

ANDROID : Requête HTTP POST

Voir tutoriel : http://wiki.frandroid.com/wiki/Requ%C3%AAte_HTTP_POST

Construction de la requête

Pour construire la requête POST, nous allons créer un objet **org.apache.http.client.methods.HttpPost** . Pour de plus amples informations à propos de cet objet, [consultez la documentation](#).

Dans cet exemple, nous allons utiliser le constructeur **HttpPost(String uri)** qui prend en paramètre une chaîne de caractères décrivant l'URL de votre cible.

```
HttpPost httpPost = new HttpPost("http://www.monsite.fr/monscript.php");
```

Maintenant, nous allons joindre à la requête des paramètres. Pour cela nous allons utiliser un objet une Liste de **org.apache.http.NameValuePair**. NameValuePair est une simple classe permettant d'encapsuler un couple de nom/valeur. C'est une classe abstraite, vous pouvez donc écrire une classe qui en dérive selon vos besoins, ou alors utiliser une de ses classes filles :

org.apache.http.message.BasicNameValuePair.

Tout d'abord nous allons instancier une liste de NameValuePair puis nous allons y rajouter des éléments.

Nous allons créer un BasicNameValuePair (ou une instance d'un autre dérivé de NameValuePair) pour chaque paramètre à envoyer par la requête puis la rajouter à notre liste grâce au constructeur **BasicNameValuePair(String name, String value)**

Par exemple si je veux envoyer deux paramètres **username** et **pwd** de valeurs respectivement **toto** et **toto** :

```
List<NameValuePair> postParameters = new ArrayList<NameValuePair>();  
//On crée la liste qui contiendra tous nos paramètres  
//Et on y rajoute nos paramètres  
postParameters.add(new BasicNameValuePair("username", "toto"));  
postParameters.add(new BasicNameValuePair("pwd", "toto"));
```

Enfin, on lie la liste de paramètres à l'instance de HttpPost créée précédemment.

```
httpPost.setEntity(new UrlEncodedFormEntity(postParameters));
```

Envoyer la requête

On envoie la requête grâce à un objet de type **org.apache.http.client.HttpClient**. Nous prendrons ici un dérivé de HttpClient : DefaultHttpClient

```
HttpClient httpClient = new DefaultHttpClient();  
httpClient.execute(httpPost); //Voilà, la requête est envoyée
```

Intercepter les exceptions

Des erreurs peuvent intervenir pendant la construction ou l'envoi de la requête. N'oubliez surtout pas d'intercepter les exceptions.

```
try {
    HttpPost httpPost = new
HttpPost("http://10.0.2.2/~stephane/test/login.php");
    httpPost.setEntity(new UrlEncodedFormEntity(postParameters));

    HttpClient httpClient = new DefaultHttpClient();
    httpClient.execute(httpPost);

} catch (Exception e) {}
```

Récupérer la réponse de la requête

Selon vos besoins, vous pourriez avoir besoin de récupérer la réponse à la requête envoyée. Il suffit de passer par une instance de **org.apache.http.HttpResponse** pour récupérer un **java.io.BufferedReader**

Il faut ensuite modifier un peu le code plus haut, en récupérant une instance de HttpResponse à partir de la méthode HttpClient.execute(httpPost)

```
HttpResponse response=httpclient.execute(httpPost);
BufferedReader reader= new BufferedReader(new
InputStreamReader(response.getEntity().getContent()));
```

Il suffit ensuite de lire depuis notre BufferedReader: voir [la documentation](#).
Si vous ne devez lire qu'une ligne, la ligne de code suivante devrait suffire:

```
String s = reader.readLine();
```

ANDROID 4

Toutes les tâches consommatrices de ressources (requêtes http, calculs lourds, ...) doivent désormais se faire dans un Thread (processus) séparé afin de ne pas bloquer l'application. Par conséquent, vous devez, comme pour la lecture d'un flux distant (voir activité : ListView) utiliser une classe qui hérite de la classe **AsyncTask** pour exécuter une requête Http.

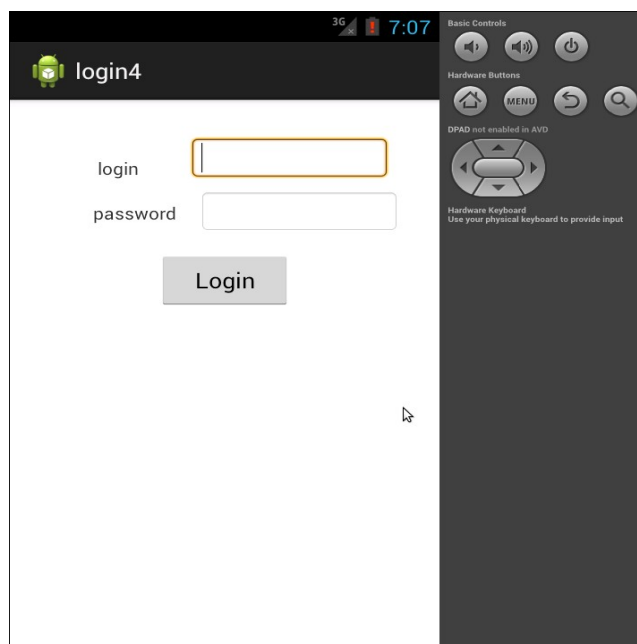
Travail à faire :

