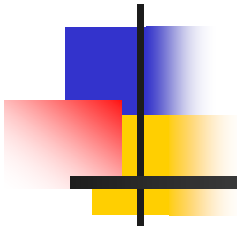


Android - poloha



Peter Borovanský
KAI, I-18

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

borovan 'at' ii.fmph.uniba.sk

- **GSM**
- **WiFi**
- **GPS**
 - Location Provider,
 - ProximityListener,
 - Geocoder

Mapy



Ako zistiť, kde sa nachádzam ?

... resp. kam smerujem...

Ak pominieme rady „starých mám“ a ešte starších moreplavcov, tak máme:

- satelity (technológia GPS a jej modernejšie klony)
- mobilnú sieť (technológia GSM, ...)
- siete (veľmi) krátkeho dosahu (napr. WiFi, Bluetooth, RFID, NFC)

Načo nám slúži/môže slúžiť informácia o polohe ?

- nájdeme sa, resp. nájde nás to, na mape ☺
- informácia, čo je okolo, POI – points of interest (v kategóriach záujmov), kam na obed, do kina, ...,
- navigácia: ako sa mám dostať na iné miesto,
- hľadanie pokladov (tzv. geocaching),
- cielená reklama ☹
- GPS tracking ☺
- SOS calling ☺

Týmto by sme mohli dnes začať aj skončiť

```
lateinit var lm: LocationManager
...
lm = getSystemService(Context.LOCATION_SERVICE) as LocationManager

val provider = LocationManager.GPS_PROVIDER    // GPS locator
              = LocationManager.NETWORK        // GSM locator

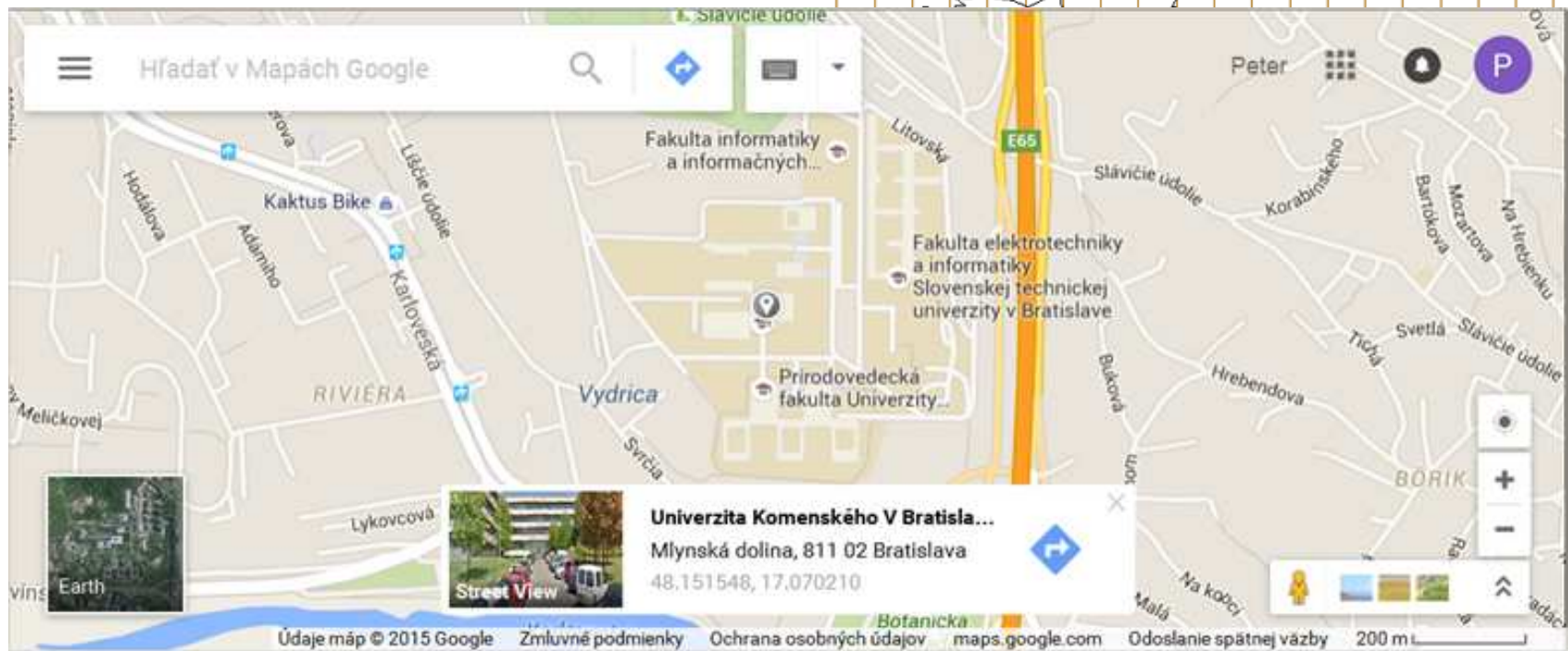
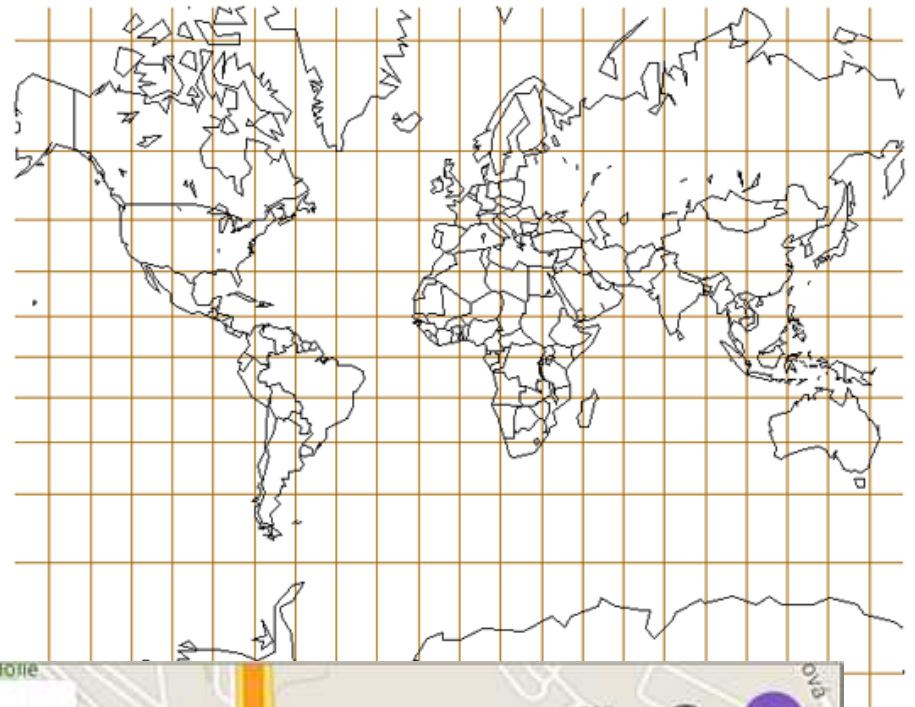
val loc : Location = lm.getLastKnownLocation(provider)

loc.latitude      - zem.šírka
loc.longitude     - zem.dĺžka
loc.altitude      - nadmorská výška, elevácia
loc.accuracy      - presnosť nameraného údaje
loc.provider      - info o provideri polohy (GPS/GSM)
loc.speed         - rýchlosť v m/s (len ak sa hýbeme)
loc.bearing       - azimut, ale musíme sa hýbať !!
loc.time          - milisek od 1970
```

Prednáška je o veciach súvisiacich s polohou

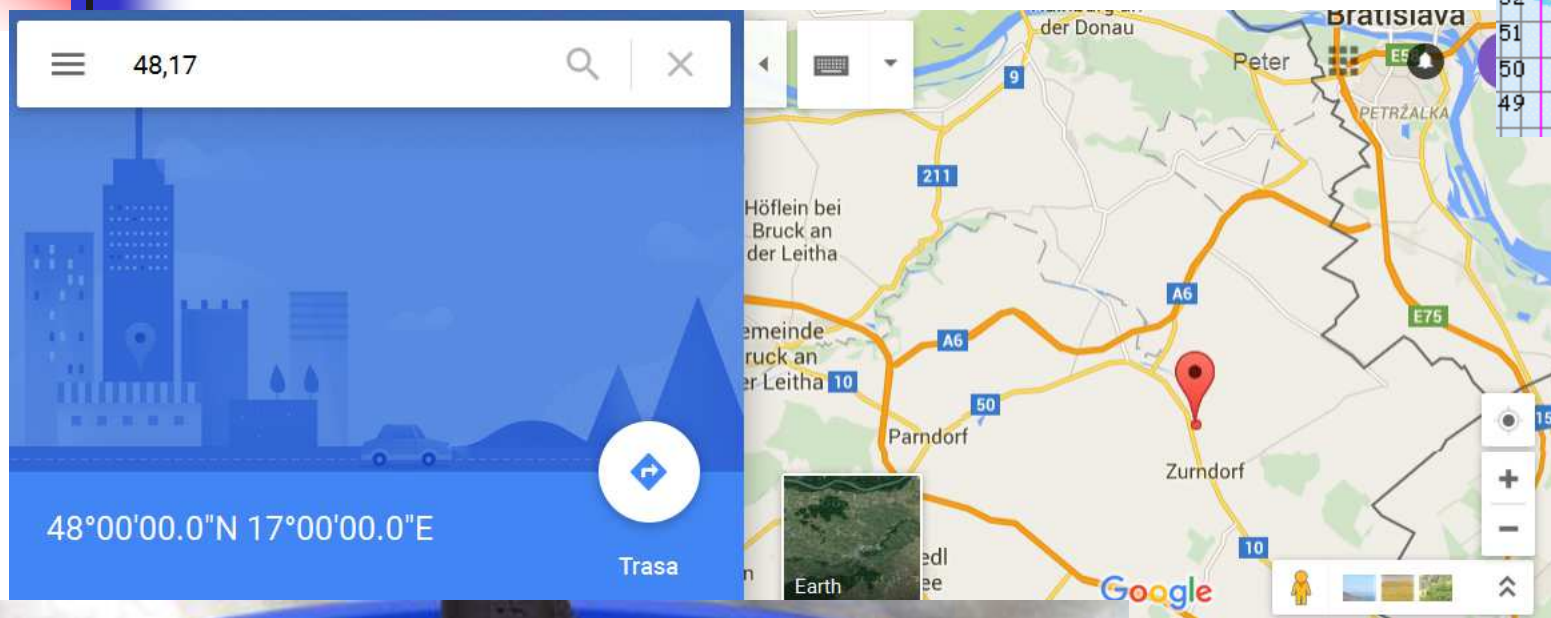
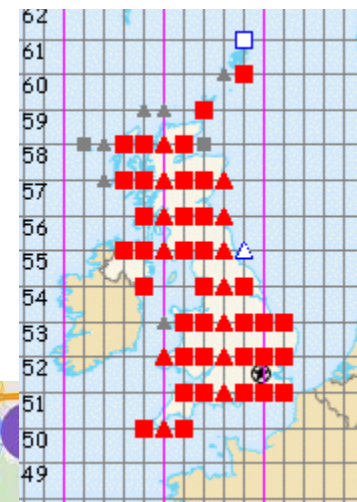
Ujasnenie

- Latitude – zem.šírka
- Longitude – zem.dĺžka
- <https://www.google.sk/maps/@48.151019,17.0707008,15z?hl=sk>



Confluence-Point

(mrežový bod - Gattendorf)

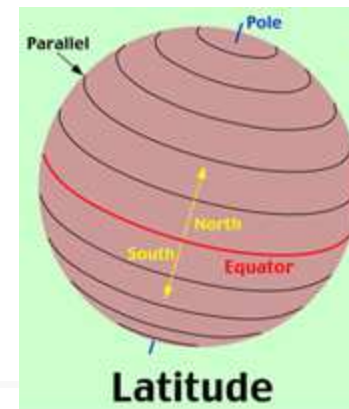


Koľko je slovenských
mrežových bodov ??

