MFF Keška, 48.151901 17.068422

- ~~zatiaľ nikto...~~

- DU3-Gameska
- MartinD
- JoaoN
- IvanaN

Android
-
mapy

Peter Borovanský
KAI, I-18

borovan 'at' ii.fmph.uniba.sk

Google Maps
- ~~V1,~~ **V2**
- **poloha ešte raz**
- **offline maps**

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*.

Postupy možno nájdete, ale pozor, **sú dva**:

- verzia **V1** (historicky zastaralá a dávno už nepodporovaná verzia GMaps): starý https://developers.google.com/maps/documentation/android/v1/

- verzia **V2** (pár rokov už jediná alternatíva Gmaps Verzia 2): *nový postup:* 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 reinstalle počítača idete s novým finger-printom)

Dôsledok: Ak si rebuildujete niektorý zo zverejených kódov, nepôjde vám skôr, ako si aplikáciu nezaregistrujete v Google Console, a vygenerovaným API key nepodpíšete váš build.

Dôsledok2: ak nerebuildujete s vašim SHA1, tak .apk je ok, a pôjde nainštalovať a spustiť.

Získanie SHA1

```
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
```

Potrebujeme získať náš SHA1 kľúč, v cmd spustíme keytool (v JRE java\bin):

```
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

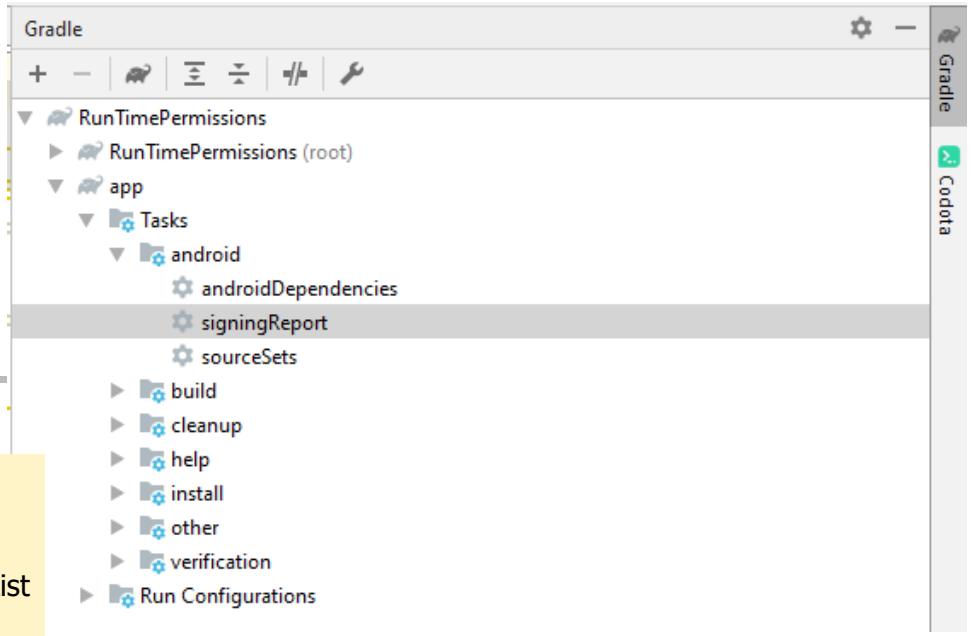
- Windows: `c:\Users\<user>\.android\debug.keystore`
- Linux: /home/*user*/.android/

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

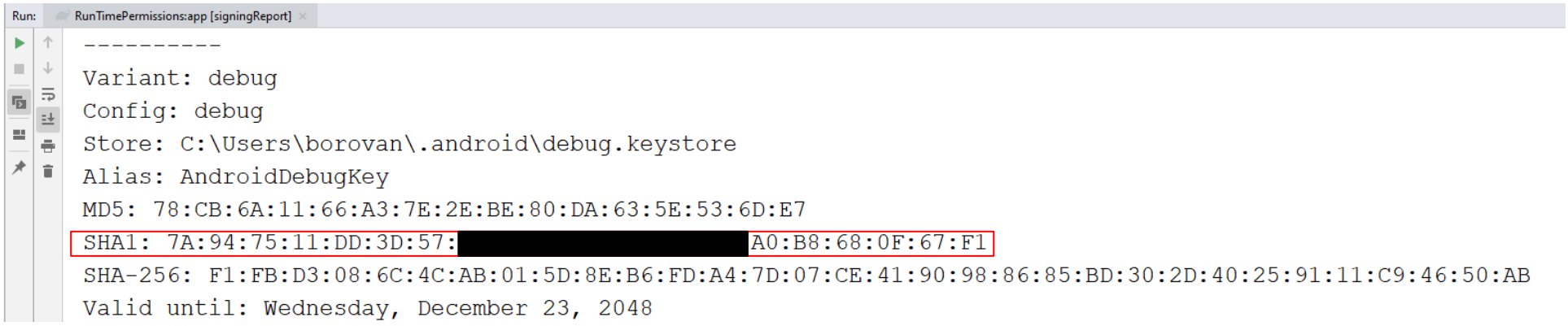
```
C:\Users\borovan>keytool -v -list -alias androiddebugkey -keystore "c:\Users\borovan\.android\debug.keystore" -storepass andr
oid -keypass android
Alias name: androiddebugkey
Creation date: Dec 31, 2018
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: C=US, O=Android, 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:[redacted]:A0:B8:68:0F: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

(Android Studio)



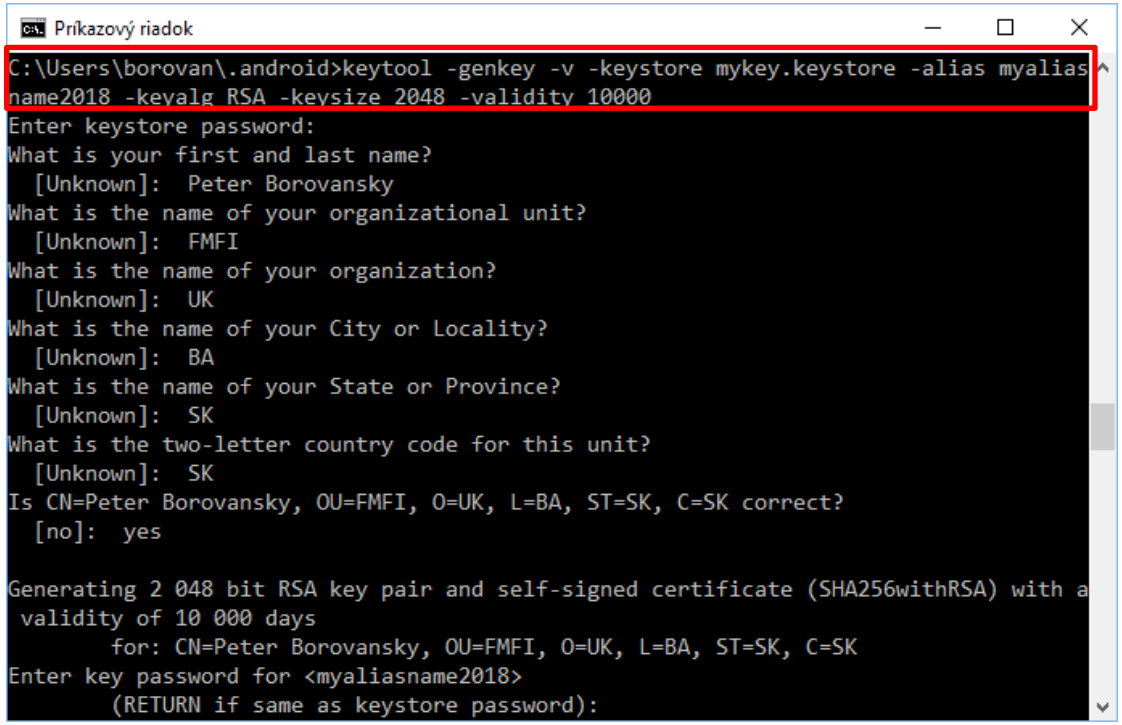
1.Open Android Studio
2.Open your Project
3.Click on Gradle (From Right Side Panel, you will see Gradle Bar)
4.Click on Refresh (Click on Refresh from Gradle Bar, you will see List Gradle scripts of your Project)
5.Click on Your Project (Your Project Name form List (root))
6.Click on Tasks
7.Click on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

