

# Programmation Android

## TP6 –Géolocalisation et Google Maps sous Android

### Partie 1 : Géolocalisation

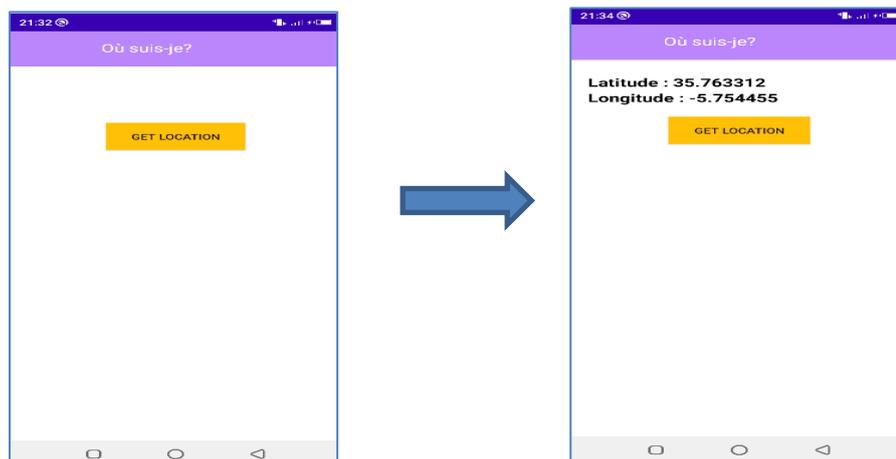
Pour réaliser la tâche de géolocalisation, on peut passer par trois étapes:

1. L'ajout de TextViews ou de Toast dans le fichier xml,
2. Déclarer dans le fichier Manifeste la permission d'utiliser le fournisseur de position (par exemple le GPS),
3. Avoir l'accès aux services Android Location via la classe LocationManager,

Dans un premier lieu, on fait l'affichage des coordonnées de votre smartphone (Latitude et Longitude) dans des TextViews

#### Etape 1 : Ajout de TextViews dans le fichier XML

Créer le fichier xml (**activity\_main.xml**) dans lequel vous déclarez les deux TextViews. L'interface ressemble à celle-ci:



#### Etape 2 : Utilisation de Location Manager

D'une part, pour utiliser le Location Manager, on doit s'assurer que le Google Play Services est disponible dans le fichier build.gradle.

D'autre part, Il faut ajouter une autorisation permettant de détecter les coordonnées de l'utilisateur. Dans un premier temps nous n'allons pas utiliser le capteur GPS mais dans l'exemple suivant on va se baser sur le réseau pour se localiser. Pour ce faire, ajoutons la ligne suivante au fichier AndroidManifest.xml :

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

Dans le fichier MainActivity.java, vous ajoutez le code suivant :

```
public class MainActivity extends Activity {  
    private TextView latitudeField;  
    private TextView longitudeField;  
    private LocationManager localisations;
```

```

private Location position;
private String provider;
private Button butlocation;
private static final int REQUEST_CODE = 101;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    latitudeField = (TextView) findViewById(R.id.TextView02);
    longitudeField = (TextView) findViewById(R.id.TextView04);
    butlocation = (Button) findViewById(R.id.but);
    butlocation.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                if
(getApplicationContext().checkSelfPermission(ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
getApplicationContext().checkSelfPermission(ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
                    requestPermissions(new
String[]{ACCESS_FINE_LOCATION, REQUEST_CODE});
                    return;
                }
            }

            LocationManager localisations = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
            Location provider =
localisations.getLastKnownLocation(NETWORK_PROVIDER);
            if (provider != null) {
                latitudeField.setText("Latitude : " + provider.getLatitude());
                longitudeField.setText("Longitude : " +
provider.getLongitude());
            } else latitudeField.setText("Location not available ");
        }
    });
}

```

### Etape 3: Ajout de la méthode de la mise à jour des coordonnées

Dans le MainActivity.java , vous ajoutez la méthode requestLocationUpdates:

```

@Override
protected void onResume() {
    super.onResume();

    localisations.requestLocationUpdates(provider, 60000, 1, this);
}

```

### Etape 4: redéfinir les méthodes obligatoires de LocationListener

l'activité MainActivity doit implémenter l'interface LocationListener et redéfinir les méthodes de cette interface :

```

public class MainActivity extends Activity implements LocationListener {

.....
@Override
protected void onPause() {
    super.onPause();
}

```

```

        localisations.removeUpdates(this);
    }

    @Override
    public void onLocationChanged(@NonNull Location location) {
        if (location != null) {
            latitudeField.setText("Latitude : " + location.getLatitude());
            longitudeField.setText("Longitude : " +
location.getLongitude());

            } else latitudeField.setText("Location not available ");

            Toast.makeText(MainActivity.this, "Nouvelle position",
Toast.LENGTH_SHORT).show();
        }

    @Override
    public void onProviderEnabled(String provider) {
        Toast.makeText(this, "Enabled new provider " + provider,
            Toast.LENGTH_SHORT).show();
    }
    @Override
    public void onProviderDisabled(String provider) {
        Toast.makeText(this, "Disabled provider " + provider,
            Toast.LENGTH_SHORT).show();
    }
}

```