

Devoir Surveillé**Classe : SEM31****Matière : Développement Mobile Avancé****Nb pages : 6****Documents Non Autorisés****Enseignant : Souissi Hafedh****Durée : 1 heure****Barème : 20 = 8 + 12****N.B. Ne donner que le code des méthodes demandées.****Exercice1 (8points = 3+5)**

« **MuseeMonde** » est une application Android qui permet de gérer les emplacements des musées du monde. « **MuseeMonde** » permet d'ajouter un ensemble de musée à une base de données **SQLite** "Musees.db", de rechercher les musées d'un pays par leurs noms et d'afficher le résultat sur une carte. Un musée est défini par un id (id), un nom (nom), un pays (pays), une latitude (lat) et une longitude (lon).

« **MuseeMonde** » contient trois activités « MainActivity », « Ajout » et « Musee » et une classe « SQLiteMusee ». Le code de ces classes et leurs interfaces sont donnés dans « Annexe1 ».

- 1- Donner le code des méthodes **ajouter()** de la classe « **Ajout** » qui permet d'ajouter le musée saisie et d'effacer les EditTexts.
- 2- Donner le code de la méthode **afficherMusees()** de la classe « **Musee** » qui permet de rechercher les musées de la table "Musee" en utilisant la requête de type « Select * From Musee where nom like ? et pays like ? » et d'afficher un marqueur pour chaque musée sur la carte avec la position, le titre et le snippet adéquats.

Exercice2 (12points = 5+7)

« **MeteoMonde** » est une application Android qui permet de gérer la météo des villes des pays du monde. « **MeteoMonde** » permet d'ajouter un ensemble de ville à une base de données **MySQL**, d'afficher la météo des villes d'un pays pour une date saisie sur une carte.

« **MeteoMonde** » contient trois activités : « **MainActivity** », « **Ajout** » et « **Meteo** ». Le code de ces classes et leurs interfaces sont donnés dans « **Annexe2** ».

« **MeteoMonde** » appelle une application Web hébergées dans l'adresse « **http://192.168.17.18 :80/MeteoMonde /** » et elle contient deux pages « **Ajout.php** » et « **Meteo.php** » qui utilisent la méthode « **POST** ».

« **Ajout.php** » prend **quatre paramètres** qui sont le nom (nom), le pays (pays), la latitude (lat) et la longitude (lon) de la ville, elle ajoute la ville à la table « **Ville** » de la base **MySQL** et retourne :

