

# Android - mapy



Peter Borovanský  
KAI, I-18

MS-Teams: [2sf3ph4](#), [List](#), [github](#)

borovan 'at' ii.fmph.uniba.sk

## Google Maps

- **API key, podpisovanie appky**
- **Google Console**
- **MarcelP – OSM, Mapbox**

# Google Maps

(api key V1 & V2)



Na prácu s balíkom `com.google.android.maps.*` potrebujeme API kľúč pre Google Maps vygenerovaný pomocou **SHA1** kľúča – čo je váš *finger-print*.  
(presnejšie finger-print vašej inštalácie na vašom počítači)



Google Maps verzia **V2** (pár rokov už jediná alternatíva Gmaps Verzia 2):

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

Google Maps API-key sa generuje pomocou **SHA1** kľúča, ktorý je vygenerovaný v súbore `.keystore` pri inštalácii Android Studio (po reinštalácii počítača idete s novým finger-printom!)

## Dôsledok1 ☹

Ak si rebuildujete niektorý zo zverejených kódov (mojich, či cudzích), nepôjde vám skôr:

- ako si aplikáciu nezaregistrujete v **Google Console** (vaším SHA1-kľúčom a package)
- vygenerovaným **API-key** z **Google Console** nepodpíšete váš build.

## Dôsledok2 ☺

ak nerebuildujete project s vaším SHA1, tak `.apk` je ok, a pôjde vám nainštalovať aj spustiť

# Získanie SHA1

androiddebugkey

```
C:\Users\borovan>dir c:\users\borovan\.android\*.keystore
Volume in drive C is System
Volume Serial Number is 0006-3F8C

Directory of c:\users\borovan\.android

12/31/2018  01:47 PM                1,259 debug.keystore
               1 File(s)                1,259 bytes
               0 Dir(s) 73,137,201,152 bytes free
```

Potrebuje získať náš **SHA1** kľúč, v cmd spustíme keytool (v JRE java\bin):  
je 40-miestny hexa, t.j. 160bitov

```
keytool -v -list -alias androiddebugkey
        -keystore "c:\Users\<user>\.android\debug.keystore"
        -storepass android -keypass android
```

Ten sa získa (pre účely ladenia) z (debug) .keystore file

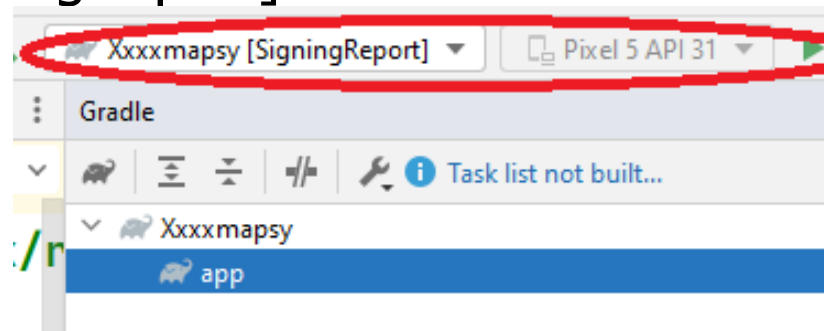
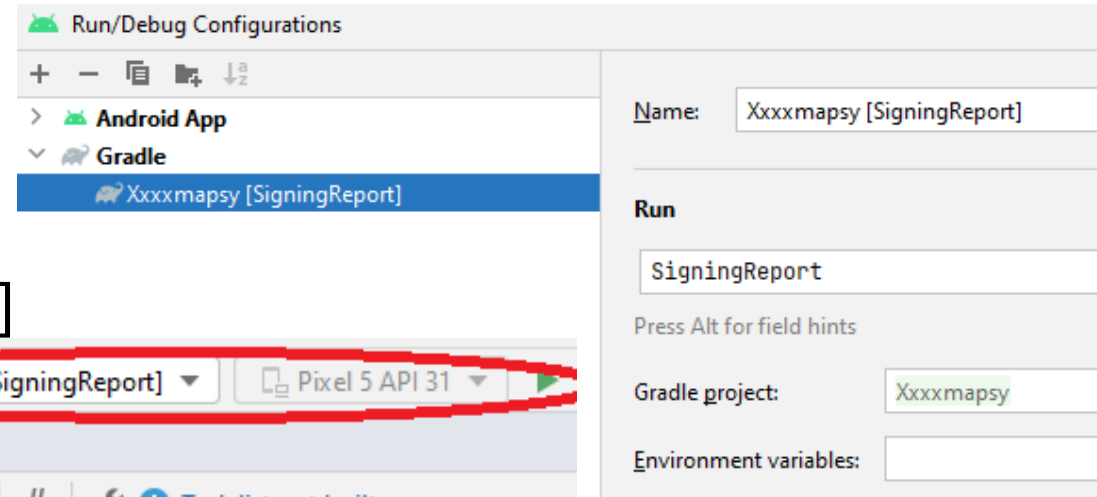
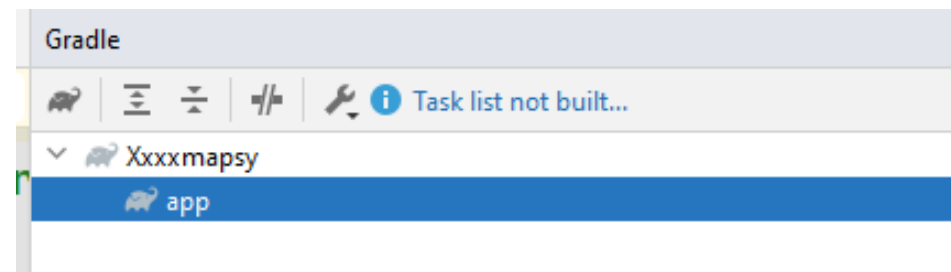
- Windows: `c:\Users\<user>\.android\debug.keystore`
- Linux: `/home/user/.android/`

```
C:\Users\borovan>keytool -v -list -alias androiddebugkey -keystore "c:\Users\borovan\.android\debug.keystore" -storepass andr
oid -keypass android
Alias name: androiddebugkey
Creation date: Dec 31, 2018
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: C=US, O=Android, CN=Android Debug
Issuer: C=US, O=Android, CN=Android Debug
Serial number: 1
Valid from: Mon Dec 31 13:47:15 CET 2018 until: Wed Dec 23 13:47:15 CET 2048
Certificate fingerprints:
    MD5: 78:CB:6A:11:66:A3:7E:2E:BE:80:DA:63:5E:53:6D:E7
    SHA1: 7A:94:75:11:DD:3D:57:                :A0:B8:68:0E:67:F1
    SHA256: F1:FB:D3:08:6C:4C:AB:01:5D:8E:B6:FD:A4:7D:07:CE:41:90:98:86:85:BD:30:2D:40:25:91:11:C9:46:50:AB
Signature algorithm name: SHA1withRSA
```

# Získanie SHA1

(v Android Studio)

- Edit Configuration/+ /Gradle
  - Gradle Project: default
  - Run: SigningReport
  - Apply/Ok
- Run Project[SigningReport]

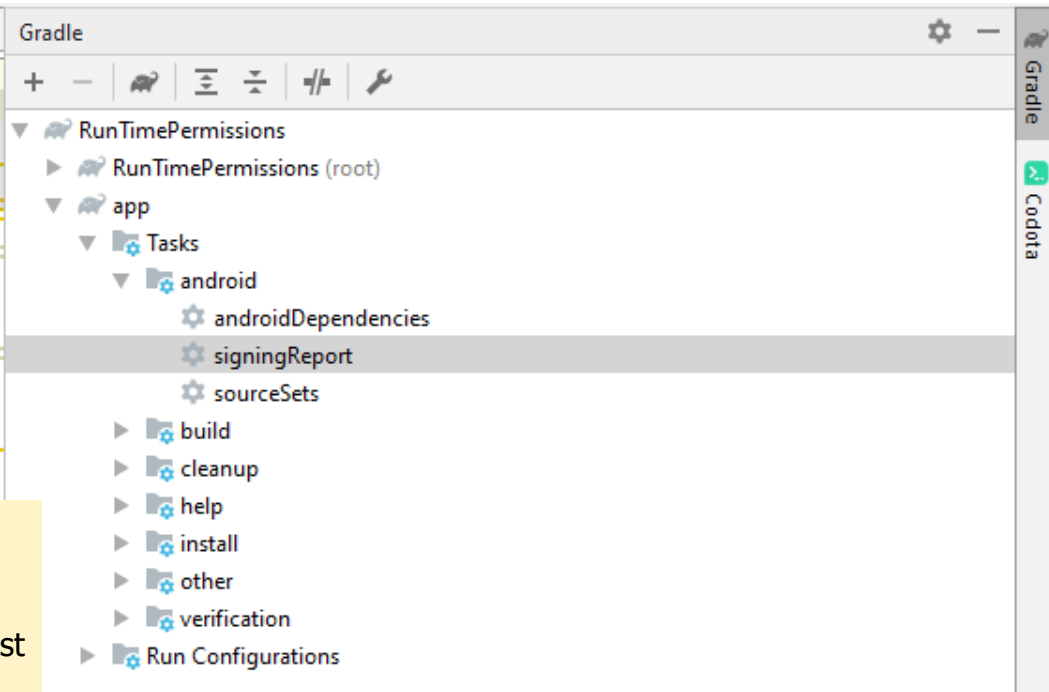


- už mám SHA-1 a čo s ním ? ...bude ho treba v Google Cloud Console



# Získanie SHA1 (staršia alternatíva)

1. Open Android Studio
2. Open your Project
3. Click on Gradle (From Right Side Panel, you will see Gradle Bar)
4. Click on Refresh (Click on Refresh from Gradle Bar, you will see List Gradle scripts of your Project)
5. Click on Your Project (Your Project Name form List (root))
6. Click on Tasks
7. Click on Android
8. Double Click on signingReport (You will get SHA1 and MD5 in Run Bar)
9. Then click this button:



