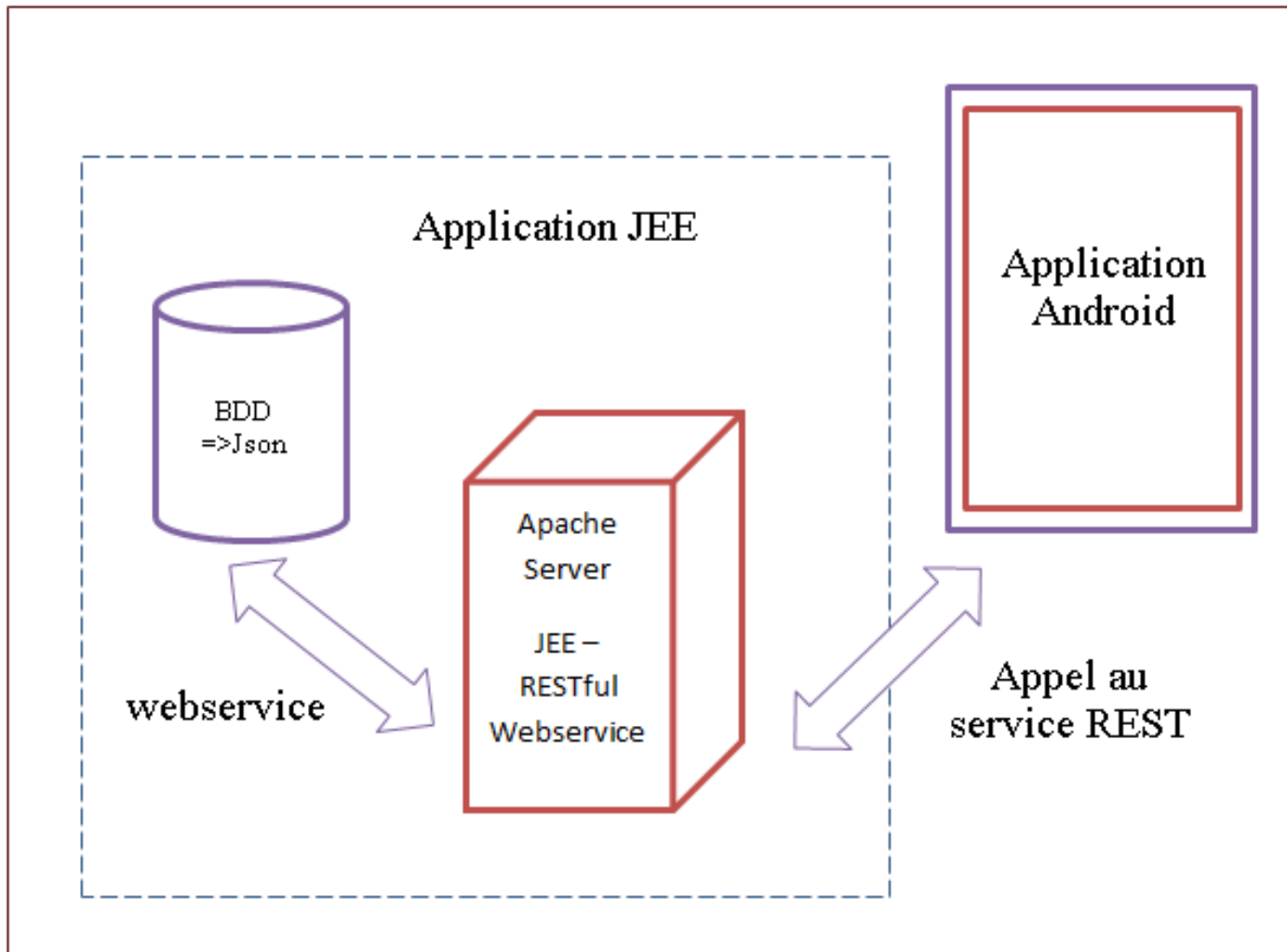


BASE DE DONNÉES EXTERNE SUR ANDROID

Pahoua YANG
18/11/2014

FONCTIONNEMENT



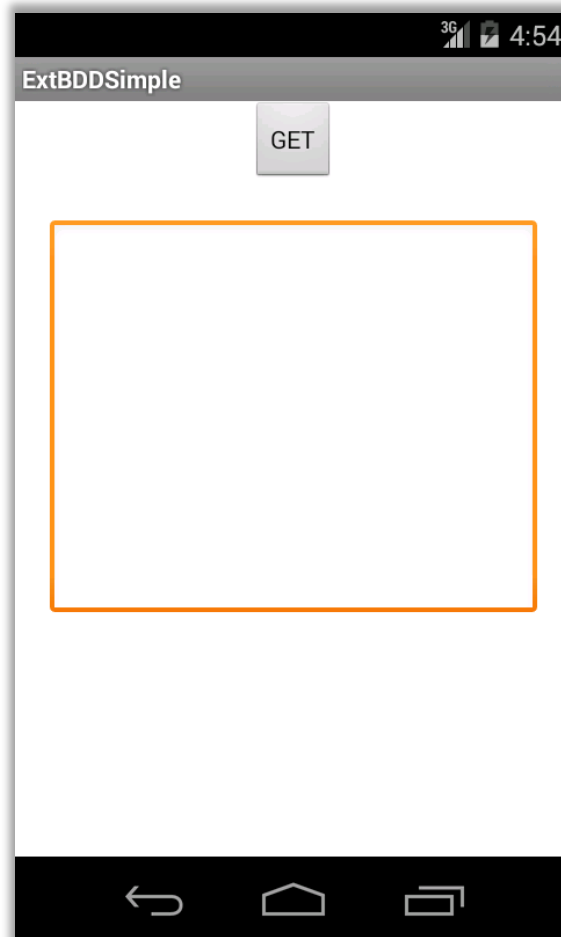
APPELER UN SERVICE REST VIA L'APPLICATION

- La permission internet : Manifest

```
<!-- pour avoir la permission d'aller chercher des infos sur internet -->  
