

Android

-

mapy



Peter Borovanský

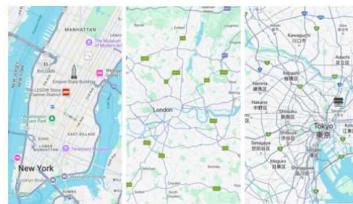
KAI, I-18

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

borovan 'at' ii.fmph.uniba.sk

Google Maps

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



Maps SDK for
Android

Build dynamic, interactive, customized maps, location, and
geospatial experiences for your Android apps.

Search Maps SDK for Android docs



Google Maps

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 reinštale počítača idete s novým finger-printom!)

Dôsledok1 ☹

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

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

Dôsledok2 ☺

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

Získanie SHA1

existuje viacero spôsobov

Potrebujeme získat' náš SHA1 kľúč, v cmdline spustíme „správny“ keytool.exe:

c:\Program Files\Android\Android Studio\jbr\bin\keytool.exe

```
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/

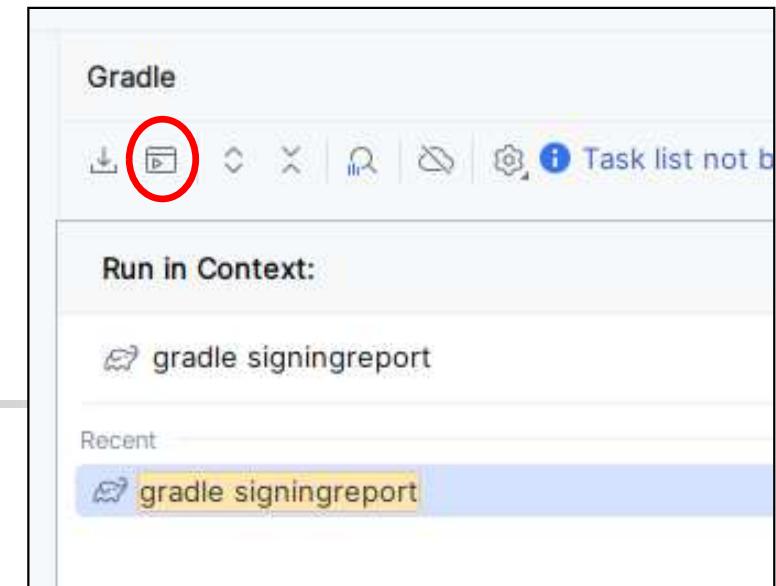
[4 files and 0 directories found]
c:\Program Files\Android\Android Studio\jbr\bin\keytool.exe
c:\Program Files\JetBrains\GoLand 2024.1.1\jbr\bin\keytool.exe
c:\Program Files\JetBrains\IntelliJ IDEA 2024.1.1\jbr\bin\keytool.exe
c:\Program Files\JetBrains\PyCharm 2024.2.2\jbr\bin\keytool.exe

```
C:\Windows\System32\cmd.exe  
  
d:\borovan\VMA\VMA\Prednasky\09>"c:\Program Files\Android\Android Studio\jbr\bin\keytool.exe" -v -list -alias androiddebugkey -keystore "c:\Users\borovan\.android\debug.keystore" -storepass android -keypass android  
Alias name: androiddebugkey  
Creation date: 2. 7. 2024  
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: Tue Jul 02 15:15:27 CEST 2024 until: Thu Jun 25 15:15:27 CEST 2054  
Certificate fingerprints:  
SHA1: 3A:CF:42:C.....:02:E1:82  
SHA256: 1B:F0:1C:E7:58:CC:1D:CE:16:7B:8B:C9:2B:0F:2D:41:38:1A:9B:12:E2:CA:C0:0D:A2:0D:96:32:35:72:CD:FF  
Signature algorithm name: SHA256withRSA  
Subject Public Key Algorithm: 2048-bit RSA key  
Version: 1  
  
d:\borovan\VMA\VMA\Prednasky\09>
```

Získanie SHA1

(v Android Studio)

- Gradle
 - Execute Gradle Task: gradle signingreport
 - Run Project[SigningReport]
 - už mám SHA-1 a čo s ním ? ...bude ho treba v Google Cloud Console