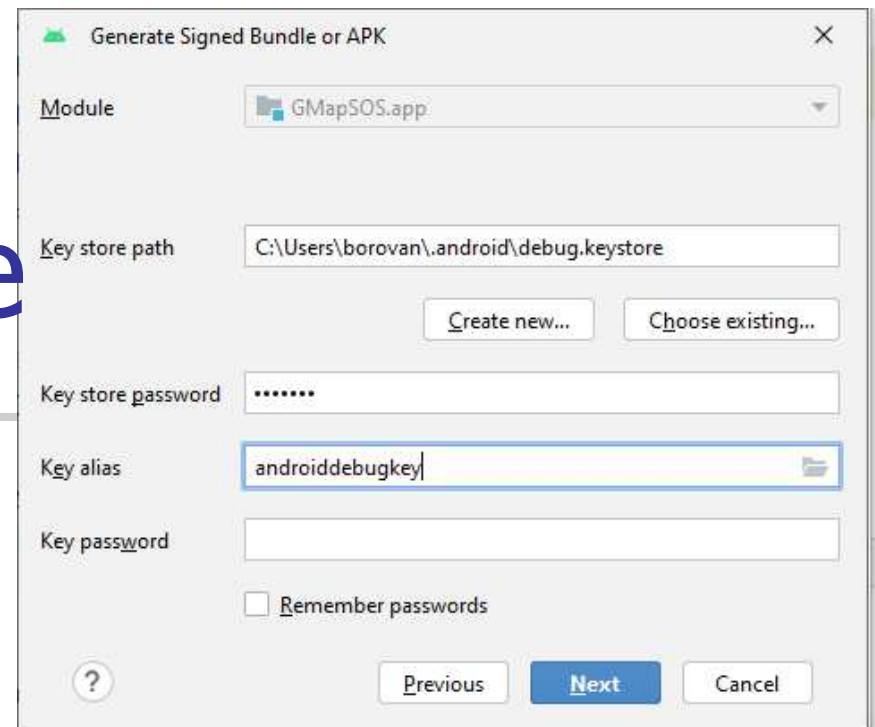
```
Run: RunTimePermissions:app [signingReport] x
-----
Variant: debug
Config: debug
Store: C:\Users\borovan\.android\debug.keystore
Alias: AndroidDebugKey
MD5: 78:CB:6A:11:66:A3:7E:2E:BE:80:DA:63:5E:53:6D:E7
SHA1: 7A:94:75:11:DD:3D:57: A0:B8:68:0F:67:F1
SHA-256: F1:FB:D3:08:6C:4C:AB:01:5D:8E:B6:FD:A4:7D:07:CE:41:90:98:86:85:BD:30:2D:40:25:91:11:C9:46:50:AB
Valid until: Wednesday, December 23, 2048
```

# Debug keystore

(manažment certifikátov)

```
$ keytool -genkey -v  
-keystore debug.keystore  
-storepass android -alias androiddebugkey -keypass android  
-keyalg RSA -keysize 2048 -validity 10000
```

Keystore name: "debug.keystore"  
Keystore password: "android"  
Key alias: "androiddebugkey"  
Key password: "android"



Generate Signed Bundle or APK

Module: GMapSOS.app

Key store path: C:\Users\borovan\.android\debug.keystore

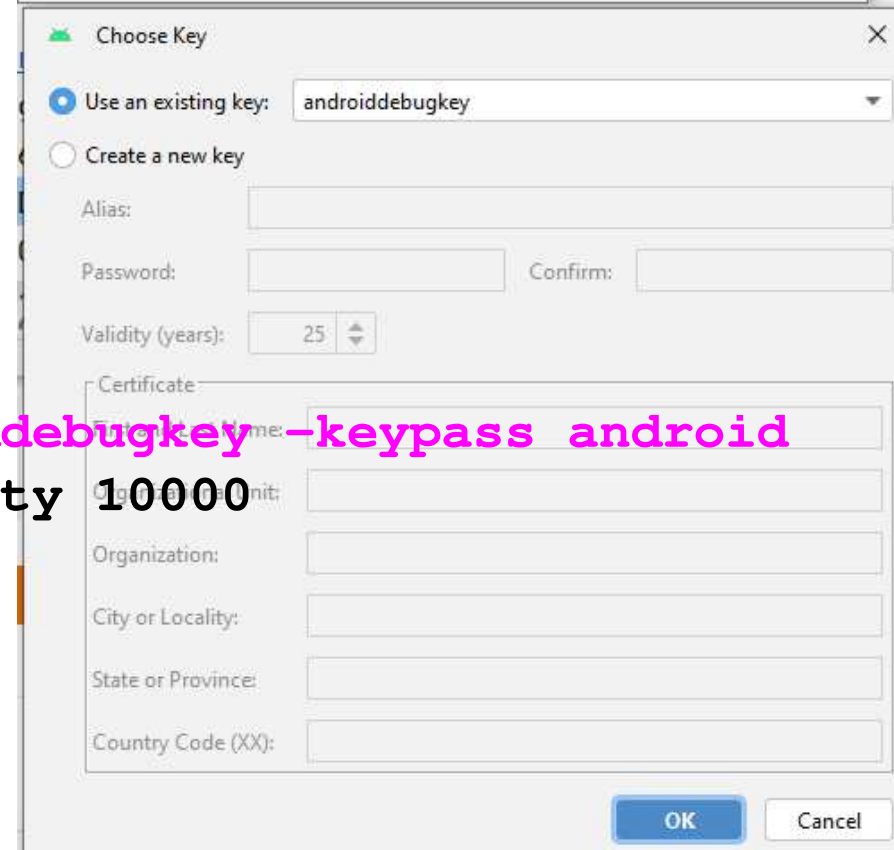
Key store password: .....

Key alias: androiddebugkey

Key password:

☐ Remember passwords

Previous Next Cancel



Choose Key

☒ Use an existing key: androiddebugkey

☐ Create a new key

Alias:

Password: Confirm:

Validity (years): 25

Certificate:

First Name:

Last Name:

Organization:

City or Locality:

State or Province:

Country Code (XX):

OK Cancel

# Debug keystore

(manažment certifikátov)

- **debug.keystore** obsahuje jeden alebo viac privátnych kľúčov (certifikátov).
- **debug.keystore** nám automaticky vyrobí Android-Studio pri inštalácii AS
- Android Studio nám automaticky podpíše každú apku kľúčom debug.keystore
- vieme vygenerovať vlastný **keystore**/certifikát, pomocou `keytool`, resp. v AS
- v AS Build/Generate Signed APK <https://developer.android.com/studio/publish/app-signing.html>

```
Prikazový riadok
C:\Users\borovan\.android>keytool -genkey -v -keystore mykey.keystore -alias myaliasname2018 -keyalg RSA -keysize 2048 -validity 10000
Enter keystore password:
What is your first and last name?
[Unknown]: Peter Borovansky
What is the name of your organizational unit?
[Unknown]: FMFI
What is the name of your organization?
[Unknown]: UK
What is the name of your City or Locality?
[Unknown]: BA
What is the name of your State or Province?
[Unknown]: SK
What is the two-letter country code for this unit?
[Unknown]: SK
Is CN=Peter Borovansky, OU=FMFI, O=UK, L=BA, ST=SK, C=SK correct?
[no]: yes

Generating 2 048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10 000 days
for: CN=Peter Borovansky, OU=FMFI, O=UK, L=BA, ST=SK, C=SK
Enter key password for <myaliasname2018>
(RETURN if same as keystore password):
```

New Key Store

Key store path: C:\Users\borovan\mykeys.jks

Password: ..... Confirm: .....

Key

Alias: myNewKeys

Password: ..... Confirm: .....