<uses-permission android:name="android.permission.INTERNET"/>
```

APPELER UN SERVICE REST VIA L'APPLICATION

- ◉ Template simple



APPELER UN SERVICE REST VIA L'APPLICATION

◉ GetUsers extends AsyncTask

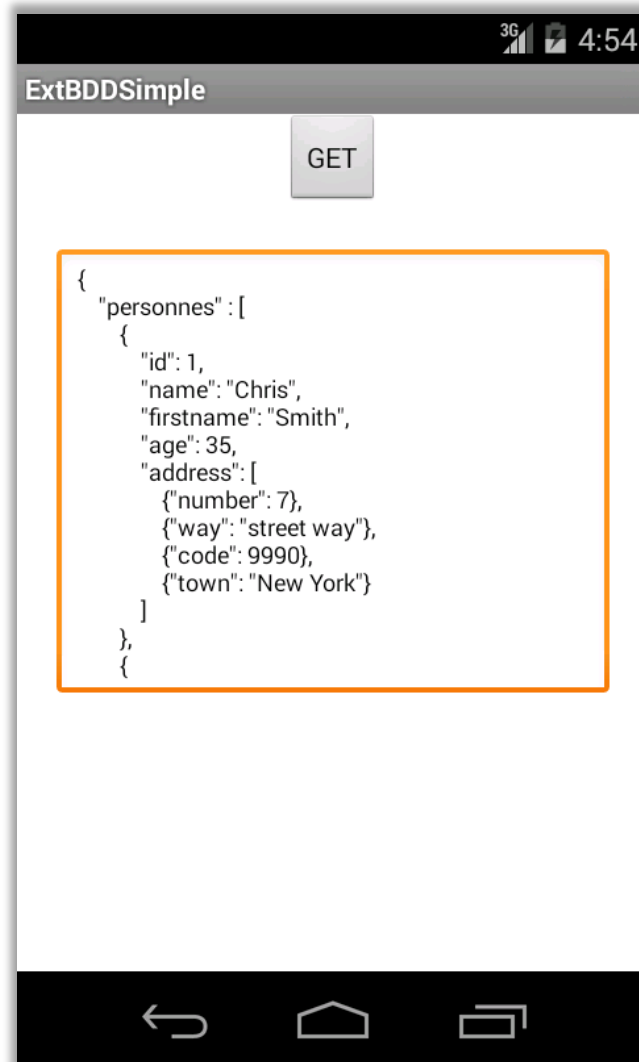
```
class GetUsers extends AsyncTask<Void, Void, String>
```