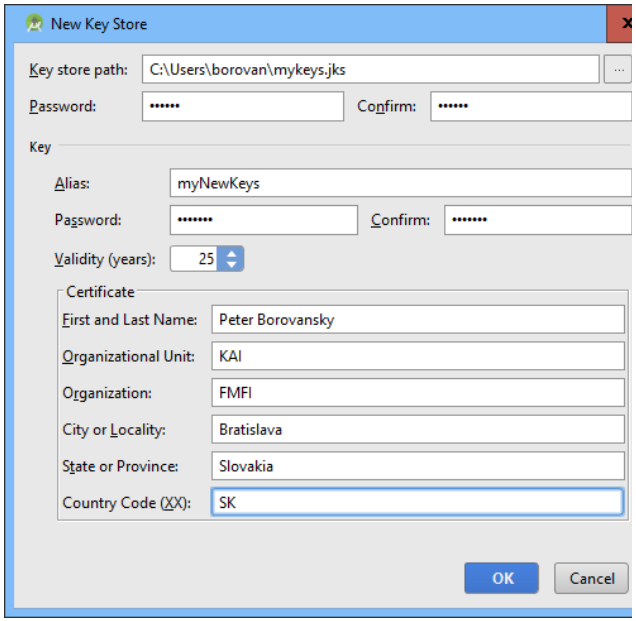


How to get the SHA-1 fingerprint certificate in Android Studio for debug mode?

Debug keystore
(manažment certifikátov)

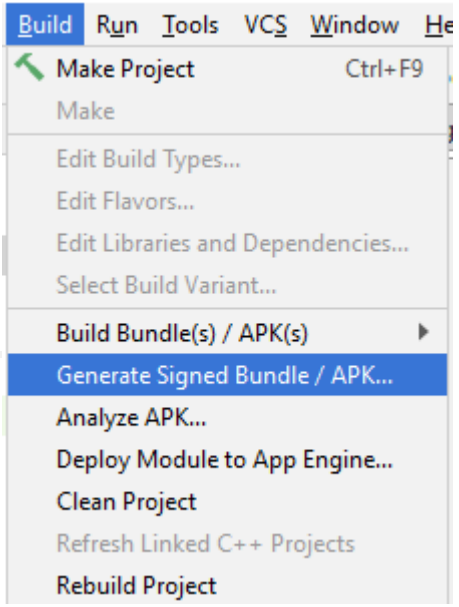
- **debug.keystore** obsahuje jeden alebo viac privátnych kľúčov (certifikátov).
- **debug.keystore** nám automaticky vyrobí Android-Studio pri inštalácii
- Android Studio nám automaticky podpíše každú apku kľúčom debug.keystore
- vieme vygenerovať vlastný keystore/certifikát, pomocou `keytool`, resp.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