Validity (years): 25

Certificate

First and Last Name: Peter Borovansky

Organizational Unit: KAI

Organization: FMFI

City or Locality: Bratislava

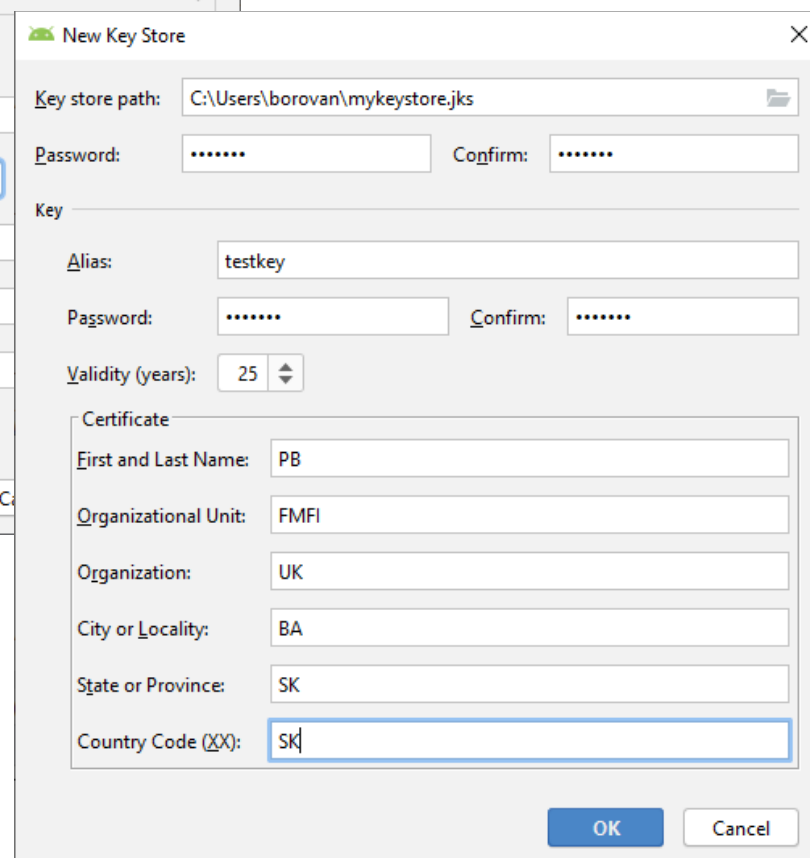
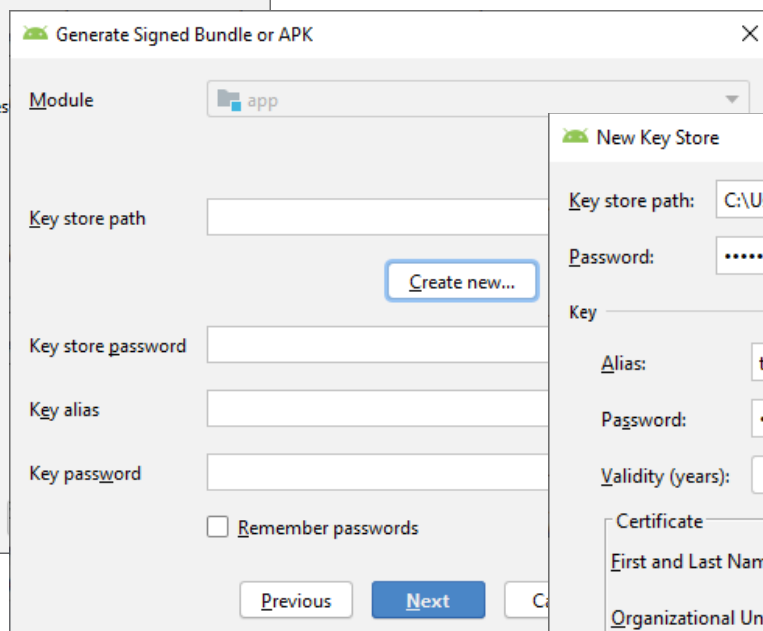
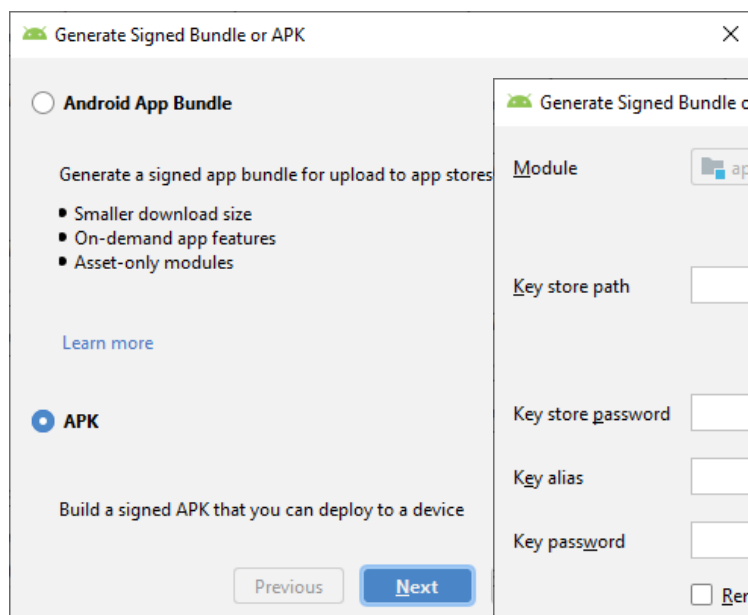
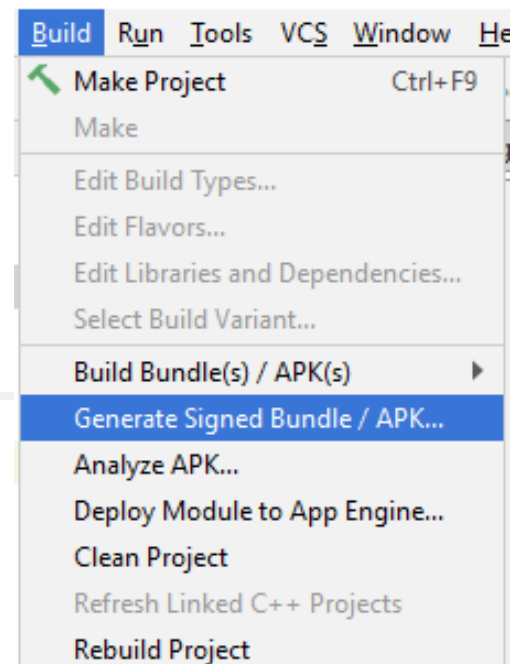
State or Province: Slovakia

Country Code (XX): SK

OK Cancel

# Generovanie kľúča

- začnite v Build/Generate Signed APK

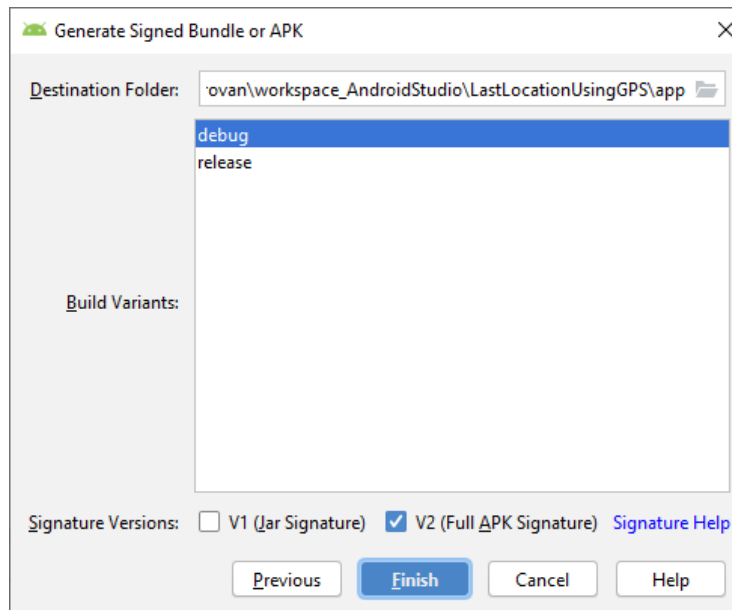
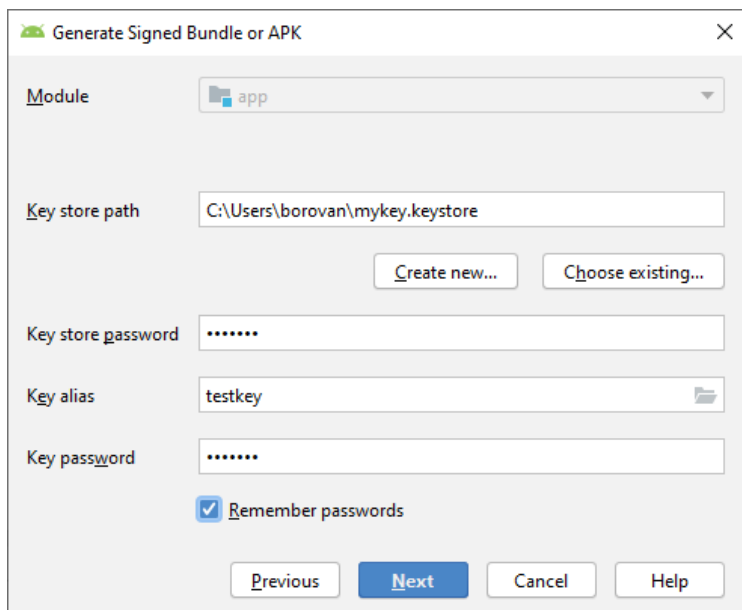




# Podpisovanie aplikácie (debug)

(použitie certifikátov)

Vygenerovaný kľúč potom slúži na podpísanie .apk  
To robí AS ale aj napr. jarsigner



AS automaticky podpisuje .apk pri builde kľúčom **androiddebugkey**

# Podpisovanie aplikácie (release)

(použitie certifikátov)

- Potrebujete release.keystore (návody):

<https://developer.android.com/studio/publish/app-signing>

<https://medium.com/mindorks/upload-your-first-android-app-on-play-store-step-by-step-ee0de9123ac>

- Potrebujete Google Play Account, Google Play Console:

<https://play.google.com/apps/publish/signup/>

Vytvorenie osobného účtu  
vývojára Play

 borovansky@gmail.com ▼

- ✓ Typ účtu
- ✓ O vás
- ✓ Účet vývojára
- ✓ Aplikácie
- Zmluvné podmienky

 Google Play Console



## Zmluvné podmienky

- ☒ Potvrďujem, že som si prečítal(a) [distribučnú zmluvu pre vývojárov v službe Google Play](#) a že s ňou súhlasím. Potvrďujem, že mám minimálne 18 rokov.
- ☒ Potvrďujem, že som si prečítal(a) [zmluvné podmienky služby Google Play Console](#) a že s nimi súhlasím.

Pokračovaním vyhlasujete a zaručujete, že: (i) máte plné oprávnenie zaviazat' organizáciu alebo osobu uvedenú vyššie distribučnou zmluvou pre vývojárov v službe Google Play a zmluvnými podmienkami služby Play Console; (ii) ste si prečítali tieto zmluvy a rozumiete im; a (iii) súhlasíte s týmito zmluvami v mene danej firmy alebo osoby



Ak si chcete vytvoriť účet, musíte zaplatiť jednorazový registračný poplatok 25 USD. Pred dokončením registrácie účtu vás môžeme požiadať o overenie identity platným dokladom totožnosti. Ak sa nám vašu identitu nepodariť overiť, registračný poplatok vám nevrátíme.

# Google (Play/Cloud) Console

- Nie je konzola ako konzola !!!