```
Store: C:\Users\borovan\.android\debug.keystore
Alias: AndroidDebugKey
MD5: 43:3E:04:B4:9E:F5:39:01:54:31:60:EF:56:1E:C0:EC
SHA1: 3A:CF:42 [REDACTED]:02:E1:82
SHA-256: 1B:F0:1C:E7:58:CC:1D:CE:16:7B:8B:C9:2B:0F:2D:41:38:1A:9B:12:E2:CA:C0:0D:A2:0D:96:32:35:72:CD:FF
Valid until: Thursday, June 25, 2054
```

Debug keystore

(maňment certifikátov)

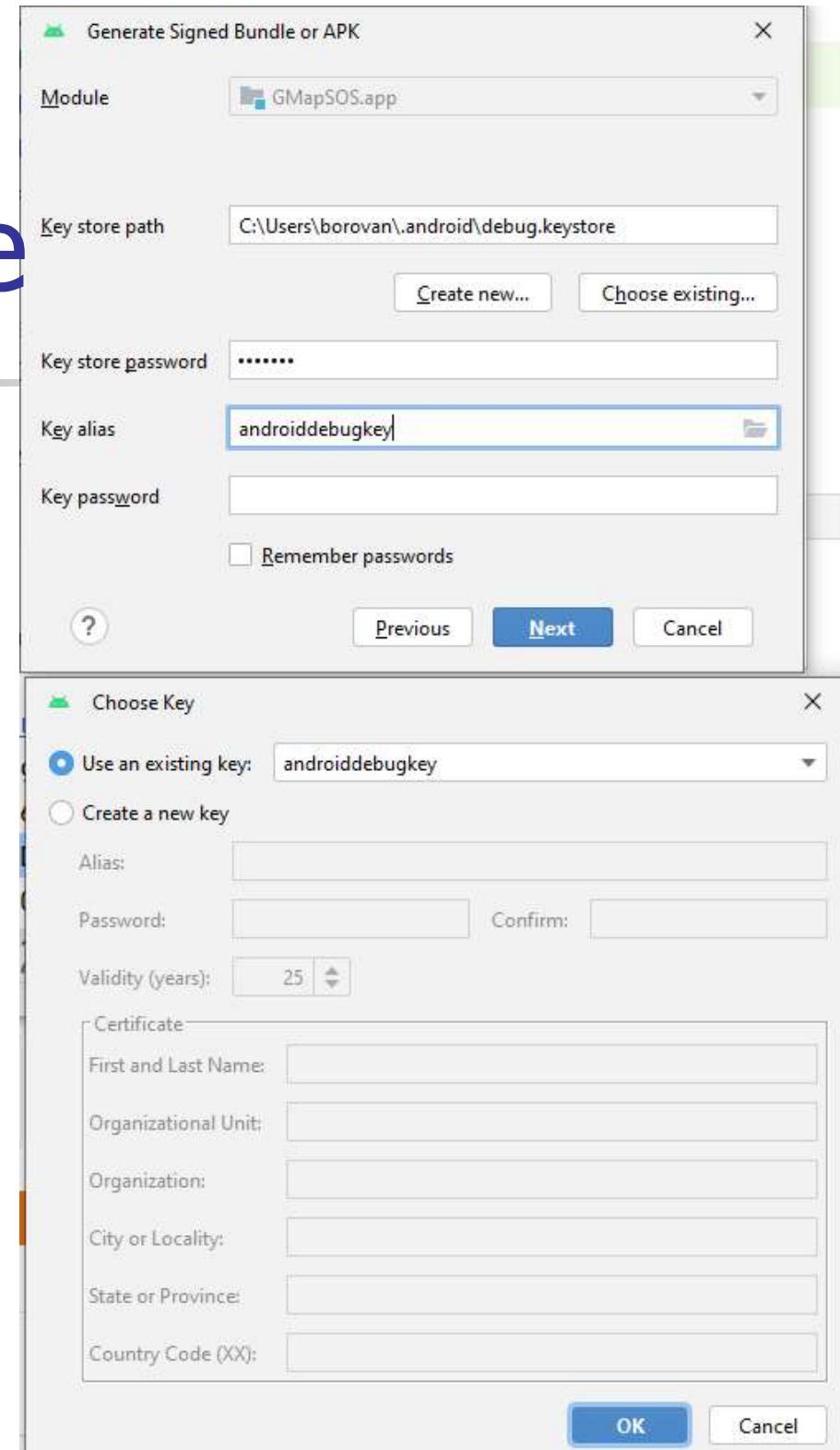
```
$ keytool -genkey -v  
-keystore debug.keystore  
-storepass android  
-alias androiddebugkey  
-keypass android  
-keyalg RSA -keysize 2048  
-validity 10000
```

Keystore name: "debug.keystore"

Keystore password: "android"

Key alias: "androiddebugkey"

Key password: "android"



Debug keystore

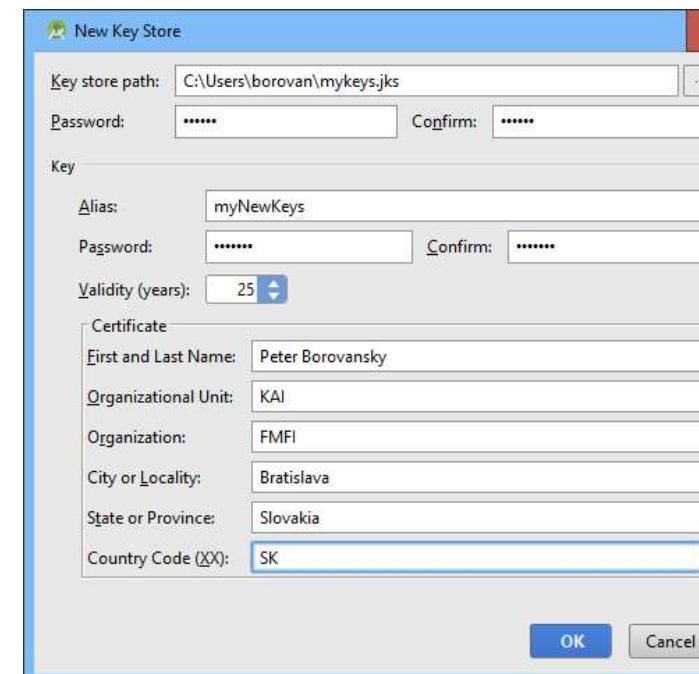
(manažment certifikátov)

- **debug.keystore** obsahuje jeden alebo viac privátnych kľúčov (certifikátov).
- **debug.keystore** nám automaticky vytvorí Android-Studio pri inštalácii AS
- Android Studio nám automaticky podpíše každú apkú 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>

```
C:\Windows\System32\cmd.exe
d:\borovan\VMA\VMA\Prednasky\09>"c:\Program Files\Android\Android Studio\jbr\bin\keytool.exe" -genkey -v -keystore debug.keystore -storepass android -alias androiddebugkey -keypass android -keyalg RSA -keysize 2048 -validity 10000
What is your first and last name?
[Unknown]: Peter Borovansky
What is the name of your organizational unit?
[Unknown]: DAI
What is the name of your organization?
[Unknown]: FMPH
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=DAI, O=FMPH, 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=DAI, O=FMPH, L=BA, ST=SK, C=SK
[Storing debug.keystore]

