

Geolocation und Maps Integration

Mobile Computing

24.06.2020 - SS 2020

Fanni Marosi, 764345

Ebru Özcelik, 764349

Agenda

1. Motivation
2. Grundlagen
3. Mobile Provider und Geodaten
4. Android API vs. Google Play API & Provider-Vergleich
5. Maps SDK
6. Übung



Motivation

Grundlagen

Provider &
Geodaten

Android API vs.
Google API

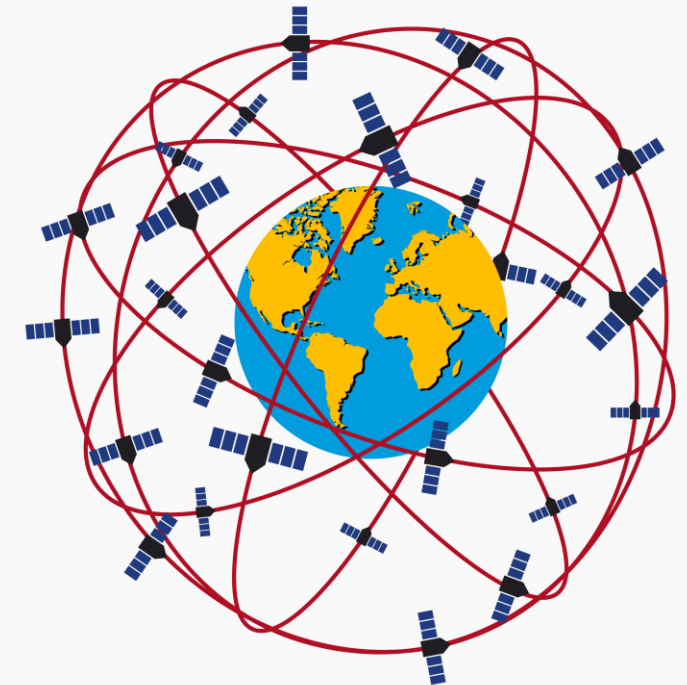
Maps SDK

Übung

- App für die Nachbarschaft
- Spätere Version:
 - ⇒ Wohnortbestimmung über Geolocation
 - ⇒ Suche mithilfe von Geolocation und Maps Integration

- **GPS (Global Positioning System):**

- ⇒ Globales Satellitennavigationssystem
- ⇒ Bestehend aus 24 Hauptsatelliten, 8 Notsatelliten
- ⇒ 3 Komponenten: Satelliten, Empfänger und Bodenstationen
- ⇒ Positionsbestimmung (mind. 4 Satelliten)
- ⇒ Position des Empfängers wird relativ zu dessen Entfernung zu den Satelliten ermittelt



Bildquelle: [1]

Motivation

Grundlagen

**Provider &
Geodaten**

Android API vs.
Google API

Maps SDK

Übung

Geodaten

Beispiel: Golden Gate Bridge (San Francisco)

⇒ Sexagesimale Schreibweise

37° 49' N

122° 29' W

⇒ Dezimale Schreibweise

37,816667

- 122,483333

Standortanbieter (Provider)

⇒ GPS

⇒ Netzwerk (Mobilfunkdaten, WLAN,
Funktürme)

⇒ Andere Apps

Location-Objekt

⇒ `getAltitude()` - Höhe

⇒ `getLatitude()` - Breitengrad

⇒ `getLongitude()` - Längengrad

Motivation

Grundlagen

Provider &
Geodaten

**Android API
vs. Google API**

Maps SDK

Übung

- **Android - Location Manager**

- ⇒ Android
- ⇒ Teil des Android-Frameworks
- ⇒ 14 Dezimalwerte
- ⇒ Logik für den Wechsel der Anbieter muss selbst geschrieben werden
- ⇒ Verfügbar in allen Android Geräten

- **Google - Fused Location Provider**

- ⇒ Google Play Services
- ⇒ Baut auf dem Android-Framework auf und ergänzt es
- ⇒ 7 Dezimalwerte
- ⇒ Die API kombiniert automatisch verschiedene Signale von unterschiedlichen Providern, um die Geodaten darzustellen
- ⇒ Google Play wird benötigt

