Objectifs :

- Récupérer les cordonnées (latitude, longitude) d'un smartphones à l'aide de GPS,
- Récupérer l'adresse ip de smartphone,
- Récupérer IMEI (International Mobile Equipment Identity) d'un smartphone,
- Enregistrer les données dans une base de données distante.

Partie 1 : Création de la base de données

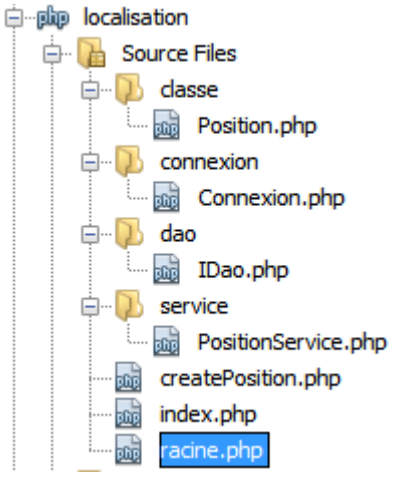
On demande de créer la table position ci-dessous dans la base de données localisation sous MySQL :

```
--
-- Structure de la table `position`
--

CREATE TABLE `position` (
  `id` int(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,
  `latitude` double NOT NULL,
  `longitude` double NOT NULL,
  `date` datetime NOT NULL,
  `imei` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Partie 2 : Développement des services avec PHP

Structure de projet PHP :



La classe Position :

```
<?php

class Position {

    private $id;
    private $latitude;
    private $longitude;
```

```
    private $date;
    private $imei;

    function __construct($id, $latitude, $longitude, $date, $imei) {
        $this->id = $id;
        $this->latitude = $latitude;
        $this->longitude = $longitude;
        $this->date = $date;
        $this->imei = $imei;
    }

    function getId() {
        return $this->id;
    }

    function getLatitude() {
        return $this->latitude;
    }

    function getLongitude() {
        return $this->longitude;
    }

    function getDate() {
        return $this->date;
    }

    function setId($id) {
        $this->id = $id;
    }

    function setLatitude($latitude) {
        $this->latitude = $latitude;
    }

    function setLongitude($longitude) {
        $this->longitude = $longitude;
    }

    function setDate($date) {
        $this->date = $date;
    }

    function getImei() {
        return $this->imei;
    }

    function setImei($imei) {
        $this->imei = $imei;
    }

}
```

La classe Connexion :

```
<?php

class Connexion {

    private $connextion;

    public function __construct() {
        $host = 'localhost';
        $dbname = 'localisation';
        $login = 'root';
        $password = '';
```

```
            try {
                $this->connextion = new PDO("mysql:host=$host;dbname=$dbname", $login, $password);
                $this->connextion->query("SET NAMES UTF8");
            } catch (Exception $e) {
                die('Erreur : ' . $e->getMessage());
            }
        }

        function getConnextion() {
            return $this->connextion;
        }

    }

    ?>
```

L'interface IDao :

```
    <?php

    interface IDao {

        public  function create($obj);

        public  function update($obj);

        public  function delete($obj);

        public  function getById($obj);

        public  function getAll();
    }
```

La classe PositionService :

```
    <?php

    include_once 'dao/IDao.php';
    include_once 'classe/Position.php';
    include_once 'connexion/Connexion.php';

    class PositionService implements IDao   {

        private $listPosition = array();
        private $connexion;
        private $position;


        public function __construct() {
            $this->connexion = new Connexion();
             $this->position = new Position("", "","","","");

        }

        public function create($position) {
              $query = "INSERT INTO position (latitude, longitude, date, imei) VALUES ("
              .$position->getLatitude(). ",".$position->getLongitude().",'".$position-
              >getDate()."','".$position->getImei()."')";
            $req = $this->connexion->getConnextion()->prepare($query);
            $req->execute() or die('SQL');
        }

        public function delete($obj) {

        }

        public function getAll() {
```

```
    }

    public function getById($obj) {

    }

    public function update($obj) {

    }

}
```

Script createPosition :