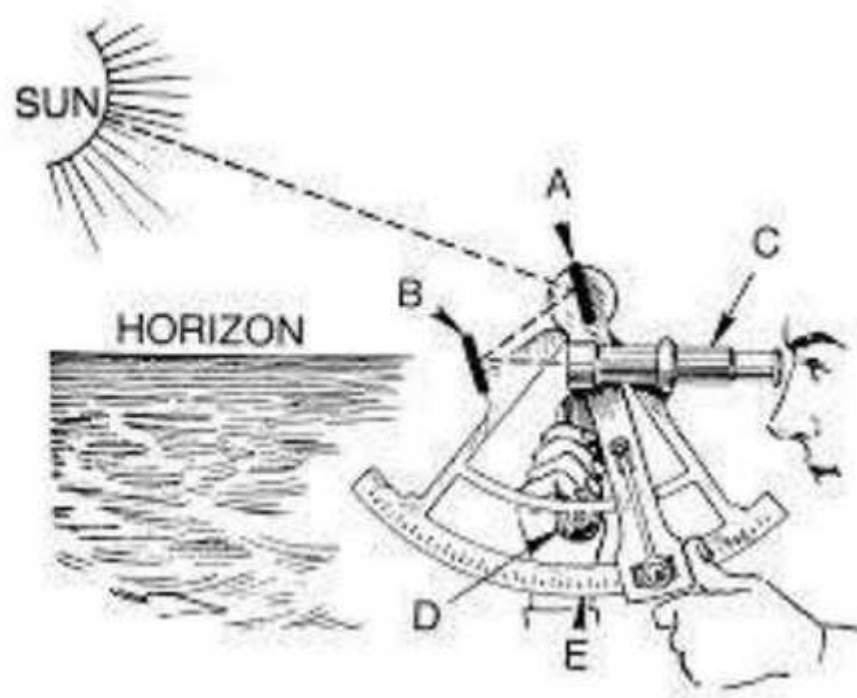
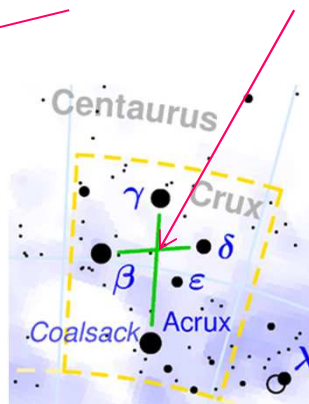
<http://www.confluence.org/country.php?id=66>

Zemepisná šírka

(latitude, 0-90°N, 0-90°S, my sme 48... °N)



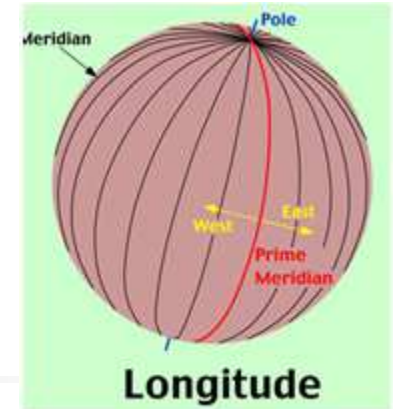
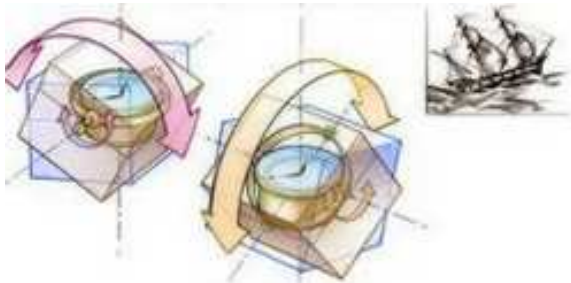
Merali uhol medzi horizontom a význačným bodom na oblohe, Slnko, Polárka (alias Severka), Južný kríž... a mali na výpočet tabuľky



Zemepisná dĺžka

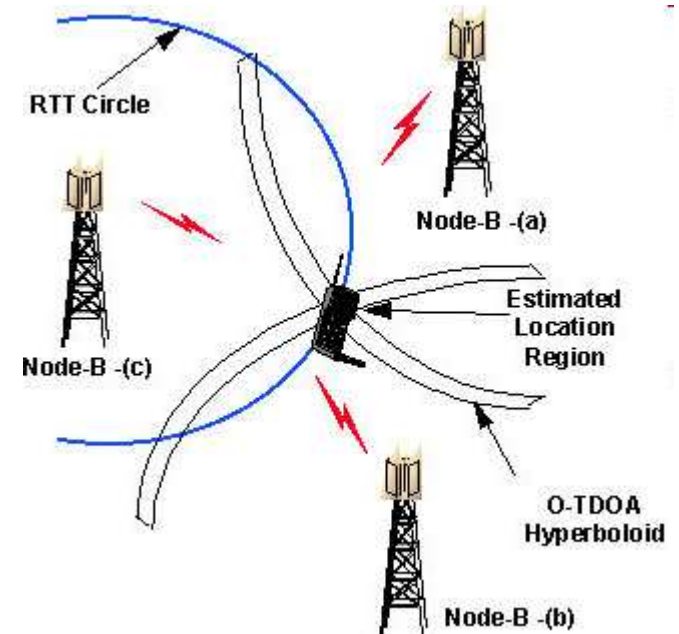
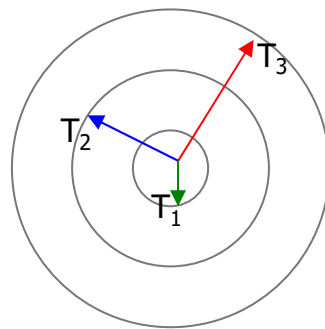
(longitude, 0-180°E, 0-180°W, my sme 17... °E)

- aby sme ju určili z hviezd, potrebujeme presný čas, resp. vedieť, koľko hodín je na Greenwich-i, lebo to súvisí s natočením Zeme
- problém sa redukuje na problém presného času
- kým na súši sa dajú použiť kyvadlové hodiny, na mori nie veľmi...



Lokalizácia (dnes) bez GPS

- mobilné zariadenie komunikuje prostredníctvom jedného (asi najbližšieho) vykrývača, to určuje okruh s presnosťou závislou od hustoty vykrývačov,
- triangulácia – z viacerých podobných informácií a polohy vykrývačov v dosahu zložíme presnejšiu aproximáciu polohy,
- time of arrival – vykrývače v rôznej vzdialenosti dostanú informáciu z mobilného zariadenia v rôznych časoch T_1 , T_2 , T_3 . Keďže signál sa šíri $\sim 300.000.000 \text{ m/s} = 300 \text{ km/ms}$



Meranie času

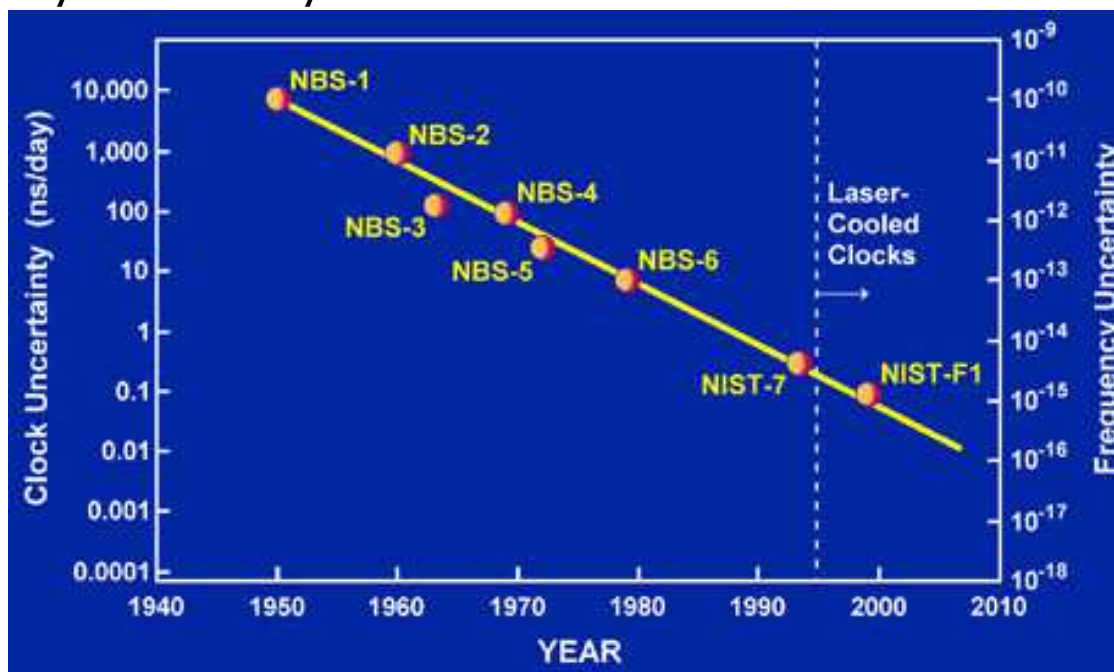
s akou presnosťou to vieme

$\sim 300.000.000 \text{ m/s} = 300 \text{ km/ms} = 30 \text{ cm/ns}$

chyba atómových hodín v ns za deň



100m/10s=
10m/1s=
10cm/0.01s
1cm/1ms



2013, the uncertainty reduced to cca 3×10^{-16} , (+/-) 1s za 100 million rokov !

Lokalizácia z GSM signálu



Nepresná informácia o polohe (mesto/mestská časť) sa dá získať z GSM siete

```
val tm = getSystemService(Context.TELEPHONY_SERVICE) as  
        TelephonyManager
```

```
tm.networkOperator - operátor = (mcc+mnc): String mcc+mnc
```

- `mcc` (mobile country code) = `networkOperator.substring(0, 3)`
- `mnc` (mobile network code) = `networkOperator.substring(3)`

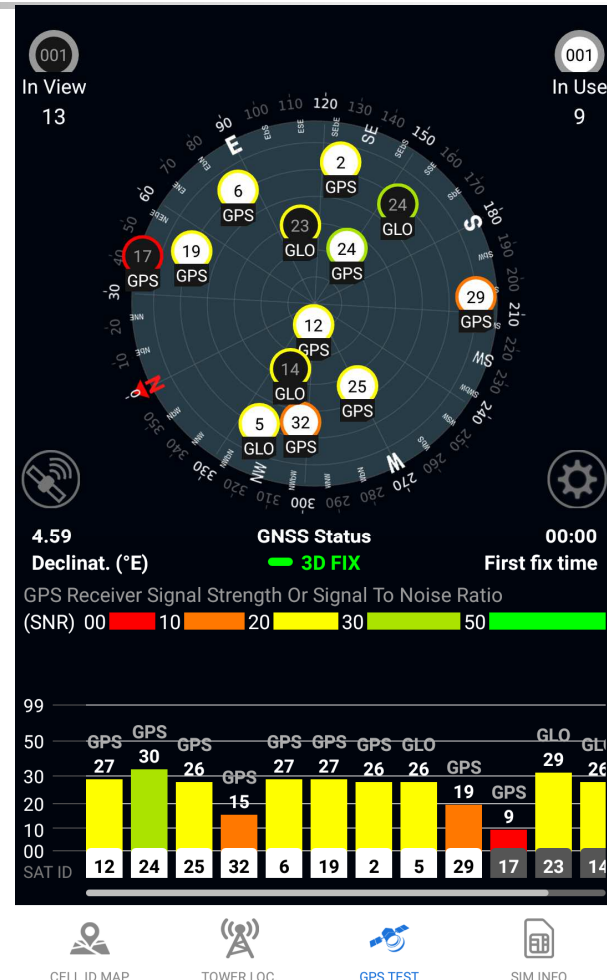
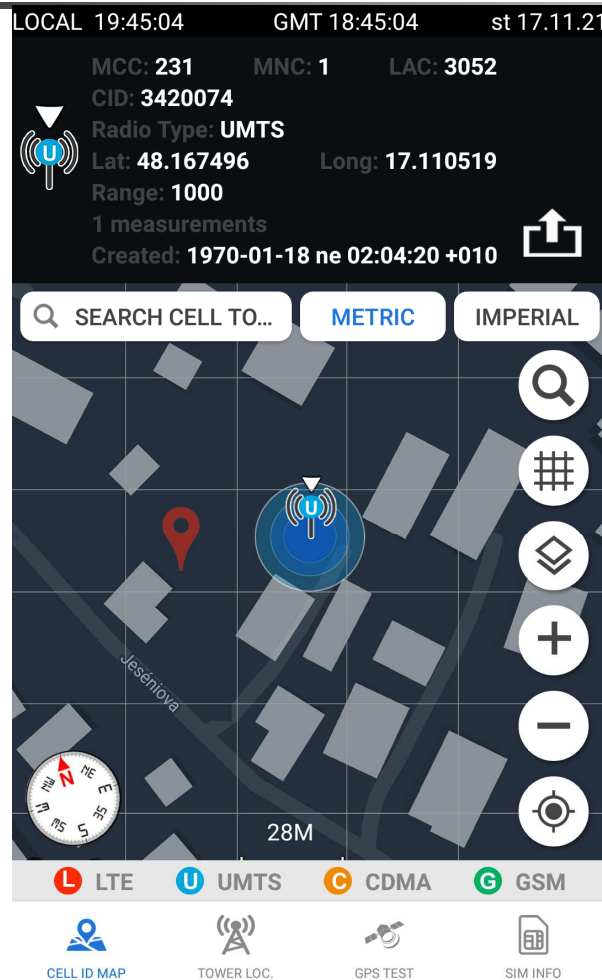
```
val gsmloc = tm.cellLocation as GsmCellLocation
```

- `gsmloc.cid` – buňka (Cell ID)
- `gsmloc.lac` – local area code

databázy GSM buniek (cid, lac, operátor)->(lat,long), nie sú úplne „free“, up-to-date