Motivation

Grundlagen

Provider &
Geodaten

**Android API
vs. Google API**

Maps SDK

Übung

API	Google Location API Services	Android Location Manager
Anbieter - GPS	PRIORITY_HIGH_ACCURACY	GPS_PROVIDER
Erforderliche Berechtigungen	<ul style="list-style-type: none">• ACCESS_FINE_LOCATION für einen genaueren Standort• ACCESS_COARSE_LOCATION für einen weniger genauen Standort	ACCESS_FINE_LOCATION
Genauigkeit	10m - 100m	10m - 100m
Leistungsbedarf	Hoch	Hoch
Verfügbarkeit	Google Play-Dienste muss verfügbar sein	Weltweit (mit freier Sicht zum Himmel)
Standortaktualisierung	<ul style="list-style-type: none">• Default: einmal pro 5 Sekunde• Werte können geändert werden	Default: einmal pro Sekunde
Bemerkung	" Urban Canyon " - Effekt -> Pokemon Go	"Urban Canyon" - Effekt

Motivation

Grundlagen

Provider &
Geodaten

**Android API
vs. Google API**

Maps SDK

Übung

API	Google Location API Services	Android Location Manager
Anbieter – Netzwerk (WLAN, Mobilfunk)	<ul style="list-style-type: none">PRIORITY_BALANCED_POWER_ACCURACYPRIORITY_LOW_POWER	NETWORK_PROVIDER
Erforderliche Berechtigungen	<ul style="list-style-type: none">ACCESS_FINE_LOCATION für einen genaueren StandortACCESS_COARSE_LOCATION für einen weniger genauen Standort	<ul style="list-style-type: none">ACCESS_COARSE_LOCATIONACCESS_FINE_LOCATION
Genauigkeit	100m - 1000m +	100m - 1000m +
Leistungsbedarf	Niedrig - Mittel	Niedrig- Mittel
Verfügbarkeit	<ul style="list-style-type: none">Google Play-Dienste müssen verfügbar sein (nicht China)In Reichweite des Zellenturm- oder WLAN-Signals	In Reichweite des Zellenturm- oder WLAN-Signals
Standortaktualisierung	Gleicher Wert wie PRIORITY_HIGH_ACCURACY	seltener als GPS ausgeführt
Bemerkung	PRIORITY_BALANCED_POWER_ACCURACY: kann immer noch GPS verwenden PRIORITY_LOW_POWER: verwendet wahrscheinlich kein GPS	Signalgenauigkeit hängt von ; Anzahl der WLAN-Signale, Signalstärke, Typ des Zellturms an



Motivation

Grundlagen

Provider &
Geodaten

Android API vs.
Google API

Maps SDK

Übung

- Basiert auf Google-Maps
- Die API bietet Zugriff zu den Google-Maps Daten
- Karten zu einer App hinzufügen
- Man kann der Karte grafische Elemente, Symbole, Markierungen und Overlays hinzufügen

1. Dependency hinzufügen

Google Play Service öffnen ->

<https://developers.google.com/android/guides/setup>

-> Herunterscrollen zu Google Location and Activity Recognition

-> API zu build.gradle (:app) hinzufügen wie auf unterer Abbildung gezeigt

```
dependencies {  
    implementation fileTree(dir: 'libs', include: ['*.jar'])  
  
    implementation 'androidx.appcompat:appcompat:1.1.0'  
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'  
  
    // API hinzufügen  
    implementation 'com.google.android.gms:play-services-location:17.0.0'  
}
```

2. LocationRequest initialisieren in der Klasse MainActivity

```
public class MainActivity extends AppCompatActivity {  
  
    private static final int PERMISSION_FINE_LOCATION = 99;  
    TextView tv_lat, tv_lon, tv_altitude, tv_accuracy, tv_speed,  
    tv_sensor, tv_updates, tv_address;  
    Switch sw_locationupdates, sw_gps;  
  
    //LocationRequest deklarieren ☺  
    //LocationRequest locationRequest;  
  
    <...>  
  
    final int DEFAULT_UPDATE_INTERVAL = 30;  
    final int FAST_UPDATE_INTERVAL = 5;  
}
```

```
sw_gps.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if(sw_gps.isChecked()){  
  
            //Anbieter GPS ☺  
            //LocationRequest.PRIORITY_HIGH_ACCURACY  
  
            locationRequest.setPriority(LocationRequest.<...>);  
            tv_sensor.setText("Using GPS sensors");  
        }else{  
  
            //Anbieter Netzwerk ☺  
            //LocationRequest.PRIORITY_BALANCED_POWER_ACCURACY  
  
            locationRequest.setPriority(LocationRequest.<...>);  
            tv_sensor.setText("Using Cell Towers or Wifi");  
        }  
    }  
});
```

