

Tp Connexion Php Mysql

Introduction

Tout d'abord , pourquoi utiliser un serveur distant?

L'intérêt d'un serveur distant, est de permettre la centralisation des données dans le cas d'une application qui se veut multi plateforme.

Par exemple:

Imaginons une application qui permet de lister des films à voir. Grâce à une bdd externe il est possible d'accéder à ses données depuis un autre appareil(tablette,pc,...)

Fonctionnement

Le client (application) fait une requête http vers une url sur le serveur, comme le fait par exemple un navigateur web avec une url.

Ensuite le fichier pointé par l'url se charge d'interroger la base de données et de renvoyer les informations en json à l'application.

Pour finir l'application se charge de traiter les données reçus et de les afficher.

Objectif du tp

Le but de ce tp est d'interagir avec une base de données externe, afin de pouvoir récupérer les informations stockées dans celle ci, mais aussi de pouvoir y ajouter des informations.

De plus dans ce tp nous utiliserons une bibliothèque appelée Okhttp qui permet de faciliter les requêtes web.

Lien du tp initial :

<https://drive.google.com/open?id=0B5mYWG5Z9R-JQmp2QUNUejV1MTQ>

Pré requis pour le tp :

Tout d'abord puisque nous souhaitons interroger une Bdd, il nous faut une Bdd. Pour ce tp nous prendrons une simple table d'utilisateur contenant 4 champs(id,pseudo,age,ville)

```
CREATE TABLE IF NOT EXISTS `utilisateur` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `pseudo` varchar(25) NOT NULL,  
  `age` int(11) NOT NULL,  
  `ville` varchar(25) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=18 ;  
  
INSERT INTO `utilisateur` (`id`, `pseudo`, `age`, `ville`) VALUES  
(1, 'riri', 20, 'lille'),  
(2, 'fifi', 21, 'paris'),  
(3, 'loulou', 25, 'lyon');
```

Ensuite il nous faut un fichier php sur ce serveur qui se charge d'interroger la bdd et de renvoyer les infos en json à notre application.

```
<?php  
try {  
    // connexion  
    $pdo_options[PDO::ATTR_ERRMODE] = PDO::ERRMODE_EXCEPTION;  
    $bdd = new PDO('mysql:host=localhost;dbname=androidtp', 'root', '', $pdo_options);  
    // requete  
    $sql = 'SELECT * FROM utilisateur;';  
    $response = $bdd->query($sql);  
    $output = $response->fetchAll(PDO::FETCH_ASSOC);  
} catch (Exception $e) {  
    die('Erreur : ' . $e->getMessage());  
}  
    // envoi du resultat en json  
    echo(json_encode($output));  
?>
```

Déroulement du tp

Première partie : récupération de données

Tout d'abord commençons par récupérer et ouvrir le projet initial.

Ensuite comme nous allons utiliser une bibliothèque facilitant les requêtes web il faut renseigner sa dépendance dans le fichier build.gradle

```
compile 'com.squareup.okhttp3:okhttp:3.3.1'
```

Ensuite sur notre première activité nous allons ajouter un bouton qui se chargera d'appeler une autre activités(consulterActivity que nous créerons plus loin) .

Activity_main.xml

```
<Button
    android:text="Consulter"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="69dp"
    android:id="@+id/button_consulter"
    android:layout_below="@+id/textView"

    android:layout_centerHorizontal="true"
/>
```

mainActivity.java

```
final Button consulterButton = (Button)
findViewById(R.id.button_consulter);
consulterButton.setOnClickListener(new
View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent intent = new
Intent(MainActivity.this,
ConsulterActivity.class);
        startActivity(intent);
    }
});
```

Afin de gérer l'affichage créons une classe Utilisateur .

Utilisateur.java

```
public class Utilisateur {
    private final int id;
    private final String pseudo;
    private final int age;
    private final String ville;

    public Utilisateur(JSONObject jsonObject) {
        this.id = jsonObject.optInt("id");
        this.pseudo = jsonObject.optString("pseudo");
        this.age = jsonObject.optInt("age");
        this.ville = jsonObject.optString("ville");
    }

    public int id() { return id; }

    public String pseudo() { return pseudo; }

    public int age() { return age; }

    public String ville() { return ville; }
}
```

Créons maintenant l'activité charger d'appeler notre fichier php
File->new->activity->EmptyActivity
Que nous appellerons ConsulterActivity

Une fois cette activité créée, il faut créer une fonction permettant de récupérer les infos via une requête http.

pour cela, créons un nouveau client

```
private final OkHttpClient client = new OkHttpClient();
```

ainsi qu'une fonction qui l'utilise:

Cette fonction prend en paramètre l'url du fichier

ConsulterActivity.java

```
public String get(String url) throws IOException {  
    // Preparation  
    Request request = new Request.Builder().url(url).build();  
    // Execution  
    Response response = client.newCall(request).execute();  
    // Recuperation  
    return response.body().string();  
}
```

Cependant comme cette opération nécessite des ressources , il faut l'exécuter dans un thread séparé(AsyncTask).