Google Cloud

Why Google Solutions Products Pricing Getting Started [Contact Us](#)



Docs Support

English

Console



<https://console.cloud.google.com>

<https://play.google.com/console/u/0/signup>



borovansky@gmail.com

Povedzte nám o sebe

Meno vývojára

Verejný názov vášho účtu vý  
Play.

Google Cloud Platform My Project

DASHBOARD

ACTIVITY

RECOMMENDATIONS

## Project info

Project name

My Project

Project number

73305277898

Project ID

handy-theory-148710

[ADD PEOPLE TO THIS PROJECT](#)

[Go to project settings](#)

# Google Maps API



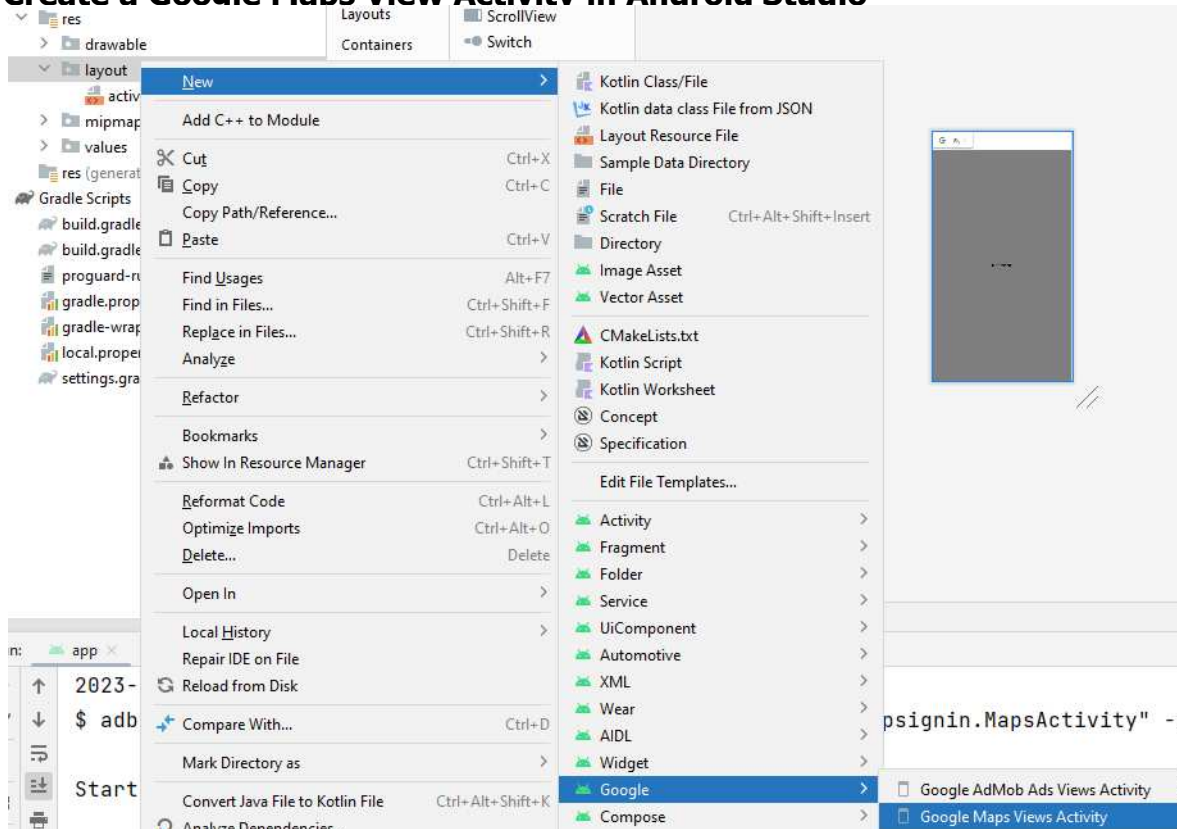
Treba **dôsledne** (!!!) prejsť oficiálnym návodom:

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

## Set up the development environment

1. Android Studio Arctic Fox or later is required. If you haven't already done so, [download](#) and [install](#) it.
2. Ensure that you are using the [Android Gradle plugin](#) version 7.0 or later in Android Studio.

## Create a Google Maps View Activity in Android Studio





# API\_KEY z Cloud Console

The AndroidManifest.xml file contains instructions on getting a Google Maps API key and then adding it to your local.properties file.

**Do not add your API key to the AndroidManifest.xml file.**

*Doing so stores your API key less securely.*

Instead, follow the instructions to create a **Cloud project** and configure an API key.

## AndroidManifest.xml

```
<!--  
  TODO: Before you run your application, you need a Google Maps API key.  
  
  To get one, follow the directions here:  
  
    https://developers.google.com/maps/documentation/android-sdk/get-api-key  
  
  Once you have your API key (it starts with "Alza"), define a new property in your  
  project's local.properties file (e.g. MAPS_API_KEY=Aiza...), and replace the  
  "YOUR_API_KEY" string in this file with "${MAPS_API_KEY}".  
-->  
<meta-data  
  android:name="com.google.android.geo.API_KEY"  
  android:value="${MAPS_API_KEY}" />
```

## local.properties

```
sdk.dir=C:\\Users\\borovan\\AppData\\Local\\Android\\Sdk  
MAPS_API_KEY=AlzaSyB31D.....XSqLKtK4SAtO
```


# Cloud Console – Step 1

- Step 1  
Set up your project
- Step 2  
Enable APIs or SDKs
- Step 3  
Get an API Key

Console Cloud SDK

1. In the Google Cloud Console, on the project selector page, click **Create Project** to begin creating a new Cloud project.

[Go to the project selector page |...](#)

2. Make sure that billing is enabled for your Cloud project. [Confirm that billing is enabled for your project](#) .

Google Cloud offers a \$0.00 charge trial. The trial expires at either end of 90 days or after the account has accrued \$300 worth of charges, whichever comes first. Cancel anytime. Google Maps Platform features a recurring \$200 monthly credit. For more information, see [Billing account credits](#) and [Billing](#).





# G-Fa



## Invoice

Invoice number: 4603377165

Google Cloud EMEA Limited

Velasco

Clanwilliam Place

Dublin 2

Ireland

VAT number: IE36689970H

### Bill to

Peter Borovansky

B

B

81105 Bratislava

Slovakia

### Details

Invoice number .....4603377165

Invoice date .....Oct 31, 2022

Billing ID .....0195-6

Account ID .....01C0FC

### Google Cloud

Total in EUR **€0.00**

### Summary for Oct 1, 2022 - Oct 31, 2022

Subtotal in EUR €0.00

VAT (20%) €0.00

Total in EUR €0.00





# Poučenie z krízového vývoja

---

Nezadávať číslo vašej karty skôr ako si neprečítate podmienky, resp. majte jednu „internetovú“ platobnú kartu s obnosom, ktorého stratu viete poľahky prežiť

...stalo sa ...





# Cloud Console – Step 2

## project-level build.gradle

```
plugins {  
    id 'com.google.android.libraries.mapsplatform.secrets-gradle-plugin' version '2.0.1'  
    apply false  
}
```

*// Top-level build file where you can add configuration options common to all sub-projects/modules.*

```
plugins {  
    id 'com.android.application' version '7.3.1' apply false  
    id 'com.android.library' version '7.3.1' apply false  
    id 'org.jetbrains.kotlin.android' version '1.7.20' apply false  
    id 'com.google.android.libraries.mapsplatform.secrets-gradle-plugin' version '2.0.1' apply false  
}
```

## module-level build.gradle

```
id 'com.google.android.libraries.mapsplatform.secrets-gradle-plugin'
```

```
plugins {  
    id 'com.android.application'  
    id 'org.jetbrains.kotlin.android'  
    id 'com.google.android.libraries.mapsplatform.secrets-gradle-plugin'  
}
```

## local.properties

```
MAPS_API_KEY=AlzaSyB31D.....XSqLKtK4SATo
```

## AndroidManifest.xml

```
<meta-data  
    android:name="com.google.android.geo.API_KEY"  
    android:value="${MAPS_API_KEY}" />
```



# Nepodpísana appka

Authorization failure. Please see <https://developers.google.com/maps/documentation/android-sdk/start> for how to correctly set up the map.

2023-11-20 17:40:42.499 3496-3615 Google Android Maps SDK com.example.xxxxx123233

E In the Google Developer Console (<https://console.developers.google.com>)

Ensure that the "Maps SDK for Android" is enabled.

Ensure that the following Android Key exists:

API Key: YOUR\_API\_KEY

Android Application (<cert\_fingerprint>;<package\_name>):

7A:94:75:11:DD:3D:57:2A:.....;com.example.xxxxx123233

Authorization failure. Please see <https://developers.google.com/maps/documentation/android-sdk/start> for how to correctly set up the map.

2023-11-20 17:45:36.578 3944-4052 Google Android Maps SDK com.example.xxxxx123233

E In the Google Developer Console (<https://console.developers.google.com>)

Ensure that the "Maps SDK for Android" is enabled.

Ensure that the following Android Key exists:

API Key: AIzaSyAJvZGfTIRxIs4oyDDb1xwJHN\_m5VL2G88

Android Application (<cert\_fingerprint>;<package\_name>):

7A:94:75:11:DD:3D:57:2A:36:ED:2A:.....;com.example.xxxxx123233

# Google Developer Console


(<https://console.developers.google.com/>)

Welcome to the API Library

The API Library has documentation, links, and a smart search experience.


Search for APIs & Services

Maps VIEW ALL (15)




**Maps SDK for Android**  
Google

Maps for your native Android app.




**Maps SDK for iOS**  
Google

Maps for your native iOS app.



**Maps JavaScript API**  
Google

Maps for your website



**Places API**  
Google

Get detailed information about 100 million places

Android > Maps Android API

Add Google Maps to your Android app.