- http://en.wikipedia.org/wiki/Cell_ID
- <https://developers.google.com/maps/documentation/business/geolocation/>
- <http://locationapi.org/>
- <https://portal.combain.com/>

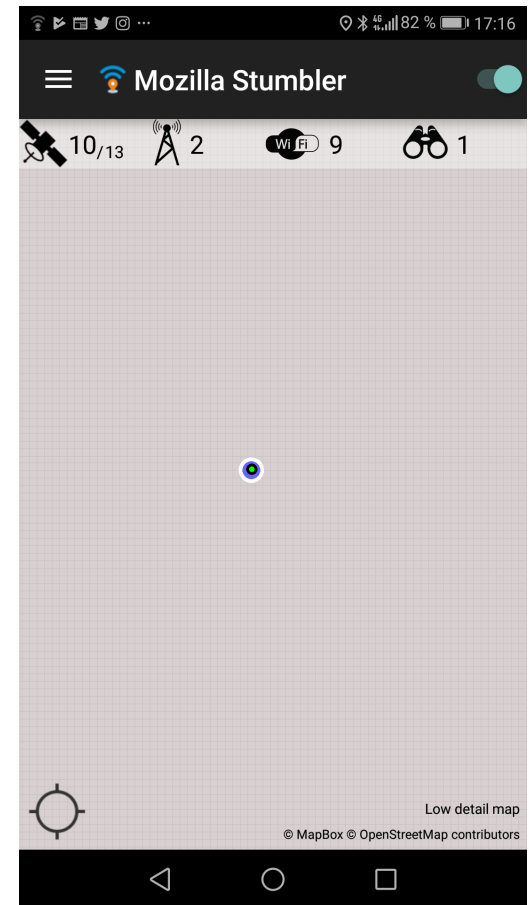
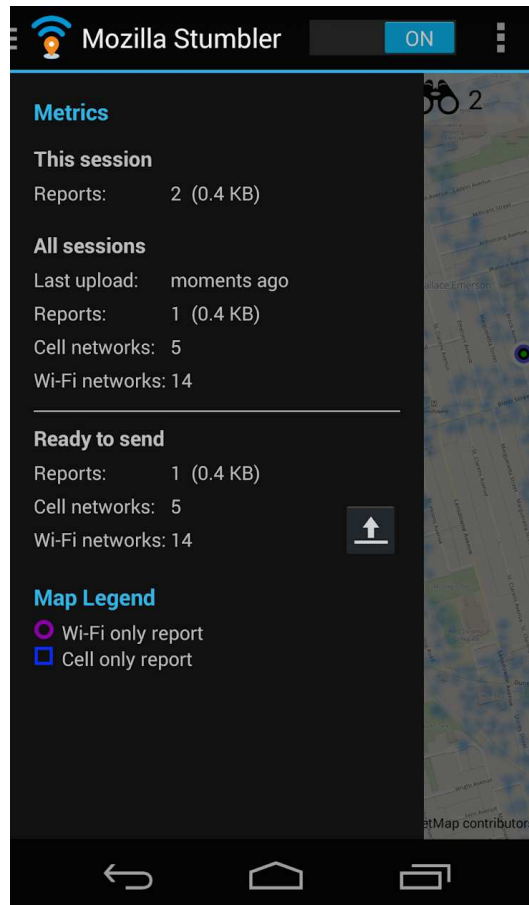
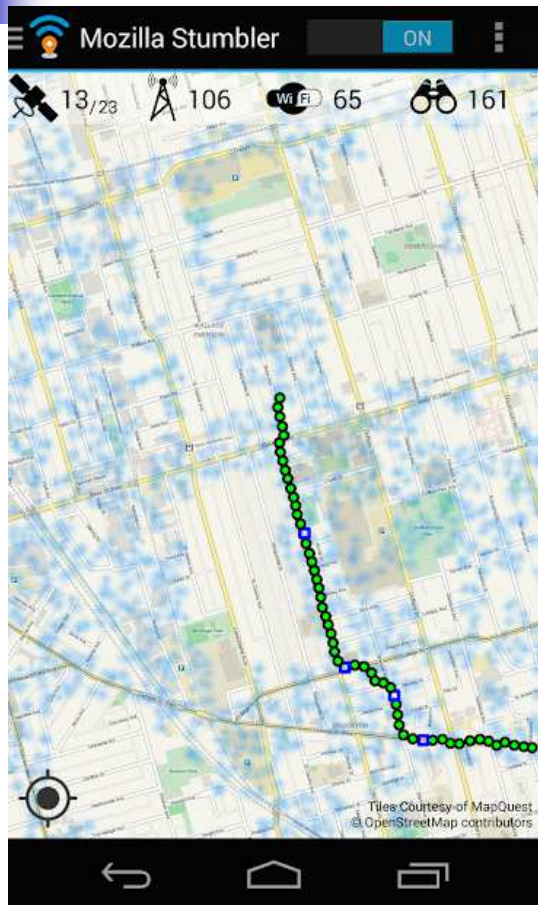
Cell & Net towers World Live map Signal and Speed



<https://play.google.com/store/apps/details?id=com.ekik.celltowers&gl=AT>

Mozilla Location Service

(Mozilla Stumbler)



Mozilla Stumbler was **retired on February 8, 2021**. This code works on Android 9, but not Android 10 or later.

<https://github.com/mozilla/MozStumbler>

https://en.wikipedia.org/wiki/Mozilla_Location_Service



```
D/MyGSMLocation(19361): gsm cid: 396517
D/MyGSMLocation(19361): gsm lac: 1001
D/MyGSMLocation(19361): operator:23102
D/MyGSMLocation(19361): network: 23102
D/MyGSMLocation(19361): mcc: 231
D/MyGSMLocation(19361): mnc: 2
```

- zaregistrujete sa napr. na free trial, max. 50 requests/day, resp. 1500 requests
- dostanete kľúč (token), 95b2941777892d (keď toto čítate, možno neplatí ☹)
- s8hmu5kq716gawzkczfz

<http://locationapi.org/site/page?view=apiv2>

Request: 1 cell | 3 cells | 7 cells

Response:

mcc:
231 SK

mnc:
01 Ora
02 Tele
04 TMO
06 O₂

```
1 {
2   "token": "1445573628",
3   "mcc": 231,
4   "mnc": 2,
5   "cells": [{
6     "cid": 396517,
7     "lac": 1001,
8     "signal": -60,
9     "tA": 13
10  }]
11 }
```

```
1 {
2   "status": "ok",
3   "balance": 45,
4   "lat": 48.16802,
5   "lon": 17.11049,
6   "accuracy": 1063,
7   "message": "Accuracy is in BETA!"
8 }
```

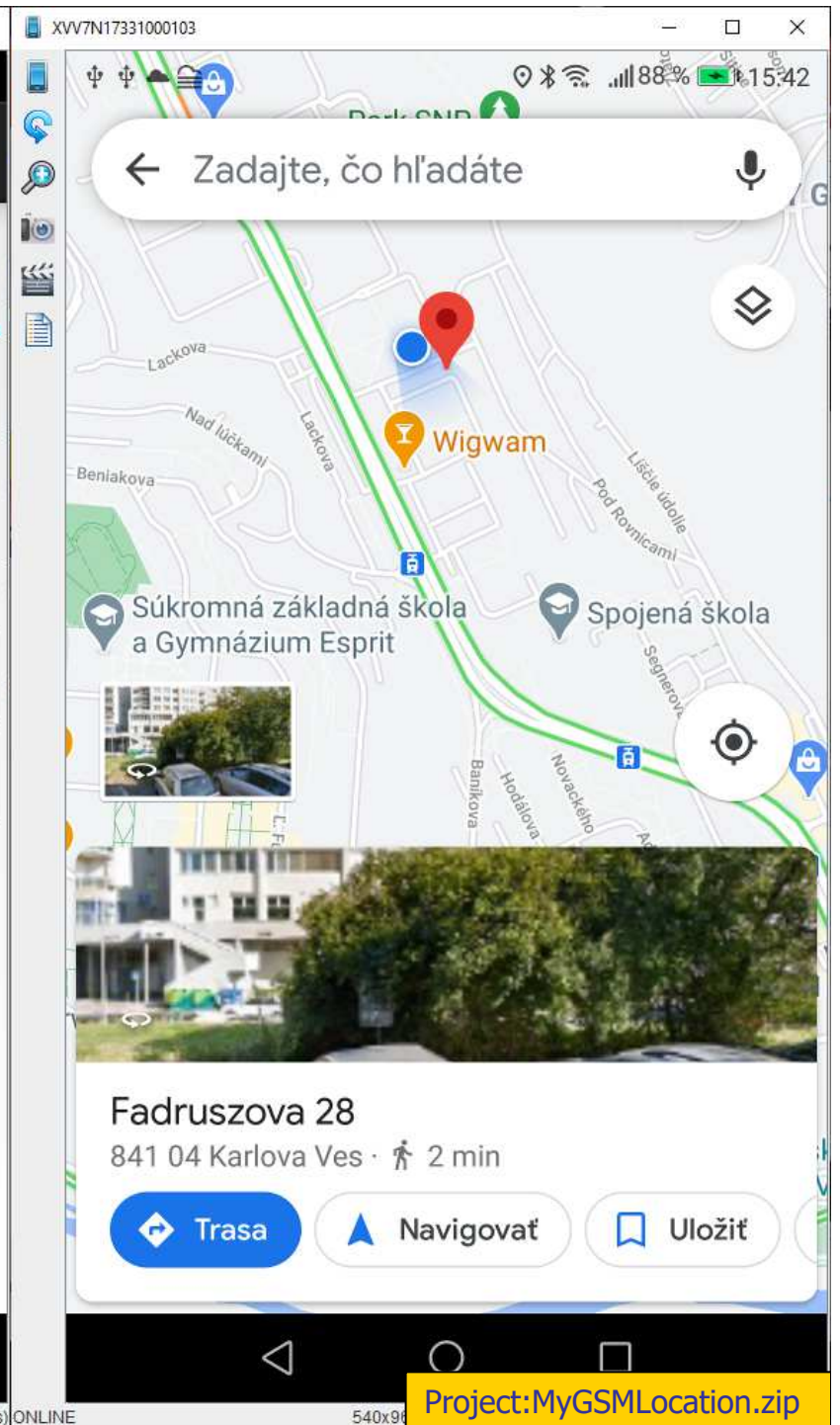
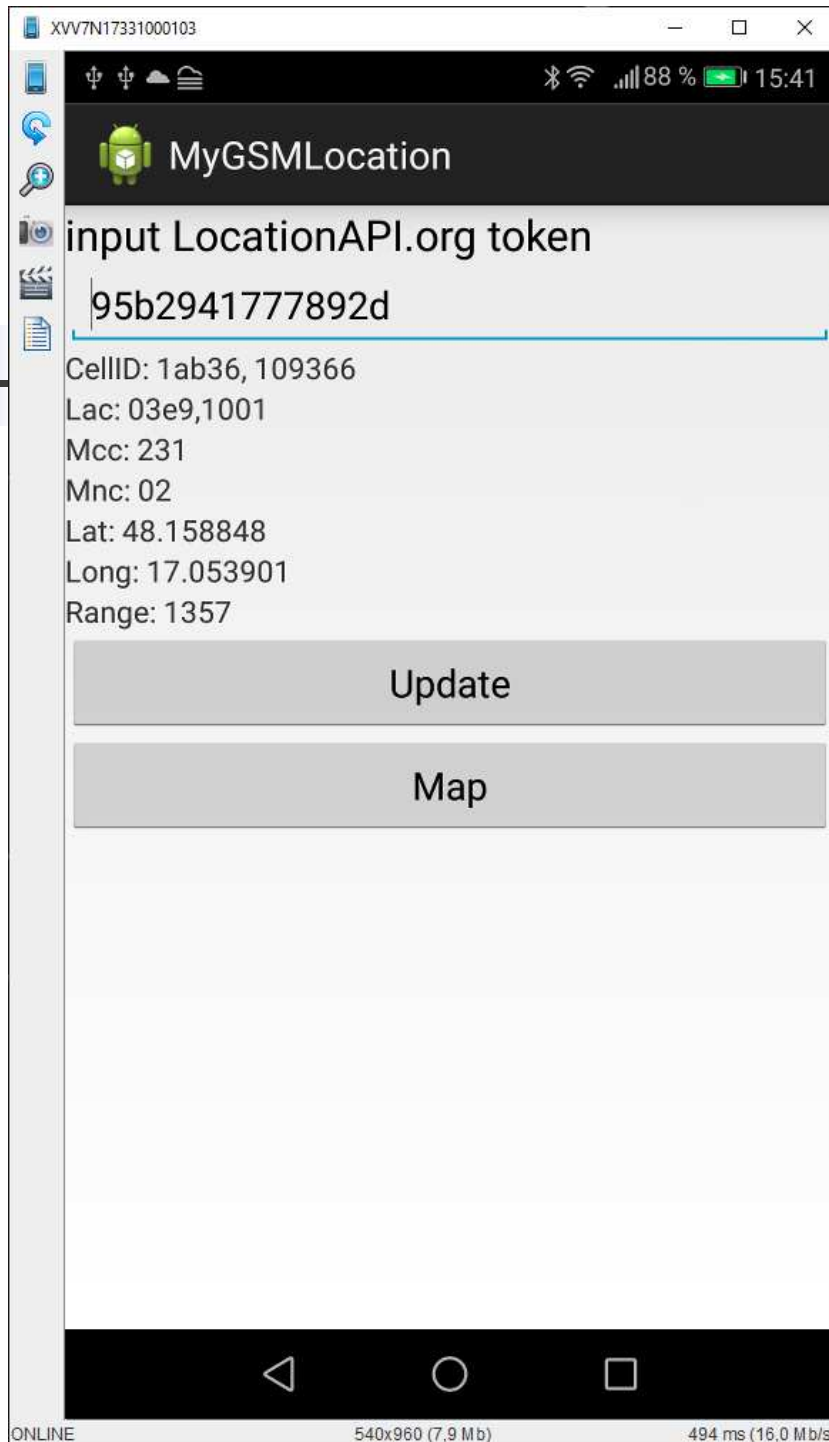
API v2 Documentation

