**GEO FEATURES (GEOCODING)**

# 1. Geocoding là gì ?

Geocoding là việc chuyển đổi một địa chỉ thông thường thành một tọa độ địa lý trên bản đồ (vĩ độ và kinh độ), qua đó giúp ta có thể đánh dấu hoặc định vị một vị trí trên bản đồ.

Reserve Coding thì ngược lại với Geocoding, chuyển một tọa độ địa lý thành một địa chỉ mà con người đọc được.

Ví dụ: Lottle Nam Sài Gòn có địa chỉ: 469 Đ.Nguyễn Hữu Thọ, Tân Hưng, Quận 7, Thành phố Hồ Chí Minh sẽ được chuyển thành: 10.740966, 106.702012

# 2. Location – Based Service (dịch vụ dựa trên vị trí)

Geocoding được sử dụng để hỗ trợ cho việc sử dụng các dịch vụ định vị trí như Location – Based Service

Dịch vụ dựa trên vị trí (LBS) là các dịch vụ được cung cấp qua thiết bị di động dựa vào vị trí của thiết bị. Một số ứng dụng LBS phổ biến như: thông tin địa phương, chỉ đường, địa điểm ưu thích, quản lý giám sát vị trí phương tiện,…

Phương thức hoạt động của LBS

Dựa vào vị trí và các yêu cầu của thiết bị, thông qua mạng Internet, nhà cung cấp gửi thông tin đến thiết bị của người dùng.

Được thực hiện thông qua 3 lớp:

1. Location: chứa thông tin vị trí (toạ độ)

2. LocationManager: cung cấp quyền truy cập vào các dịch vụ định vị trí

3. Location Provoder: cung cấp dữ liệu vị trí trong androi. Gồm 2 loại:

+ GPS Location Provider

+ Netwwork Provider

# 3. Ứng dụng Google Maps API vào Android

- Google Maps API là dịch vụ do Google Play Service hỗ trợ dùng để cung cấp các thao tác với bản đồ và thông tin vị trí.

- Ứng dụng vào Android:

+ Tạo Project trong Google Console để kích hoạt thư viện API và tạo API Key

\_B1: Vào trang Google Console

\_B2: Tạo Project: Chọn Select Project -> New Project

A screenshot of a computer

Description automatically generated with medium confidence

\_B3: Chọn Project vừa tạo -> Chọn APIs and Services -> Chọn Library

Graphical user interface, application, email, website

Description automatically generated

\_B4: Chọn Maps SDK for Android -> Nhấn Enable

Graphical user interface, text, application, email

Description automatically generated

\_B5: Tạo API key: Chọn APIs and Services -> Credential ->   
Create Credentials -> API key -> tạo xong bấm Close

A screenshot of a computer

Description automatically generated

Graphical user interface, application

Description automatically generated

+ Tạo Project Android Studio mới và khởi tạo Google Map:

\_B1: Thêm Library Dependences:

**implementation 'com.google.android.gms:play-services-maps:17.0.0'**

\_B2: Thêm Permission và meta-data trong AndroidManifest:

**<uses-permission android:name="android.permission.INTERNET"/>**

**<meta-data  
 android:name="com.google.android.geo.API\_KEY"  
 android:value="YOUR\_API\_KEY"/>**

YOUR\_API\_KEY là Key đã tạo được ở phần trước, copy bỏ vào.

\_B3: Tạo fragment cho layout: activity\_main:

**<fragment  
 android:id="@+id/my\_map"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" android:name="com.google.android.gms.maps.SupportMapFragment“/>**

+ Các chức năng của Google Maps API (Cách sử dụng):

Để sử dụng được các chức năng của Google Maps API, trước hết ta cần thực hiện:

\_B1: Kế thừa Interface: OnMapReadyCallback trong MainActivity

\_B2: Override lại phương thức onMapReady(GoogleMap googleMap)

\_B3: Tạo biến biến ánh xạ tới fragment đã tạo bên layout: main\_activity

**public class MainActivity extends AppCompatActivity implements OnMapReadyCallback {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 SupportMapFragment supportMapFragment = (SupportMapFragment) getSupportFragmentManager()  
 .findFragmentById(R.id.*my\_map*);   
 supportMapFragment.getMapAsync(this);  
   
 }  
  
 @Override  
 public void onMapReady(GoogleMap googleMap) {  
   
 }  
}**

Các chức năng của Google Maps API:

1. Một số chức năng cơ bản

+ Thêm Marker vào Maps: Dựa vào vĩ độ và kinh độ, ta có thể đánh đấu vị trí của một điểm cụ thể trên bản đồ với dòng lệnh tại hàm onMapReady()

**@Override  
public void onMapReady(GoogleMap googleMap) {  
  
 LatLng tdtu = new LatLng(10.740966, 106.702012);  
  
 googleMap.addMarker(new MarkerOptions()  
 .position(tdtu)  
 .title("Lottle Nam Sài Gòn")  
 .snippet("lottle"));  
  
 googleMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(tdtu, 18));  
}**

+ Đổi type hiển thị với dòng lệnh:

**googleMap.setMapType(GoogleMap.*MAP\_TYPE\_SATELLITE*);**

\_ GoogleMap.MAP\_TYPE\_SATELLITE : kiểu vệ tinh.

\_ GoogleMap.MAP\_TYPE\_TERRAIN : kiểu địa hình.