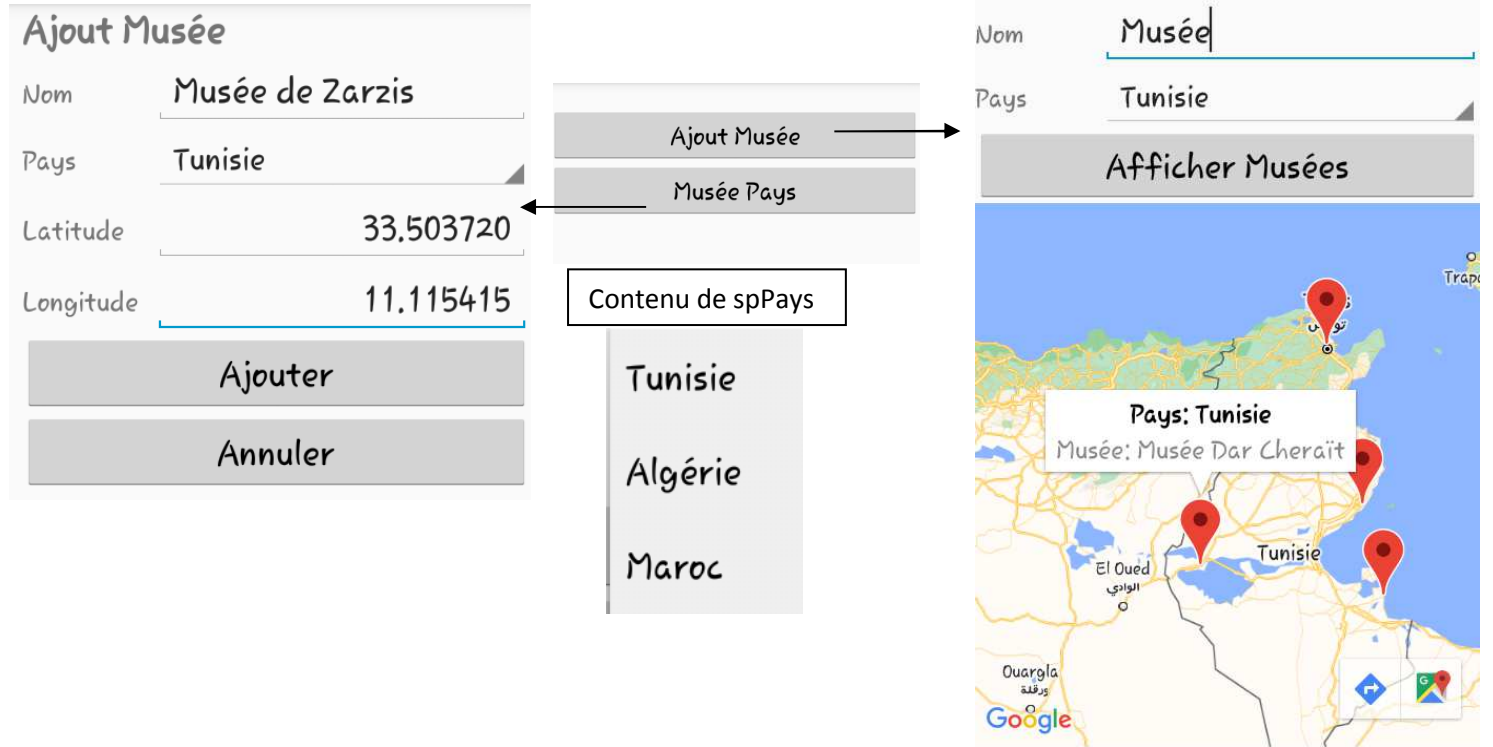
- { **"ETAT" : "SUCCES"** } en cas de succès.
- { **"ETAT" : "ECHEC"** } en cas d'échec.

« **Meteo.php** » prend **deux paramètres** qui sont la date (date) et le pays(pays), elle effectue une recherche dans la table « **Meteo** » de la base **MySQL** et retourne un objet **JSON** de la forme suivante :

```
{  "meteo": [  
    { "ville": "Sfax", "ciel": "Soleil", "temp": "35°C", "vent": "Sud 15Km/h"  
      , "lat": "34.7615155", "lon" : "10.6630579" },  
    { "ville": "Gabès", "ciel": "Soleil", "temp": "36°C", "vent": "Sud/Est 10Km/h"  
      , "lat": "33.8893134", "lon" : "10.0851486" },  
    { "ville": " Tozeur", "ciel": "Soleil", "temp": "39°C", "vent": "Sud 15Km/h"  
      , "lat": "33.9187468", "lon" : "8.084821" }  
  ]  
}
```

- 1- Donner le code de la méthode **ajouter()** de la classe « **Ajout** » qui permet d'ajouter la ville saisie, si la réponse est { **"ETAT" : "SUCCES"** } elle ferme l'activité, sinon elle affiche un message d'erreur.
- 2- Donner le code de la méthode **afficherMeteo()** de la classe « **Meteo** » qui permet de :
 - a. rechercher la météo des villes qui correspondent à la date et au pays saisies,
 - b. analyser la réponse **JSON**,
 - c. d'afficher un marqueur pour la météo de chaque ville sur la carte avec la position, le title et le snippet adéquats.

