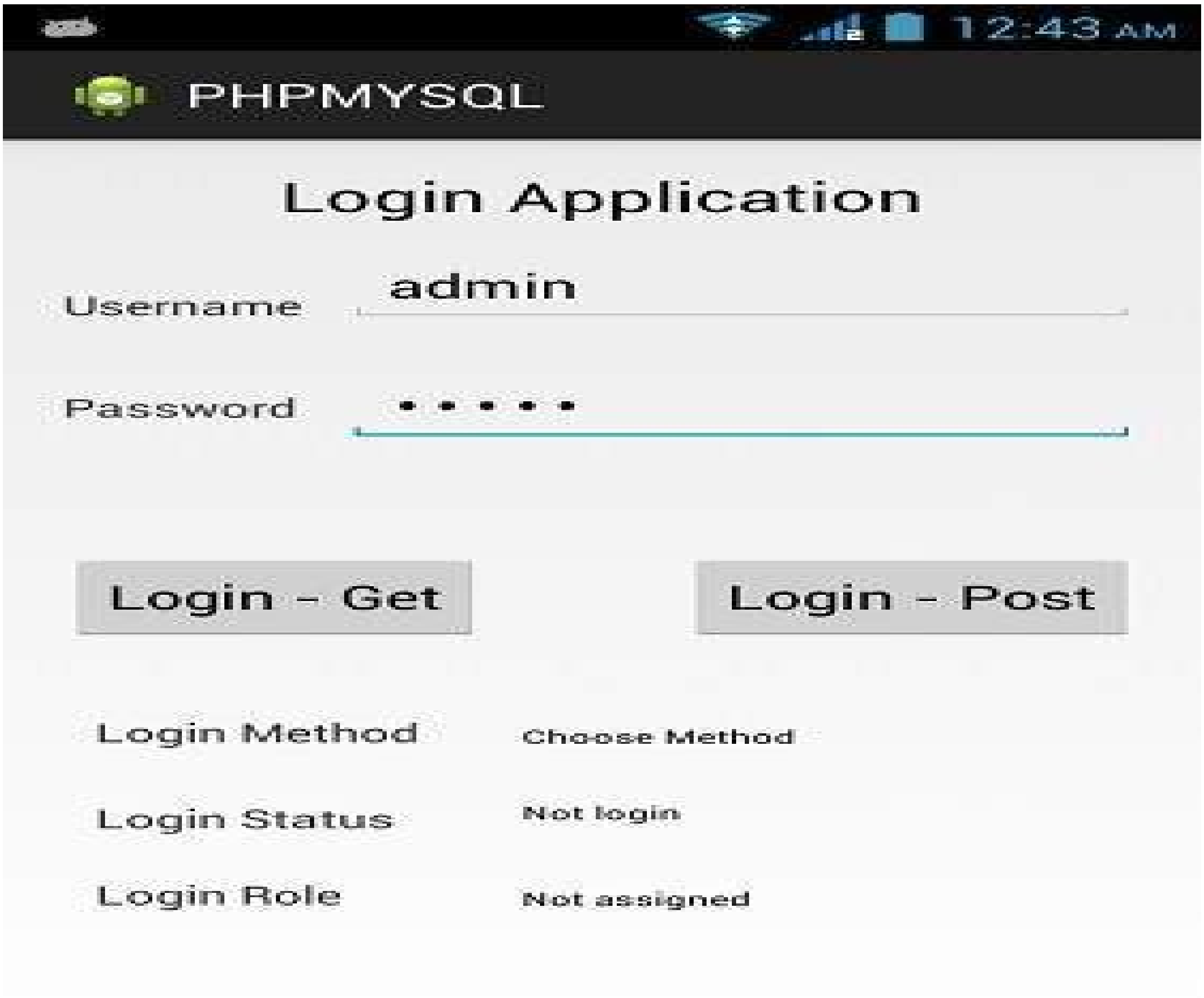
```
<?php
 $con=mysqli_connect("localhost","root","","base");
  if (mysqli_connect_errno($con)) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
  }
```

```
   $username = $_GET['username'];

   $password = $_GET['password'];

 //  $result = mysqli_query($con,"SELECT role FROM login where username='$username' and
password='$password'");

 //  $result = mysqli_query($con,"SELECT * FROM login WHERE  username='".$username."'"." AND
password ='".$password."'");

$row = mysqli_fetch_array($result);

   $data = $row[0];

   if($data){

      echo $data;

   }

  mysqli_close($con);

?>
```

