

# GPS und ortsbezogene Dienste



# Vorgehen um Standort zu ermitteln

LocationManager besorgen

Liste der verfügbaren Provider anzeigen, oder  
direkt nach Kriterien filtern

Einen oder mehrere Provider wählen, welche/r genutzt werden soll/en

Letzten Bekannten Standort abfragen, oder  
LocationListener registrieren



# Benötigte Rechte und Manifest

Um der App den Zugriff auf den aktuellen Standort zu ermöglichen müssen folgende Einträge in der AndroidManifest.xml vorhanden sein:

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

<uses-feature android:name="android.hardware.location.gps" />
<uses-feature android:name="android.hardware.location.network" />
```

Ortungsmethode:	Netzwerk (WIFI + mobil)	GPS
'uses-premission' :	ACCESS_COARSE_LOCATION oder: ACCESS_FINE_LOCATION	ACCESS_FINE_LOCATION
'uses-feature' :	android.hardware.location.network	android.hardware.location.gps



# LocationManager

Ermöglicht Zugriff auf die Standortanbieter (locationProvider)

Context.getSystemService(LocationManager.class) liefert diesen

Wichtige Methoden:

getAllProviders()

getBestProvider(Criteria, Bool)

requestLocationUpdates(LocationProvider, time, dist, LocationListener)

removeUpdates()



# LocationListener

Wird an `LocationManager.requestLocationUpdates()` übergeben

Beschreibt was im Falle einer Ortsänderung geschehen soll

Methoden die implementiert werden müssen:

`onLocationChanged(location)`

`onStatusChanged(provider, status, extras)`

`onProviderEnabled(provider)`

`onProviderDisabled(provider)`



# Google-Maps in App integrieren

Einfügen der Abhängigkeit in Project → app → build.gradle  
dependencies {

```
    ...  
    compile 'com.google.android.gms:play-services-maps:11.8.0'  
    ...  
}
```

API-Key auf [console.developers.google.com](https://console.developers.google.com) erstellen und im AndroidManifest.xml einfügen:

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

Oder als @string Verweis für alternative Ressourcen (seperater Key für debug Release)



# Mit der Karte interagieren

Selbstverständlich arbeitet die Google-Maps Integration auch mit den Längen und Breitengraden. Um auf der Karte zu navigieren verwendet man die lat. und long. um ein LatLng Objekt zu erzeugen. Dieses können wir für Kameraänderungen und Kartenmarkierungen verwenden.