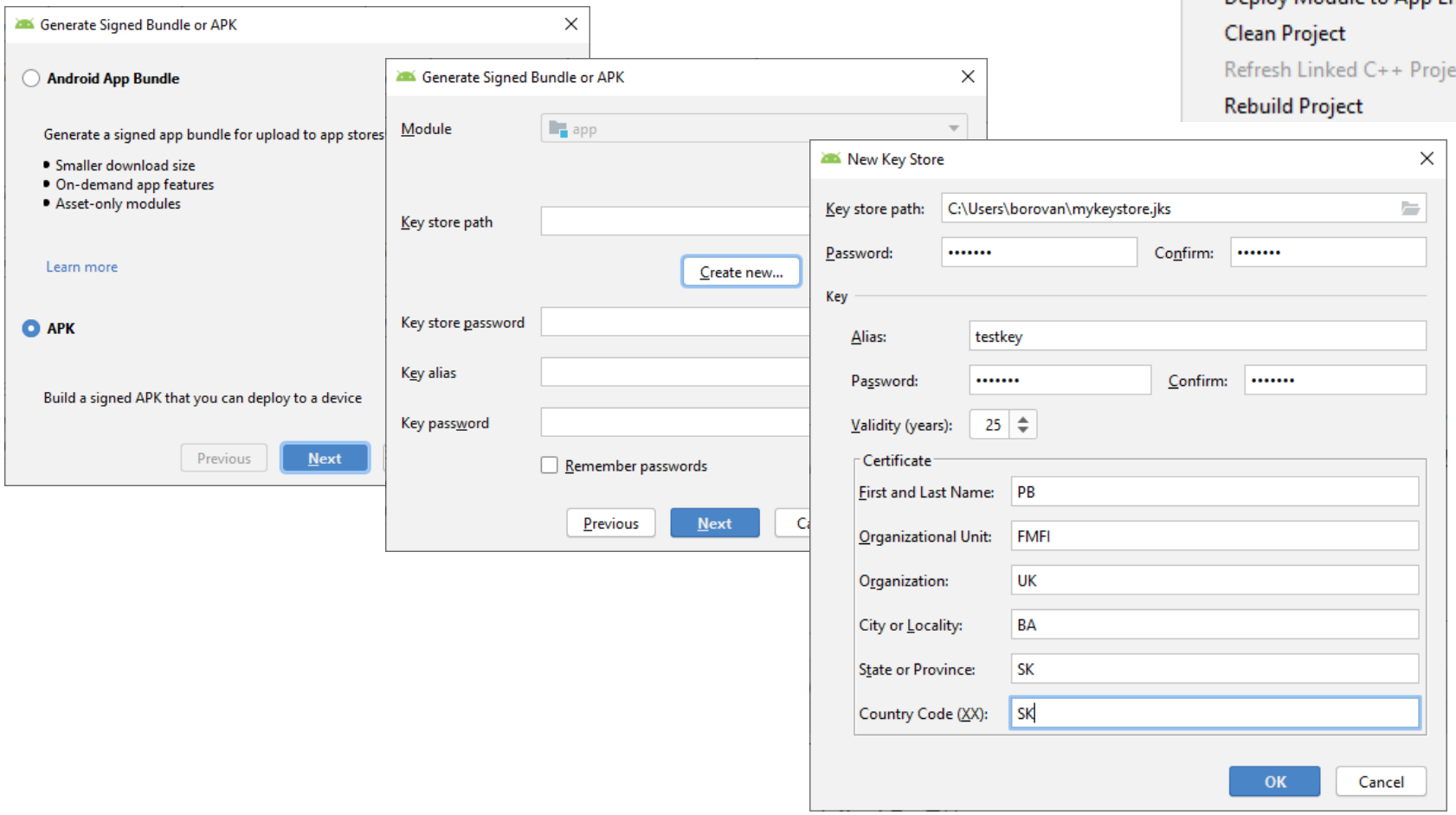




Generovanie kľúča

- začnite v Build/Generate Signed APK

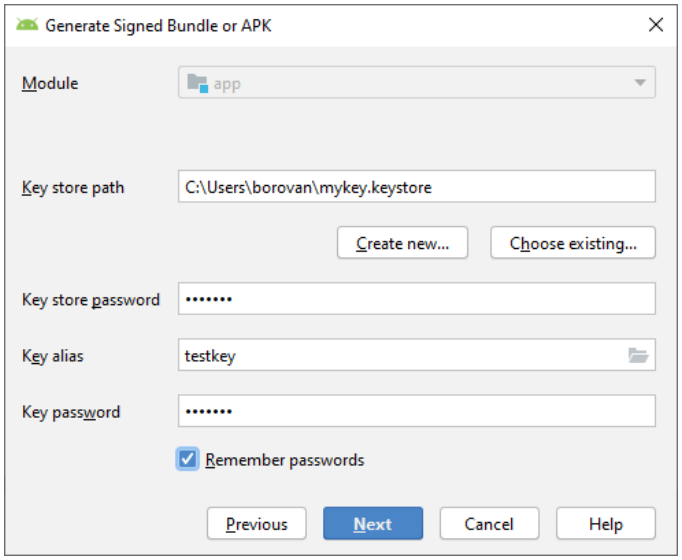


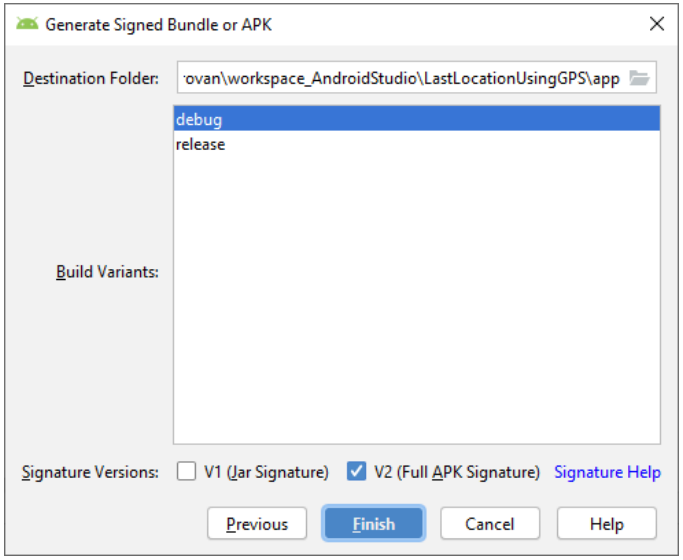


Podpisovanie aplikácie (debug)

(použitie certifikátov)

Vygenerovaný kľúč potom slúži na podpísanie .apk, napr. pomocou `jarsigner`





resp. AS automaticky podpisuje .apk pri builde

```
build.gradle, resp. signing-config.json:
signingConfigs {
    config {
      keyAlias 'myaliasname,
      keyPassword 'manager'
      storeFile file('C:/Users/borovan/mykey.keystore')
      storePassword 'manager'
    }
}
```

https://developer.android.com/studio/publish/app-signing.html#release-mode

Podpisovanie aplikácie (release)

(použitie certifikátov)

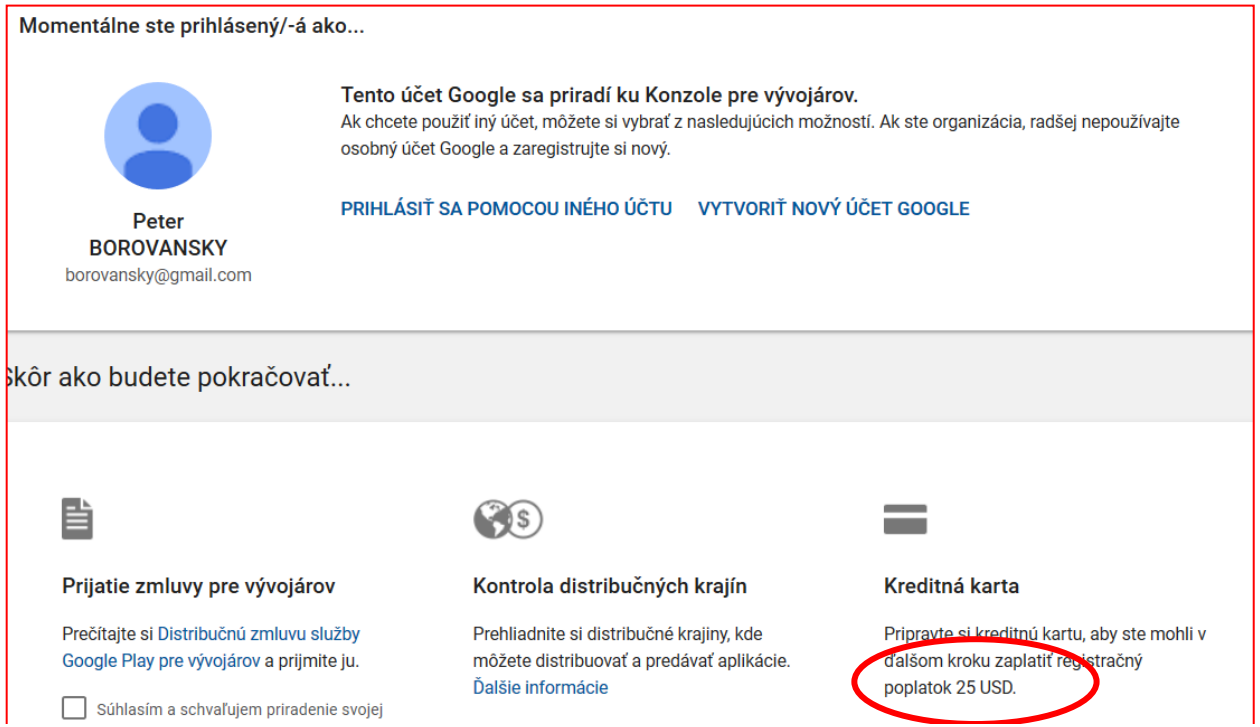
- Potrebujete release.keystore (návody):

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:

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



Google Maps API key V2

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

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

resp (alternatíva, 2016!):

- http://www.vogella.com/articles/AndroidGoogleMaps/article.html

alebo podľa mňa, veľmi názorný tutorial (step-by-step, 2017!):

