



# Services de Localisation



# Etapes d'utilisation des services de localisation

---

## 1. Obtenir une instance de LocationManager

```
LocationManager lm= (LocationManager) getSystemService(LOCATION_SERVICE);
```

Définie dans la classe Context, elle permet d'obtenir les services systèmes.

Nom du service de localisation:  
Context.LOCATION\_SERVICE

La classe LocationManager permet l'accès au service de localisation

- Obtenir la position courante
- Supervision des mises à jour de la position courante.
- Demander une notification lorsqu'une position donnée est atteinte.

## 2. Choisir le fournisseur de localisation

- La plupart des terminaux Android supportent deux fournisseurs de positionnement
- GPS
  - LocationManager.GPS\_PROVIDER
  - Une grande précision (peut atteindre 3 m)
- Réseau
  - LocationManager.NETWORK\_PROVIDER
  - Utilise les relais mobiles et les nœuds WI-FI.
  - La précision dépend du nombre de nœuds WI-FI et des relais mobiles.



### 3. La classe Location

la classe Location contient des informations d'un point géographiques

- Position
  - `getLatitude`, `getLongitude`, `getAccuracy` (en mètres)
- Temps
  - `getTime`
    - Dépend du type de fournisseur
  - `getElapsedRealTimeNanos`
    - Retourne le temps en nanosecondes depuis le démarrage du terminal.
    - Indépendante du fournisseur de localisation
    - Depuis Android 4.2 (API Level 17)
- Autres informations qui dépendent du fournisseurs
  - `getSpeed`, `getBearing`, `getAltitude`
  - Il faut utiliser les méthodes `hasSpeed`, `hasBearing` et `hasAltitude` pour déterminer si la valeur est définie

#### 4. Ajouter les permissions appropriées dans le fichier AndroidManifest.xml

Pour utiliser les services de localisation l'application doit avoir les permissions nécessaires qui dépendent du niveau de précision souhaité.

- Pour une précision approximative (WI-FI, relais)

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

- Pour une précision élevée (GPS)

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

## 5. Implémenter les fonctions de rappel

Pour obtenir les mises à jour, il faut implémenter l'interface `LocationListener` qui définit les méthodes:

- `public void onLocationChanged(Location location)`
- `public void onProviderDisabled(String provider)`
- `public void onProviderEnabled(String provider)`
- `public void onStatusChanged(String provider, int status, Bundle extras):`

```
public class MLocationListener implements LocationListener {
    final String _logTag = "Localisation";
}

@Override
public void onLocationChanged(Location location) {
    String provider = location.getProvider();
    double lat = location.getLatitude();
    double lon = location.getLongitude();
    float precision = location.getAccuracy();
    long temps = location.getTime();
    String logMessage = String
        .format("Fournisseur:%s;Latitude:%s;Longitude:%s;Précision:%s;Durée:%s",
            provider, lat, lon, precision, temps);
    Log.d(_logTag, "Nouvelle position:" + logMessage);
}

@Override
public void onProviderDisabled(String provider) {
    Log.d(_logTag, "Provider Désactivé:" + provider);
}

@Override
public void onProviderEnabled(String provider) {
    Log.d(_logTag, "Monitor Location - Provider Activé:" + provider);
}
}
```

## 6. Demander les mises à jour de position

Pour obtenir les informations de localisation la classe `LocationManager` fournit les méthodes suivantes:

- `requestLocationUpdates`: obtenir les mises à jour à intervalle régulier.
  - Paramètres:
    - Fournisseur: `LocationManager.GPS_PROVIDER` ou bien `LocationManager.NETWORK_PROVIDER`
    - Période de l'intervalle en ms
    - Distance minimale du déplacement en m
    - Une instance de `LocationListener`
  - La méthode `removeUpdates` permet d'arrêter l'écoute des mises à jour.
- `getLastKnownLocation`: retourne la dernière position reçu par le fournisseur.
  - Paramètre: le fournisseur.
- `requestSingleUpdate`:

## 7. La classe Address

La classe Address contient des informations détaillées sur un emplacement

- Informations détaillées
  - getThoroughfare: rue
  - getLocality: ville
  - getAdminArea: province ou état
  - getPostalCode: code postal
  - getCountryName: pays
  - getCountryCode: code du pays
- Afficher les données associées à une adresse:
  - getMaxAddressLineIndex: indice de la dernière ligne dans l'adresse
  - getAddressLine( indice): retourne une ligne
- Coordonnées de l'adresse
  - getLatitude/getLongitude

## 9. La classe Geocoder

La classe Geocoder permet de récupérer une liste d'adresses associées à un emplacement.

Méthodes:

`getFromLocation (latitute, longitude, nombre_adresses)`: retourne une liste d'adresses, le dernier paramètre définit le nombre maximal d'adresses à retourner.

`getFromLocationName(String adresse)`

`GeoCoder.isPresent`: une méthode statique qui retourne true, si les bibliothèques google nécessaires sont présentes dans le terminal.

Obtenir une instance de Geocoder

`new Geocoder (Context context)`