1. [Usage](#)
2. [Test it out](#)
3. [Request body](#)
4. [Response body](#)
5. [Example Script - PHP](#)
6. [Example Script - Python](#)

Usage

Requests are sent using POST to the following url:

<http://locationapi.org/v2/process.php>



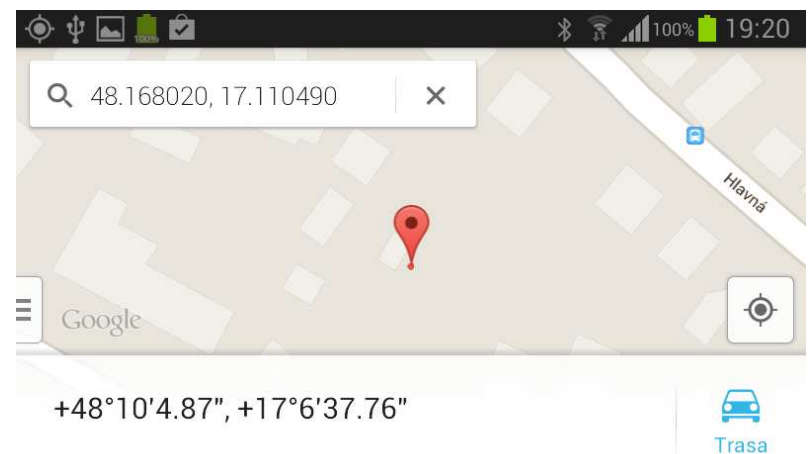
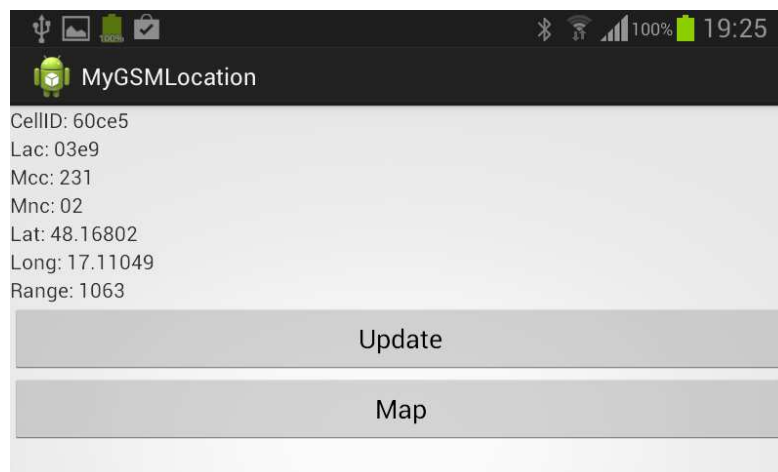
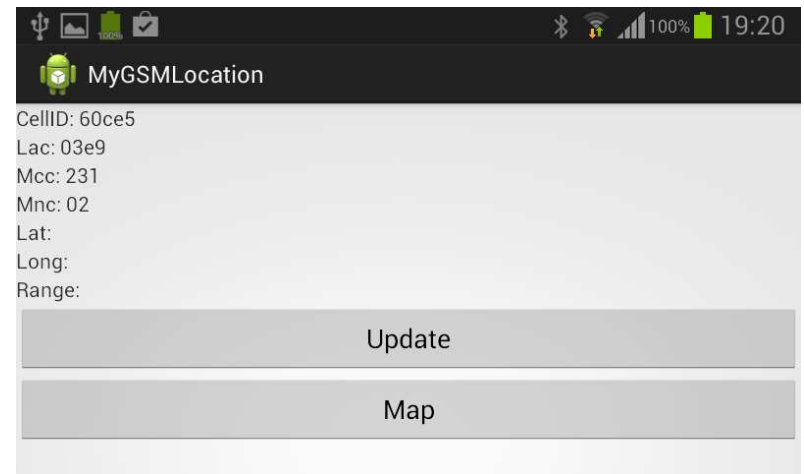
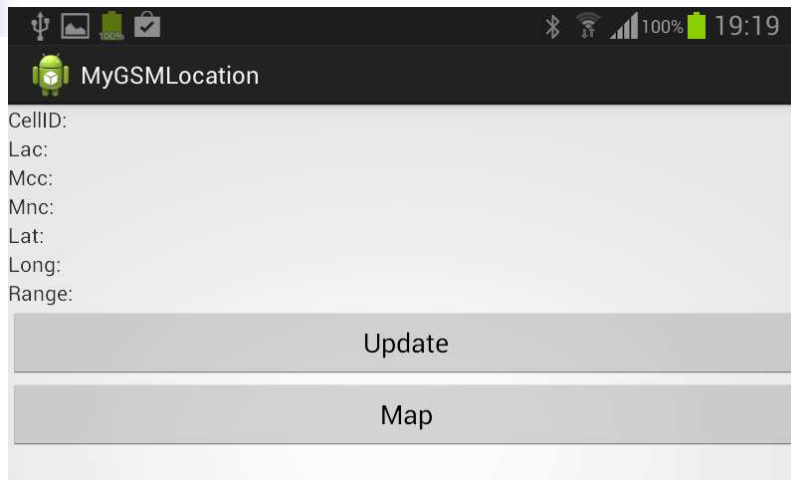


LocationAPI z aplikácie

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

- potrebujeme urobiť http-POST request na
<https://eu1.unwiredlabs.com/v2/process.php>
... na budúce (v prednáške http)
- keďže to niečo trvá, **nesmieme to robiť v hlavnom vlákne** – AsyncTask
... na budúce
- do tela dotazu (requestu) potrebujeme **zakódovať** (cellID, lac, mcc, mnc + môj token) hoc jednoduchý, ale predsa-len **JSON** object->json
... na budúce (v prednáške http)
- z tela odpovede (responzu) potrebujeme **dekódovať** hoc jednoduchý, ale **JSON** objekt, t.j. prečítať latitude-longitude, json -> object
... na budúce (v prednáške http)
- no a **zobraziť na mape**, vytvoríme, zavoláme intent `startActivity(Intent(android.content.Intent.ACTION_VIEW, Uri.parse("geo:0,0?q=$latit,$longit")))`
... alebo na budúce (v prednáške Google MAPS)
- resp. všetko je v priloženom kóde, resp. z php/pythonovských príkladov

GSM LocationAPI.org



Project: MyGSMLocation.zip

Cell Tower & WiFi Coverage



Pokrytie

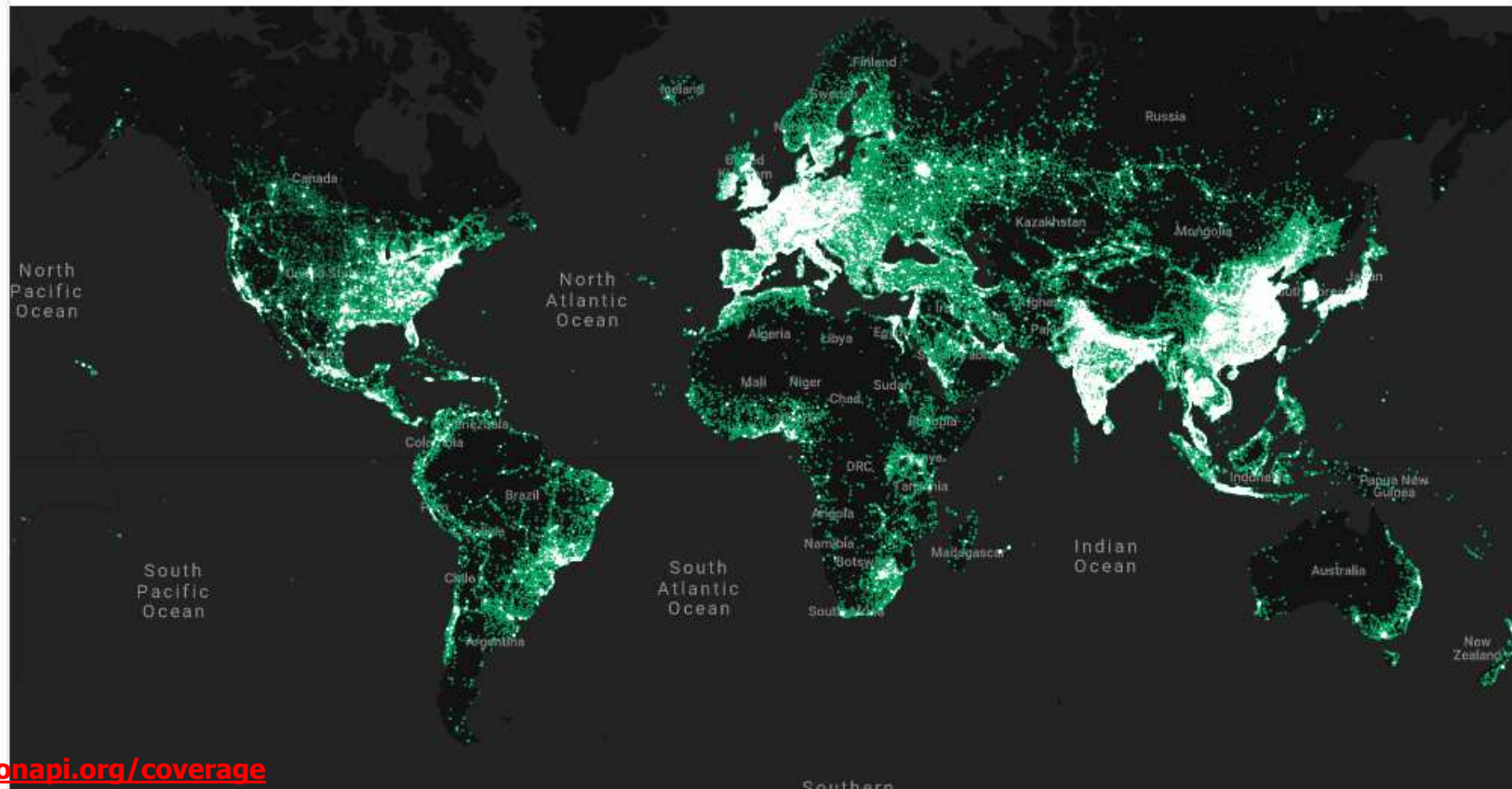
204.27 million

Cells

4.15 billion

WiFi Access Points

This interactive map shows the physical locations of Cell towers & Wifi APs from our database. Data has been rounded off to resolution of ~100 meters.