ConsulterActivity.java

```
private class Affichage extends AsyncTask<Void, Void, String>  
{  
    private volatile ConsulterActivity screen;  
    public Affichage(ConsulterActivity s)  
    {  
        this.screen = s;  
    }  
    @Override  
    protected String doInBackground(Void... params)  
    {  
        String rep = "";  
        try {  
            rep = get("http://tpandroid.esy.es/recuperation.php");  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
        return rep;  
    }  
    @Override  
    protected void onPostExecute(String result)  
    {  
        this.screen.populate(parse(result));  
    }  
    public String get(String url) throws IOException {  
        // Preparation  
        Request request = new Request.Builder().url(url).build();
```

```

        // Execution
        Response response = client.newCall(request).execute();
        // Recuperation
        return response.body().string();
    }
    private List<Utilisateur> parse(final String json) {
        try {
            final List<Utilisateur> products = new ArrayList<>();
            final JSONArray jProductArray = new JSONArray(json);
            for (int i = 0; i < jProductArray.length(); i++) {
                products.add(new Utilisateur(jProductArray.optJSONObject(i)));
            }
            return products;
        } catch (JSONException e) {
        }
        return null;
    }
}

```

Enfin vous pouvez constater qu'à la fin (onPostExecute) une fonction *populate* permet d'afficher le résultat sous forme d'un tableau, il faut donc ajouter cette méthode afin d'avoir un résultat visuel.

ConsulterActivity.java

```

public void populate(List<Utilisateur> data) {

    TableLayout table = (TableLayout) findViewById(R.id.idTable); // on prend le
    tableau défini dans le layout

    for (Utilisateur u : data) {
        row = new TableRow(this); // création d'une nouvelle ligne

        c1 = new TextView(this); // création cellule
        c1.setText(Integer.toString(u.id())); // ajout du texte
        c1.setGravity(Gravity.CENTER); // centrage dans la cellule
        // adaptation de la largeur de colonne à l'écran :
        c1.setLayoutParams( new TableRow.LayoutParams( 0,
        android.view.ViewGroup.LayoutParams.WRAP_CONTENT, 1 ) );

        c2 = new TextView(this); // création cellule
        c2.setText(u.pseudo()); // ajout du texte
        c2.setGravity(Gravity.CENTER); // centrage dans la cellule
        // adaptation de la largeur de colonne à l'écran :
        c2.setLayoutParams( new TableRow.LayoutParams( 0,
        android.view.ViewGroup.LayoutParams.WRAP_CONTENT, 1 ) );

        c3 = new TextView(this); // création cellule
        c3.setText(Integer.toString(u.age())); // ajout du texte
        c3.setGravity(Gravity.CENTER); // centrage dans la cellule
        // adaptation de la largeur de colonne à l'écran :
        c3.setLayoutParams( new TableRow.LayoutParams( 0,
        android.view.ViewGroup.LayoutParams.WRAP_CONTENT, 1 ) );

        c4 = new TextView(this); // création cellule
        c4.setText(u.ville()); // ajout du texte
        c4.setGravity(Gravity.CENTER); // centrage dans la cellule
        // adaptation de la largeur de colonne à l'écran :
        c4.setLayoutParams( new TableRow.LayoutParams( 0,

```

```

android.view.ViewGroup.LayoutParams.WRAP_CONTENT, 1 ) );

    // ajout des cellules à la ligne
    row.addView(c1);
    row.addView(c2);
    row.addView(c3);
    row.addView(c4);

    // ajout de la ligne au tableau
    table.addView(row);
}
}

```

Enfin on appelle ce thread dans onCreate:

```

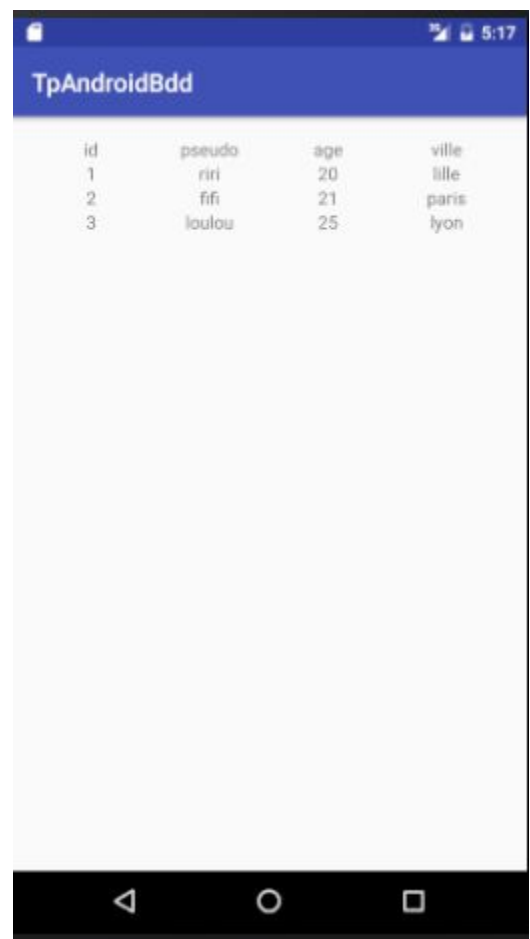
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_consulter);

    new Affichage(this).execute() ;
}