GET A KEY

VIEW PRICING AND PLANS





# Cloud Console – Step 3

---

1. Go to the Google Maps Platform/Credentials page, click Create credentials/API key.  
The API key created dialog displays your newly created API key.
2. Click Close.
3. The new API key is listed on the Credentials page under API keys.

## API key created


Use this key in your application by passing it with the `key=API_KEY` parameter.

Your API key

AIza:

0M



 This key is unrestricted. To prevent unauthorized use, we recommend restricting where and for which APIs it can be used. [Edit API key](#) to add restrictions. [Learn more](#)

[CLOSE](#)

Name \*  
API key 18

API Key `AIzaSyAQHz1aQu_1VG2N1`

Use this key in your application

Creation date 

### Key restrictions



This key is unrestricted. To prevent unauthorized use, we recommend restricting where and for which APIs it can be used. [Learn more](#)

### Application restrictions

An application restriction controls which websites, IP addresses, or applications can use your API key. You can set one application restriction per key.

- ☐ None
- ☐ HTTP referrers (web sites)
- ☐ IP addresses (web servers, cron jobs, etc.)
- ☒ Android apps
- ☐ iOS apps

### Restrict usage to your Android apps

Add your package name and SHA-1 signing-certificate fingerprint to restrict usage to your Android apps:

ADD AN ITEM

## API restrictions

API restrictions specify the enabled APIs that this key can call

- ☒ Don't restrict key  
This key can call any API
- ☐ Restrict key

How do I restrict my A

You can restrict an API key to certificate fingerprint or a rele

### Debug certificate fingerprint

For Linux or macOS:

```
$ keytool -list -v -ke
```

For Windows:

```
$ keytool -list -v -ke
```

## Release certificate fingerprint

```
$ keytool -list -v -ke
```

Replace `your_keystore_name` including the `.keystore` extension assigned to the certificate with



# Restrict API Key

---

## Application restrictions

An application restriction controls which websites, IP addresses, or applications can use your API key. You can set one application restriction per key.

- ☐ None
- ☐ HTTP referrers (web sites)
- ☐ IP addresses (web servers, cron jobs, etc.)
- ☒ Android apps
- ☐ iOS apps

## Restrict usage to your Android apps

Add your package name and SHA-1 signing-certificate fingerprint to restrict usage to your Android apps

### Edit item



Package name \*

com.example.gmapapp

SHA-1 certificate fingerprint \*

7A:94:75:11:D[

:F1

# API Key

## API key

This API key can be used in this project and with any API that supports it. To use this key in your application, pass it with the `key=API_KEY` parameter.

Creation date Nov 6, 2016, 12:26:20 PM  
Created by borovansky@gmail.com (you)

## API key

AIzaSyB6e5 [redacted] 30EeC4QD1r8

## Name

API key VMA 2016

## Key restriction

Key restriction lets you specify which web sites, IP addresses, or apps can use this key. [Learn more](#)

- ☐ None
- ☐ HTTP referrers (web sites)
- ☐ IP addresses (web servers, cron jobs, etc.)
- ☒ Android apps
- ☐ iOS apps

## Restrict usage to your Android apps (Optional)

Add your package name and SHA-1 signing-certificate fingerprint to restrict usage to your Android apps  
Get the package name from your AndroidManifest.xml file. Then use the following command to get the fingerprint:

```
$ keytool -list -v -keystore mystore.keystore
```

## Package name

pokus.example.com.myapplicationx

## SHA-1 certificate fingerprint

05:0F:34:5E:FD:E [redacted] 6:8E:57:A2:80:1C:D1:91

pokus.example.com.gmapv2

05:0F:34:5E:FD:E [redacted] 6:8E:57:A2:80:1C:D1:91

com.example.mapdemo

05:0F:34:5E:FD:E [redacted] 6:8E:57:A2:80:1C:D1:91



Your free trial is waiting: activate now to get \$300 credit to explore Google Cloud



Google Cloud Platform



GMaps2018

<https://console.developers.google.com/apis/credentials/>



## Project info

Project name  
GMaps2018  
Project ID  
gmaps2018-223015  
Project number  
294916644121



Go to project settings

API key pre Google Maps sa pridela pre dvojicu, alebo niekoľko dvojíc, (package name, SHA1-v prostredí, kde kompilujete)

# Najčastejšie chyby s GMapsAPI

(jemne serioznejší pohľad)

<http://ddewaele.github.io/GoogleMapsV2WithActionBarSherlock/part6>

## Beware of Map API key caching

ak package name, zmenu aplikujte aj na Google Console  
ak zmeníte API key/package name, radšej odinštalujte starú verziu m-appky,  
nainštalujte opäť novú, API kľúč môže byť nacachovaný...

## Ak u seba prekompilujete (napr. moju) m-appku, tak nepôjde...

musíte jej vygenerovať API key zodpovedajúci vášmu SHA1

## Failing to provide the correct map permissions, API Key

najčastejšie sa vám nezobrazí mapa, okopírujte potrebné permissions napr. zo súboru, resp. iného, čo funguje

<https://github.com/ddewaele/GoogleMapsV2WithActionBarSherlock/blob/master/GoogleMapsV2WithActionBarSherlock/AndroidManifest.xml>

E/Google Maps Android API: Authorization failure. Please see  
<https://developers.google.com/maps/documentation/android-api/start> for how to correctly set up the map.

E/Google Maps Android API: In the Google Developer Console (<https://console.developers.google.com>)

Ensure that the "Google Maps Android API v2" is enabled.

Ensure that the following Android Key exists:





# Najčastejšie chyby s GMapsAPI

(jemne serioznejší pohľad)

<http://ddewaele.github.io/GoogleMapsV2WithActionBarSherlock/part6>

## API key problem

váš package name-SHA1-GMAPS\_API\_Key musia súvisieť cez Google Console:

- Binary XML file line #2: Error inflating class fragment
- Caused by: java.lang.RuntimeException: API key not found. Check that
- Google Maps Android API(4040): Failed to contact Google servers.

## Zlé SDK

java.lang.RuntimeException: Unable to start activity

ComponentInfo{com.ecs.google.maps.v2.actionbarsherlock/com.ecs.google.maps.v2.simple.SimpleMapActivity}:

android.view.InflateException: Binary XML file line #2: Error inflating class fragment at  
android.app.ActivityThread.performLaunchActivity(ActivityThread.java:1651)

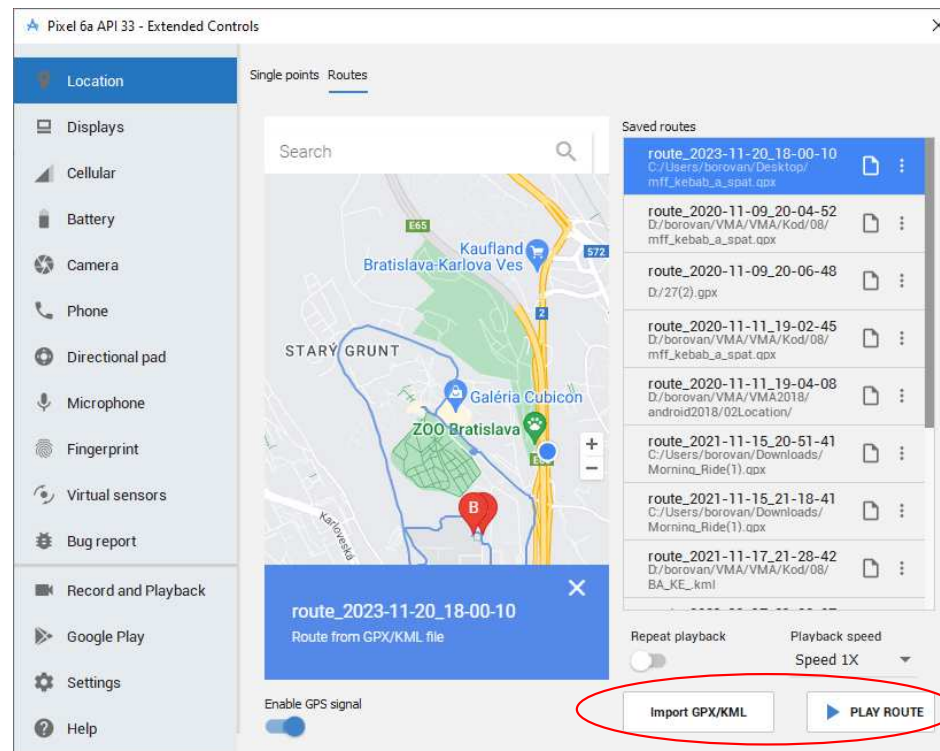
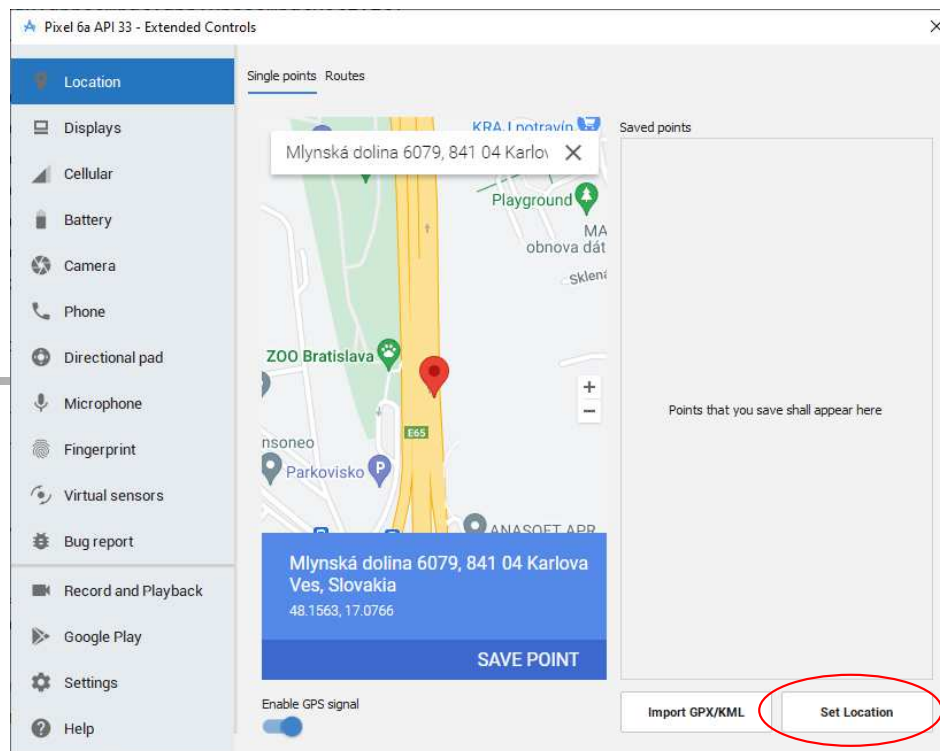
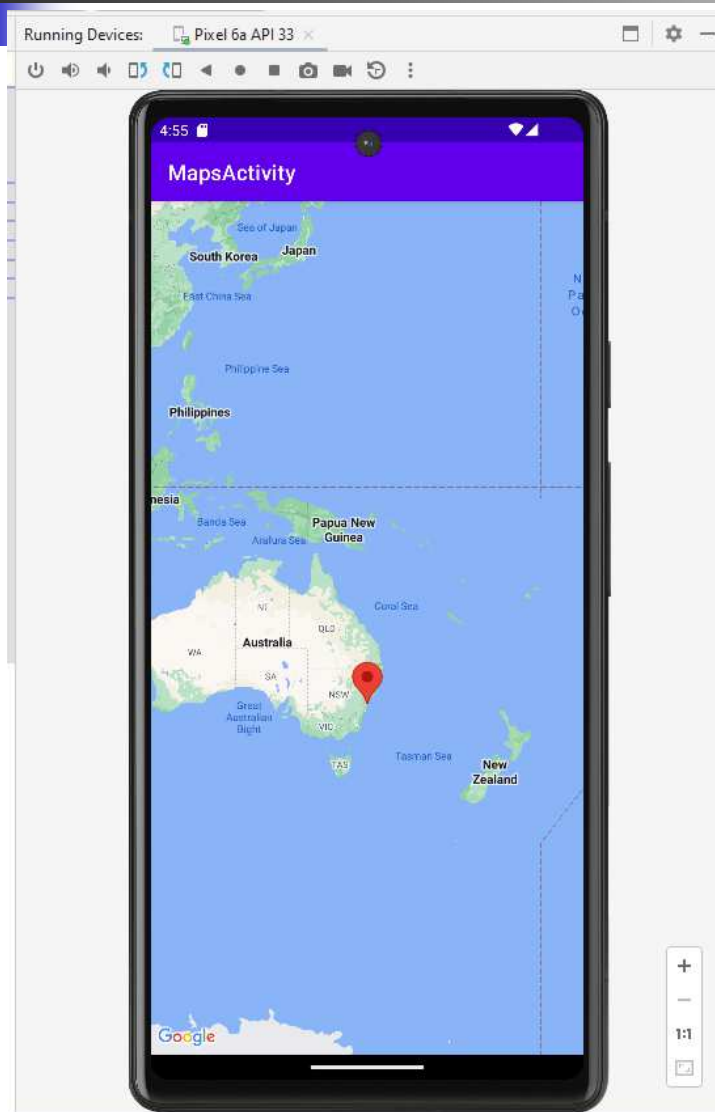
## A mnohé iné problémy

