

---

### **Atelier 3 : Création de l'activité Google Maps**

#### **Projet : Gestion de syndicat de copropriété**

---

**Objectif :** Intégrer Google maps dans un projet Android

#### **Etape 1 : Configuration de l'API Google Maps**

- 1.1 Créer un compte sur Google Cloud Console
- 1.2 Créez un nouveau projet
- 1.3 Activez l'API Google Maps Android dans le tableau de bord du projet.
- 1.4 Générez une clé d'API pour votre projet.
- 1.5 Stocker la clé dans le fichier `local.properties`,  
`MAPS_API_KEY=YOUR_API_KEY`

#### **Etape 2 : Dépendances de Google Maps**

2.1 Ouvrez votre fichier `build.gradle` (Module: app) et ajoutez la dépendance pour Google Maps

```
implementation 'com.google.android.gms:play-services-maps:18.1.0'
```

2.2 Dans votre fichier `AndroidManifest.xml`, ajouter les lignes suivantes :

```
<meta-data  
    android:name="com.google.android.geo.API_KEY"  
    android:value="{MAPS_API_KEY}" />
```

Et les permissions nécessaires :

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

**Etape 3 :** Créer un nouveau Layout pour votre carte (à vous de choisir le type de

```
<com.google.android.gms.maps.MapView  
    android:id="@+id/mapView"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"/>
```

layout approprié)

### Etape 5 : Initialisation de Google Maps dans l'Activité

```
public class MapsActivity extends AppCompatActivity implements
OnMapReadyCallback {

    private GoogleMap map;
    MapView mapView;
    private LocationManager locationManager;
    private static final int REQUEST_LOCATION_PERMISSION = 1;    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        mMapView = findViewById(R.id.mapView);
        mMapView.onCreate(savedInstanceState);
        mMapView.getMapAsync(this);
    // Initialize LocationManager
    locationManager = (LocationManager)
    getSystemService(Context.LOCATION_SERVICE);
    }
    //initialize the map and request location updates.
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        // Configurer la carte comme vous le souhaitez
        // Add a marker in your place and move the camera
        LatLng place = new LatLng(33.594584, -6.712530); //just example
        mMap.addMarker(new MarkerOptions().position(place).title("Marker in Rabat"));
        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(place,12));
    //to get current Location
        if (ActivityCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION) !=
        PackageManager.PERMISSION_GRANTED
        && ActivityCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_COARSE_LOCATION) !=
        PackageManager.PERMISSION_GRANTED) {
            // Request runtime permissions if not granted
            ActivityCompat.requestPermissions(this, new
            String[]{Manifest.permission.ACCESS_FINE_LOCATION},
            REQUEST_LOCATION_PERMISSION);
            return;
        }
    }
```

// is a callback method triggered whenever the location changes. Here, you add a marker for the current location and move the camera to focus on it.

```
@Override
public void onLocationChanged(@NonNull Location location) {
    map.clear();
```

**6.1** Utiliser les méthodes de l'objet GoogleMap pour personnaliser la carte selon vos besoins