<http://locationapi.org/coverage>

Wifi

- 00:0f:f7:2e:16:a0,"FMFI_UK",48.152248500846326,17.07091609016061,5,100,FMFI_UK,1,FMFI_UK
- 00:0f:f7:2e:16:a2,"eduroam",48.1524363392964,17.070470340549946,5,100,eduroam,1,eduroam

Request: 2 WiFis

```
1 {
2   "token": "95b2941777892d",
3   "wifi": [{
4     "bssid": "00:0f:f7:2e:16:a0"
5   }, {
6     "bssid": "00:0f:f7:2e:16:a2"
7   }],
8   "address": 1
9 }
```

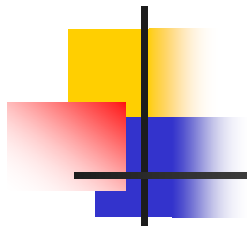
Response:

```
1 {
2   "status": "ok",
3   "balance": 48,
4   "lat": 48.15225561,
5   "lon": 17.06998922,
6   "accuracy": 10,
7   "address": "Staré grunty, Švédske d
8 }
```

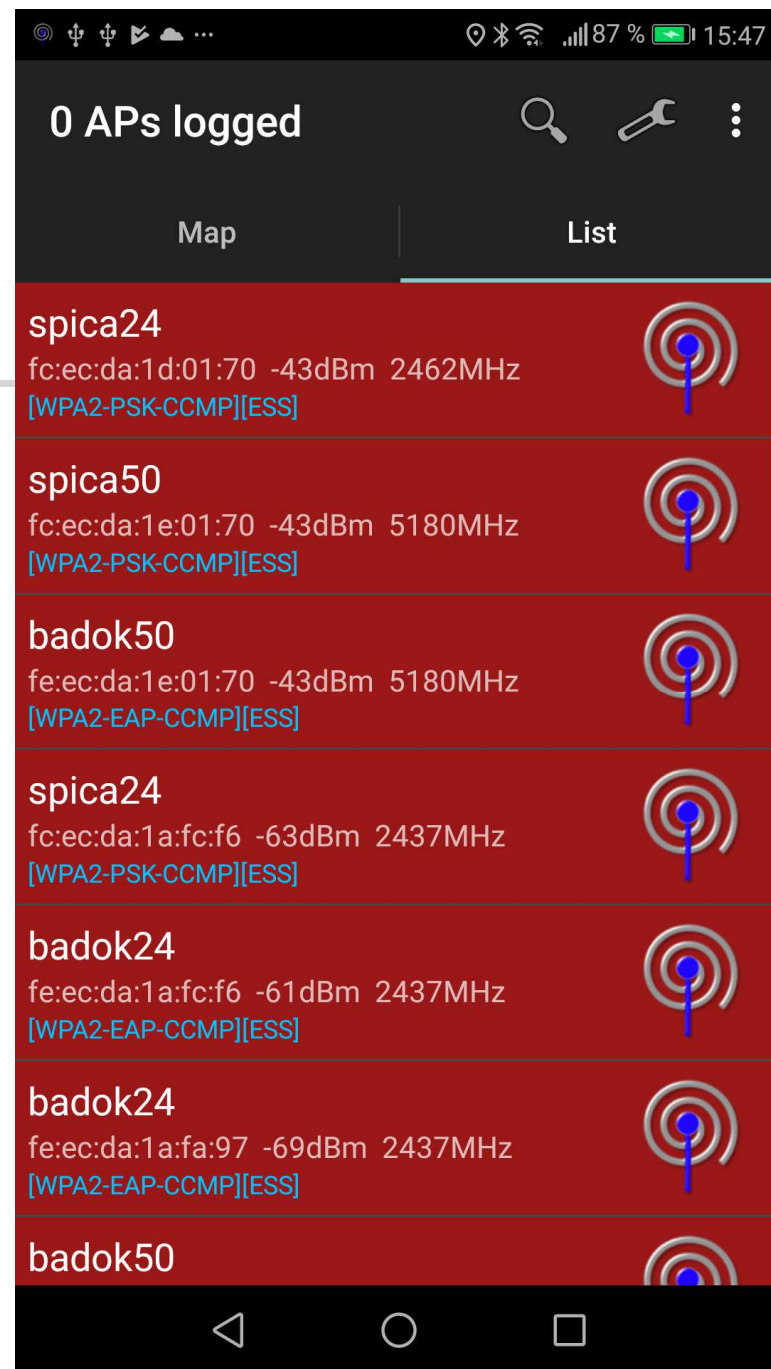
Location:



- 8c:59:c3:90:71:81
- 0c:84:dc:96:7b:06
- <https://www.google.com/maps/place/48%C2%B010'02.9%22N+17%C2%B006'36.0%22E/@48.1674834,17.1078093,17z/data=!3m1!4b1!4m5!3m4!1s0x0:0x0!8m2!3d48.1674834!4d17.109998>



```
{  
  "token": "95b2941777892d",  
  "mcc": 231,  
  "mnc": 2,  
  "cells": [{  
    "lac": 1001,  
    "cid": 109366  
  }],  
  "address": 1  
}
```



Iné služba, iná DB, iné API

(podobné výsledky)

Request

GSM

```
1 {
2   "radioType": "gsm",
3   "cellTowers": [{
4     "mobileCountryCode": 231,
5     "mobileNetworkCode": 2,
6     "locationAreaCode": 1001,
7     "cellId": 396517 }]
8 }
```

Response

```
1 {
2   "location": {
3     "lat": 48.16677,
4     "lng": 17.11046
5   },
6   "accuracy": 254
7 }
```

Location



`https://cps.combain.com?key=YOUR_API_KEY`

Method: POST

Content-Type: application/json

... registrujte sa a vyskúšajte

<https://portal.combain.com/>

WiFi

- 00:0f:f7:2e:16:a0,"FMFI_UK",48.152248500846326,17.07091609016061,5,100,FMFI_UK,1,FMFI_UK
- 00:0f:f7:2e:16:a2,"eduroam",48.1524363392964,17.070470340549946,5,100,eduroam,1,eduroam

Request WIFI

```

1 {
2   "wifiAccessPoints": [{
3     "macAddress": "00:0f:f7:2e:16:a0",
4     "signalStrength": -61
5   },
6   {
7     "macAddress": "00:0f:f7:2e:16:a2",
8     "signalStrength": -62
9   }]
10 }

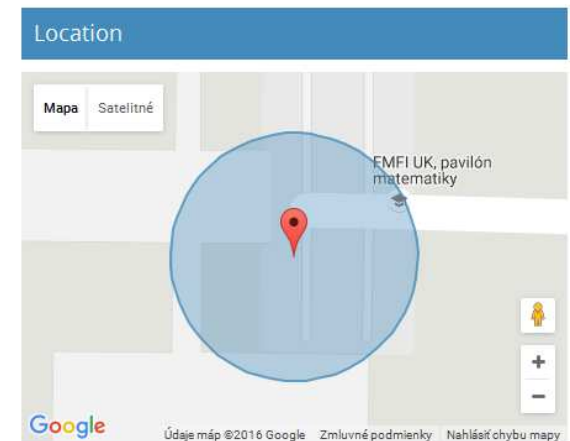
```

Response

```

1 {
2   "location": {
3     "lat": 48.15225,
4     "lng": 17.07092
5   },
6   "accuracy": 20
7 }

```



231 - Slovakia

Total number of GSM:	56493
Total number of WCDMA:	84208
Total number of LTE:	24465
Total number of CDMA:	0
Total number of WIFI:	2189255



Pokrytie

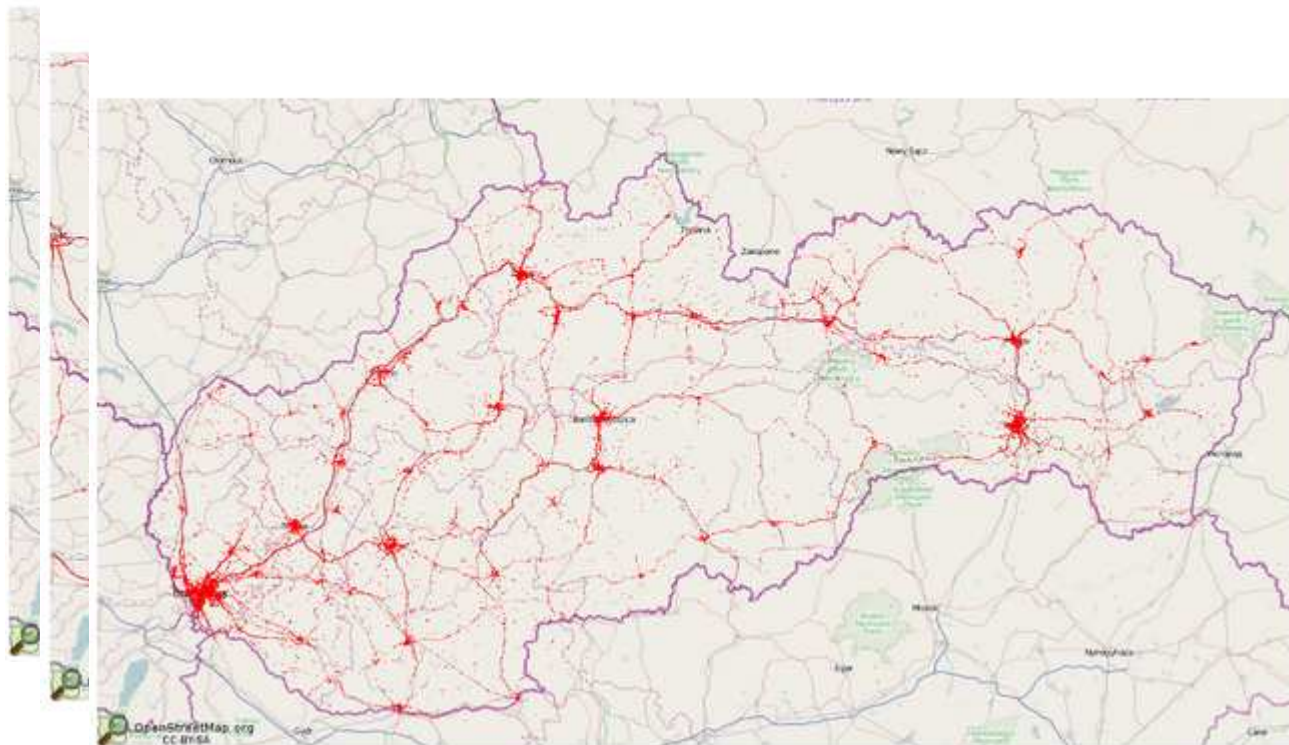
Ra

Radio Type:

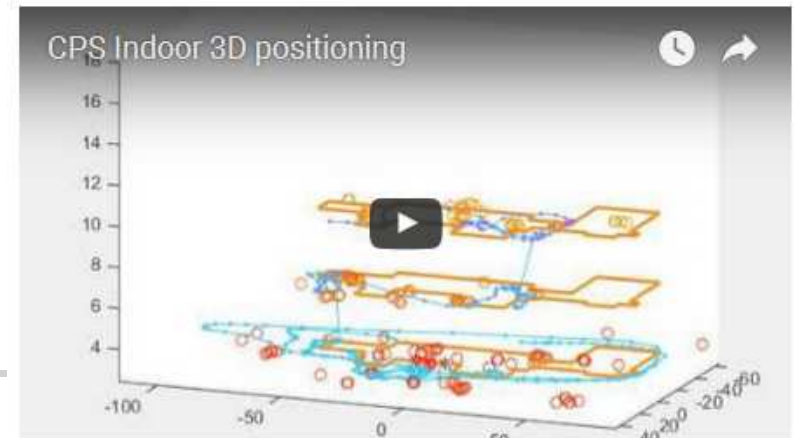
Radio Type:

All GSM

All LTE



Indoor



See how our SLAM creates a 3D map of all WiFis in a building

<https://youtu.be/49rXHpScAyc>



Mattehuset

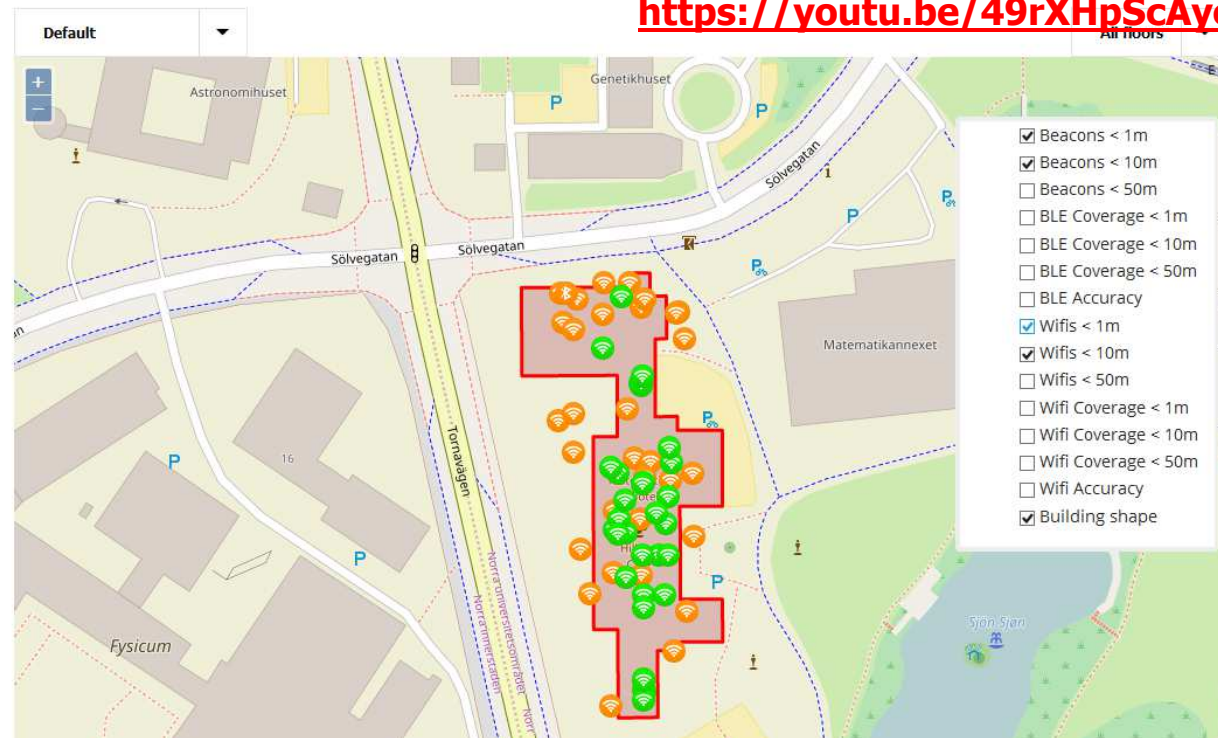
Address: Sölvegatan 18

Zip code: 22363

City: Lund

Country: Sweden

Type: School

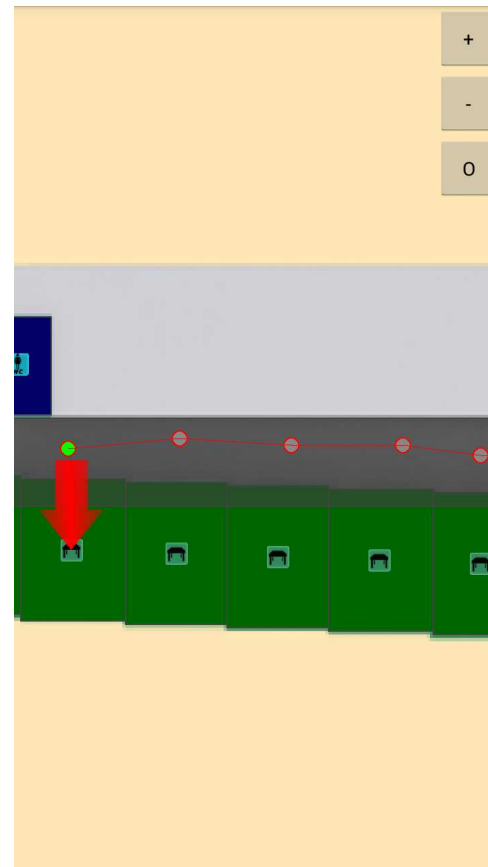
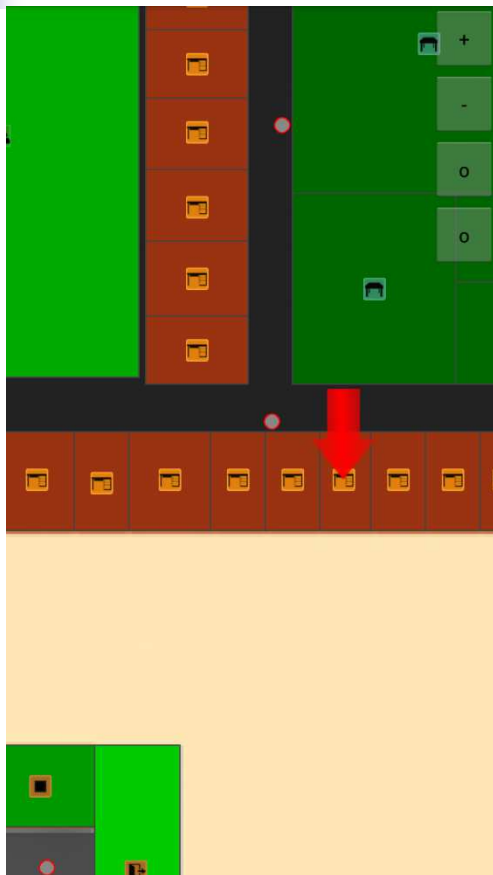