d:\borovan\VMA\VMA\Prednasky\09>
```



Generovanie kl'úča

začnite v Build/Generate Signed APK

The screenshot shows the "Generate Signed Bundle or APK" wizard in Android Studio. It has two tabs: "Android App Bundle" (selected) and "APK". The "Android App Bundle" tab indicates it generates a signed app bundle for upload to the Play Store, mentioning smaller download size, on-demand app features, and asset-only modules. The "APK" tab indicates it builds a signed APK for deployment to a device. Both tabs have "Next" and "Previous" buttons. A central modal window titled "Generate Signed Bundle or APK" shows the "Module" dropdown set to "app", the "Key store path" field empty, and two buttons: "Create new..." and "Choose existing...". Below these are fields for "Key store password", "Key alias", and "Key password", along with a "Remember passwords" checkbox. At the bottom are "Previous", "Next", "Cancel", and "Help" buttons.

This dialog box is used to create a new key store. It requires the "Key store path" (C:\Users\borovan\mykeystore.jks), "Password" (.....), and "Confirm" (.....). It also allows setting the "Validity (years)" to 25. Under the "Certificate" section, developer information is provided: First and Last Name (PB), Organizational Unit (FMFI), Organization (UK), City or Locality (BA), State or Province (SK), and Country Code (XX) (SK). "OK" and "Cancel" buttons are at the bottom.

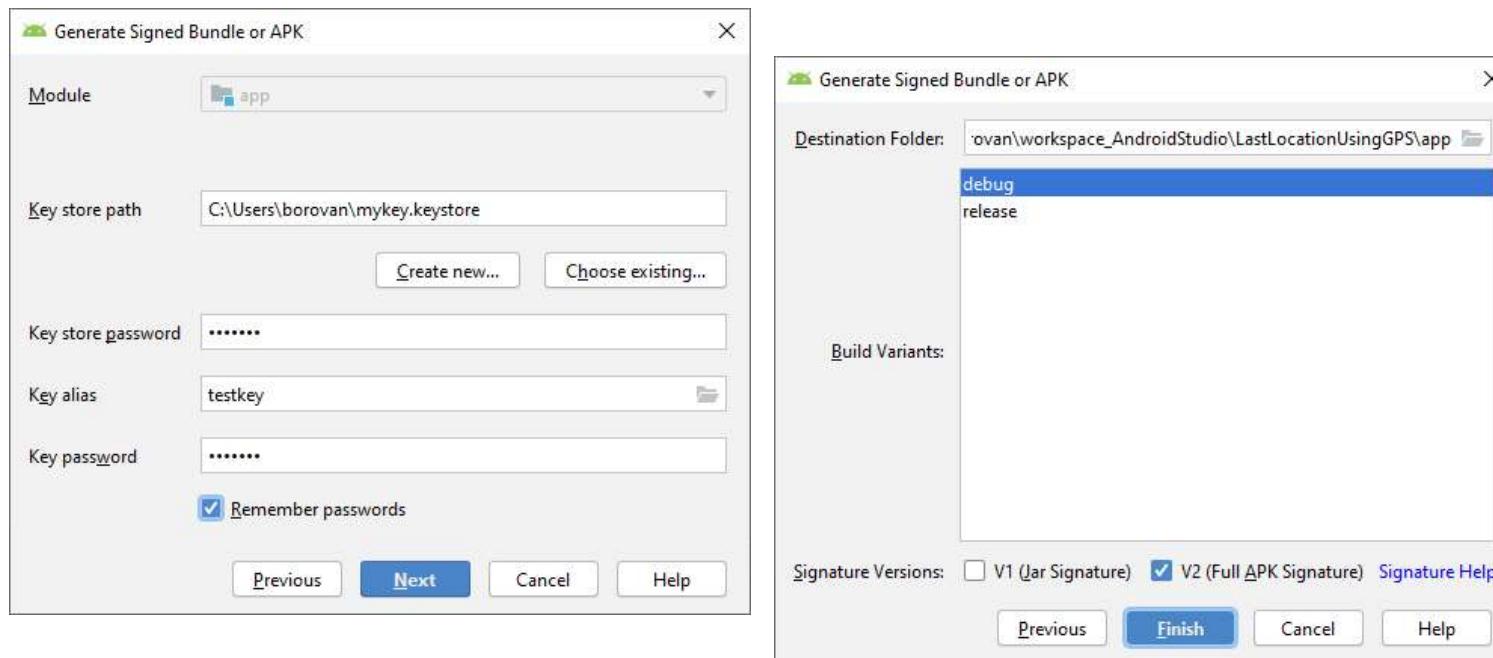
App bundles are publishing format, whereas **APK (Android application Package)** is the packaging format which eventually will be installed on device. Google uses app bundle to generate and serve optimized APKs for each user's device configuration, so they download only the code and resources they need to run your app. Therefore, users can get smaller and more optimized downloads.



Podpisovanie aplikácie (debug)

(použitie certifikátov)

Vygenerovaný kľúč potom slúži na podpísanie .apk

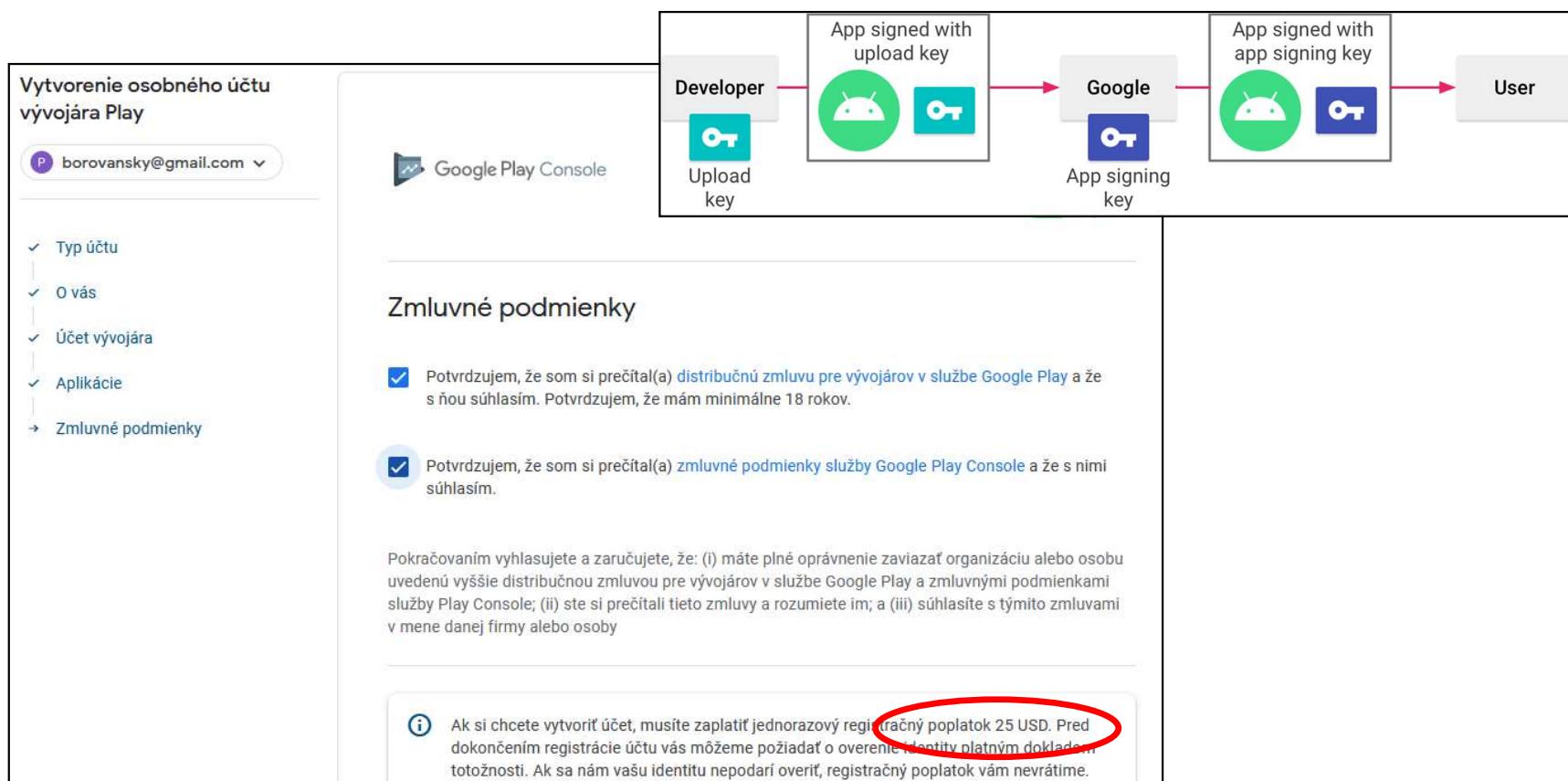


AS automaticky podpisuje .apk pri builde kľúčom **androiddebugkey**

Podpisovanie aplikácie (release)

(použitie certifikátov)

- Potrebujete Upload key a App signing key:
<https://developer.android.com/studio/publish/app-signing>
- Potrebujete Google Play Account, Google Play Console:
<https://play.google.com/apps/publish/signup/>



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>

Google Play Console

borovansky@gmail.com

Povedzte nám o sebe

Meno vývojára

Verejný názov vášho účtu vývojára na 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

Cloud Console – Step 1

Step 1 Step 2 Step 3
[Set up your project](#) [Enable APIs or SDKs](#) [Get an API Key](#)

Console Cloud SDK

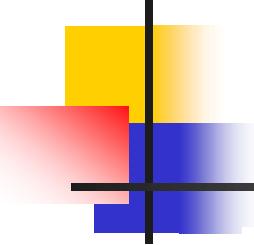
1. In the Google Cloud Console, on the project selector page, click **Create Project** to begin creating a new Cloud project.

[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: IE3668997OH

Bill to

Peter Borovansky

B

B

81105 Bratislava

Slovakia

Details

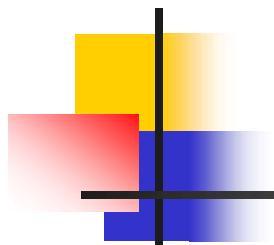
Invoice number 4603377165
Invoice date Oct 31, 2022
Billing ID 0195-6
Account ID 01C0FC

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

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

...stalo sa ...



Google Maps API

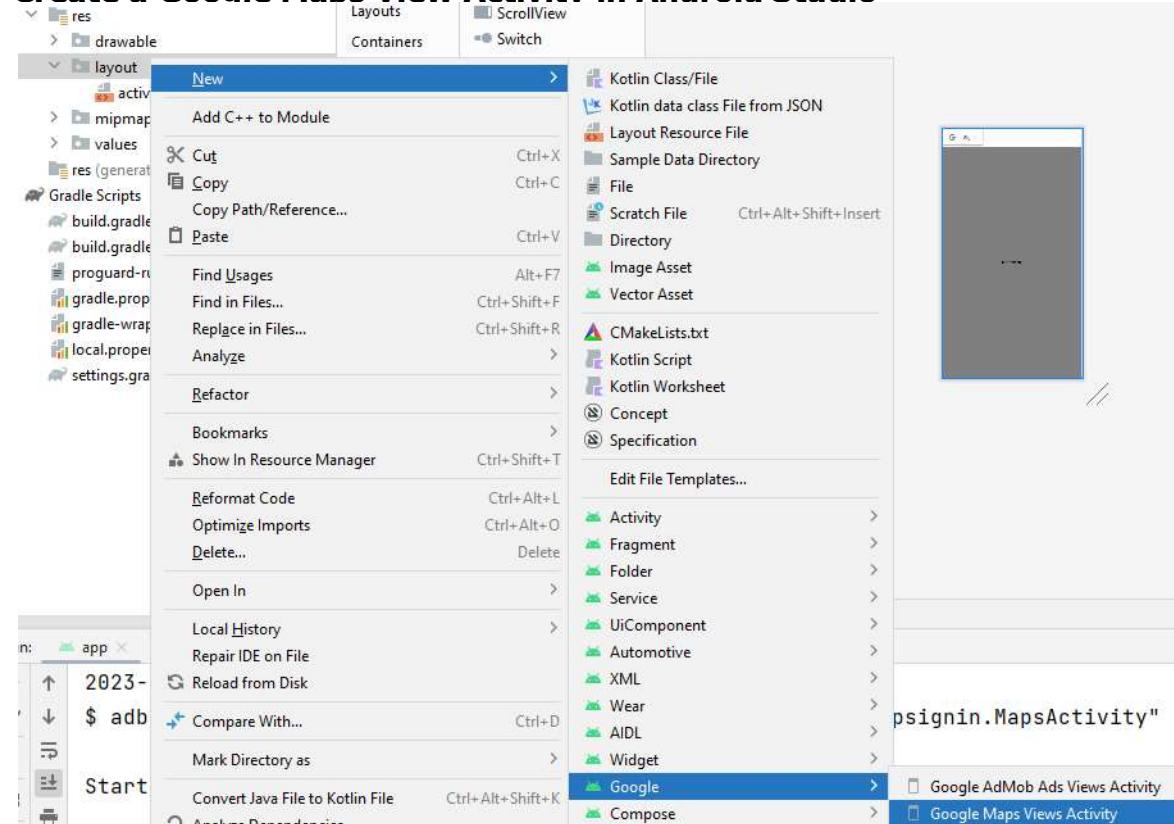
Treba **dôsledne** (!!!) prejst' oficiálnym návodom:

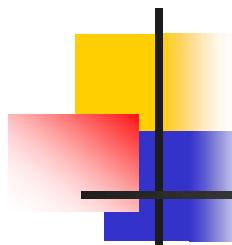
- <https://developers.google.com/maps/documentation/android-api/start>

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>

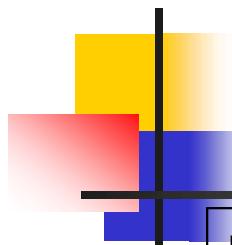
Once you have your API key (it starts with "Alza"), define a new property in your project's local.properties file (e.g. MAPS_API_KEY=Alza...), and replace the "YOUR_API_KEY" string in this file with "\${MAPS_API_KEY}".

