Location-based services in Android
Presentation Notes

Jan Bremauer, Matthias Rupp

05/2020

Contents

1 Motivation

> *Quarter of all Android apps ask for users' GPS location data.* [3]

This quote is based on the Rew Research Study from 2015, where they analyzed the app permissions of 1 million apps in the play store.
It shows that even back then location awareness was a important feature.

2 Introduction

A location based service are basicly just a service that provides information based on the geographical location of your android device.
It was initialy driven by emergency assistance applications to locate a mobile devies by measuring its distance to a basestations.
So to use the location sensors android provides various APIs to make it easy to implement location based services.

3 Android Location APIs

3.1 Location Manager

Included in the Android API since version 1

- Request location updates (periodically)
- Get last known location of this provider
- Use different providers
- Retrieve information about GPS chipset

Providers:

- `LocationManager.GPS_PROVIDER`
- `LocationManager.NETWORK_PROVIDER`
- `LocationManager.PASSIVE_PROVIDER`

3.2 Fused Location API

Part of Google Play Services

- Simple and battery-efficient
- Request location updates
- Get last known location (system-wide)
- Automaticly changes to the appropiate location source
- Can deliver updates to a callback at specific intervals
  ⇒ Provide additional information like direction
- Location is queried with a *LocationRequest*
  - Interval settings
  - Priorities
  - Displacements
  - ...
- Initialized via *GoogleApiClient.Builder*

Priorities:

- `PRIORITY_BALANCED_POWER_ACCURACY`
- `PRIORITY_HIGH_ACCURACY`
- `PRIORITY_LOW_POWER`
- `PRIORITY_NO_POWER`

3.3 Geofencing API

Part of Google Play Services

- Recognize when user enters/leaves a predefined circular region
- Simple but limited API
- Very power efficient
- Up to 100 geofences per app, per device

4 Power Usage

Location based services can cause a huge battery drain when done wrong.
The location is related to the battery drain in the following aspects:

- Higher accuracy → higher battery drain
- Higher frequency → higher battery drain
- Less latency → higher battery drain

The developer can reduce the battery drain of his application by...

- ...using geofencing whenever possible
- ...using getLastLocation() instead of requesting a new location
- ...tweaking the frequency and accuracy

Best practices:

- Remove location updates if no longer needed
- Set timeouts for location updates
- Batch requests together
- Use passive location updates

5 Access Guidelines

Permissions:

```
ACCESS_FINE_LOCATION
ACCESS_COARSE_LOCATION
ACCESS_BACKGROUND_LOCATION
```

The permissions should be requested by using *requestPermission()*

6 Example Implementation

To show an exmaple implementation for two of the location apis we thought out a simple geofencing app.
We basicly track detailed location data of the user once he enters the city center (the geofence).
With this location data we can later create a heatmap of the shopping behavior of the different users.

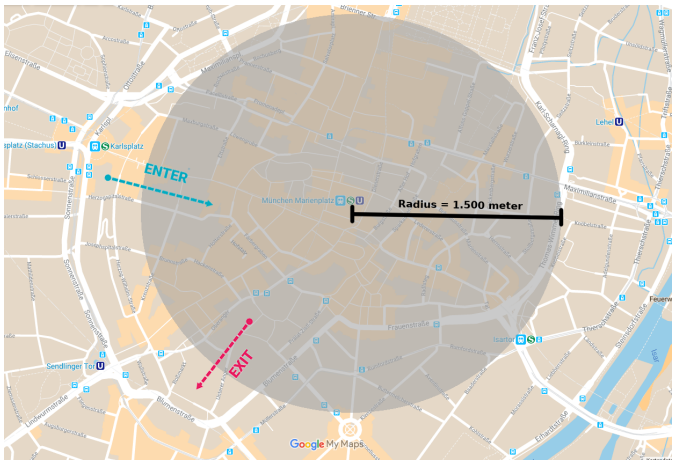
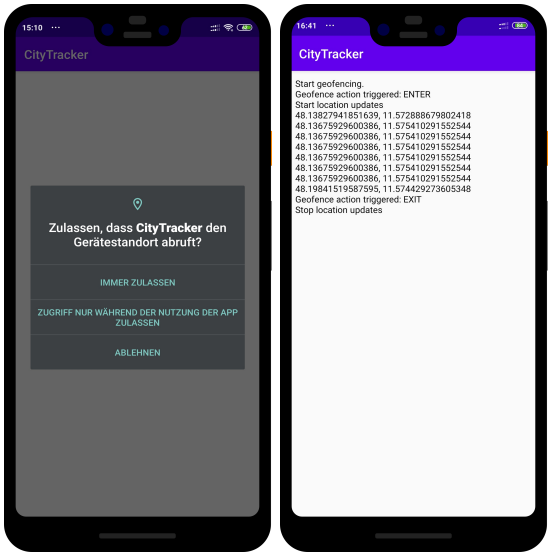


Figure 1: Our Geofence



(a) (b)

Figure 2: https://github.com/mobileappdevhm20/CityTracker

7 Summary

- Important feature for android development
- Android provides many different possibilities to implement location awareness
- Keep the battery usage in mind and use the best practices
- Pay attention to the access guidelines

8 References

[1] https://developer.android.com/training/location
[2] https://developers.google.com/location-context/fused-location-provider
[3] https://thehill.com/policy/technology/259655-quarter-of-all-android-apps-ask-for-gps-data
[4] https://developers.google.com/maps/documentation/android-sdk/location
[5] https://developer.android.com/training/location/geofencing