◉ HttpGet

```
protected String doInBackground(Void... params) {  
    String uri = "http://pahoua-yang.fr/projets/android/personne.json";  
    HttpClient httpClient = new DefaultHttpClient();  
    HttpContext localContext = new BasicHttpContext();  
    HttpGet httpGet = new HttpGet(uri);  
    String text = null;  
    try {  
        HttpResponse response = httpClient.execute(httpGet, localContext);  
        HttpEntity entity = response.getEntity();  
        text = getASCIIContentFromEntity(entity);  
    } catch (Exception e) {  
        return e.getLocalizedMessage();  
    }  
    return text;  
}
```

APPELER UN SERVICE REST VIA L'APPLICATION

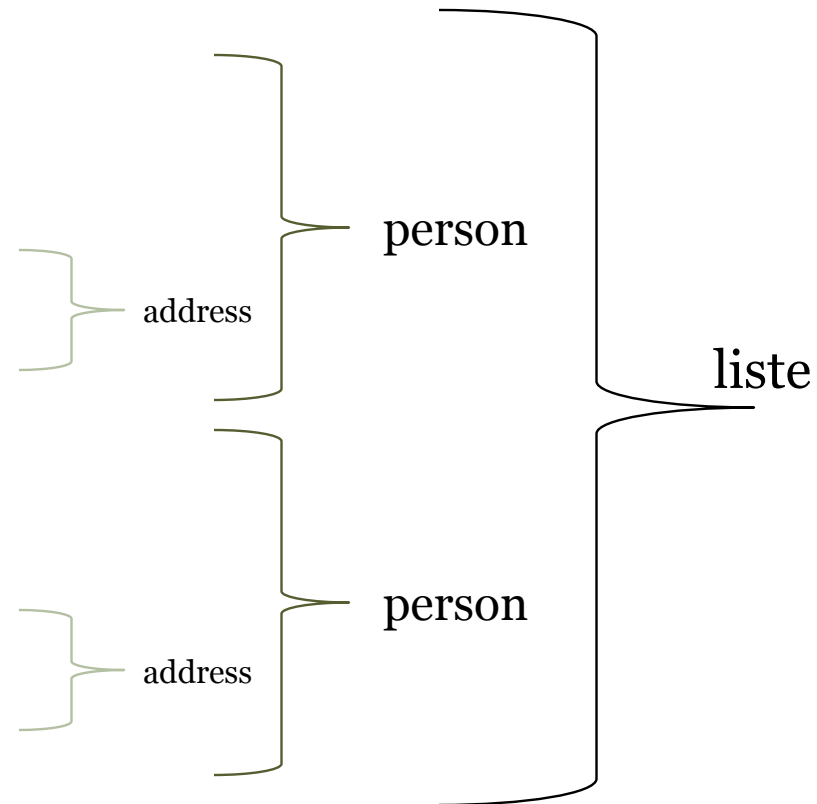
○ Résultat



TRAITEMENT DES DONNÉES RÉCUPÉRÉES - JSON

● Le Json

```
{
  "personnes" : [
    {
      "id": 1,
      "name": "Chris",
      "firstname": "Smith",
      "age": 35,
      "address": [
        {"number": 7},
        {"way": "street way"},
        {"code": 9990},
        {"town": "New York"}
      ]
    },
    {
      "id": 1,
      "name": "Mary",
      "firstname": "Tomson",
      "age": 35,
      "address": [
        {"number": 5},
        {"way": "avenue du parc"},
        {"code": 7590},
        {"town": "san fransisco"}
      ]
    }
  ]
}
```