```
-->
```

```
<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 2

project-level build.gradle

```
plugins {
    alias(Libs.plugins.google.android.libraries.mapsplatform.secrets.gradle.plugin) apply false
}

// Top-Level build file where you can add configuration options common to all sub-
// projects/modules.
plugins {
    alias(Libs.plugins.android.application) apply false
    alias(Libs.plugins.kotlin.android) apply false
    alias(Libs.plugins.google.android.libraries.mapsplatform.secrets.gradle.plugin) apply
false
}
```

module-level build.gradle

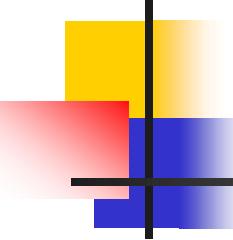
```
alias(Libs.plugins.google.android.libraries.mapsplatform.secrets.gradle.plugin)
```

local.properties

```
MAPS_API_KEY=AlzaSyB31D.....XSqlKtK4SATo
```

AndroidManifest.xml

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



Nepodpisana 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.xxxxxx123233

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.xxxxxx123233

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.xxxxxx123233

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.xxxxxx123233

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

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

VIEW ALL (15)

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-----9M 

⚠ 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](#)

Restrict API Key

Name *
API key 18

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

API Key
AIzaSyAQHz1aQu_1VG2N1

Use this key in your application

Creation date 

How do I restrict my API key?