\_ GoogleMap.MAP\_TYPE\_HYBRID : kiểu hỗn hợp.

\_ GoogleMap.MAP\_TYPE\_NORMAL : kiểu đơn giản.

+ Đổi syle hiển thị với dòng lệnh:

**googleMap.setMapStyle(MapStyleOptions.  
 *loadRawResourceStyle*(this, R.raw.*map\_style\_aubergine*));**

Để tiến hành đổi style, trước hết t cần tải file Json từ link:

<https://mapstyle.withgoogle.com/>

Sau đó copy code bỏ vào thư mục raw trong res của Androi Studio

+ Polyline: Chức năng giúp nối tọa độ được chỉ định trên bản đồ (áp dụng vào việc chỉ đường)

**googleMap.addPolyline(new PolylineOptions()  
 .add( new LatLng(10.73384803463463, 106.69754329737148),  
 new LatLng(10.731998266154712, 106.69903894678845),  
 new LatLng(10.73280116932693, 106.70129176274014),  
 new LatLng(10.729274889512252, 106.70171140492721)));**

2. Autocomplete Place

Hỗ trợ gợi ý các địa chỉ khi ta tìm kiếm vị trí trên Google Maps bằng cách sử dụng thư viện được hỗ trợ là Places API. Để sử dụng chức năng này cần kích hoạt Places API tại Project ta đã tạo trước đó ở Google console. Kích hoạt giống như đã làm với chức năng Maps SDK for Android đã làm trước đó.

Để sử dụng Autocomplete Place, trước hết ta cần thêm thư viện:

**implementation 'com.google.android.libraries.places:places:2.4.0'**

Sau đó tại hàm onCreate của MainActivity.java, khởi tạo Autocomplete Place và chạy:

**placeSearch = (TextView) findViewById(R.id.*place\_search*);  
  
//Kiem tra khoi tao dia diem  
if (!Places.*isInitialized*()) {  
 Places.*initialize*(MainActivity.this, api\_keys);  
}  
  
placeSearch.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
  
 List<Place.Field> fields = Arrays.*asList*(Place.Field.*ID*, Place.Field.*NAME*,  
 Place.Field.*ADDRESS*, Place.Field.*LAT\_LNG*);  
  
 Intent intent = new Autocomplete.IntentBuilder(AutocompleteActivityMode.*OVERLAY* , fields).build(MainActivity.this);  
  
 startActivityForResult(intent, 1);  
 }  
});**

Kết quả sau khi thực hiện Autocomplete Place sẽ được trả về tại hàm onActivityResult():

**@Override  
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
  
 if (requestCode == 1 && resultCode == *RESULT\_OK*) {  
 Place place = Autocomplete.*getPlaceFromIntent*(data);  
  
 myMap.addMarker(new MarkerOptions()  
 .position(place.getLatLng())  
 .title(place.getName()));  
  
 myMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(place.getLatLng(), 12));  
 }  
}**

**Ghi chú: Set các thuộc tính trước khi thực hiện các hàm trên:**

**Private String api\_keys = “ lấy từ project đã tạo”**

**Private textView placeSearch**

**List<Place.Field> fields**

**Private Google myMap**

3**. Direction:** sử dụng HTTP để lấy về file dữ liệu dạng JSON hoặc XML chứa tọa độ của các vị trí nối điểm đầu và điểm cuối tạo thành một đường đi hoàn chỉnh bằng cách sử dụng thư viện Direction API là một Web service. Ta có thể hiểu đơn giản Direction là việc ta nối các điểm vị trí lại với nhau bằng chức năng Polyline để tạo thành đường đi hoàn chỉnh.

Để sử dụng Direction, ta thực hiện theo các bước sau:

+ Bước 1: Kích hoạt Direction API trên Project đã tạo. (giống với cách kích hoạt SDK Maps for Android)

+ Bước 2: Gửi yêu cầu lấy dữ liệu là file JSON chứa tọa độ đường đi.

Cú pháp có dạng:

**https://maps.googleapis.com/maps/api/directions/json?origin=vị trí &destination=vị trí&key= YOUR\_API\_KEY**

Ví dụ ta đi từ Lottle Nam Sài Gòn tới trường Tôn Đức Thắng Q7:

**https://maps.googleapis.com/maps/api/directions/json?origin= 10.7410, 106.7020 &destination= 10.7320, 106.6990&key= YOUR\_API\_KEY(lấy từ project ở google console)**

+ Bước 3: Tiến hành đọc và lưu dữ liệu từ file JSON vừa lấy được.

+ Bước 4: Từ các tọa độ đọc được, tiến hành vẽ Polyline.

4**. Nearby Search:** Cũng giống như AutoComplete Place, sử dụng thư viện Place API, nhưng Nearby Search sử dụng HTTP để lấy về file dữ liệu dạng JSON chứa tọa độ của các vị trí trong vòng bán kính mà chúng ta muốn.

Để sử dụng Nearby Search, ta cần thực hiện theo các bước:

+ Bước 1: Kích hoạt Place API trên Project đã tạo. (giống với cách kích hoạt SDK Maps for Android)

+ Bước 2: Gửi yêu cầu lấy dữ liệu là file JSON chứa tọa độ các địa điểm nằm trong bán kính.

+ Bước 3: Tiến hành đọc và lưu dữ liệu từ file JSON vừa lấy được.

+ Bước 4: Từ các tọa độ đọc được, tiến hành thêm những marker để đánh dấu.