- http://www.androidhive.info/2013/08/android-working-with-google-maps-v2/

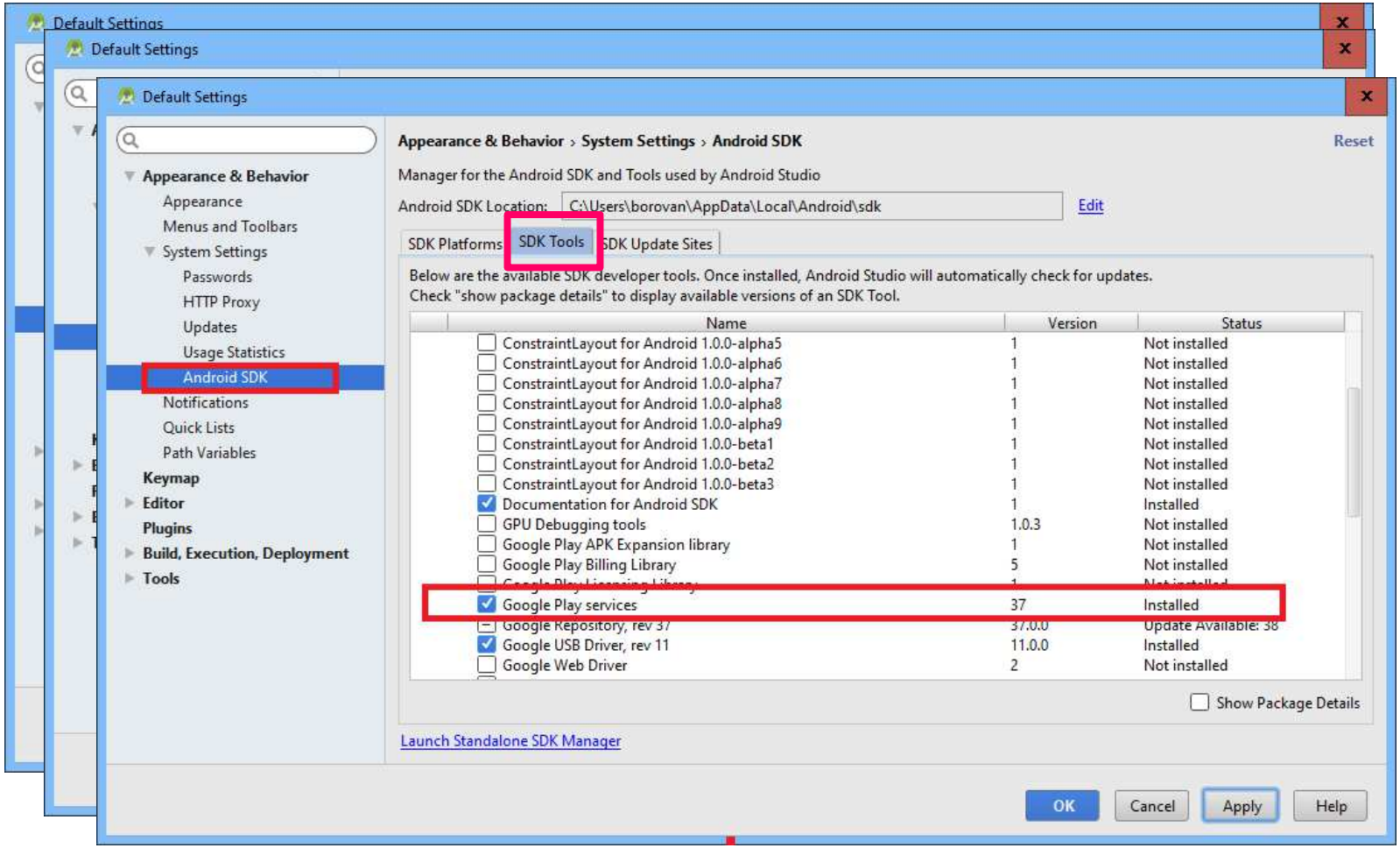
toto robí AS Automaticky☺

1. cez SDK manager doinštalovať Google Play Services,
2. pridať Google Play Services do vášho mapového projektu,
3. vygenerovať z vášho SHA1 + **package name** Google MAP API V2 kľúč
4. nastaviť všetky práva v AndroidManifest.xml (najlepšie skopírovať z príkladu)
5. vložiť vygenerovaný kľúč V2 do google_maps_api.xml ako meta-tag aplikácie

```
<string name="google_maps_key" ...>
   AIzaSyBsGY9grgC****l8gGz****njva8hmXSpQ
</string>
```

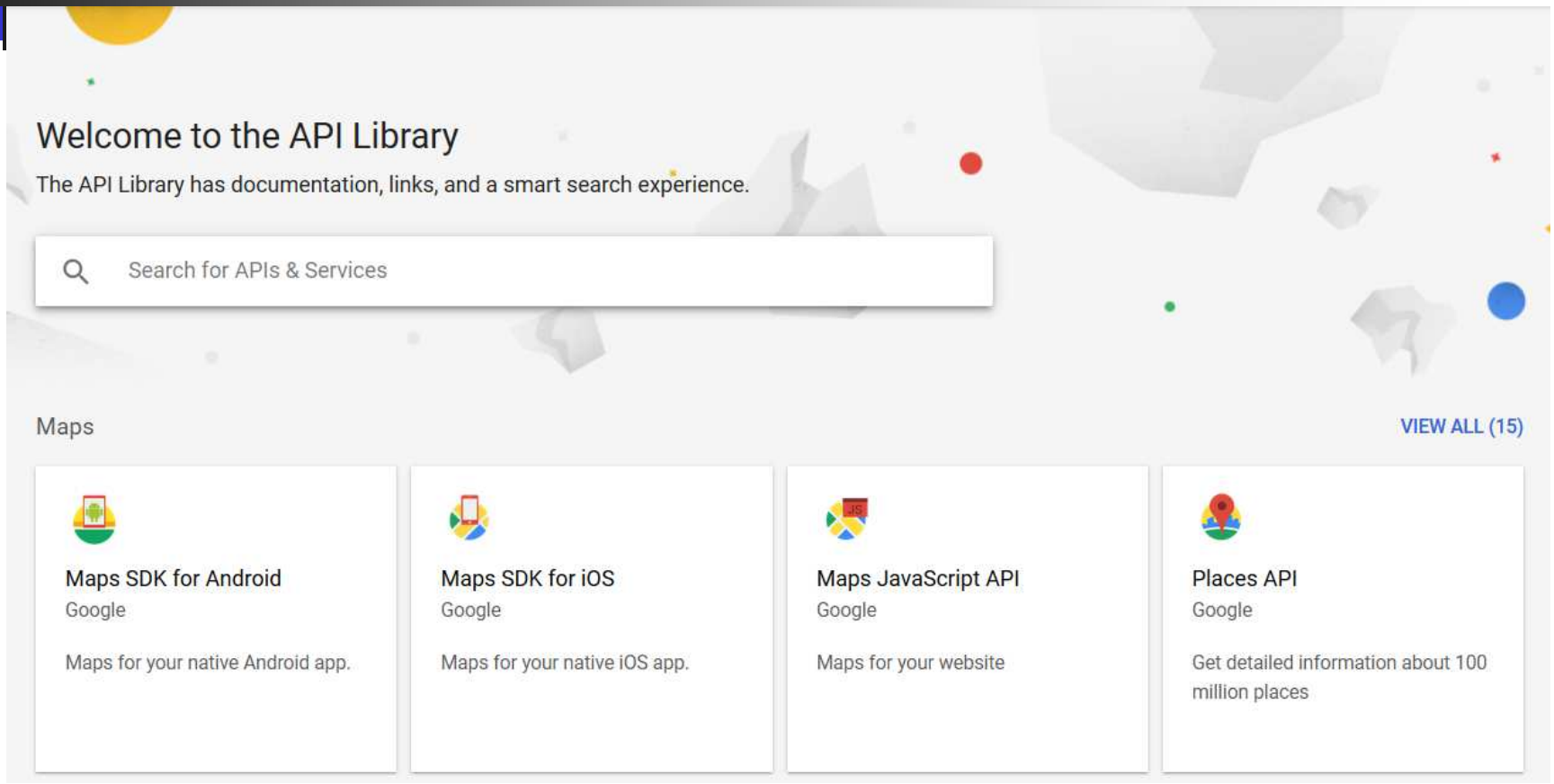
Google API/Google Play Services

Doinštaluje sa cez Window/Android SDK Manager

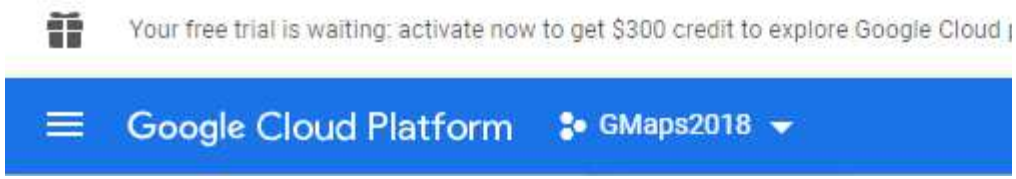


Google Developer Console

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



API Key V2

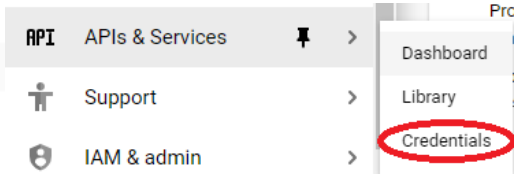


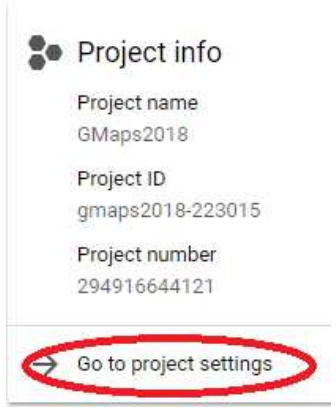
https://console.developers.google.com/apis/credentials/

