

TP Google Map

0. Prérequis :

- Assurez-vous que le téléphone a bien connecté à l'internet pour afficher la carte.

1. Objectif du TP

L'objectif de ce TP est d'apprendre à utiliser l'API de Google Map sur Android. On va réaliser une activité qui présente l'utilisation courante de l'API.

2. Introduction

Code initial de TP :

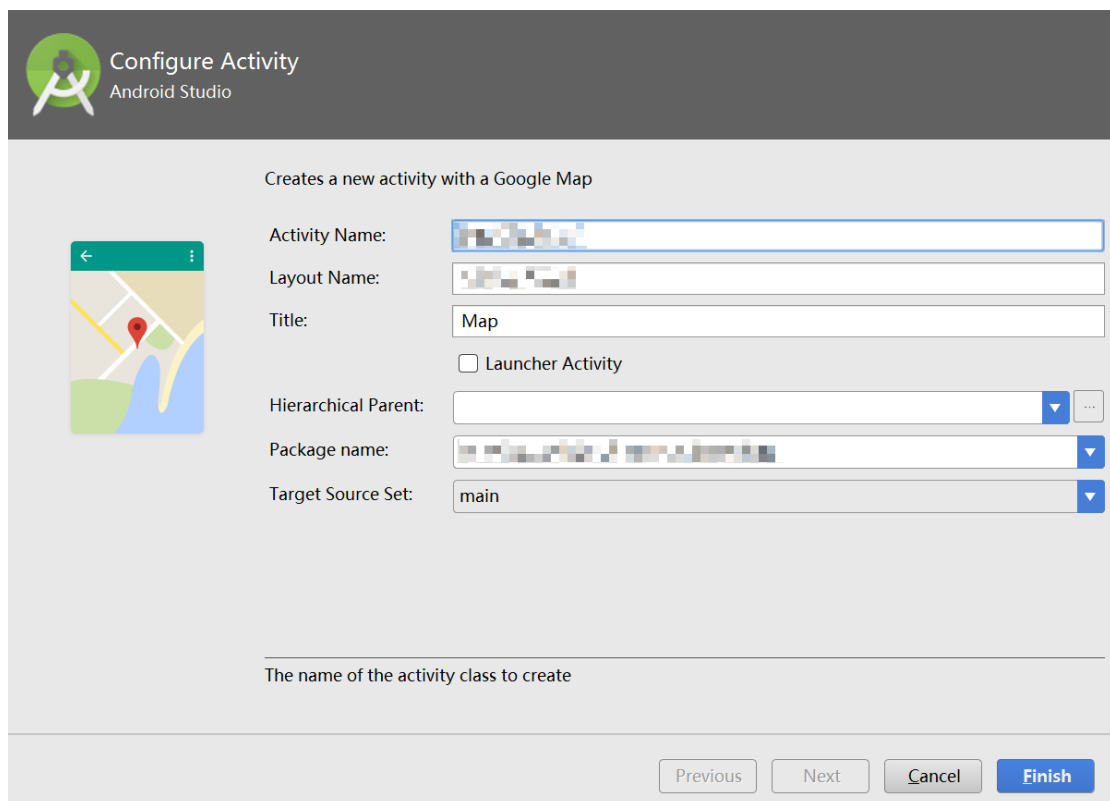
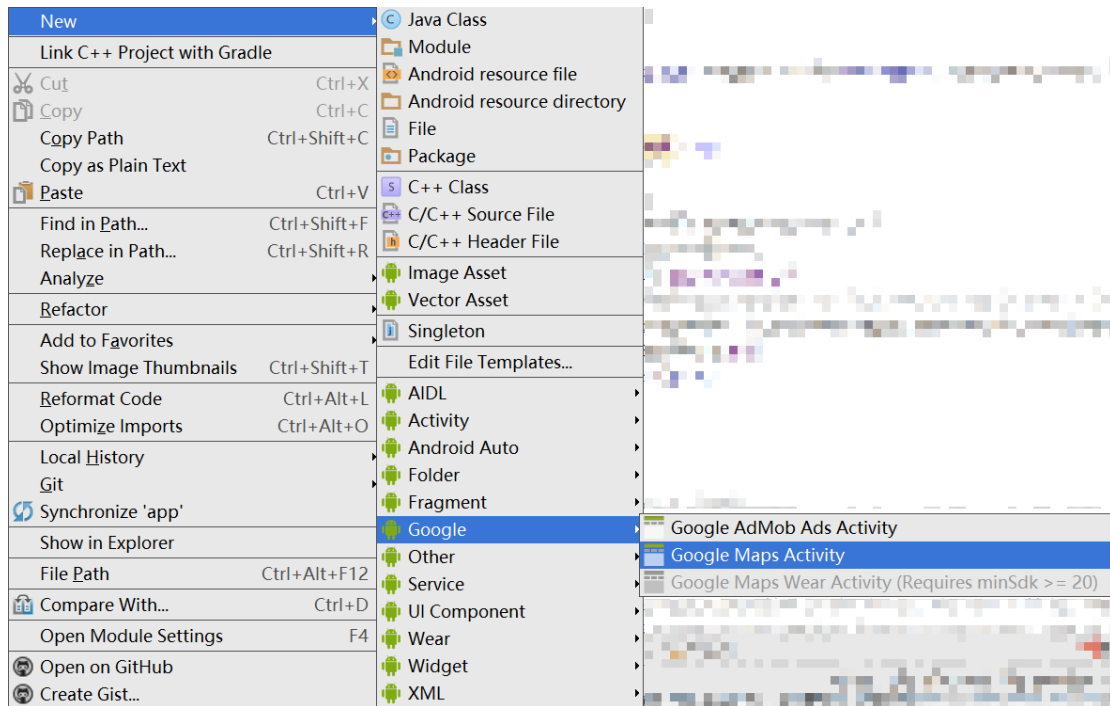
https://drive.google.com/file/d/0B_jq0BJo4ikCRDF1UFVKOVhfVWM/view?usp=sharing