Annexe1



Placer un marqueur sur une carte

```
static final float LAT =34.756932f;
static final float LONG =10.772176f;
MarkerOptions marker = new MarkerOptions().position(new LatLng(LAT, LONG));
marker.title("Iset Sfax");
marker.snippet("Sfax");
mMap.addMarker(marker);
```

```
package com.musee.monde;
```

```
public class SQLiteMusee extends SQLiteOpenHelper {
    public SQLiteMusee(Context context, String name, SQLiteDatabase.CursorFactory factory,
        int version) {
        super(context, name, factory, version);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        String sql="create table Musee (id INTEGER PRIMARY KEY AUTOINCREMENT,nom TEXT NOT
        NULL,pays NOT NULL,lat REAL,lon REAL);";
        db.execSQL(sql);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    }
}
```

```

package com.musee.monde;
public class MainActivity extends AppCompatActivity{

    ...
}
package com.musee.monde;
public class Ajout extends AppCompatActivity {

    private EditText edNom;
    private Spinner spPays;
    private EditText edLat;
    private EditText edLon;
    private Button btnAjouter;
    private Button btnAnnuler;

    ...

private void ajouter() {
    // Permet d'ajouter le musée saisie à la base SQLite,
    // efface les trois EditText

}

}
package com.musee.monde;
public class Musees extends FragmentActivity implements OnMapReadyCallback {

    private EditText edNom;
    private Spinner spPays;
    private Button btnAfficher;
    private GoogleMap mMap;

    ...

private void afficherMusees() {
    // Exécute une requête de type
    // « Select * From Musee where nom like ? et pays like ? »
    // et affiche un marqueur pour chaque musée sur la carte
    // avec la position, le title et le snippet adéquats

}

}

}

```

Annexe2

Ajout Ville

Nom

Pays

Latitude

Longitude

Ajouter

Annuler

Ajout Ville

Météo Pays

Contenu de spPays

Tunisie

Algérie

Maroc

Ajout Ville

Date

Pays

Afficher Météo



Volley

```
RequestQueue queue = Volley.newRequestQueue(this);
String url = "http://100.20.12.2:80/Page.php";
StringRequest sr = new StringRequest(Request.Method.POST,
    url, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            //Traitement de la réponse
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError e) {
            //Traitement de l'erreur
        }
    }) {
    @Override
    public Map<String, String> getParams() throws AuthFailureError {
        HashMap<String, String> headers = new HashMap<String, String>();
        headers.put("param", "val");
        return headers;
    }
};
queue.add(sr);
```

JSONObject

```
JSONObject json = new JSONObject("{\"c1\":\"v1\",\"c2\":\"12\",\"c3\":\"12.4\"}");
String s = json.getString("c1");
int s = json.getInt("c2");
float s = Float.parseFloat(json.getString("c3"));
```

JSONArray

```
try {
    JSONArray a = json.getJSONArray("liste");
    for (inti = 0; i<a.length(); i++) {
        JSONObject o = a.getJSONObject(i);
    }
} catch (JSONException e) {
    e.printStackTrace();
}
```

```

package com.meteo.monde;
//imports
public class MainActivity extends AppCompatActivity {
    ...
}
package com.meteo.monde;
//imports
public class Ajout extends AppCompatActivity {

    private EditText edNom;
    private Spinner spPays;
    private EditText edLat;
    private EditText edLon;
    private Button btnAjouter;
    private Button btnAnnuler;

    ...

    protected void ajouter() {
        // permet de verifier les données, d'ajouter la ville saisie et d'analyser la réponse
    }

}
package com.meteo.monde;
//imports
public class Meteo extends FragmentActivity implements OnMapReadyCallback {

    private EditText edDate;
    private Spinner spPays;
    private Button btnAfficher;
    private GoogleMap mMap;

    ...

}

    protected void afficherMeteo() {
        //permet de :
        // rechercher la météo des villes qui correspondent à la date et au pays saisies,
        // analyser la réponse JSON,
        // afficher un marqueur pour la météo de chaque ville sur la carte avec la
        // position, le title et le snippet adéquats

    }
}

```