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]8OEeC4QD1r8

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

API key pre Google Maps (V2) sa prideľuje pre dvojicu, alebo niekoľko dvojíc, (package name, SHA1-v prostredí, kde kompilujete)

Restrict usage to your Android apps (Optional)

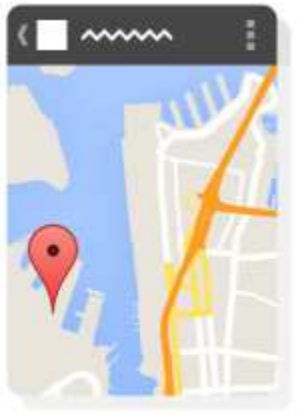
Add your package name and SHA-1 signing-certificate fingerprint to restrict usage to your Android apps

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	SHA-1 certificate fingerprint
pokus.example.com.myapplicationx	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

Maps v Android Studio



Google Maps Activity

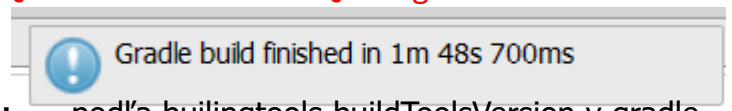
- inštalujte the Google Play services SDK
- vytvorte a Google Maps projekt
  File/New/New Project/**Google Maps Activity**
- získajte a Google Maps API key
  čítajte komentár a klinite na link v google_maps_api.xml
- vygenerujete link priamo do Google Developers Console

https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&keyType=CLIENT_SIDE_ANDROID&r=7A:94:75:11:DD:3D:57:[redacted]13:A0:B8:68:0F:67:F1%3Bcom.example.gmapfirstapp

- nakopírujte vygenerovaný Google Maps API key
  do súboru google_maps_api.xml

```
<resources>
    <string name="google_maps_key" translatable="false"
       templateMergeStrategy="preserve">
        YOUR_KEY_HERE                    ← AIzaSyC-_5QmNIi0v7favcJCfVZBQxl9RgcZKVM
    </string>
</resources>
```

- skompilujte (čakajte.....)/pustite projekt



podľa builingtools buildToolsVersion v gradle

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

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.`

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))
}
```

Project:GMAPFirstApp.zip

Layout s MapFragmentom

```
<fragment xmlns:android=http://schemas.an
    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)
    GooglemMap.MAP_TYPE_NONE)
```

Project:GMAPFirstApp.zip

Permissions

- AndroidManifest.xml

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

- Žiadanie povolenia v runtime

```
if (mMap != null) {
    val permission = ContextCompat.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 -> { ....
```

Project:GMAPFirstApp.zip

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`** – poloha ikony definovaná latitude+longitude
- **`Shapes`** – Polyline (lomená čiara), Polygon (n-uholník)
- **`UiSettings`** – nastavenie užívateľského rozhrania, napr. zoom-level, ...
- vrstva My Location – ak je povolená, zobrazuje sa button MyLocation, ktorý vycentruje mapu podľa aktuálnej polohy

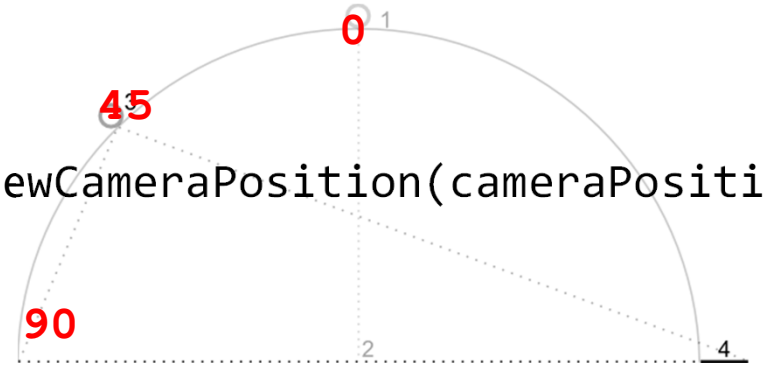
Marker/MarkerOptions

```
                                        // zobrazí moju polohu
mMap.isMyLocationEnabled = true         // button na mape
val mff = LatLng(48.151901, 17.068422) // 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
)
```

Project:GMapV2.zip

Camera

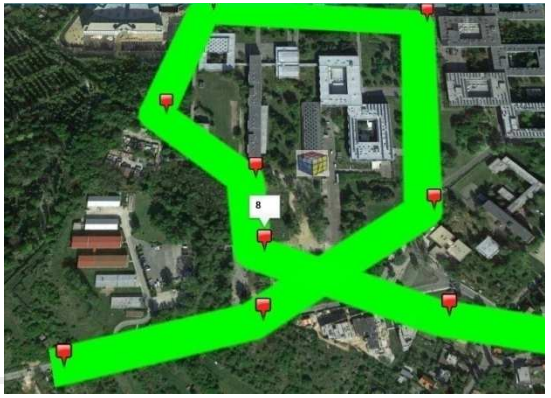
```
mMap.isBuildingsEnabled = true
mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(mff, 6))
mMap.animateCamera(CameraUpdateFactory.zoomIn()) //zoomOut
mMap.animateCamera(CameraUpdateFactory.zoomTo(13),2000,null)

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 0-90
    .build()
mMap.animateCamera(
   CameraUpdateFactory.newCameraPosition(cameraPosition))
```



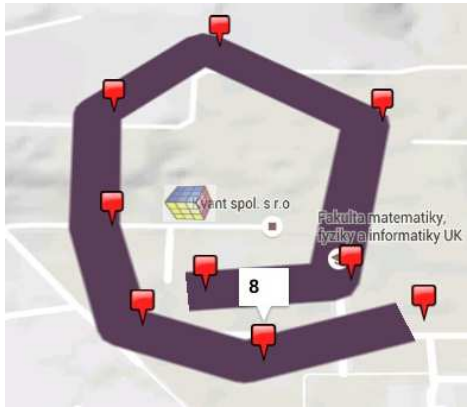
Porovnanie: ESB vs. FMFI

Project:GMapV2.zip

onMapClick, PolyLine



```
clickedLine = mMap.addPolyline(PolylineOptions())
                                  // ak chceme klikať do mapy
mMap.setOnMapClickListener {
    latlng ->   // onClickListener – nefunguje pri satelitných m.
        val clickedPoints = clickedLine.getPoints()
        clickedPoints.add(latlng)
        clickedLine.setPoints(clickedPoints)
        clickedLine.setColor(...)
        clickedLine.setWidth(...)

        mMap.addMarker(MarkerOptions()
          .title("" + clickedPoints.size())
          .icon(BitmapDescriptorFactory
                .fromResource(R.drawable.marker))
          .position(latlng))
```



Project:GMapV2.zip

GPS Location

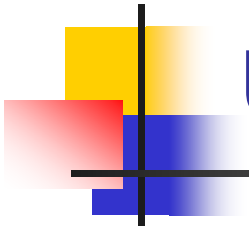
(prvá možnosť - GPS)

Project:GMapV2.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)
```



uiSettings

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

LocationServices API

(druhá možnosť - 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,
- používajú Google Play Services, do build.gradle doplnite:

```
dependencies {
    compile 'com.google.android.gms:play-services-location:16.0.0'
}
```