<https://youtu.be/Ca05ffE78so>

<https://combain.com/building/?id=171>

Lokalizácia a navigácia v interieri

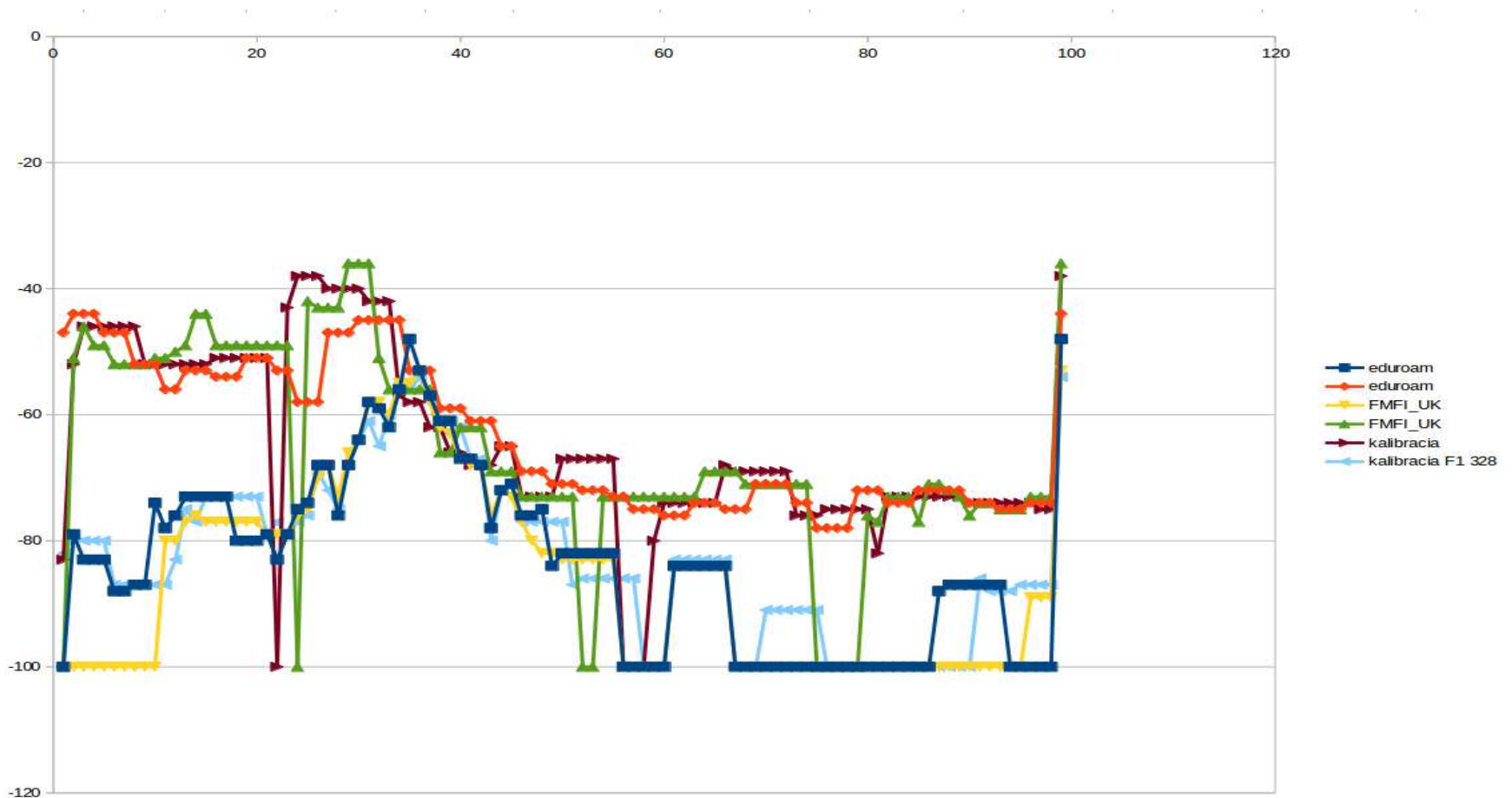
Bc. Martin Šuník



Lokalizácia a navigácia v interieri

Bc. Martin Šuník

■ Pohyb po chodbe, Y je sila signálu, X je čas

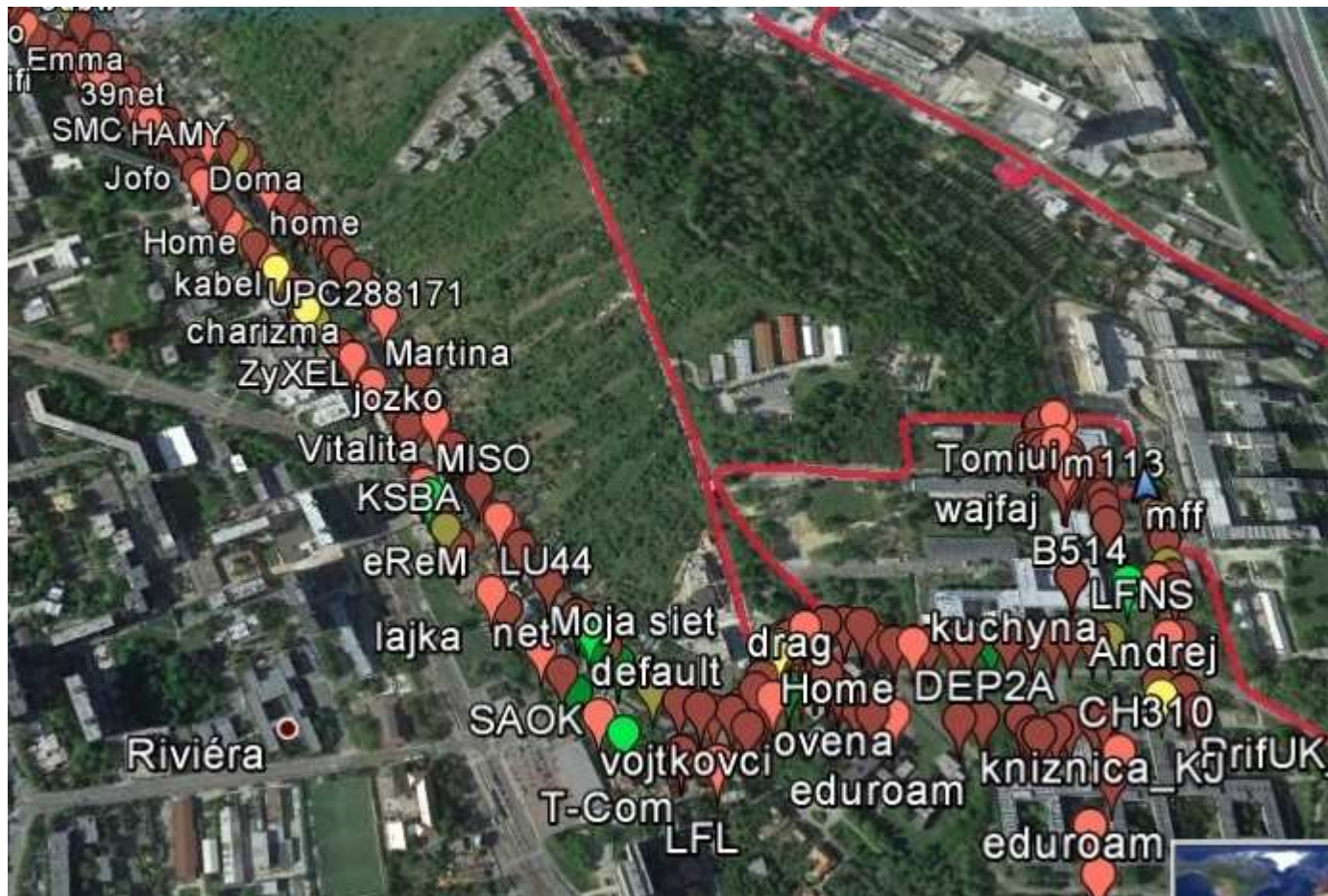




Wifi tracker

<https://play.google.com/store/apps/details?id=org.prowl.wifiscanner>

Alternatíva, použijeme hotový software



Exportujeme do

- .CSV
wifiscan-export.csv
- .kml
wifiscan-export.kml

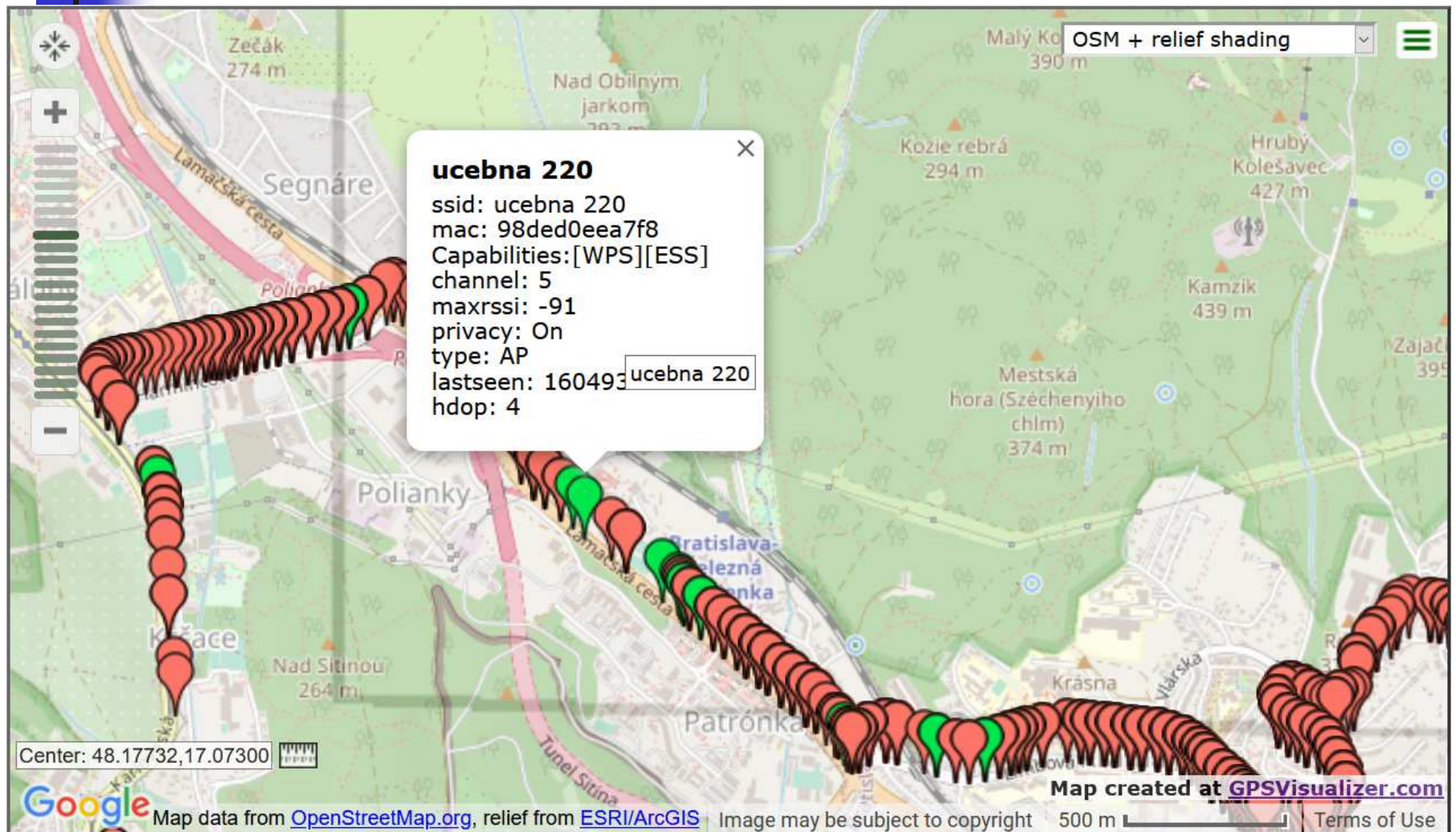
transformujeme
databázu:

- mac addr
 - gps
 - ssid
- wifiscan-export_redukovany*

Nakrmíme combian

wifi-scan-export.kml, csv, .txt

<https://www.gpsvisualizer.com/>





WiFi positioning system (WPS)

(prípade zlyhávajúceho GPS, resp. v kombinácii s GSM, GPS)

```
wifiManager =  
    applicationContext.getSystemService(Context.WIFI_SERVICE) as  
        WifiManager  
  
if (!wifiManager.isWifiEnabled) {  
    wifiManager.setWifiEnabled(true)  
}  
  
if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.M  
    &&                                // dôležité od verzie Marshmallow  
    checkSelfPermission(Manifest.permission.ACCESS_COARSE_LOCATION)  
    != PackageManager.PERMISSION_GRANTED) {  
  
    requestPermissions(arrayOf(  
        Manifest.permission.ACCESS_WIFI_STATE,  
        Manifest.permission.CHANGE_NETWORK_STATE,  
        Manifest.permission.ACCESS_FINE_LOCATION,  
        Manifest.permission.ACCESS_COARSE_LOCATION),  
        RUNTIME_PERMISSION_REQUEST_CODE) // callback pre  
    }  
    onRequestPermissionsResult
```




WiFi positioning system (WPS)

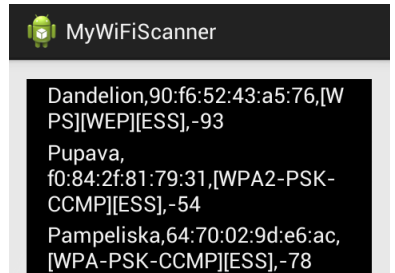
(requestPermissions callback)

```
// callback
override fun onRequestPermissionsResult(
    requestCode: Int,
    permissions: Array<String>,
    grantResults: IntArray ) {
    when (requestCode) {
        RUNTIME_PERMISSION_REQUEST_CODE -> {
            for (i in grantResults.indices) {
                if (grantResults[i]==PackageManager.PERMISSION_GRANTED) {
                    Log.d(TAG, "GRANTED")
                } else {
                    Log.d(TAG, "DENIED")
                }
            }
            return
        }
    }
}
```

WiFi positioning system (WPS)

(broadcast receiver)

```
broadcastReceiver = object : BroadcastReceiver() {  
    override fun onReceive(context: Context?, intent: Intent?) {  
        val scanList = wifiManager.scanResults  
        val wifis = arrayOfNulls<String>(scanList.size)  
        for (i in scanList.indices) {  
            val sr = scanList[i]  
            wifis[i] = sr.SSID +           // network name  
                    "," + sr.BSSID + // mac addr  
                    "," + sr.capabilities + // authent, key,  
                    "," + sr.level // db  
        }  
        listView1.adapter = ArrayAdapter<String>( applicationContext,  
            android.R.layout.simple_list_item_1, wifis)  
    }  
}  
v onCreate zaregistrujte broadcastReceiver pre intent SCAN_RESULTS_AVAILABLE_ACTION  
    registerReceiver(broadcastReceiver, IntentFilter(  
        WifiManager.SCAN_RESULTS_AVAILABLE_ACTION))  
v spustite skenovanie  
    wifiManager.startScan()
```





Capabilities

[WPA2-PSK-CCMP][ESS]
[WPA2-PSK-CCMP+TKIP][ESS]
[WPA-PSK-CCMP+TKIP]
[WPA2-PSK-CCMP+TKIP][ESS]
[WPA-PSK-TKIP CCMP][WPA2-PSK-TKIP-CCMP][WPS][ESS]

[Authentication Algorithm - Key Management Algorithm - Pairwise Cipher]

- Authentication Algorithm

- EAP
- WPA
- WEP

- Pairwise Cipher

- CCMP
- TKIP

```
//A capability of [ESS] represents an open access point
if (scanResult.capabilities != null &&
    scanResult.capabilities.contains("[ESS]"))
    openNetworks++
```



Žiadosť o povolenie

Ak SDK 23 (Marshmallow) alebo vyššie:

```
if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.M)
```

- musíte deklarovať napr. `ACCESS_COARSE/FINE_LOCATION` alebo v A-Manifest.xml
- musíte žiadať o povolenie dynamicky, ktoré užívateľ schváli alebo zamietne

```
requestPermissions(arrayOf(  
    Manifest.permission.ACCESS_FINE_LOCATION,  
    ...  
    Manifest.permission.ACCESS_COARSE_LOCATION  
),  
    RUNTIME_PERMISSION_REQUEST_CODE)
```

Ak SDK 22 alebo nižšie:

- deklarujete `ACCESS_COARSE/FINE_LOCATION` len v AndroidManifest.xml



Permissions do Manifest.xml

(toto je/bolo pravda do API <23)

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

a veľmi skoro budeme potrebovať ...

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

MapActivity

Accur:23.345236

Lat:48.167550833333344

Long:17.109992333333333

Altitude:287.2379524372518

Provider:gps

Speed:0.34313443

Bearing:0.0

Time:1353692380000 ...Nov 23, 2012 6:39:40 PM

Project:MyLocation1.zip



Permissions do Manifest.xml

(ak API ≥ 23)

Okrem tohoto:

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
a veľmi skoro budeme potrebovať ...
<uses-permission android:name="android.permission.INTERNET"/>
```

treba v kóde dynamicky žiadať o povolenie (zjednodušený kód):

```
if (android.os.Build.VERSION.SDK_INT >= 23) {
    if (getApplicationContext().checkSelfPermission(permission) !=
        PackageManager.PERMISSION_GRANTED)
        permissionsList.add(permission)
}
```

... a následne požiadať o povolenia:

```
requestPermissions(permissionsList.toArray(),
    REQUEST_CODE_ASK_MULTIPLE_PERMISSIONS)
```



Úrovně povolení

Normal Permissions –

nízka úroveň narušení soukromí

- ACCESS_NETWORK_STATE
- CHANGE_NETWORK_STATE
- ACCESS_WIFI_STATE
- CHANGE_WIFI_STATE
- CHANGE_WIFI_MULTICAST_STATE
- BLUETOOTH
- BLUETOOTH_ADMIN
- INTERNET
- SET_ALARM
- SET_WALLPAPER
- VIBRATE
- WAKE_LOCK

Signature Permissions –

appka musí být podepsána autoritou

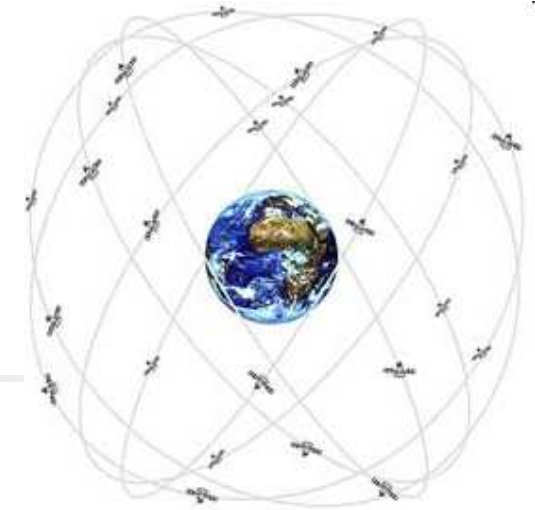
- BIND_ACCESSIBILITY_SERVICE
- BIND_NFC_SERVICE
- BIND_TV_INPUT
- BIND_WALLPAPER
- READ/WRITE_VOICEMAIL
- WRITE_SETTINGS

Dangerous Permissions –

appka musí explicitně žádat povolení

- READ/WRITE_CALENDAR
- CAMERA
- READ/WRITE_CALL_LOG
- READ/WRITE_CONTACTS
- GET_ACCOUNTS
- ACCESS_FINE_LOCATION
- ACCESS_COARSE_LOCATION
- SEND/RECEIVE_SMS

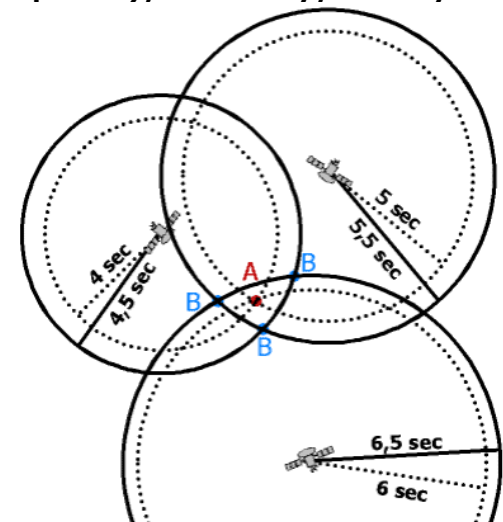
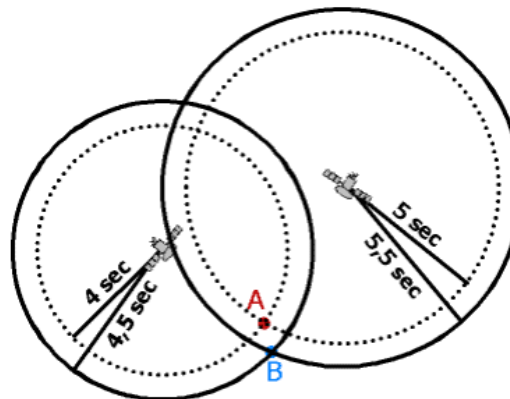
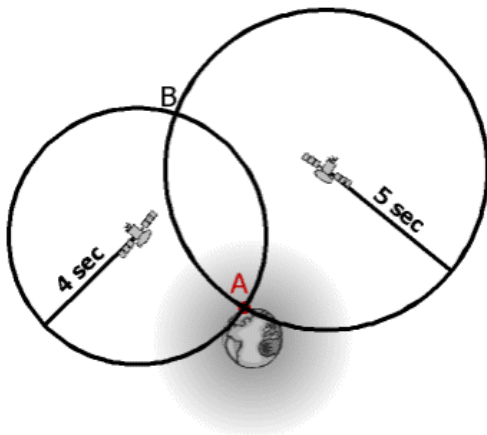
World Geodetic System



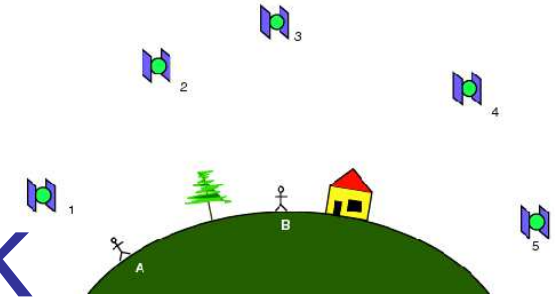
- viac ako 24 satelitov v niekoľkých rovinách,
- vysielajú v signáli svoje súradnice a presný čas,
- prijímač zistí časový posun, za ktorý správa príde k prijímaču,
- ako ? prijímač nemá presný čas... (<http://gpsinformation.net/main/gpslock.htm>)
- to aproximuje vzdialenosť od satelitu,
- satelity disponujú presným časom [merania s presnosťou 1ns=30cm],
- 3 satelity určujú 3 guľové sféry, ktorých prieniku zodpovedajú 2 body,
- preto sa používajú minimálne 4 satelity na určenie polohy,
- problémy: atmosferická refrakcia, počasie (vodná para), stromy/domy

<http://en.wikipedia.org/wiki/Geoid>

<http://www.kowoma.de/en/gps/orbits.htm>



Ostáva mnoho otázok



- ako vysoko lietajú satelity ?
- stoja satelity, alebo sa pohybujú vzhľadom na Zem ?
- kto vie/určuje polohu satelitov ?
- čo, ak satelit má posunuté hodinky ?
- vieme síce vypočítať vzdialenosť dvoch bodov na zemi (Great Circle Distance, http://en.wikipedia.org/wiki/Great-circle_distance), ale ako vypočítať vzdialenosť bodu na zemi od satelitu ?
- ako je možné, že presnosť, ktorú dosiahneme je v rádoch metre-desiatky vojaci/geodeti používajú gps s presnosťou na cm. Čo majú iné ? Prijímač ? Vlastné satelity ?

Pre zvyšok prednášky považujme za štandard: WGS-84

- latitude (zem.šírka) $-90 \dots +90$
- longitude (zem.dĺžka) $-180 \dots +180$
- altitude (nadmorská výška) oproti ideálnemu elipsoidu





GPS - NMEA