<http://ddewaele.github.io/GoogleMapsV2WithActionBarSherlock/part6>



# Gmap

## emulator 2023



# V úspešnom prípade

(v Android Studiu)



Ak sa všetko podarí, dostanete Layout s SupportMapFragmentom, a zobrazí sa mapa

```
class MapsActivity : AppCompatActivity(), OnMapReadyCallback {  
    private lateinit var mMap: GoogleMap  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_maps) -- obsahuje Map fragment  
        val mapFragment = supportFragmentManager  
            .findFragmentById(R.id.map) as SupportMapFragment  
        mapFragment.getMapAsync(this) -- otvorenie Gmaps chvíľku trvá  
    } -- keď sa naložia, zavolá sa callback onMapReady v tejto triede  
  
    override fun onMapReady(googleMap: GoogleMap) { -- tu začína život m-appky  
        mMap = googleMap -- referenciu na Gmapu si odložíme  
        // Add a marker in Sydney and move the camera  
        val sydney = LatLng(-34.0, 151.0) -- Sydney  
        mMap.addMarker(MarkerOptions() -- balónik  
            .position(sydney)  
            .title("Marker in Sydney"))  
        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney))  
    }  
}
```

# Layout s MapFragmentom

<fragment

```
xmlns:android=http://schemas.android.com/apk/res/android
xmlns:map=http://schemas.android.com/apk/res-auto
xmlns:tools=http://schemas.android.com/tools
android:id="@+id/map"
android:name="com.google.android.gms.maps.SupportMapFragment"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="pokus.example.com...MapsActivity" />
```

typy mapových podkladov:

```
mMap.setMapType(
    GoogleMap.MAP_TYPE_SATELLITE)
GoogleMap.MAP_TYPE_HYBRID)
GoogleMap.MAP_TYPE_TERRAIN)
GoogleMap.MAP_TYPE_NORMAL)
-- pre offline mapy (zložitejšie)
GoogleMap.MAP_TYPE_NONE)
```





# Permissions

---

- **AndroidManifest.xml**

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

- Žiadanie povolenia v runtime

```
val permission = ActivityCompat.checkSelfPermission(this,
                                                    Manifest.permission.ACCESS_FINE_LOCATION)
if (permission == PackageManager.PERMISSION_GRANTED) {
    mMap.isMyLocationEnabled = true
} else {
    requestPermissions(
        arrayOf(Manifest.permission.ACCESS_FINE_LOCATION),
        LOCATION_REQUEST_CODE)
}
```

- Callback

```
override fun onRequestPermissionsResult(requestCode: Int,
                                         permissions: Array<String>, grantResults: IntArray) {
    when (requestCode) {
        LOCATION_REQUEST_CODE -> { ....

```



# Elementy Gmap API

---

- **MapView** : **View**, ktorá vie zobrazit' Gmaps
- **SupportMapFragment:Fragment** - môže byť umiestnený v rámci fragmentu
- **GoogleMap** – hlavná trieda sa vytvorí s vytvorením MapView, SupportMF
- **Marker** – ikona na mape definovaná polohou latitude+longitude
- **Shapes** – útvary, napr. Polyline (lomená čiara), Polygon (n-uholník)
- **UiSettings** – nastavenie užívateľského rozhrania, napr. zoom-level, ...
- **Overlays** - vrstvy
- **My Location** – ak je povolená, zobrazuje sa button MyLocation, ktorý vycentruje mapu podľa aktuálnej polohy





# Marker/MarkerOptions



```
mMap.isMyLocationEnabled = true // zobrazí moju polohu
val mff = LatLng(48.151901, 17.068422) // button na mape // cache z prémie
val MFF = mMap.addMarker(MarkerOptions()
    .position(mff) // žiadne 1E6, ale slušná trieda LatLng
    .icon(BitmapDescriptorFactory // ikona markera
        .fromResource(R.drawable.andro_cube))
    .draggable(true) // vieme marker posúvať
    .alpha(0.5f) // 0=transparent, 1=nontransparent
    .flat(true) // marker sa nezoomuje s mapou
    .title("MFF") // popis markera
    .anchor(0.0f, 1.0f) // pozícia ikony relatívne k position
    .snippet("Kockáči") // popis
    .rotation(90.0f) // natočenie
)
```



# Camera



```
mMap.isBuildingsEnabled = true
```

```
mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(mff, 6))
```

```
mMap.animateCamera(CameraUpdateFactory.zoomIn()) //zoomOut
```

```
mMap.animateCamera(CameraUpdateFactory.zoomTo(13), 2000, null)
```

```
val cameraPosition = CameraPosition.Builder()
```

```
.target(mff) // kamera nasmerovaná na cieľ
```

```
.zoom(17) // finálny zoom level
```

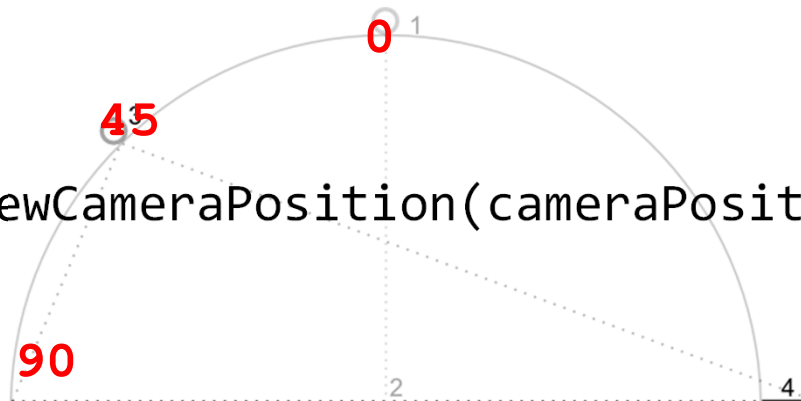
```
.bearing(90) // azimut kamery, 90=východ
```

```
.tilt(30) // horizontálne natočenie 30-90
```

```
.build()
```

```
mMap.animateCamera(
```