1 - dans votre répertoire public_html/jukebox, créer le script PHP suivant : androidAuthentification.php

```
<?php
    $login = $_POST['username'];
    $pwd = $_POST['pwd'];
    // vous pouvez mettre ici le code mysql
    if($login == "toto" AND $pwd == "toto")
        echo 1; // for correct login pwd
    else
        echo 0; // for incorrect login pwd
?>
```

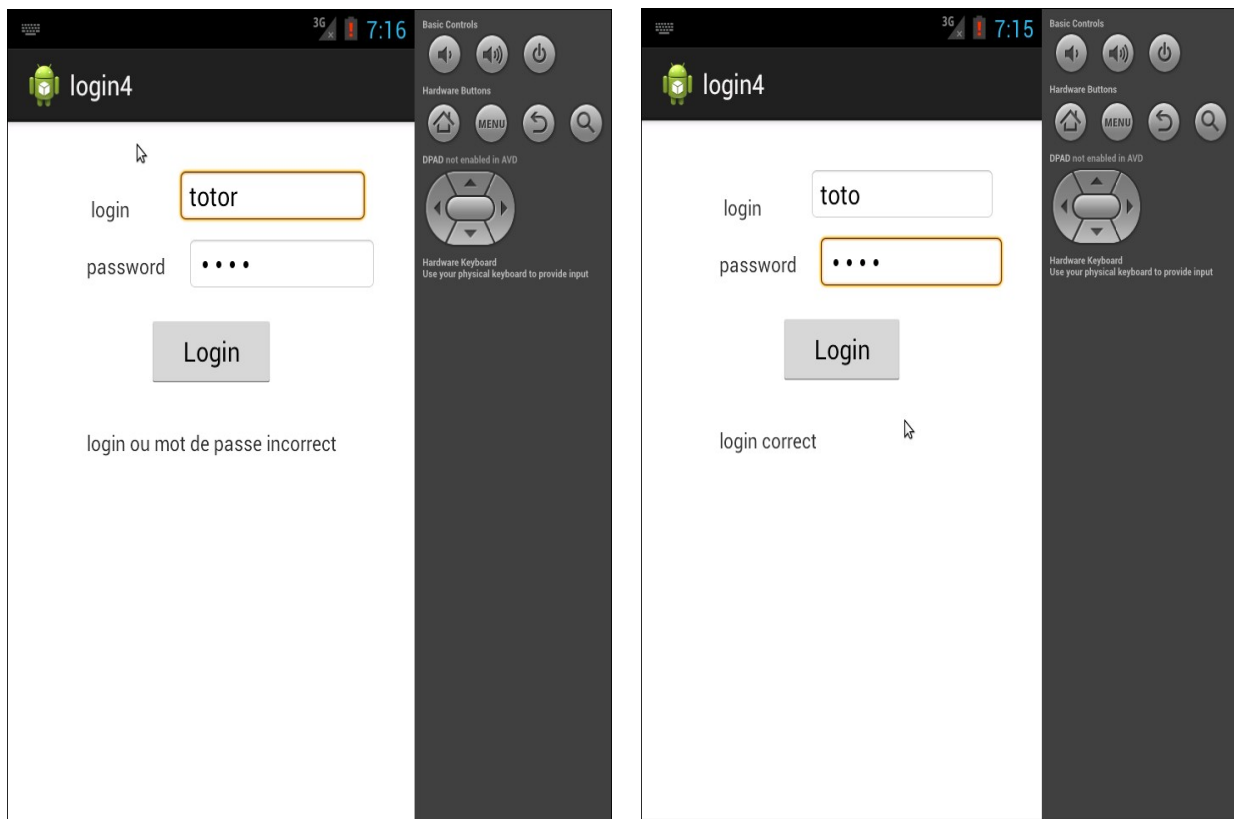
2 – Créer un nouveau projet Android : Login

3 – Créer l'interface suivante :



4 – Faire en sorte que votre application envoie une requête POST, contenant le login et le password saisis, au serveur hébergeant votre script : androidAuthentification.php

En fonction de la réponse du serveur vous devez afficher dans votre application :



5 – Créer une table users avec comme champs : id, login, password et remplacer dans votre script PHP, une authentification Mysql (voir cours : <http://guyonst.free.fr/php-cours.php> : session)