You can restrict an API key to a specific certificate fingerprint or a release key.

Debug certificate fingerprints

For Linux or macOS:

```
$ keytool -list -v -keystore /path/to/debug_keystore
```

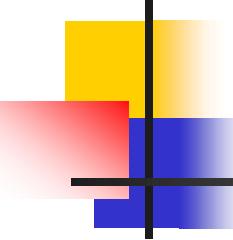
For Windows:

```
$ keytool -list -v -keystore %USERPROFILE%\.android\debug.keystore
```

Release certificate fingerprints

```
$ keytool -list -v -keystore /path/to/release_keystore
```

Replace your_keystore_name with the name of the keystore you assigned to the certificate when you signed your APK.



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:D1 :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]

[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:[REDACTED] 6:8E:57:A2:80:1C:D1:91

pokus.example.com.gmapv2

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

com.example.mapdemo

05:0F:34:5E:FD:[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 pridel'uje
pre dvojicu, alebo niekol'ko dvojíc,
(package name, SHA1-v prostredí, kde komplujete)

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 klúč 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>)

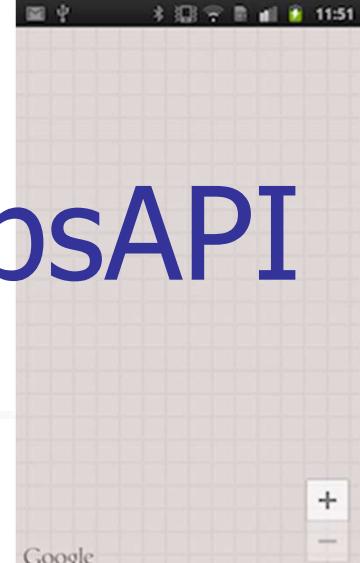
Ensure that the "Google Maps Android API v2" is enabled.

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 súvisiet' 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

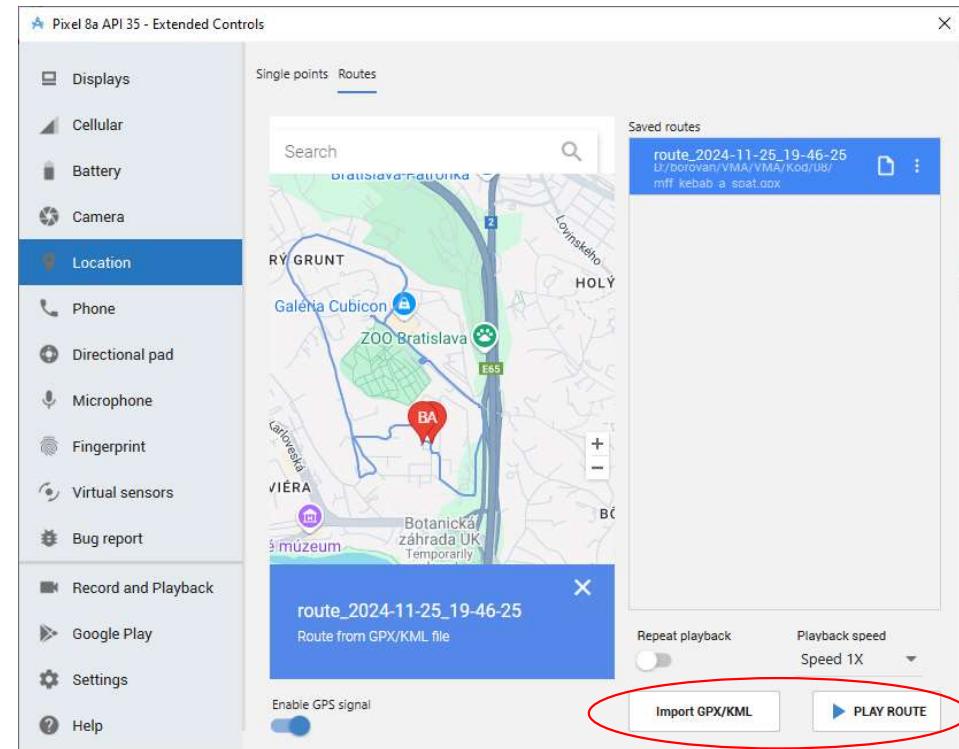
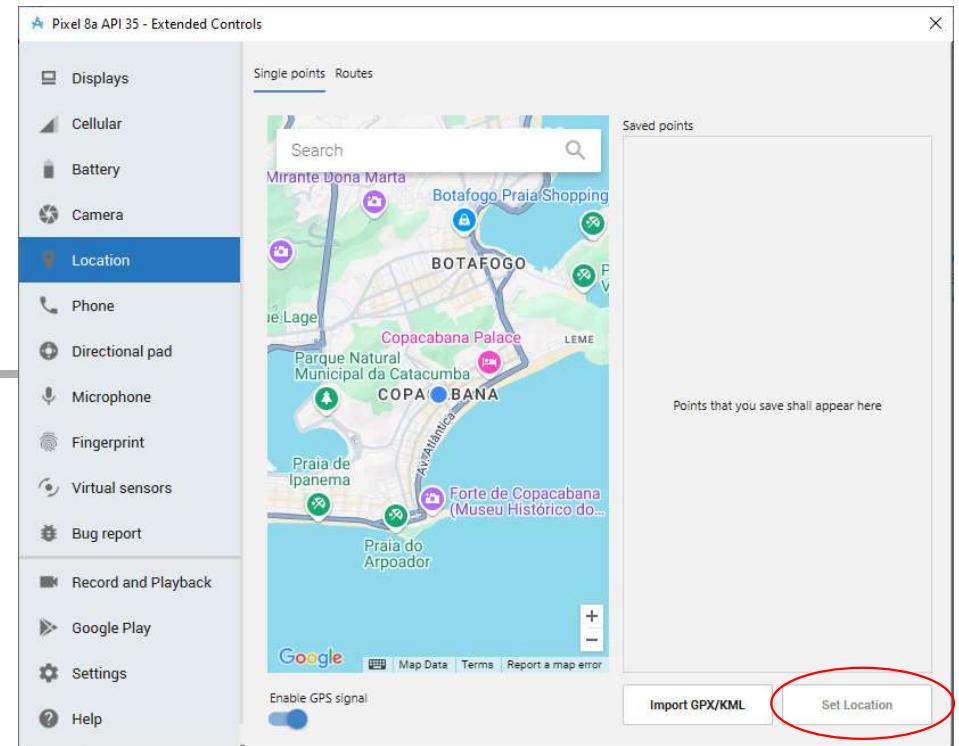
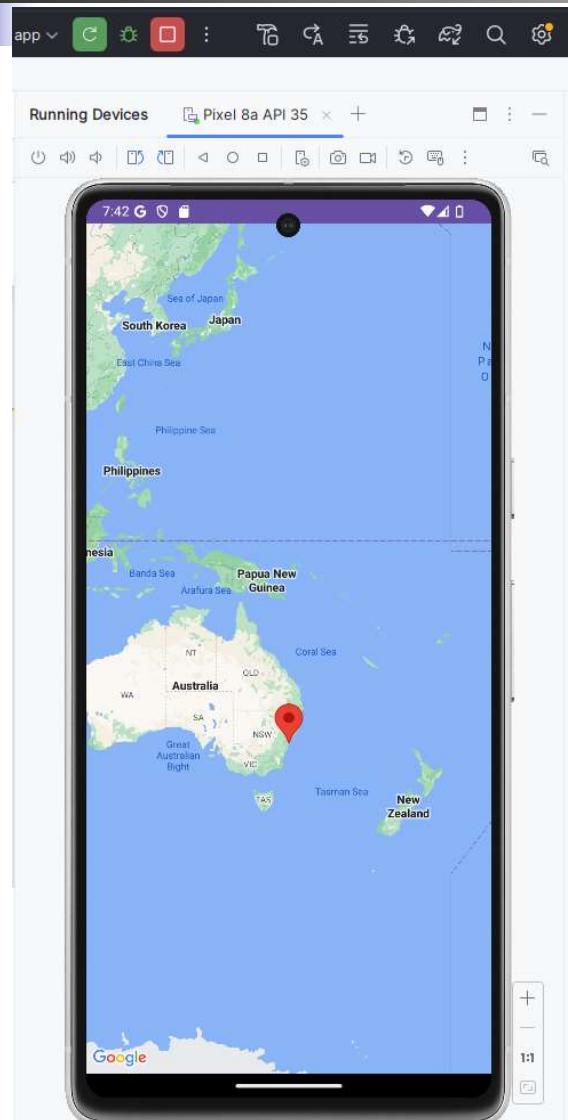
ComponentInfo{com.ecs.google.maps.v2.actionbarsherlock/com.ecs.google.maps.v2.simple.SimpleMapActivity}:

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 2024



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()  
            .position(sydney)  
            .title("Marker in Sydney"))  
        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney))  
    }  
}
```

Layout s MapFragmentom

<fragment

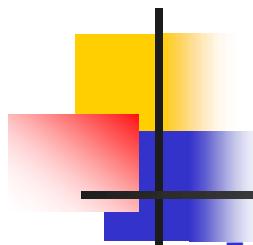
```
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:map="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/map"
android:name="com.google.android.gms.maps.SupportMapFragment"
android:layout_width="match_parent"
android:layout_height="match_parent"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



Project:GMAPEmpty.zip



Permissions

AndroidManifest.xml

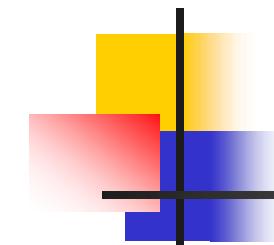
```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

- 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 zobraziť 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



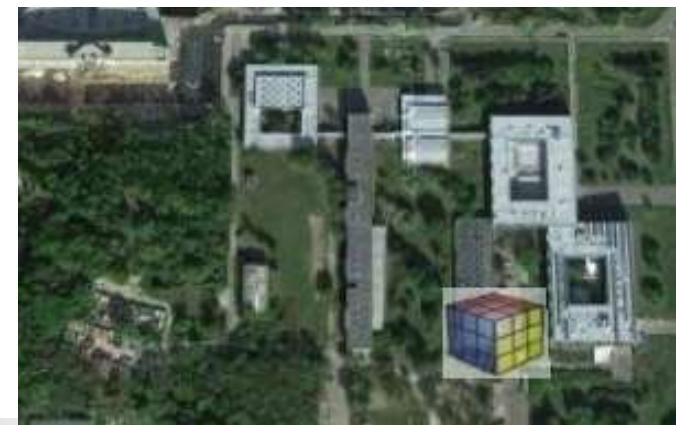
<https://developers.google.com/maps/documentation/android-sdk/groundoverlay>

Marker/MarkerOptions



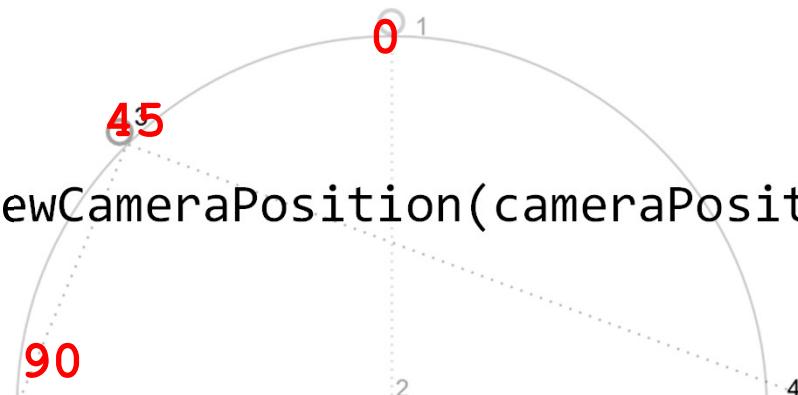
```
// zobrazí moju polohu
mMap.isMyLocationEnabled = true      // button na mape
val mff = LatLng(48.151901, 17.068422) // cache z premie
val MFF = mMap.addMarker(MarkerOptions()
    .position(mff)           // žiadne 1E6, ale slušná trieda LatLng
    .icon(BitmapDescriptorFactory // ikona markera
          .fromResource(R.drawable.andro_cube))           // Rubik's cube icon
    .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)              // horizontalne natočenie 30-90  
    .build()  
  