- do onCreate pribudne **GoogleApiClient**

```
mGoogleApiClient = new GoogleApiClient.Builder(MapsActivity.this)
        .addApi(LocationServices.API)      -- používa location services
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .build()
```

```
@Override
override fun onResume() {
    super.onResume()
    mGoogleApiClient.connect()
}
```

```
@Override
override fun onPause() {
  super.onPause()
  mGoogleApiClient.disconnect()
}
```

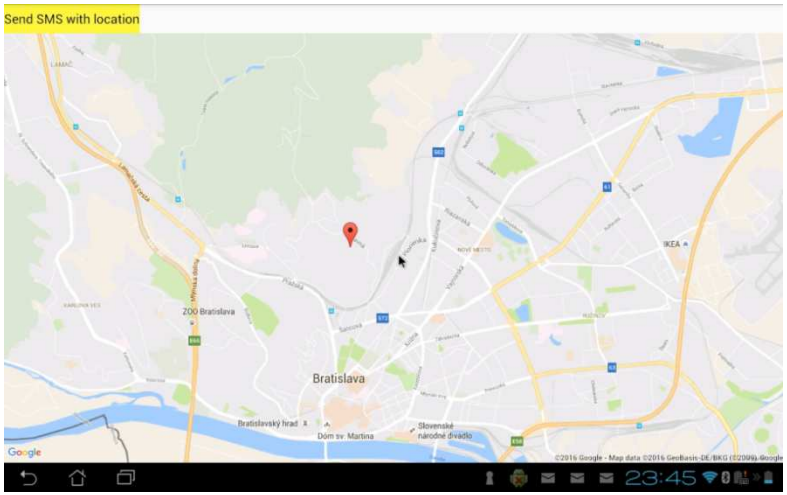
Project:GMapSOSLocation.zip

Posledná poloha

https://developer.android.com/training/location/change-location-settings.html

- Posledná poloha (synchronne):

```
val mLastLocation =
    LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient)
if (mLastLocation != null) {
    ...mLastLocation.getLatitude(),mLastLocation.getLongitude()...
}
```



- Poslanie Správy (SMS):

```
val intent = Intent(Intent.ACTION_VIEW)
intent.data = Uri.parse("sms:")
intent.putExtra("sms_body", "My location: $lat, $long")
startActivity(sendIntent)
```

- Presnosť: ???

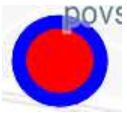
Project:GMapSOSLocation.zip

Zmenená poloha

https://developer.android.com/training/location/change-location-settings.html

- Posledná poloha (synchronne):

```
val mLastLocation =
    LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient)
if (mLastLocation != null) {
     ...mLastLocation.getLatitude(),mLastLocation.getLongitude()...
}
```



Zmená poloha (asynchronne):

- vyrobíme `LocationRequest` popisujúci naše potreby o polohe (ako `Criteria` pre LM):

```
val locRequest = LocationRequest.create()
    .setPriority(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
    .setInterval(5 * 1000/*ms*/) -- ako často chceme update polohy
    .setFastestInterval(2*1000/*ms*/) -- ako často vieme spracovať update
    polohy-pre prípad,že iná apka intenzívnejšie používa GPS
```

- vyrobený `LocationRequest` podhodíme LocationServices s *mGoogleApiClientom*

Project:GMapSOSLocation.zip

Zmenená poloha

https://developer.android.com/training/location/receive-location-updates.html

Zmená poloha (asynchronne): -- pokračovanie

```
override fun onConnected(bundle: Bundle?) {
...
  LocationServices.FusedLocationApi.requestLocationUpdates(
     mGoogleApiClient, locRequest, location -> {
         String msg = "Updated Location: " +
            ... location.getLatitude()
            ... location.getLongitude())
         }
   )
override fun onPause() { ...
   if (mGoogleApiClient.isConnected) {
      LocationServices.FusedLocationApi
                      .removeLocationUpdates(mGoogleApiClient, this)
      mGoogleApiClient.disconnect()
   }
}
```

Project:GMapSOSLocation.zip

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

Poloha vs. Poloha

(a máme dve polohy)

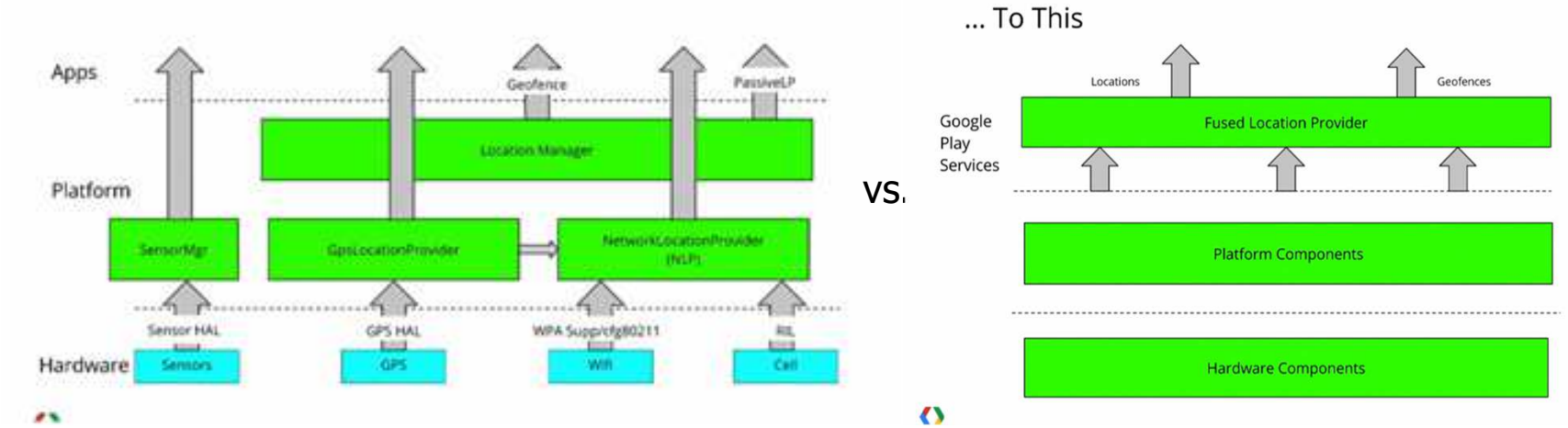
https://antoniohongkr.wordpress.com/2013/08/19/google-play-service-analysis-4-choice-between-google-play-location-service-and-android-location-service/

- android.location.LocationListener.onLocationChanged(Location)
- com.google.android.gms.location.LocationListener.onLocationChanged(...)

Dokumentácia hovorí:

The Google Play services location APIs are preferred over the Android framework location APIs (android.location) as a way of adding location awareness to your app. If you are currently using the Android framework location APIs, you are strongly encouraged to switch to the Google Play services location APIs as soon as possible.