Login_post.php

```
<?php

   $con=mysqli_connect("localhost","root","","base");

   if (mysqli_connect_errno($con))  {

     echo "Failed to connect to MySQL: " . mysqli_connect_error();

   }


   $username = $_POST['username'];

   $password = $_POST['password'];

   $result = mysqli_query($con,"SELECT role FROM login where username='$username' and
password='$password'");

  $row = mysqli_fetch_array($result);

   $data = $row[0];

   if($data){

      echo $data;

   }
```

```
mysqli_close($con);

?>
```



Login Application

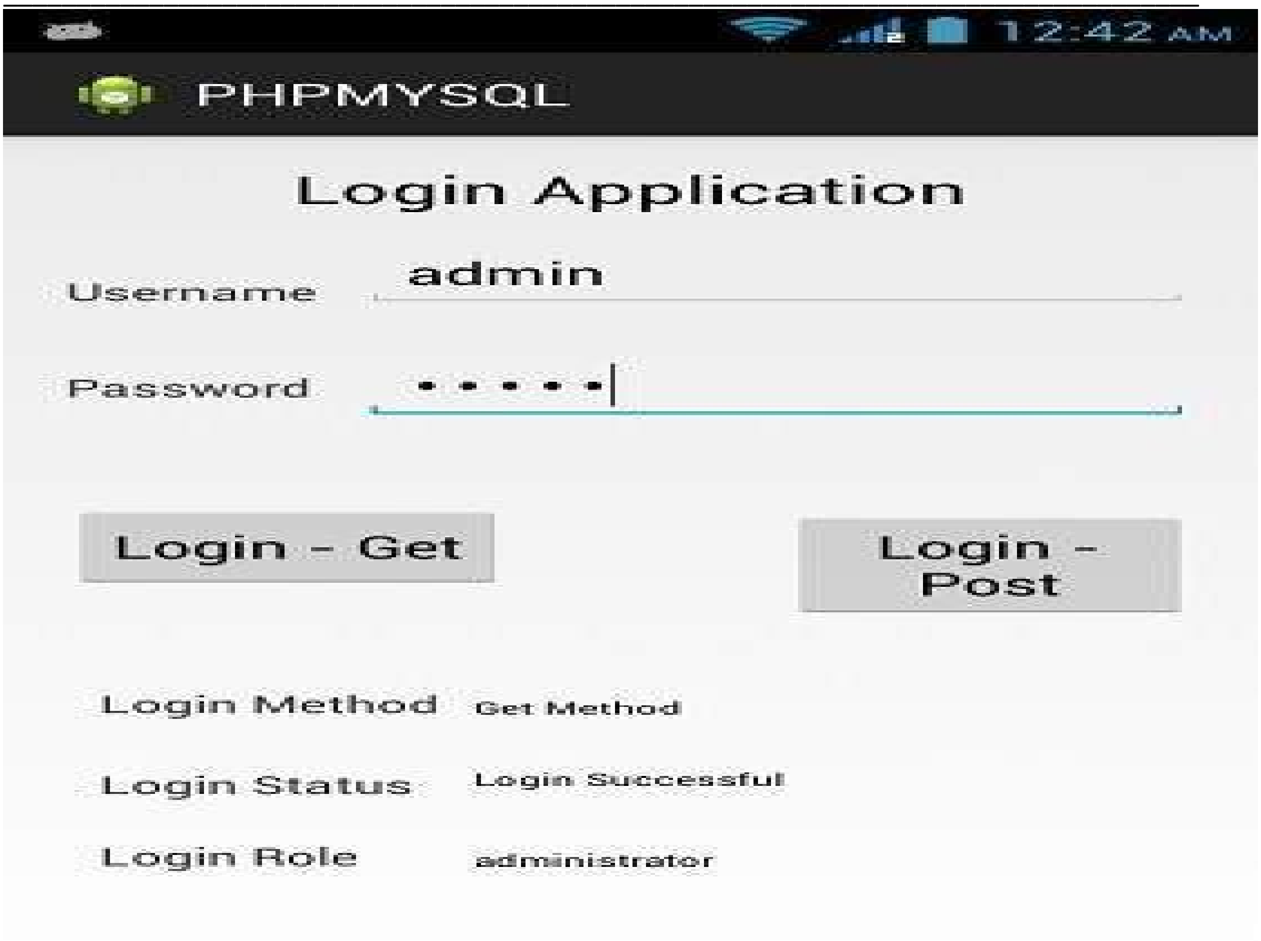
Username admin

Password •••••

Login Method Choose Method

Login Status Not login

Login Role Not assigned



Username admin

Password •••••

Login Method Get Method

Login Status Login Successful

Login Role administrator

NB : on peut accèder à un service web RestFull JAX-RS + JERSEY depuis un client Android