mMap.animateCamera(  
    CameraUpdateFactory.newCameraPosition(cameraPosition))
```



Keška MFF

LukášM.



FilipB.



Google Maps

google.com/maps/@48.1512272,17.0688162,59a,35y,350.06h,66.13t/data=!3m1!1e3

Search Google Maps

See travel times, traffic and nearby places

FreeFood

Map

2D

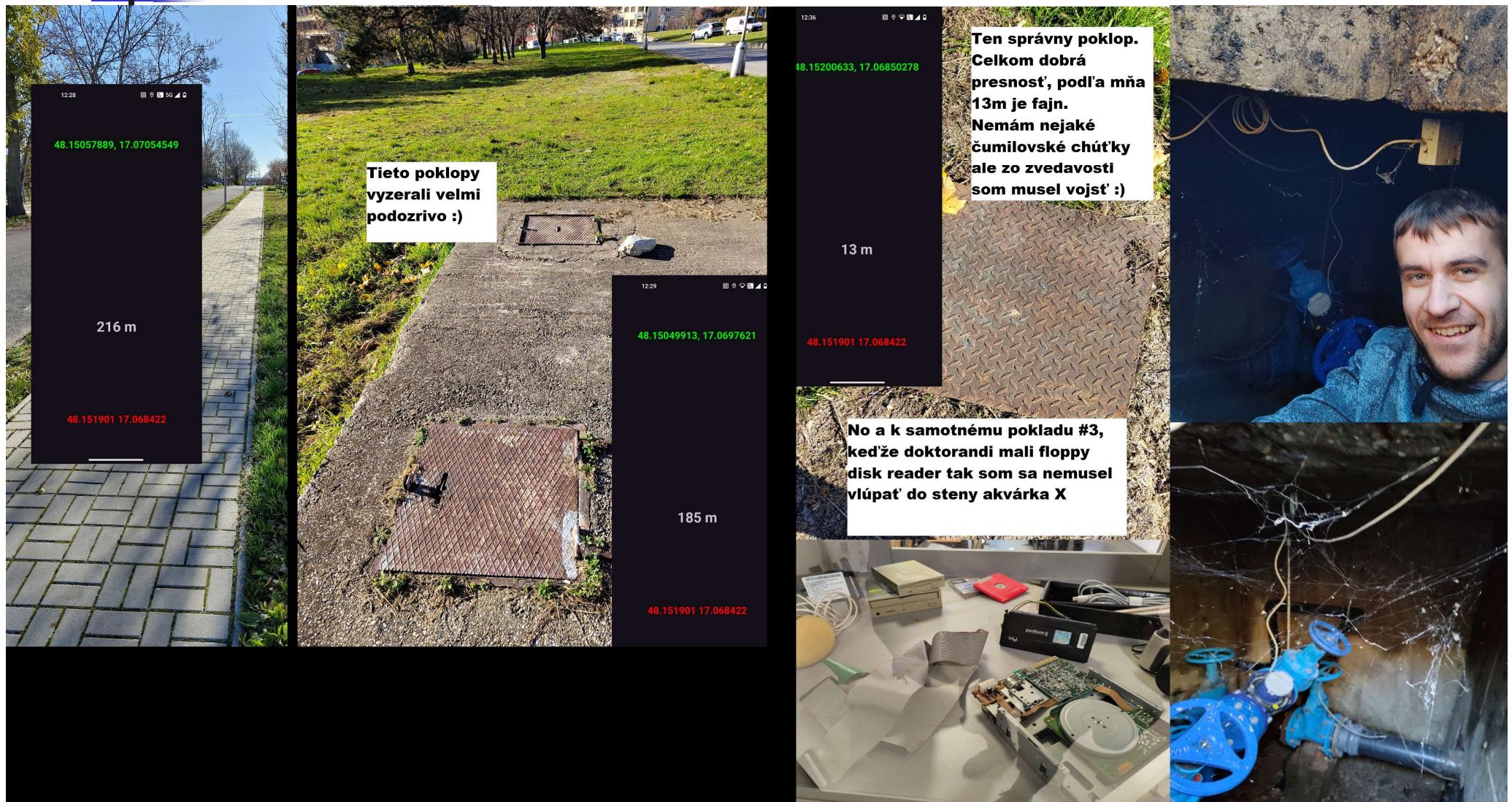
Imagery ©2020 Google. Data SIO, NOAA, U.S. Navy NGA, GEBCO, Landsat / Copernicus. Map data ©2020 Maxar Technologies. Map data ©2020

Slovakia Terms Send feedback 10 m L

An aerial photograph of a university campus. In the foreground, there are several large, leafy trees and a grassy area. To the right, a long, modern building with a light-colored facade and a solar panel array on its roof is visible. A small orange location marker with the text 'FreeFood' is placed on the building. In the background, there are more buildings, a river, and a dense forest. The image is overlaid with a Google Maps interface, including a search bar, travel time information, and various map controls like zoom and orientation.

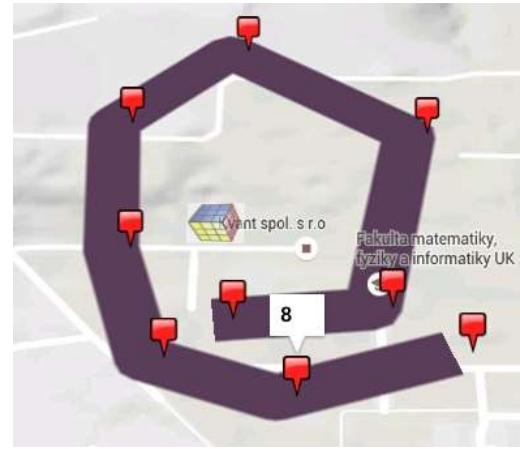
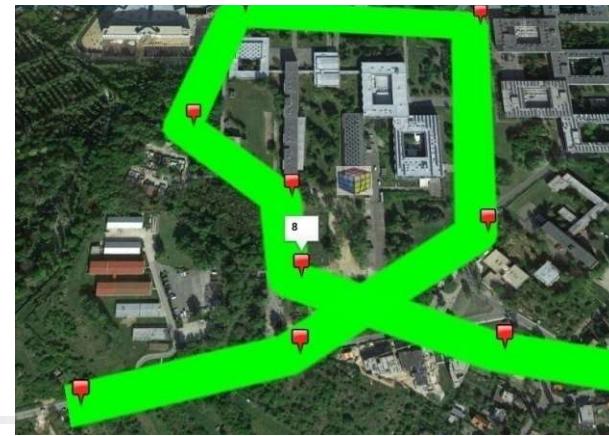
MichalV

problém, že „Darovanej kobyle do zubov nehl'adia“



onMapClick, PolyLine

