ISET SFAX

AU 2022/2023 SEMESTRE I

Correction Devoir Surveillé		
Classe : *SEM31*	***Matière :*** *Développement Mobile Avancé*	***Nb pages :*** *3*
Documents Non Autorisés	***Enseignant :*** *Souissi Hafedh*	***Durée :*** *1 heure*
Barème : 20 = 8 + 12		

Exercice1(8points = 3+5)

```
private void ajouter() {
     SQLiteMusee b= new SQLiteMusee(this, "Musees.db", null, 1);
     SQLiteDatabase db = b.getWritableDatabase();
     ContentValues v = new ContentValues();
     v.put("nom",  edNom.getText().toString());
     v.put("pays", spPays.getSelectedItem().toString());
     v.put("lat",  Double.parseDouble(edLat.getText().toString()));
     v.put("lon",  Double.parseDouble(edLon.getText().toString()));
     db.insert("Musee", null, v);
     db.close();
     edNom.setText("");
     edLat.setText("");
     edLon.setText("");
 }

 private void afficherMusees()  {
     SQLiteTache b = new SQLiteTache(this, "Musees.db", null, 1);
     SQLiteDatabase db = b.getWritableDatabase();
     String sql="Select * From Musee where nom like ?  et pays like  ?";
     String nom="%"+edNom.getText().toString()+"%";
     String pays="%"+ spPays.getSelectedItem().toString()+"%";
     Cursor c = db.rawQuery(sql,new String[]{nom,pays});
     while (c.moveToNext()) {
       String nom  = c.getString(1);
       String pays = c.getString(2);
       double lat  = c.getDouble(3);
       double lon  = c.getDouble(4);
       MarkerOptions m = new MarkerOptions().position(new LatLng(lat,
       lon));
       marker.title("Pays: "+pays);
       marker.snippet("Musée: "+nom);
       mMap.addMarker(m);
     }
     c.close();
     db.close();
 }
```

Exercice2 (12points = 5 + 7)

```
  private void ajouter() {
      RequestQueue queue = Volley.newRequestQueue(this);
      String url = "http://192.168.17.18:80/MeteoMonde/Ajout.php";
      StringRequest sr = new StringRequest(Request.Method.POST, url,
     new Response.Listener<String>() {
       @Override
       public void onResponse(String response) {
       try{
         JSONObject json = new JSONObject(response);
            String reponse = json.getString("ETAT");
            if (reponse.equals("SUCCES"))
               finish();
            else {
           Toast t = Toast.makeToast(Ajout.this,"Problème dans
                           Ajout!",Toast.LENGTH_LONG);
                t.show() ;
            }
       } catch (JSONException error) {
            Toast t = Toast.makeText(Ajout.this, "Problème d'analyse JSON: " +
                    error.getMessage(), Toast.LENGTH_LONG);
               t.show();
       }
       }
     }, new Response.ErrorListener() {
          @Override
          public void onErrorResponse(VolleyError e) {
              Toast t = Toast.makeText(Ajout.this, "Problème d'appel HTTP: " +
                      e.getMessage(), Toast.LENGTH_LONG);
          t.show();
          }
        }) {
  @Override
  public Map<String, String> getParams() throws AuthFailureError {
    HashMap<String, String> headers = new HashMap<String, String>();
    headers.put("nom",  edNom.getText().toString());
    headers.put("pays", spPays.getSelectedItem().toString());
    headers.put("lat",  edLat.getText().toString());
    headers.put("lon",  edLon.getText().toString());
    return headers;
  }
  };
  queue.add(sr);
}
```

```
private void afficherMeteo() {

RequestQueue queue = Volley.newRequestQueue(this);
String url = "http://192.168.17.18:80/MeteoMonde/Meteo.php";
StringRequest sr = new StringRequest(Request.Method.POST, url,
     new Response.Listener<String>() {
       @Override
       public void onResponse(String response) {
try{
         JSONObject json = new JSONObject(response);
           JSONArray aPr = json.getJSONArray("meteo");
           for (int i = 0; i < aPH.length(); i++) {
             JSONObject o = aPr.getJSONObject(i);
             String ville = o.getString("ville");
             String ciel = o.getString("ciel");
             String temp = o.getString("temp");
             String vent = o.getString("vent");
             double lat = Double.parseDouble(o.getString("lat"));
             double lon = Double.parseDouble(o.getString("long"));
             //afficher un marqueur
             MarkerOptions m = new MarkerOptions().position(new LatLng(lat, lon));
             m.title("Météo: "+ville);
             m.snippet("Ciel: "+ciel+" Temp: "+temp+" Vent: "+vent);
             mMap.addMarker(m);
           }
} catch (JSONException error) {
                Toast t = Toast.makeText(Meteo.this, "Problème d'analyse JSON: " +
error.getMessage(), Toast.LENGTH_LONG);
                t.show();
              }
       }
     }, new Response.ErrorListener() {
          @Override
          public void onErrorResponse(VolleyError e) {
              Toast t = Toast.makeText(Meteo.this, "Problème d'appel HTTP: " +
e.getMessage(), Toast.LENGTH_LONG);
          t.show();
          }
        }) {
  @Override
  public Map<String, String> getParams() throws AuthFailureError {
    HashMap<String, String> headers = new HashMap<String, String>();
    headers.put("date", edDate.getText().toString());
    headers.put("pays", (String) spPays.getSelectedItem());
    return headers;
  }
  };
  queue.add(sr);

   }
```