```

Une fois tout ceci fait il est maintenant temps de tester notre application.

Pour cela lancer votre émulateur, vous devriez avoir un résultat similaire a celui ci lorsque vous cliquez sur consulter.



Deuxième partie : envoi de données

Tout d'abord pour pouvoir envoyer des données, il nous faut un formulaire:

AjouterActivity.xml

```
<TextView
    android:text="Pseudo : "
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_alignParentStart="true"
    android:layout_marginStart="29dp"
    android:layout_marginTop="71dp"
    android:id="@+id/textView2"
    android:textSize="18sp" />

<TextView
    android:text="Age : "
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView2"
    android:layout_alignStart="@+id/textView2"
    android:layout_marginTop="54dp"
    android:id="@+id/textView3"
    android:textSize="18sp" />

<TextView
    android:text="Ville : "
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textView4"
    android:textSize="18sp"
    android:layout_centerVertical="true"
    android:layout_alignStart="@+id/textView3" />

<Button
    android:text="Ajouter"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="83dp"
    android:id="@+id/button_ajouter" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="textPersonName"
    android:text="pseudo"
    android:ems="10"
    android:layout_alignBottom="@+id/textView2"
    android:layout_alignStart="@+id/button_ajouter"
    android:id="@+id/editText_pseudo" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="textPersonName|number"
    android:text="24"
    android:ems="10"
    android:layout_alignBottom="@+id/textView3" />
```



```
android:layout_alignStart="@+id/editText_pseudo"
android:id="@+id/editText_age" />
```

```
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="textPersonName"
    android:text="ville"
    android:ems="10"
    android:layout_alignBottom="@+id/textView4"
    android:layout_alignEnd="@+id/editText_age"
    android:id="@+id/editText_ville" />
```

Comme pour la récupération, il nous faut une AsyncTask pour effectuer l'ajout

AjouterActivity.java

```
private class Ajouter extends AsyncTask<Void, Void, String>
{
    private volatile View.OnClickListener screen;
    private volatile EditText ETpseudo,ETage,ETville;
    public Ajouter(View.OnClickListener s,EditText ETpseudo,EditText ETage,EditText ETville)
    {
        this.screen = s;
        this.ETpseudo = ETpseudo;
        this.ETage = ETage;
        this.ETville = ETville;
    }
    @Override
    protected String doInBackground(Void... params)
    {
        String rep = "";
        try {
            rep = get("http://tpandroid.esy.es/ajouter.php");
        } catch (IOException e) {
            e.printStackTrace();
        }
        return rep;
    }

    @Override
    protected void onPostExecute(String result)
    {
    }

    public String get(String url) throws IOException {
        RequestBody formBody = new FormBody.Builder()
            .add("pseudo", ETpseudo.getText().toString())
            .add("age", ETage.getText().toString())
            .add("ville", ETville.getText().toString())
            .build();
        // Prepare la requete
        Request request = new Request.Builder().url(url).post(formBody).build();
        // Execute la requete.
        Response response = client.newCall(request).execute();
        // Recuperation des resultats
        return response.body().string();
    }
}
```


Enfin pour finir la fonction *onCreate*:

AjouterActivity.java

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_ajouter);  
    final Button ajouterButton = (Button) findViewById(R.id.button_ajouter);  
    ajouterButton.setOnClickListener(new View.OnClickListener() {  
  
        public EditText ETpseudo,ETage,ETville;  
  
        @Override  
        public void onClick(View v) {  
            this.ETpseudo = (EditText) findViewById(R.id.editText_pseudo);  
            this.ETage = (EditText) findViewById(R.id.editText_age);  
            this.ETville = (EditText) findViewById(R.id.editText_ville);  
            new Ajouter(this,ETpseudo,ETage,ETville).execute() ;  
            AjouterActivity.this.finish();  
        }  
    });  
}
```

Voilà vous pouvez maintenant essayer d'ajouter une entrée et de voir qu'elle a bien été ajoutée en utilisant le bouton consulter.

Lien du tp final :

<https://drive.google.com/open?id=0B5mYWG5Z9R-JZmNROEVVcVfVNDg>