TRAITEMENT DES DONNÉES RÉCUPÉRÉES - JSON

○ Traitement d'une liste

■ Classe GetUsers

```
protected void onPostExecute(String results) {  
  
    vue = (ListView) this.activity.findViewById(R.id.testPage_listView);  
    final String res = results;  
    JSONObject obj;  
    JSONArray personnes = null;  
    String[][] repertoire = null;  
    try {  
        obj = new JSONObject(results);  
        personnes = obj.getJSONArray("personnes");  
        repertoire = new String[personnes.length()][2];  
  
        for (int i = 0; i < personnes.length(); i++) {  
            final JSONObject personne = personnes.getJSONObject(i);  
            repertoire[i][0] = personne.getString("firstname");  
            repertoire[i][1] = personne.getString("name");  
        }  
    } catch (JSONException e) {  
        e.printStackTrace();  
    }  
  
    List<HashMap<String, String>> liste = new ArrayList<HashMap<String, String>>();  
    HashMap<String, String> element;  
    for (int i = 0; i < repertoire.length; i++) {  
        element = new HashMap<String, String>();  
        element.put("donnee1", repertoire[i][0]);  
        element.put("donnee2", repertoire[i][1]);  
        liste.add(element);  
    }  
  
    ListAdapter adapter = new SimpleAdapter(this.activity, liste,  
        android.R.layout.simple_list_item_2, new String[] { "donnee1",  
            "donnee2" }, new int[] { android.R.id.text1,  
                android.R.id.text2 });  
  
    vue.setAdapter(adapter);  
}
```

Traitement
de la liste
Json de
personnes

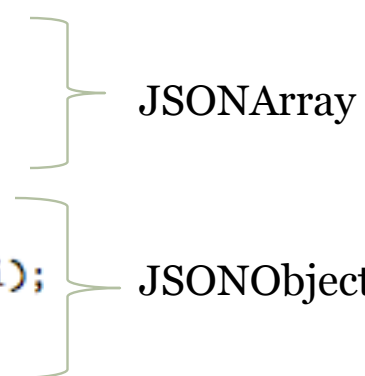
Affichage
de la liste

TRAITEMENT DES DONNÉES RÉCUPÉRÉES - JSON

- Traitement d'une liste
 - Récupération du Json

```
final String res = results;
JSONObject obj;
JSONArray personnes = null;
String[][] repertoire = null;
try {
    obj = new JSONObject(results);
    personnes = obj.getJSONArray("personnes");
    repertoire = new String[personnes.length()][2];

    for (int i = 0; i < personnes.length(); i++) {
        final JSONObject personne = personnes.getJSONObject(i);
        repertoire[i][0] = personne.getString("firstname");
        repertoire[i][1] = personne.getString("name");
    }
}
```



The diagram consists of two curly braces on the right side of the code. The first brace groups the lines: `obj = new JSONObject(results);`, `personnes = obj.getJSONArray("personnes");`, and `repertoire = new String[personnes.length()][2];`. To the right of this brace is the label "JSONArray". The second brace groups the lines: `final JSONObject personne = personnes.getJSONObject(i);`, `repertoire[i][0] = personne.getString("firstname");`, and `repertoire[i][1] = personne.getString("name");`. To the right of this brace is the label "JSONObject".

TRAITEMENT DES DONNÉES RÉCUPÉRÉES - JSON

○ Traitement d'une liste

■ Affichage

```
List<HashMap<String, String>> liste = new ArrayList<HashMap<String, String>>();
HashMap<String, String> element;
for (int i = 0; i < repertoire.length; i++) {
    element = new HashMap<String, String>();
    element.put("donnee1", repertoire[i][0]);
    element.put("donnee2", repertoire[i][1]);
    liste.add(element);
}
```