3. FusedLocationProviderClient initialisieren

```
private void updateGPS(){  
  
    // FusedLocationProviderClient initialisieren  
    // = LocationServices.getFusedLocationProviderClient(MainActivity.this);  
  
    fusedLocationProviderClient = <...>  
  
    if(ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED){  
        fusedLocationProviderClient.getLastLocation().addOnSuccessListener(this, new OnSuccessListener<Location>(){  
            @Override  
            public void onSuccess(Location location) {  
                updateUIValues(location);  
            }  
        });  
    }  
    else{  
        if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M){  
            requestPermissions(new String[] {Manifest.permission.ACCESS_FINE_LOCATION}, PERMISSION_FINE_LOCATION);  
        }  
    }  
}}
```

Motivation

Grundlagen

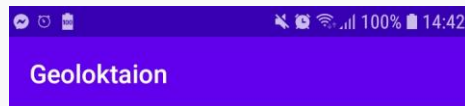
Provider &
Geodaten

Android API vs.
Google API

Maps SDK

Übung

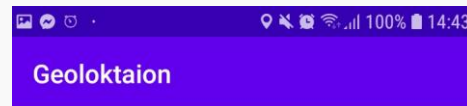
3. GUI der App



Latitude: 48.5170887
Longitude: 8.8261989
Altitude: 527.8999633789062
Accuracy: 19.601
Speed: Not avaiible
Address:
Bondorf, Deutschland

Location Updates ☒
Updates: Off
GPS/SavePower ☐
Sensor: Cell Tower + Wifi

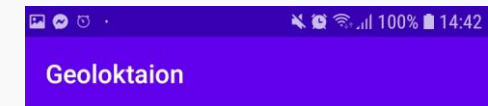
Geo-Adventures of Ebru and Fanni



Latitude: 48.5170843
Longitude: 8.8262188
Altitude: 527.8999633789062
Accuracy: 19.104
Speed: Not avaiible
Address:
Bondorf, Deutschland

Location Updates ☒
Updates: Location is being tracked!
GPS/SavePower ☒
Sensor: Using GPS sensors

Geo-Adventures of Ebru and Fanni



Latitude: Not tracking location
Longitude: Not tracking location
Altitude: Not tracking location
Accuracy: Not tracking location
Speed: Not tracking location
Address: Not tracking location

Location Updates ☐
Updates: Location is NOT being tracked
GPS/SavePower ☐
Sensor: Not tracking location

Geo-Adventures of Ebru and Fanni

Danke für eure Aufmerksamkeit!

Quellenverzeichnis

Capderou, M.: Handbook of Satellite Orbits. From Kepler to GPS. Springer, 2014. DOI 10.1007/978-3-319-03416-4

Doberstein, D.: Fundamentals of GPS Receivers. A Hardware Approach. Springer, 2012. DOI 10.1007/978-1-4614-0409-5

Schüttler, T.: Satellitennavigation. Wie sie funktioniert und wie sie unseren Alltag beeinflusst. Springer, 2014. DOI 10.1007/978-3-642-53887-2

Louis, D., Müller, L.: Android, Kapitel 17. Hanser, 2016. DOI 10.3139/978-3-446-45112-4

<https://developer.android.com/reference/kotlin/android/location/Location>

<https://developer.android.com/reference/kotlin/android/location/LocationManager>

<https://developers.google.com/location-context/fused-location-provider>

<https://developer.android.com/training/location/geofencing>

<https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient>

<https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest>

<https://riptutorial.com/de/android/topic/1837/ort>

<https://developers.google.com/maps/documentation/android-sdk/intro?hl=de>

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

<https://appus.software/blog/difference-between-locationmanager-and-google-location-api-services>

<https://www.youtube.com/watch?v=K8oNMCWZlmQ&t=150s>

Quellenverzeichnis

Android Studio GPS location tracker tutorial 01 - 05

<https://www.youtube.com/watch?v=V62sxpypxapU>

<https://www.youtube.com/watch?v=2ibBng2eJJA&t=336s>

<https://www.youtube.com/watch?v=CdZ3xURK-c>

<https://www.youtube.com/watch?v=uTxltxrDVSk>

https://www.youtube.com/watch?v=0Bqd_Mv2uI

Bildquellen

[1] https://www.google.com/search?q=konstellation+satelliten&client=firefox-b-d&sxsrf=ALeKk00ZUA073CYcaiWj2aHFM1lpucBQ4w:1592930681247&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiU6qCbsZjqAhWt5KYKHQHTCLsQ_AUoAnoECBAQBA&biw=1252&bih=574#imgsrc=k5_gs_B0JHOMKM