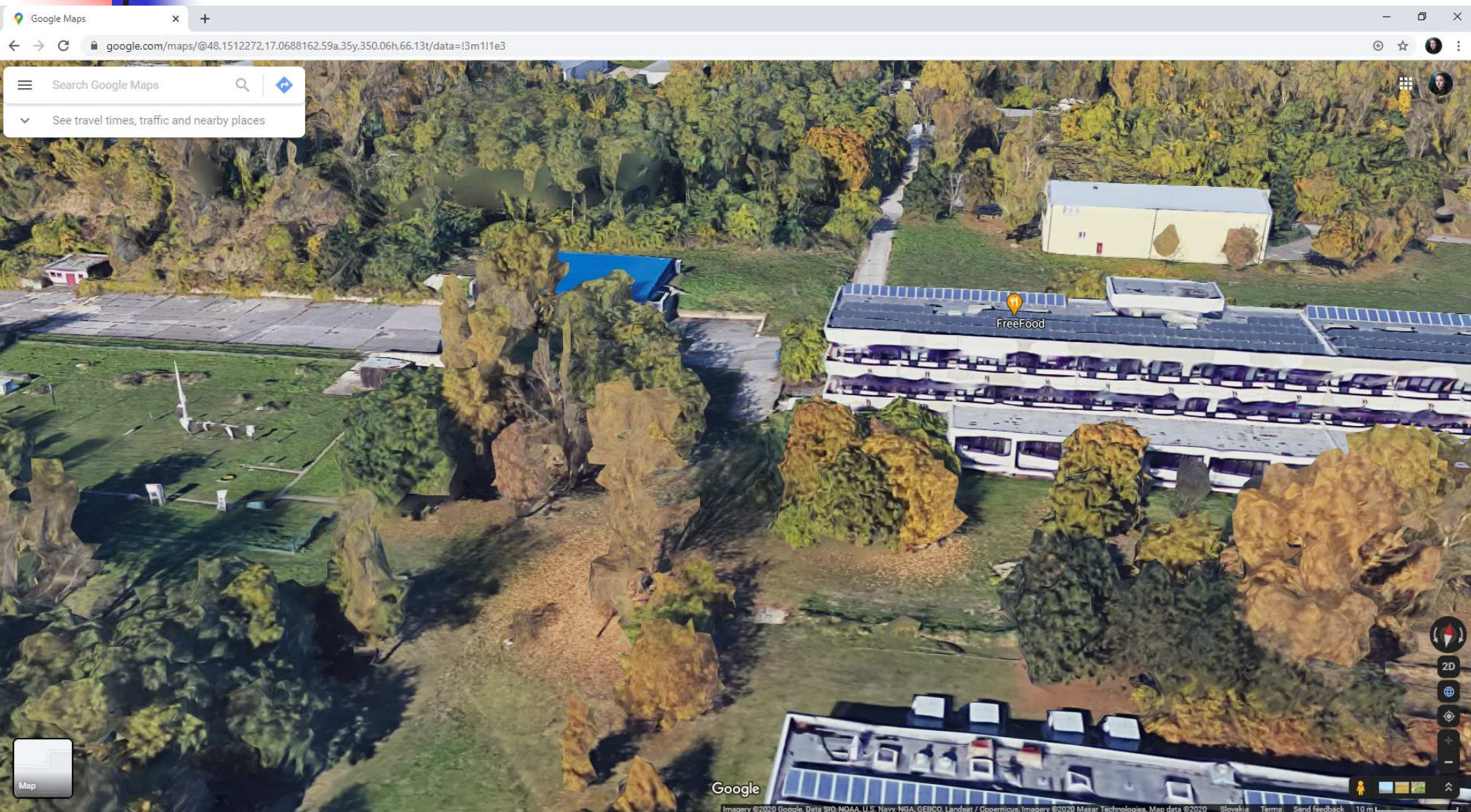
```
CameraUpdateFactory.newCameraPosition(cameraPosition))
```



Porovnanie: ESB vs. FMFI

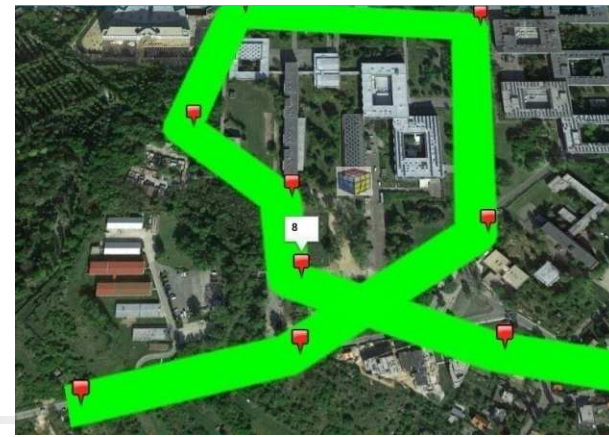


# Keška MFF





# onMapClick, PolyLine



```
clickedLine = mMap.addPolyline(PolylineOptions())
```

// ak chceme klikať do mapy

```
mMap.setOnMapClickListener {
```

```
    latlng -> // onClickListener - nefunguje pri satelitných m.
```

```
    val clickedPoints = clickedLine.points
```

```
    clickedPoints.add(latlng)
```

```
    clickedLine.points = clickedPoints
```

```
    clickedLine.color = ...
```

```
    clickedLine.width = ...
```

```
mMap.addMarker(MarkerOptions()
```

```
    .title("" + clickedPoints.size())
```

```
    .icon(BitmapDescriptorFactory
```

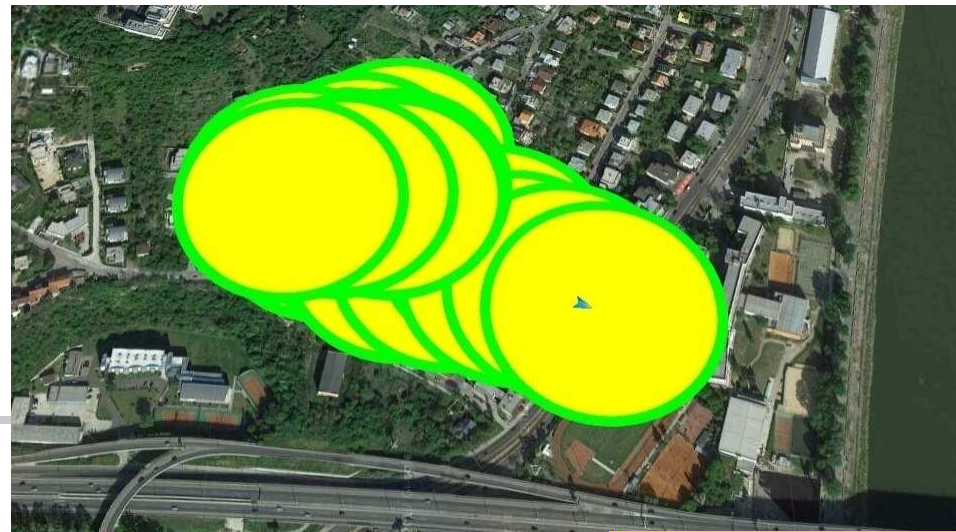
```
        .fromResource(R.drawable.marker))
```

```
    .position(latlng))
```



# GPS Location

(prvá možnosť - GPS)



Project:GMap.zip

Použijeme LocationManager, ako v minulej prednáške, a LocationListener

```
override fun onLocationChanged(loc:Location) {  
    val latlng = LatLng(loc.getLatitude(), loc.getLongitude())  
    val circle = mMap.addCircle(CircleOptions(latlng)  
        .center()  
        .radius(15)           // polomer kruhu v metroch  
        .strokeColor(Color.GREEN)  
        .fillColor(Color.YELLOW))  
    circle.setZIndex((float)System.currentTimeMillis()) }  

```

```
val bounds =                // ak si mimo...  
    mMap.getProjection().getVisibleRegion().latLngBounds  
if(!bounds.contains(latlng)) // preanimuj na novú polohu  
    mMap.animateCamera(CameraUpdateFactory.newLatLng(LatLng)
```

Project:GMap.zip



# uiSettings

---

```
map.uiSettings  
    .isMyLocationButtonEnabled  
    .isCompassEnabled  
    .isRotateGesturesEnabled  
    .isScrollGesturesEnabled  
    .isScrollGesturesEnabledDuringRotateOrZoom  
    .isZoomGesturesEnabled  
    .isTiltGesturesEnabled  
    .isMapToolbarEnabled
```

# LocationServices API

(Fused Location API)

<https://developer.android.com/training/location/retrieve-current.html>

Iná možnosť, ako získať poslednú polohu na vyššej úrovni ako od GPS senzora

- neriešite location providera (GPS/NETWORK/wifi),
- vysoká presnosť, nízka spotreba baterky,
- ale používajú Google Play Services, do build.gradle doplníte:

dependencies {

```
    implementation 'com.google.android.gms:play-services-location:18.0.0'
```


```
    implementation 'com.google.android.gms:play-services-maps:18.0.0'
```

```
}
```

- `import com.google.android.gms.*...`

- `fusedLocationProviderClient =`

```
    LocationServices.getFusedLocationProviderClient(this@MapsActivity)
```

- `locationCallback = object : LocationCallback() {`   
 `override fun onLocationResult(locationResult: LocationResult) {`  
 `for (location in locationResult.locations) {` *// pri zmene polohy*  
 `... location je aktuálna poloha`  
 `}`  
 `}`  
 `}` `}`

# Fused Location API

(onResume, onPause)

<https://developer.android.com/training/location/retrieve-current.html>

```
override fun onResume() {
    super.onResume()
    val interval = (10 * 1000).toLong()           // 10 seconds, in milliseconds
    val fastestInterval = (1 * 1000).toLong()     // 1 second, in milliseconds
    val minDisplacement = 0f

    // Create the LocationRequest object
    val mLocationRequest = LocationRequest.create() // criteria
        .setPriority(LocationRequest.PRIORITY_BALANCED_POWER_ACCURACY)
        .setInterval(interval)           -- ako často chceme update polohy
        .setFastestInterval(fastestInterval) -- ako často vieme spracovať update
        .setSmallestDisplacement(minDisplacement)

    fusedLocationProviderClient.requestLocationUpdates(
        mLocationRequest,
        locationCallback,
        Looper.getMainLooper())

    override fun onPause() {
        super.onPause()
        fusedLocationProviderClient.removeLocationUpdates(locationCallback)
    }
}
```