```
<?php
if($_SERVER["REQUEST_METHOD"] == "POST"){
    include_once 'service/PositionService.php';
    create();
}

function create (){
    $latitude = $_POST['latitude'];
    $longitude = $_POST['longitude'];
    $date = $_POST['date'];
    $imei = $_POST['imei'];
    $ss = new PositionService();

    $ss->create(new Position(1, $latitude, $longitude, $date, $imei));

}
```

Partie 3 : Développement de l'application mobile

Les permissions

Rajouter les permissions suivantes dans le fichier de configuration de votre projet « AndroidManifest.xml » :

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

Intégration de la bibliothèque Volley

Rajouter la dépendance dans « build.gradle » pour intégrer la bibliothèque Volley:

```
  compile 'com.mcxiaoke.volley:library:1.0.18'
```

Déclaration des messages dans le fichier strings.xml

Rajouter les déclarations des chaines formatées dans le fichier strings.xml :

```
<string name="provider_enabled">The provider %s is now enabled</string>
<string name="provider_disabled">The provider %s is now disabled</string>
<string name="provider_new_status">The provider %1$s has now a new status %2$s</string>
<string name="new_location">New Location : Latitude = %1$s, Longitude = %2$s, Altitude = %3$s
avec une précision de %4$s mètres </string>
```

Récupération des cordonnées avec GPS

```
public class MainActivity extends ActionBarActivity {
    private double latitude;
    private double longitude;
    private double altitude;
    private float accuracy;

    RequestQueue requestQueue;

    String insertUrl = "http://192.168.43.228/localisation/createPosition.php";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        requestQueue = Volley.newRequestQueue(getApplicationContext());

        LocationManager locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 60000, 150, new
        LocationListener() {
            @Override
            public void onLocationChanged(Location location) {
                latitude = location.getLatitude();
                longitude = location.getLongitude();
                altitude = location.getAltitude();
                accuracy = location.getAccuracy();
                String msg = String.format(
                        getResources().getString(R.string.new_location), latitude,
                        longitude, altitude, accuracy);

                addPosition(latitude, longitude);

                Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_LONG).show();
            }

            @Override
            public void onStatusChanged(String provider, int status, Bundle extras) {
                String newStatus = "";
                switch (status) {
                    case LocationProvider.OUT_OF_SERVICE:
                        newStatus = "OUT_OF_SERVICE";
                        break;
                    case LocationProvider.TEMPORARILY_UNAVAILABLE:
                        newStatus = "TEMPORARILY_UNAVAILABLE";
                        break;
                    case LocationProvider.AVAILABLE:
                        newStatus = "AVAILABLE";
                        break;
                }
                String msg = String.format(getResources().getString(R.string.provider_new_status),
                provider, newStatus);
                Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_SHORT).show();

            }

            @Override
            public void onProviderEnabled(String provider) {
                String msg = String.format(getResources().getString(R.string.provider_enabled),
                provider);
                Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_SHORT).show();
            }

            @Override
            public void onProviderDisabled(String provider) {
```

```
                String msg = String.format(getResources().getString(R.string.provider_disabled),
                provider);
                Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_SHORT).show();
            }
        });

    }
    void addPosition(final double lat, final double lon) {
        StringRequest request = new StringRequest(Request.Method.POST,
                insertUrl, new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {

            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {

            }
        }) {
            @Override
            protected Map<String, String> getParams() throws AuthFailureError {
                TelephonyManager telephonyManager =
                (TelephonyManager)getSystemService(Context.TELEPHONY_SERVICE);