Code final de TP :

https://drive.google.com/file/d/0B_jq0BJo4ikCd2J0QkNzSTloUEE/view?usp=sharing

2.1 Comment créer une activité Google Map ?

Dans un projet existant, on peut créer une activité en clic droit sur l'**app**.



Si c'est la première fois qu'on crée une activité de Google Map, Android Studio va générer un fichier « **google_maps_api.xml** » dans le répertoire « **values** ». Le contenu indiqué comme ci-après :

```

<resources>
  <!--
    TODO: Before you run your application, you need a Google Maps API key.

    To get one, follow this link, follow the directions and press "Create" at the end:

    https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&keyType=CLIENT_

    You can also add your credentials to an existing key, using this line:
    [REDACTED]

    Alternatively, follow the directions here:
    https://developers.google.com/maps/documentation/android/start#get-key

    Once you have your key (it starts with "AIza"), replace the "google_maps_key"
    string in this file.
  -->
  <string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">
    [REDACTED]
  </string>
</resources>

```

Copier-coller le lien du fichier google_maps_api.xml ; Vous suivrez les indications jusqu'à la fin pour avoir la clé d'API.

Remarque : Si vous avez déjà une clé d'API de Google Map, Vous devez aller au Google Developer Console pour ajouter votre nouvelle application dans ce API.

Si tous fonctionnent bien, l'activité doit charger les données de Google Map.



Après créer l'activité de Map, Android Studio va générer automatiquement la méthode `onCreate()` et `onMapReady()`.

La méthode `getMapAsync(this)` enregistre une méthode `onMapReadyCallback` ;

Quand la carte est prête, la méthode `onMapReady()` va être appelée.