LocationRequest.PRIORITY\_HIGH\_ACCURACY ~GPS  
LocationRequest.PRIORITY\_BALANCED\_POWER\_ACCURACY; ~100m,block  
LocationRequest.PRIORITY\_LOW\_POWER; ~ 10km,mesto  
LocationRequest.PRIORITY\_NO\_POWER; berie polohu od iných klientov

# Fused Location API

(LocationCallback)

<https://developer.android.com/training/location/retrieve-current.html>

```
locationCallback = object : LocationCallback() {  
    // pri zmene polohy podľa LocationRequest  
    override fun onLocationResult(locationResult: LocationResult) {  
        for (location in locationResult.locations){  
            Log.w(TAG, "onLocationResult= ${location?.latitude}, ${location?.longitude}")  
            val savedLastLocation = LatLng(location.latitude, location.longitude)  
            val options = MarkerOptions()  
                .position(savedLastLocation)  
                .title("I am here!")  
            mMap.addMarker(options)  
            mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(savedLastLocation, 15f))  
        }  
    }  
}  
  
binding.sendSosBTN.setOnClickListener {  
    val sendIntent = Intent(Intent.ACTION_VIEW)  
    sendIntent.data = Uri.parse("sms:")  
    sendIntent.putExtra("sms_body",  
        "My location at \nLatitude: ${savedLastLocation.latitude} \n" +  
        "Longitude: ${savedLastLocation.longitude}")  
    startActivity(sendIntent)  
}
```



# Poloha vs. poloha

<https://antoniohongkr.wordpress.com/2013/08/19/google-play-service-analysis-4-choice-between-google-play-location-service-and-android-location-service/>

Priority	Typical location update interval	Battery drain per hour (%)	Accuracy
HIGH_ACCURACY	5 seconds	7.25% <b>~1/14</b>	~10 meters
BALANCED_POWER	20 seconds	0.6%	~40 meters
NO_POWER	N/A	small	~1 mile



# Demo pre inšpirácie

(kód je v java)

<https://github.com/googlesamples/android-play-location>

## GMapsAPI

### Basic Map

Launches a map.

### Camera

Demonstrates camera functions.

### Camera Clamping

Demonstrates how to constrain the camera to specific boundaries and zoom levels.

### Circles

Demonstrates how to add Circles to a map.

### Events

Demonstrates event handling.

### Ground Overlays

Demonstrates how to add a GroundOverlay to a map.

### Indoor

Demonstrates how to use the Indoor API.

### Layers

Demonstrates the different map layers.

### Lite Mode

Demonstrates some features on a map in lite mode.

### Lite Mode ListView

Demonstrates using maps in lite mode in a ListView.

### Location Source Demo

Demonstrates how to use a custom location source.

### Map In Pager

Demonstrates how to add a map to a ViewPager.

### Markers

## GMapsAPI

Demonstrates how to save the state of a MapFragment upon rotation of the device.

### Snapshot

Demonstrates how to take a snapshot of the map.

### Street View Panorama and Map

Demonstrates how to show a Street View panorama and map.

### Street View Panorama

Standard Street View Panorama using a Fragment.

### Street View Panorama events

Standard Street View Panorama with event handling.

### Street View Panorama navigation

Street View Panorama with programmatic navigation.

### Street View Panorama options

Street View Panorama with toggles for options.

### Street View Panorama View

Standard Street View Panorama using a View.

### Styled Map

Demonstrates how to style a map.

### Tile Coordinate Overlay

Demonstrates how to add a tile overlay with tile coordinates to a map.

### Tile Overlays

Demonstrates how to add a tile overlay to a map.

### UI Settings

Demonstrates how to alter user interface settings.

### Visible Regions

Demonstrates how to use Visible Regions.



# Break – Marcel Mapbox/OSM

---

# GoogleSignIn

```
val gso = GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestEmail()
    .build()
```

*// Build a GoogleSignInClient with the options specified by gso.*

```
val mGoogleSignInClient = GoogleSignIn.getClient(this, gso)
```

*// Set the dimensions of the sign-in button.*

```
sign_in_button.setSize(SignInButton.SIZE_STANDARD)
```

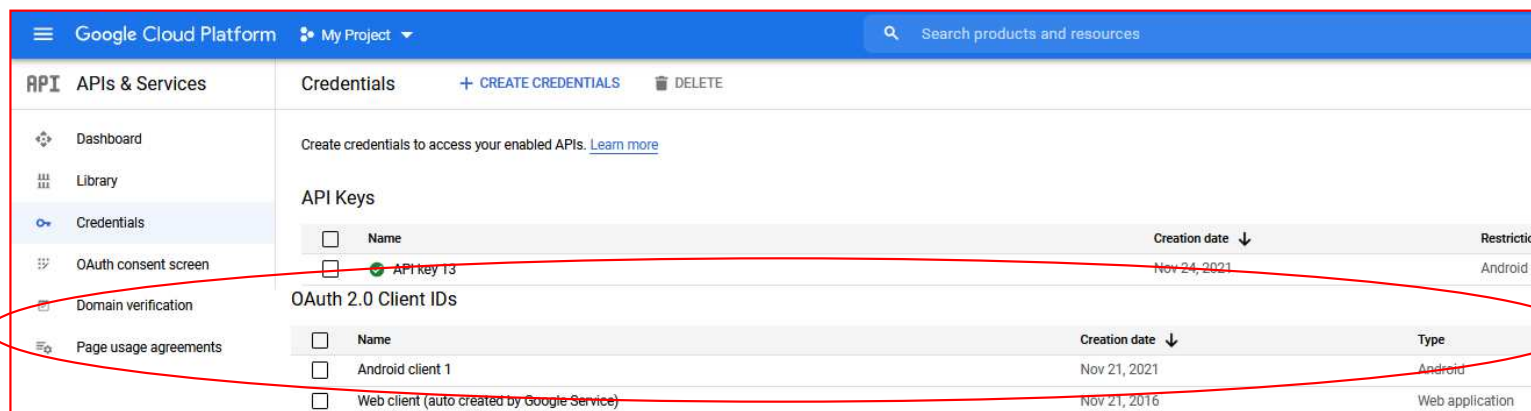
```
sign_in_button.setOnClickListener {
```

```
    val signInIntent = mGoogleSignInClient.signInIntent
```

```
    startActivityForResult(signInIntent, SIGN_IN_REQUESTCODE)
```

```
}
```

```
val mGoogleApiClient = mGoogleSignInClient.asGoogleApiClient()
```



The screenshot shows the Google Cloud Platform console. The left sidebar has a menu with 'APIs & Services' selected. The main content area is titled 'Credentials' and includes a '+ CREATE CREDENTIALS' button and a 'DELETE' button. Below this, there is a section for 'API Keys' with a table showing one key named 'API key 13' created on 'Nov 24, 2021' with a restriction of 'Android app'. Below the API Keys section is the 'OAuth 2.0 Client IDs' section with a table showing three client IDs: 'Android client 1' (created Nov 21, 2021, type Android), and 'Web client (auto created by Google Service)' (created Nov 21, 2016, type Web application). A red circle highlights the 'API Keys' and 'OAuth 2.0 Client IDs' sections.

Name	Creation date	Restriction
API key 13	Nov 24, 2021	Android app

Name	Creation date	Type
Android client 1	Nov 21, 2021	Android
Web client (auto created by Google Service)	Nov 21, 2016	Web application



# GoogleSignIn

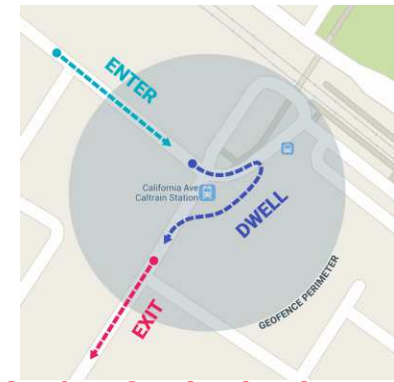
```
override fun onActivityResult(requestCode : Int, resultCode : Int, data : Intent?) {
    super.onActivityResult(requestCode, resultCode, data)

    if (requestCode == SIGN_IN_REQUESTCODE) {
        // The Task returned from this call is always completed, no need to attach
        // a listener.
        val task : Task<GoogleSignInAccount> =
            GoogleSignIn.getSignedInAccountFromIntent(data)
        handleSignInResult(task)
    }
}

private fun handleSignInResult(completedTask: Task<GoogleSignInAccount>) {
    try {
        val account = completedTask.getResult(ApiException::class.java)
        Log.w(TAG, "signInResult:success account= ${account.displayName}")
    } catch (e: ApiException) {
        Log.w(TAG, "signInResult:failed code=" + e.statusCode)
    }
}
```

# Geofencing

ploty a zábradlia



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

- detekuje vchod/východ/trvanie v oblasti, ktorá môže expirovať=zmiznúť
- AndroidManifest.xml:

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

Kruhová ohrada (zoznam ohraničený na 100 oblastí):

```
mGeofenceList.add(
    Geofence.Builder()
        .setRequestId(entry.getKey())
        .setCircularRegion(entry.getValue().latitude,
                           entry.getValue().longitude,
                           Constants.GEOFENCE_RADIUS_IN_METERS)
        .setExpirationDuration(Constants.GEOFENCE_EXPIRATION_IN_MILLISECONDS)
        .setTransitionTypes(Geofence.GEOFENCE_TRANSITION_ENTER |
                             Geofence.GEOFENCE_TRANSITION_EXIT)
        .build())
```