                HashMap<String, String> params = new HashMap<String, String>();
                SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
                params.put("latitude", lat + "");
                params.put("longitude", lon + "");
                params.put("date", sdf.format(new Date()) + "");
                params.put("imei", telephonyManager.getDeviceId());

                return params;
            }
        };
        requestQueue.add(request);
    }

}
```

Exemple d'insertion :

id	latitude	longitude	date	imei
9	31.64467319	-8.01915503	2016-12-25 17:02:23	359707060971683
10	31.64469002	-8.01915993	2016-12-25 17:18:58	359707060971683
11	31.64468709	-8.01915946	2016-12-25 17:19:48	359707060971683
12	31.64464335	-8.01940147	2016-12-25 18:09:00	359707060971683

Pour récupérer l'adresse ip du client : $_SERVER['REMOTE_ADDR']

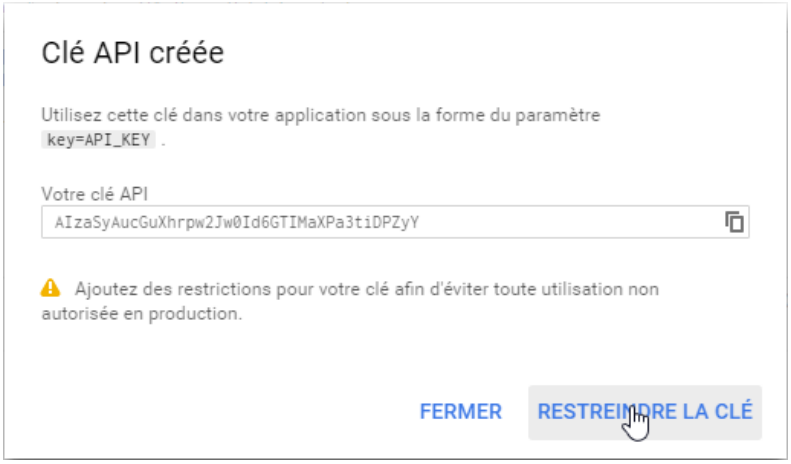
TP : Google Map

Objectifs :

- Intégration d’une Map,
- Afficher les positions dans une Map,

Enoncé :

1. Dans le projet précédant créer une activité de type « Google Map Activity ».
2. Récupérer la clé « google_maps_key », pour ce faire, vous pouvez suivre les étapes dans le fichier xml « google_maps_api.xml ».



3. Implémenter la méthode « getAll () » dans la classe PositionService :

```
public function getAll() {
        $query = "select * from position";
        $req = $this->connexion->getConnextion()->prepare($query);
        $req->execute();
        return $req->fetchAll(PDO::FETCH_ASSOC);
}
```

4. Développer le service permettant de renvoyer les positions sous format Json :

```
<?php

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    include_once 'service/PositionService.php';
    showPositions();
}

function showPositions() {
    $cs = new PositionService();
    header('Content-type: application/json');
    echo json_encode(array("positions" => $cs->getAll()));
}
```

5. Dans l'activité « Main » ajouter un bouton pour naviguer vers l'activité Map.

6. Dans l'activité Google Map Rajouter les déclarations :

```
String showUrl = "http://192.168.43.228/localisation/showPositions.php";
RequestQueue requestQueue;
@Override
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        requestQueue = Volley.newRequestQueue(getApplicationContext());
        setUpMapIfNeeded();
}
```

7. Ré-implémenter la méthode setUpMap() :

```
private void setUpMap() {
  JsonObjectRequest jsonObjectRequest = new JsonObjectRequest(Request.Method.POST,
       showUrl, new Response.Listener<JSONObject>() {
       @Override
       public void onResponse(JSONObject response) {
          try {
               JSONArray positions = response.getJSONArray("positions");
               for (int i = 0; i < positions.length(); i++) {
                  JSONObject position = positions.getJSONObject(i);
                   mMap.addMarker(new MarkerOptions().position(new
                   LatLng(position.getDouble("latitude"),
                   position.getDouble("longitude"))).title("Marker"));
               }
           } catch (JSONException e) {
                    e.printStackTrace();
           }
```

```
        }
        }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {

        }
        });
        requestQueue.add(jsonObjectRequest);
    }
```