```
clickedLine = mMap.addPolyline(PolylineOptions())
                    // ak chceme klikat' do mapy
mMap.setOnMapClickListener {
    latlng -> // onClickListener - nefunguje pri satelitnych 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))
```



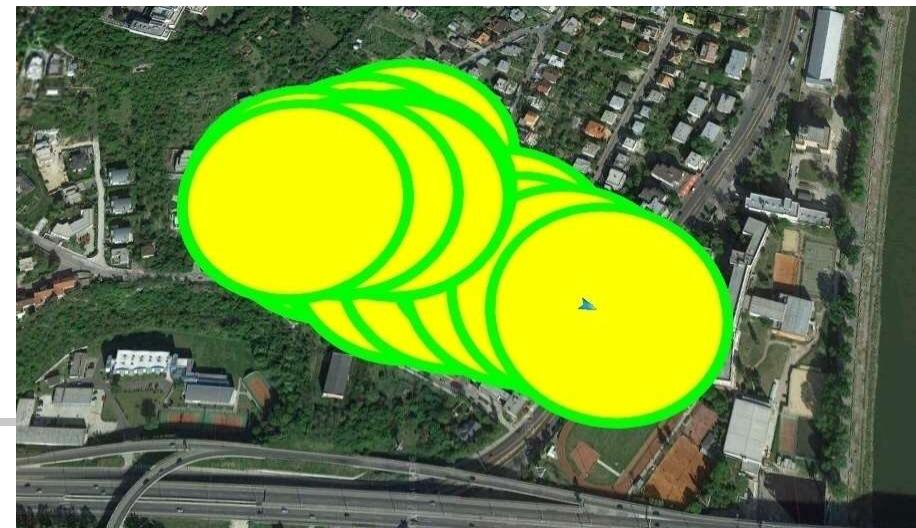
GPS Location

(prvá možnosť - GPS)

Použijeme LocationManager, ako v minulej prednáške, a LocationListener

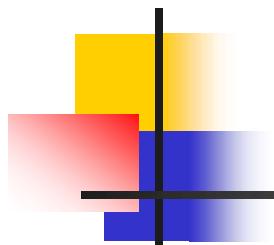
```
override fun onLocationChanged(loc:Location) {  
    val latlng = LatLng(loc.getLatitude(), loc.getLongitude())  
    val circle = mMap.addCircle(CircleOptions(latlng)  
        .center()  
        .radius(15)           // polomer kruhu v metrech  
        .strokeColor(Color.GREEN)  
        .fillColor(Color.YELLOW))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

Project:GMap.zip



uiSettings

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

LocationServices API

(Fused Location API)

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

Simple, battery-efficient location API for Android

Apps can take advantage of the signals provided by multiple sensors in the device to determine device location. However, choosing the right combination of signals for a specific task in different conditions is not simple. Finding a solution that is also battery-efficient is even more complicated.

The fused location provider is a location API in Google Play services that intelligently combines different signals to provide the location information that your app needs.

The fused location provider manages the underlying location technologies, such as GPS and Wi-Fi, and provides a simple API that you can use to specify the required quality of service. For example, you can request the most accurate data available, or the best accuracy possible with no additional power consumption.

```
/*
 * Provides the entry point to the Fused Location Provider API.
 */
private FusedLocationProviderClient mFusedLocationClient;

/**
 * Represents a geographical location.
 */
protected Location mLastLocation;

private String mLatitudeLabel;
private String mLongitudeLabel;
private TextView mLatitudeText;
private TextView mLongitudeText;

/**
 * Provides a simple way of getting a device's location and is well
 * applications that do not require a fine-grained location and the
 * updates. Gets the best and most recent location currently available.
 */
private void getLastLocation() {
    if(ContextCompat.checkSelfPermission(context, Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED) {
        mFusedLocationClient.getLastLocation()
            .addOnCompleteListener(activity, new OnCompleteListener<Location>() {
                @Override
                public void onComplete(@NonNull Task<Location> task) {
                    if (task.isSuccessful() && task.getResult() != null) {
                        mLastLocation = task.getResult();
                    }
                    mLatitudeText.setText(String.format(Locale.getDefault(), "%s", mLastLocation.getLatitude()));
                    mLongitudeText.setText(String.format(Locale.getDefault(), "%s", mLastLocation.getLongitude()));
                }
            });
    }
}
```

Support for common location scenarios

Last Known Location

Location Settings

Project:GMapSOS.zip

LocationServices API

(Fused Location API)

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

Iná možnosť, ako získat' 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  
        }  
    } }
```

Project:GMapSOS.zip

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
                                            polohy-pre prípad, že iná apka intenzívnejšie používa GPS
        .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

Project:GMapSOS.zip

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% ~1/14 | ~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.



Project:ApiDemos.zip

GoogleSignIn

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

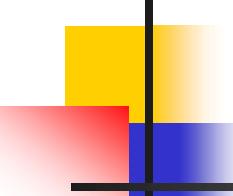
```
// Build a GoogleSignInClient with the options specified by gso.
val mGoogleSignInClient = GoogleSignIn.getClient(this, gso)
// Set the dimensions of the sign-in button.
sign_in_button.setSize(SignInButton.SIZE_STANDARD)
sign_in_button.setOnClickListener {
    val signInIntent = mGoogleSignInClient.signInIntent
    startActivityForResult(signInIntent, SIGN_IN_REQUESTCODE)
}
val mGoogleApiClient = mGoogleSignInClient.asGoogleApiClient()
```

The screenshot shows the Google Cloud Platform API & Services page under the 'Credentials' section. It displays two tables: 'API Keys' and 'OAuth 2.0 Client IDs'. A red oval highlights the 'API Keys' table, which contains one entry: 'API key 13' (Creation date: Nov 24, 2021, Restriction: Android app). Another red oval highlights the 'OAuth 2.0 Client IDs' table, which contains three entries: 'Android client 1' (Creation date: Nov 21, 2021, Type: Android), 'Web client (auto created by Google Service)' (Creation date: Nov 21, 2016, Type: Web application), and 'Page usage agreements' (which is collapsed).

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

| OAuth 2.0 Client IDs | | | |
|---|---------------|-----------------|--|
| Name | Creation date | Type | |
| Android client 1 | Nov 21, 2021 | Android | |
| Web client (auto created by Google Service) | Nov 21, 2016 | Web application | |

GMapsSignIn.zip



GoogleSignIn

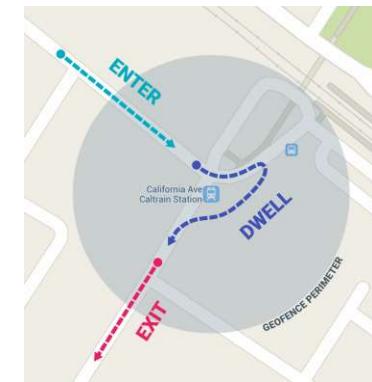
```
override fun onActivityResult(requestCode : Int, resultCode : Int, data : Intent?) {
    super.onActivityResult(requestCode, resultCode, data)

    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>

- detektuje 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())
```