HashMap

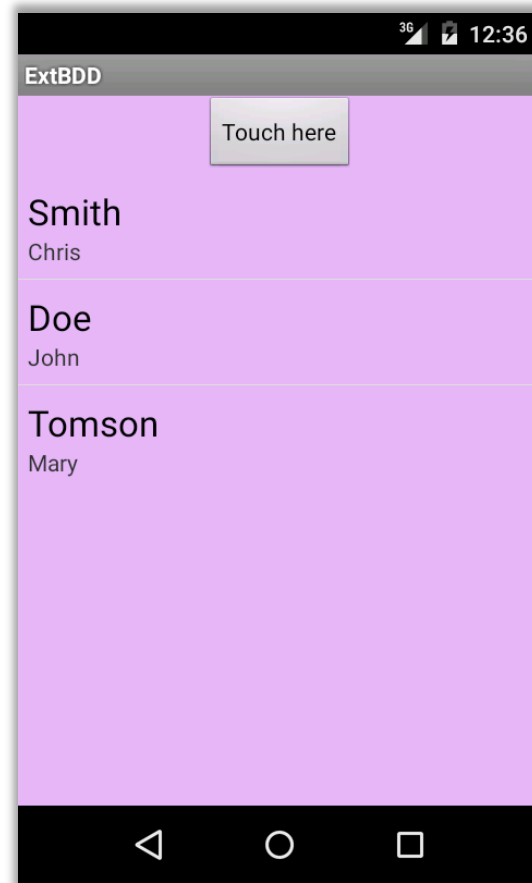
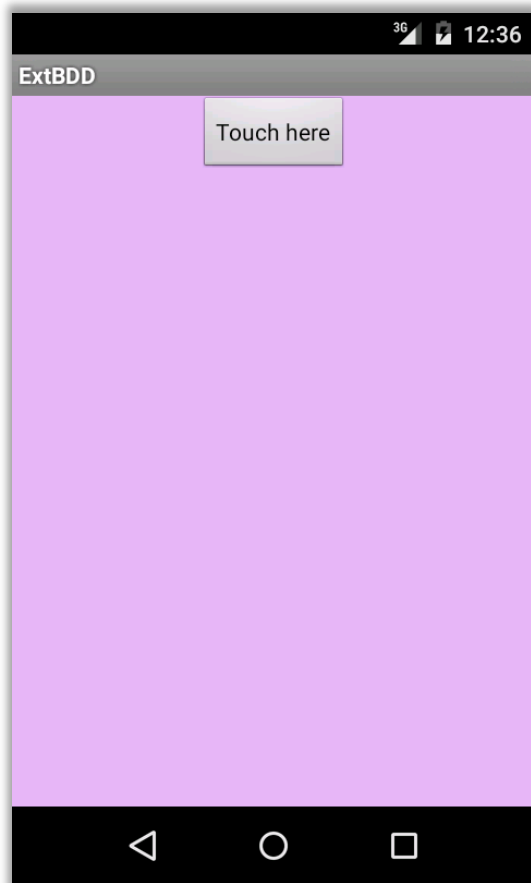
```
ListAdapter adapter = new SimpleAdapter(this.activity, liste,
    android.R.layout.simple_list_item_2, new String[] { "donnee1",
    "donnee2" }, new int[] { android.R.id.text1,
    android.R.id.text2 });
```

ListAdapter

```
vue.setAdapter(adapter);
```

TRAITEMENT DES DONNÉES RÉCUPÉRÉES - JSON

○ Résultat



TRAITEMENT DES DONNÉES RÉCUPÉRÉES - JSON

○ Traitement d'un user

- Envoi d'un Json entre deux activity

```
ListView lv = (ListView) activity.findViewById(R.id.testPage_listView);
lv.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> parent, View view,
        int position, long id) {
        Intent intent = new Intent(activity, UserDetails.class);
        try {
            JSONObject obj = new JSONObject(res);
            JSONArray personnes = obj.getJSONArray("personnes");
            JSONObject personne = personnes.getJSONObject(position);

            // Create a Bundle and Put Bundle in to it
            Bundle bundleObject = new Bundle();
            bundleObject.putString("personne", personne.toString());
            intent.putExtras(bundleObject);
            activity.startActivity(intent);

        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
});
```

putExtras

TRAITEMENT DES DONNÉES RÉCUPÉRÉES - JSON

- Traitement d'un user
 - Réception d'un Json

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

```
    Bundle b = getIntent().getExtras();  
    String pers = b.getString("personne");  
    JSONObject personne;  
    try {
```

```
        personne = new JSONObject(pers);
```

```
        • • •
```

```
    } catch (JSONException e) {  
        e.printStackTrace();  
    }
```