Remarque : si on manipule la carte dans la méthode `onCreate()`, cela risque que la carte n'est pas encore créée. Il est possible de poser l'erreur de `NullPointerException`.

2.2 Google Map c'est un objet.

La classe de Google Map décompilé :

```
public final class GoogleMap {
    public static final int MAP_TYPE_NONE = 0;
    public static final int MAP_TYPE_NORMAL = 1;
    public static final int MAP_TYPE_SATELLITE = 2;
    public static final int MAP_TYPE_TERRAIN = 3;
    public static final int MAP_TYPE_HYBRID = 4;
    private final IGoogleMapDelegate anP;
    private UiSettings anQ;

    protected GoogleMap(IGoogleMapDelegate var1) {
        this.anP = (IGoogleMapDelegate)zzaa.zzy(var1);
    }

    public final CameraPosition getCameraPosition() {
        try {
            return this.anP.getCameraPosition();
        } catch (RemoteException var2) {
            throw new RuntimeRemoteException(var2);
        }
    }
}
```

Quand on définit l'objet de Map :

```
private GoogleMap mMap;
```

Google Map est un objet qui est placé dans un `MapView` ou un `MapFragment` selon l'utilisation différente. :

- Si vous voulez juste ajouter le Map dans une Activité, utiliser le `MapFragment`.
- Si vous voulez ajouter un Map dans un Fragment, utiliser le `MapView`.

Toutes les modifications sur l'objet de Google Map.

On ajoute un marker :

```
mMap.addMarker(new MarkerOptions().position(latLng).title("hahaha"));
```

On dessine un polygone :

```
PolygonOptions lille1Surface = new PolygonOptions()
    .add(new LatLng(50.612532, 3.142158),
        new LatLng(50.610291, 3.13722),
        new LatLng(50.606268, 3.135161),
        new LatLng(50.606823, 3.144594),
        new LatLng(50.610788, 3.146718));
Polygon polygon = mMap.addPolygon(lille1Surface, strokeColor(Color.RED));
```



Comment peut-on initialiser la carte ? La carte peut positionner selon Longitude et Latitude. Imaginer si on positionne sur *Learning Center*. Il faut juste deux étapes :

- Créer un objet LatLng avec les données de longitude et latitude. Pour *Learning Center*, c'est (50.609291, 3.142015).
- Bouger et zoomer la carte au Learning Center.

```
public void onMapReady(GoogleMap googleMap) {  
    mMap = googleMap;  
  
    // Add a marker in Sydney and move the camera  
    LatLng sydney = new LatLng(-34, 151);  
    mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));  
    mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));  
}
```

Remarque : les modifications doit être dans la méthode `onMapReady()`

2.3 L'interaction avec le Map

Google vous propose les APIs différents pour l'interaction. Vous pouvez activer ou désactiver la geste ou l'opération selon vos demandes.

Si vous voulez désactiver la rotation de la carte, utilise la méthode `UiSettings.setZoomGesturesEnabled(Boolean)` ; Si vous voulez désactiver le zoom, utilise la méthode `UiSettings.setRotateGesturesEnabled(Boolean)`.

Les widgets de Google Map se trouve dans l'`UiSettings` : Commandes de zoom, Boussole, Bouton Ma position etc. Vous pouvez décider de les afficher ou pas.

3. à vous de jouer !

- Positionner la carte au *Learning Center*, Campus Lille 1 ; LatLng = (50.609291, 3.142015)

- Réaliser la fonctionnalité d'Activer/Désactiver le zoom et rotation de la carte.
- Ajouter un évènement de clic long. C'est évènement est pour ajouter un marker dans la carte. Changer la couleur quand on clique sur le marker.
- Dessiner le contour de Campus en polygone.

Fin du TP

C'est tout pour ce TP, j'espère que vous comprenez comment l'API fonctionne. Il existe d'autre choses dans ce API, mais nous n'avons pas de temps de les aborder ici.

Références

<https://developers.google.com/maps/documentation/android-api>