FusedLocationProvider „fúzuje"=integruje všetky druhy určenia polohy:



Poloha a mockovanie
(LocationClient)

http://android.xsoftlab.net/training/location/location-testing.html

FusedLocation a mocking

- android.location – bez problémov
- LocationClient – dalo sa
- com.google.android.gms.location - ??? asi nejde …

public class

Summary: Nested Classes | Constants | Ctors | Methods | Inherited Methods | [Expand All]

LocationClient

extends Object

implements GooglePlayServicesClient

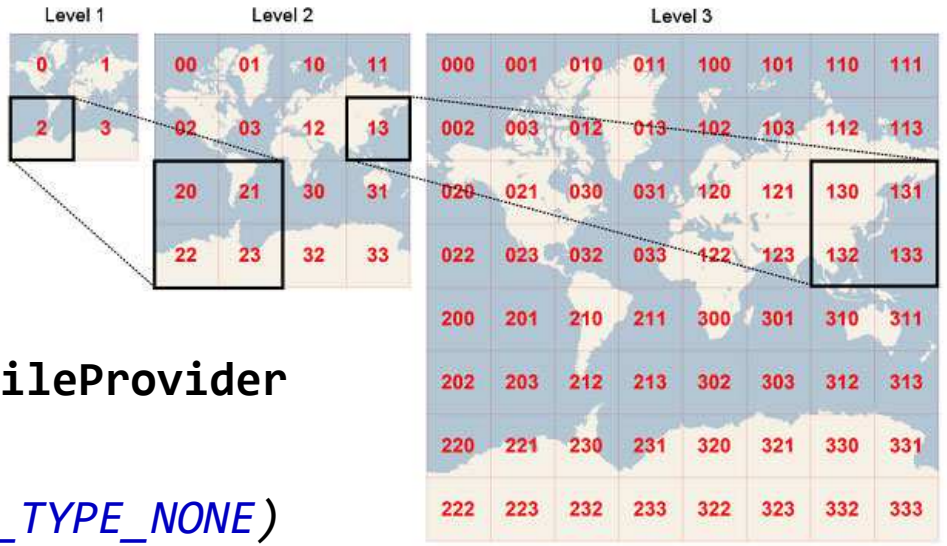
java.lang.Object
↳com.google.android.gms.location.LocationClient

This class is deprecated.
Use `LocationServices`.

void	setMockLocation(Location mockLocation) Sets the mock location to be used for the location provider.
void	setMockMode(boolean isMockMode) Sets whether or not the location provider is in mock mode.

GMaps Offline

(pre API key V2)



Offline Maps stačí urobiť vlastný **tileProvider**

a treba mať vlastné mapy ;-)

```
mMap.setMapType(GooglemMap.MAP_TYPE_NONE)
mMap.addTileOverlay(
    TileOverlayOptions().tileProvider(
        MyMapTileProvider(
            getResources().getAssets(),
        getApplicationContext())))
```

ale musíme naprogramovať TileProvider:

```
class MyMapTileProvider : TileProvider {
 ... // ktorá sprístupní vygenerované OpenStreetMap tiles
 fun getTile(x:Int, y:Int, zoom:Int) : Tile {...
```

Map Tiles

(pre API key V2)

```
/Androzic/tiles/osm/13/4484-2844, exist?:true
/Androzic/tiles/osm/13/4485-2843, exist?:true
/Androzic/tiles/osm/13/4484-2840, exist?:true
/Androzic/tiles/osm/13/4485-2841, exist?:true
/Androzic/tiles/osm/13/4483-2841, exist?:true
/Androzic/tiles/osm/13/4483-2844, exist?:true
/Androzic/tiles/osm/13/4484-2839, exist?:true
```

Project:MapV2_LocationClient_Offline.zip

```
class MyMapTileProvider : TileProvider {
 fun getTile(x:Int, y:Int, zoom:Int) : Tile {
      image = readTileImage(x, y, zoom)
     return Tile(TILE_WIDTH, TILE_HEIGHT, image)
  }
  fun readTileImage(x:Int, y:Int, zoom:Int) : Bitmap ...
  fun getTileFile(x:Int, y:Int, zoom:Int) : File {
   sdcard = Environment.getExternalStorageDirectory()
   tileFile =                    // mapy musia byť na SDkarte
       "/Androzic/tiles/osm/" + zoom + '/' + x + '-' + y
   file = File(sdcard,tileFile)
   Log.d("MapV2_tile", tileFile+",exist?:" + file.exists())
   return file
}
```

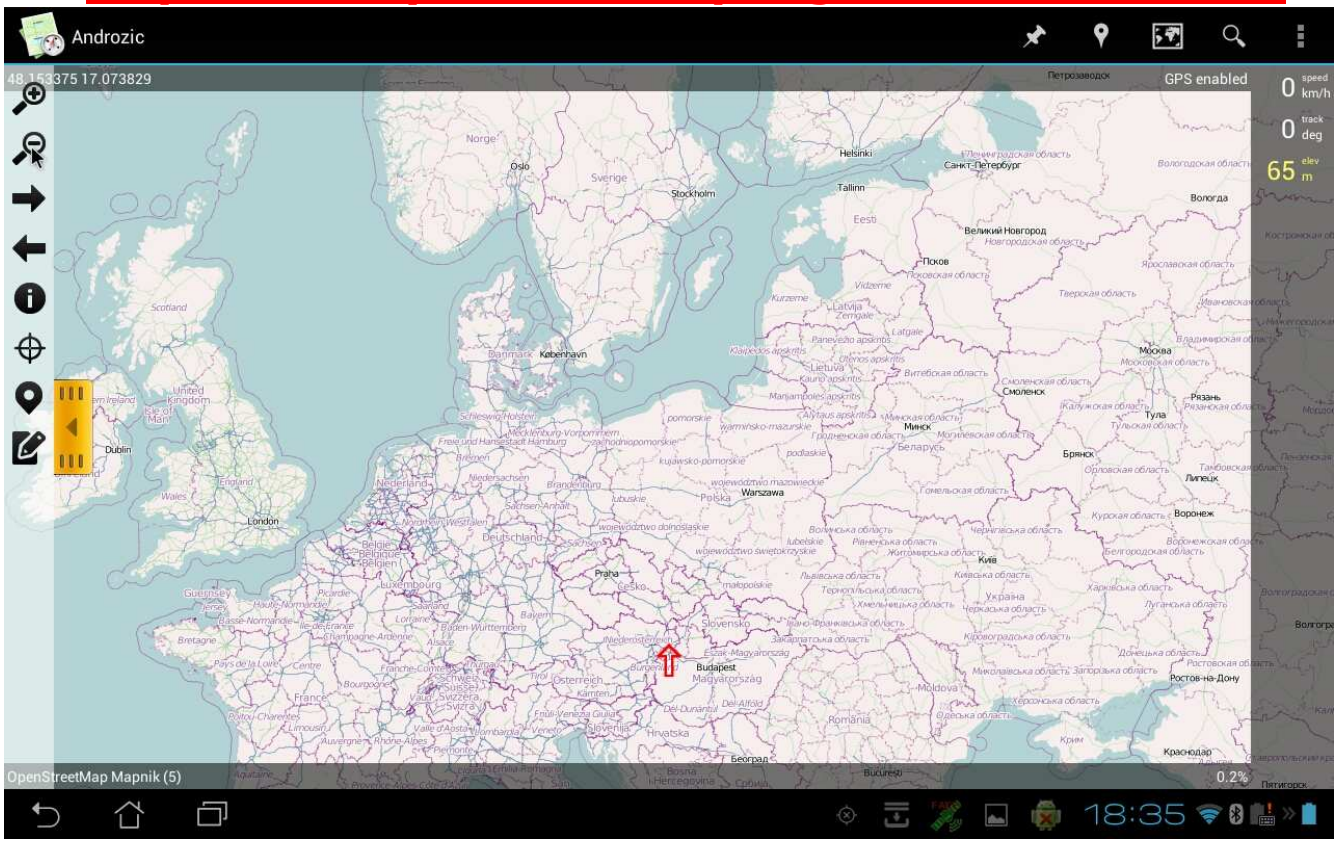
Open Street Maps (OSM)
Open Cycle Maps (OCM)

OSM (Generované pomocou OziMapper):