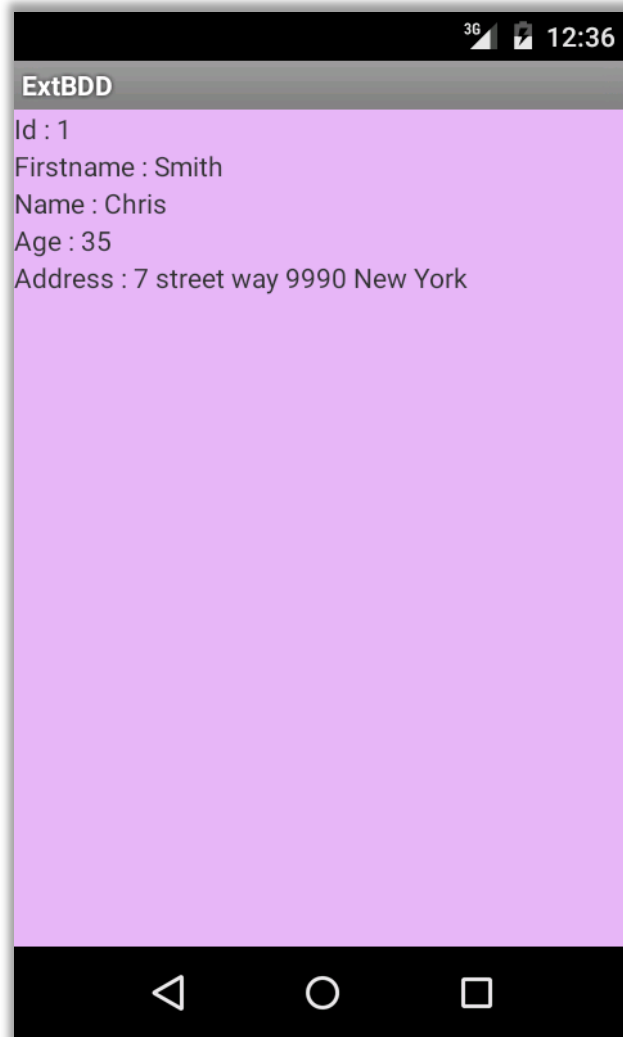
```
}
```



getExtras

TRAITEMENT DES DONNÉES RÉCUPÉRÉES - JSON

○ Résultat



POUR ALLER PLUS LOIN

⊙ Put : la théorie

- Même principe que le HttpGet
- Faire le traitement de la requête
- Il faut créer le webservice adapté

POUR ALLER PLUS LOIN

◎ Put : quelques lignes de code (non-testés)

```
// Creating HTTP client
HttpClient httpClient = new DefaultHttpClient();
// Creating HTTP Post
HttpPost httpPost = new HttpPost(
    "http://www.example.com/login");

// Building post parameters
// key and value pair
List<NameValuePair> nameValuePair = new ArrayList<NameValuePair>(2);
nameValuePair.add(new BasicNameValuePair("email", "user@gmail.com"));
nameValuePair.add(new BasicNameValuePair("message",
    "Hi, trying Android HTTP post!"));

// Url Encoding the POST parameters
try {
    httpPost.setEntity(new UrlEncodedFormEntity(nameValuePair));
} catch (UnsupportedEncodingException e) {
    // writing error to Log
    e.printStackTrace();
}
```


A VOUS DE JOUER

- ◎ Liens pour le tester
 - Téléchargement:
 - pahoua-yang.fr/projets/android/bddExt.zip
 - Tutorial:
 - pahoua-yang.fr/projets/android/bddExtTuto.pdf