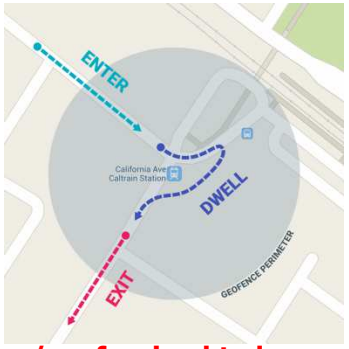
- Level2, 16 files
- Level3, 47 files
- Level4, 101 files
- Level5, 167 files
- Level6, 109 files
- Level7, 210 files
- Level8, 276 files
- Level9, 192 files
- Level10, 346 files
- Level11, 218 files
- Level12, 310 files
- Level13, 299 files
- Level14, 3427 files
- Level15, 226 files
- Level16, 72 files
- Level17, 40 files

http://download.mapsforge.org/maps/europe/

http://wiki.openstreetmap.org/wiki/Tile_servers



Geofencing



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

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

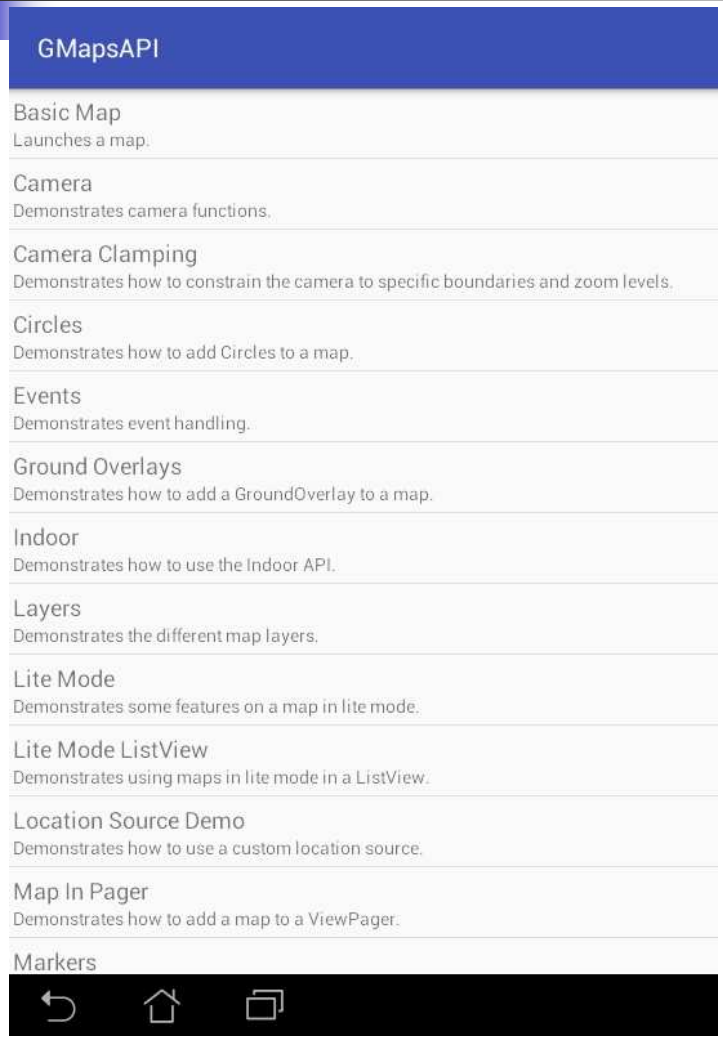
```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<application
      <service android:name=".GeofenceTransitionsIntentService"/>
<application/>
```

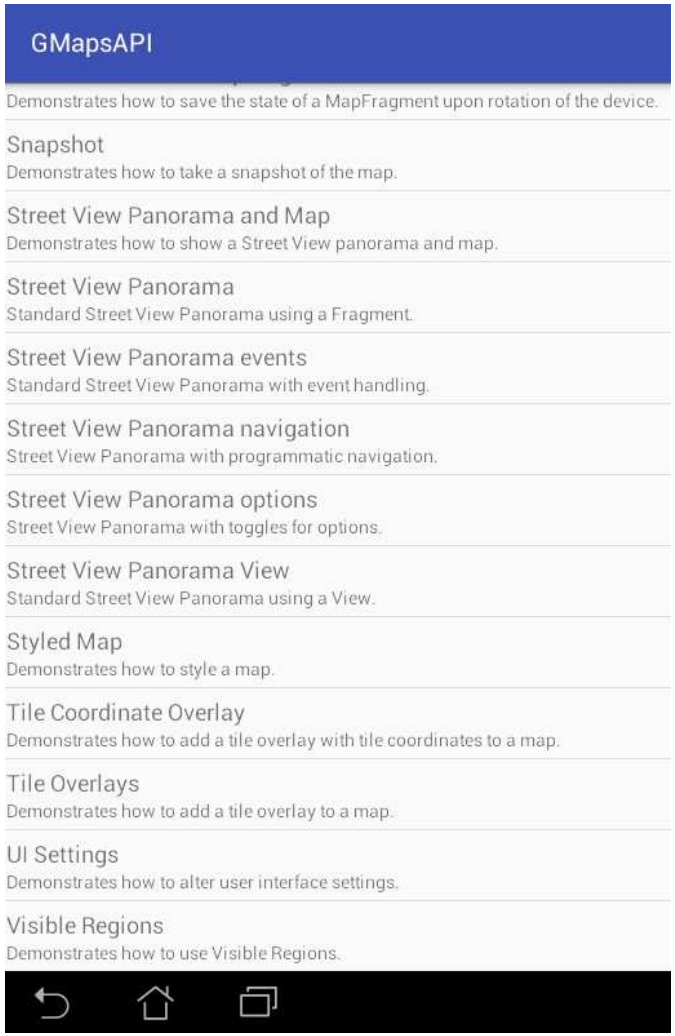
Kruhová ohrada (zoznam ohraničený na 100 oblastí):

```
mGeofenceList.add(new 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())
```

Demo pre inšpirácie

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





Detekcia aktivity

(ActivityRecognition.API)

```
= GoogleApiClient.Builder(context)
        .addApi(ActivityRecognition.API)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .build()
```

Metódy:

`.getType()-IN_VEHICLE/ON_BICYCLE/ON_FOOT/RUNNING/STILL/WALKING/…`

`.getConfidence()  0..100`

```
D/ActivityRecogition: onConnected
E/ActivityRecogition: onHandleIntent
E/ActivityRecogition:  Still: 100
E/ActivityRecogition: onHandleIntent
E/ActivityRecogition:  Still: 100
E/ActivityRecogition:  On Foot: 92
E/ActivityRecogition:  Running: 87
E/ActivityRecogition:  On Bicycle: 8
E/ActivityRecogition:  Walking: 5
```

Project:ActivityRecognition.zip

Detekcia aktivity

(ActivityRecognition.API)

```
Intent intent =
    new Intent(this,ActivityRecognizedIntentService.class );
PendingIntent pendingIntent =
    PendingIntent.getService(this,0,intent,
    PendingIntent.FLAG_UPDATE_CURRENT );
ActivityRecognition.ActivityRecognitionApi
    .requestActivityUpdates( mApiClient, 1000, pendingIntent );

public class ActivityRecognizedIntentService extends IntentService {
    protected void onHandleIntent(Intent intent) {
    if(ActivityRecognitionResult.hasResult(intent)) {
      ActivityRecognitionResult result =
      ActivityRecognitionResult.extractResult(intent);
      String act = "";
      for(DetectedActivity activity:result.getProbableActivities()){
        switch( activity.getType() ) {
          case DetectedActivity.IN_VEHICLE: {
             act += " In Vehicle: " + activity.getConfidence();
             break;
```

Project:ActivityRecognition.zip