```
$GPGGA,184357.08,1928.967,S,02410.530,E,1,04,1.9,100.00,M,-33.9,M,,0000*67
$GPGLL,1928.947,S,02410.536,E,184358.08,A,A*70
$GPVTG,16.78,T,,M,74.00,N,137.05,K,A*36
$GPRMC,184400.08,A,1928.907,S,02410.547,E,74.00,16.78,210410,0.0,E,A*29
$GPGGA,184401.08,1928.888,S,02410.553,E,1,04,2.2,100.00,M,-33.9,M,,0000*6E
$GPGLL,1928.868,S,02410.559,E,184402.08,A,A*7D
$GPVTG,16.78,T,,M,74.00,N,137.05,K,A*36
$GPRMC,184404.08,A,1928.829,S,02410.571,E,74.00,16.78,210410,0.0,E,A*25
$GPGGA,184405.08,1928.809,S,02410.577,E,1,04,1.1,100.00,M,-33.9,M,,0000*65
$GPGLL,1928.789,S,02410.583,E,184406.08,A,A*7E
$GPVTG,16.78,T,,M,74.00,N,137.05,K,A*36
$GPRMC,184408.08,A,1928.750,S,02410.595,E,74.00,16.78,210410,0.0,E,A*22
$GPGGA,184409.08,1928.730,S,02410.601,E,1,04,2.8,100.00,M,-33.9,M,,0000*64
$GPGLL,1928.710,S,02410.607,E,184410.08,A,A*76
```

Ked' „preskočíme“ fyziku, čo a aké informácie GPS prijímač produkuje ?

GPS signál obsahuje vety (>19) rôznych formátov, napr. niektoré z nich:

- \$GPGGA - Global Positioning System Fix Data
- \$GPGLL - Geographic position, latitude / longitude
- \$GPGSV - GPS Satellites in view
- \$GPZDA - Date & Time

\$GPGLL,5133.81,N,00042.25,W*75

- 1 5133.81 Current latitude
- 2 N North/South
- 3 00042.25 Current longitude
- 4 W East/West
- 5 *75 checksum

Knižnica pre prácu s GPS poskytuje API zakrývajúce tieto raw-formáty

<http://aprs.gids.nl/nmea/>

Prémia "Slné hodiny"

V „čiernej skrinke“ bol externý Bluetooth GPS prijímač, ktorý generuje:

```
$GPGGA,202013.000,4810.0583,N,01706.6030,E,1,05,3.9,306.5,M,,,,0000*04
```

```
$GPGSA,A,3,07,30,28,08,13,,,,,,,,,5.8,3.9,4.3*36
```

```
$GPRMC,202013.000,A,4810.0583,N,01706.6030,E,0.18,170.37,021116,,*09
```

```
$GPGGA,202014.000,4810.0578,N,01706.6030,E,1,05,3.9,306.4,M,,,,0000*06
```

```
$GPGSA,A,3,07,30,28,08,13,,,,,,,,,5.8,3.9,4.3*36
```

**Takže, keď vznikal tento .log,
Bolo 2.11.2016 presne 20:20:13 UTC**





LocationManager LocationProvider

Location API zakrýva NMEA
ale keď potrebujete,
viete sa k nemu dostať...

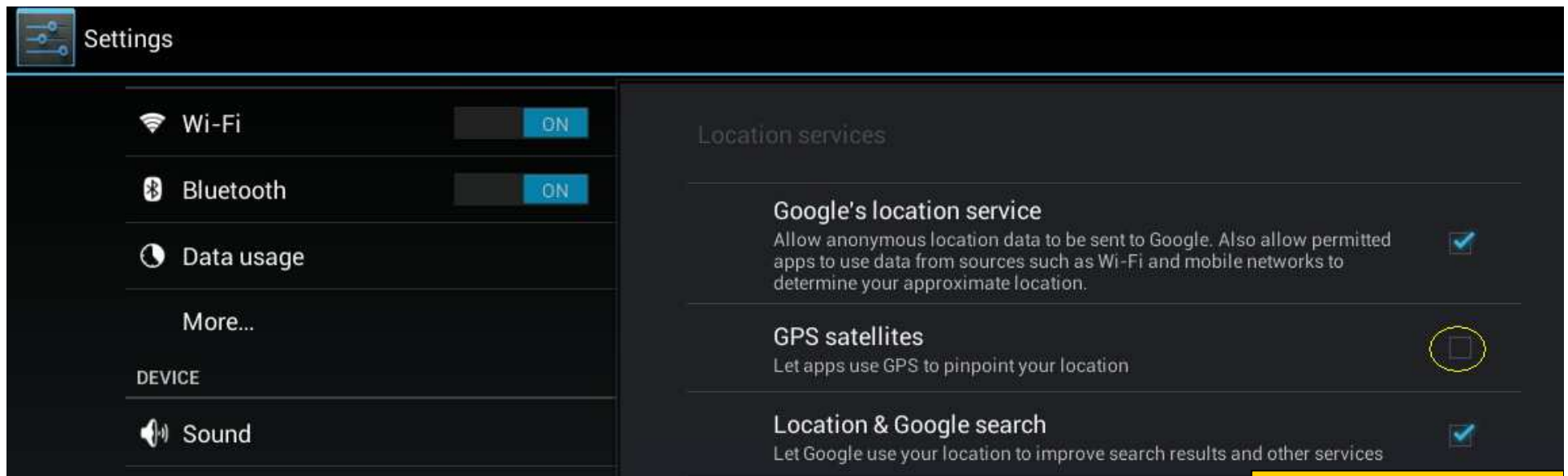
```
lateinit var lm: LocationManager
...
lm = getSystemService(Context.LOCATION_SERVICE) as LocationManager

val cr = Criteria()
    cr.isAltitudeRequired = false
    cr.accuracy = Criteria.ACCURACY_COARSE // hrubá presnosť
    // cr.setAccuracy(Criteria.ACCURACY_FINE); // fajnová presnosť
    cr.powerRequirement = Criteria.POWER_LOW
    cr.isSpeedRequired = false

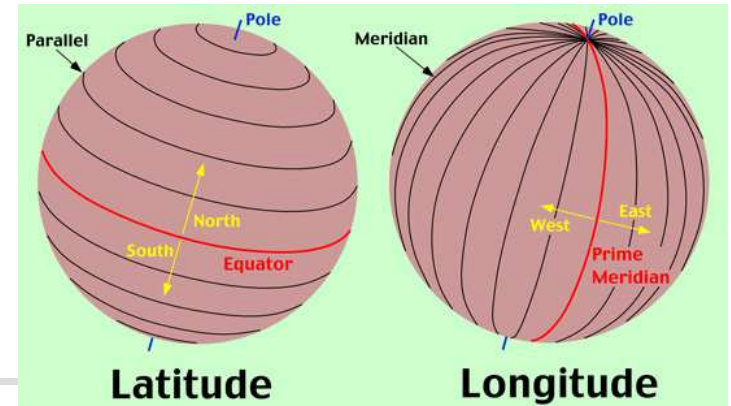
Log.d("MyLoc", "all providers:${lm.allProviders}")
String provider = LocationManager.GPS_PROVIDER;
    // LocationManager.NETWORK_PROVIDER
    // lm.getBestProvider(cr, false); // enabled or disables
    // lm.getBestProvider(cr, true); // enabled providers
```

Ak máme GPS vypnuté

```
if (! lm.isProviderEnabled(LocationManager.GPS_PROVIDER) ) {  
    val settingsIntent = Intent(  
        android.provider.Settings.ACTION_LOCATION_SOURCE_SETTINGS)  
    startActivity(settingsIntent)  
} else  
    Log.d("MyLoc", "GPS is enabled")
```



Location



```
val loc = lm.getLastKnownLocation(provider);
if (loc != null) {
    loc.latitude        longitude        altitude
    loc.accuracy        provider
    loc.speed           - rýchlosť v m/s
    loc.bearing         - azimut, ale musíme sa hýbať !! nepresnosť
                        - niekedy môže zapríčiniť, že sa hýbeme, ak keď stojíme !
    loc.time            - milisec od 1970
```

MapActivity	Activity
Accur:30.0	ur:20.0
Lat:48.1672728	48.16756483633071
Long:17.1103865	g:17.11011447943747
Altitude:0.0	itude:290.6
Provider:network	vider:gps
Speed:0.0	ed:1.8125
Bearing:0.0	ring:323.77
Time:1353530211083 ... Nov 21, 2012 9:36:51 PM	e:1383471502000 ... 3.11.2013 10:38:22

Project: MyLocation1.zip

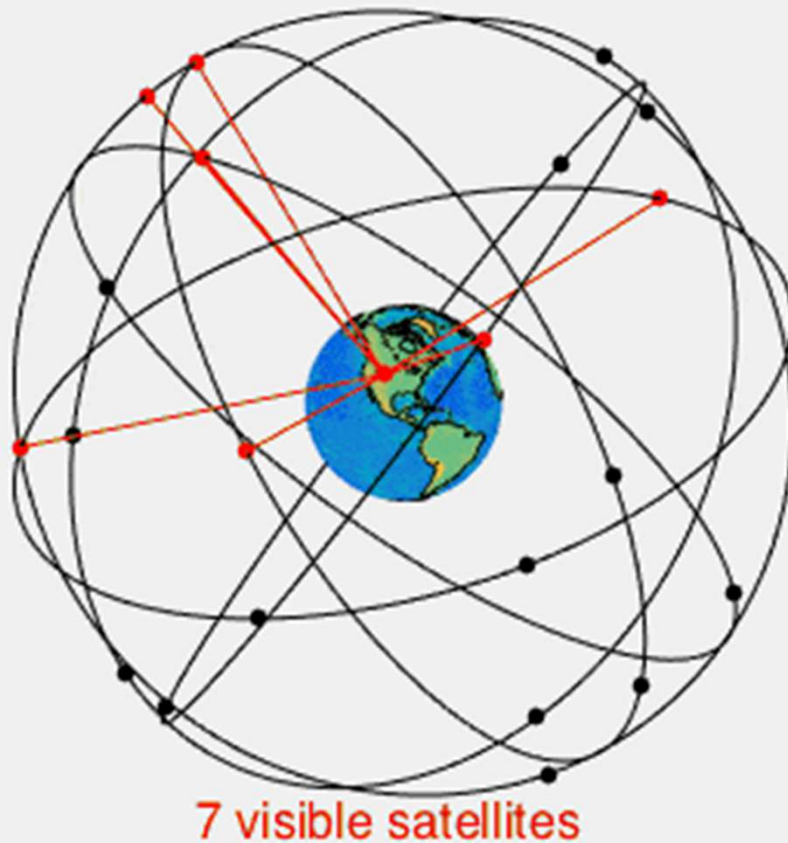
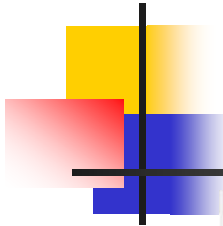


LocationListener - updates

```
override fun onResume() { ←
    // každú sekundu, resp. ak sme prešli aspoň 10 metrov
    lm.requestLocationUpdates(provider, 1000, 10, this)
    // minDistance = 0, minTime = 0
    lm.requestLocationUpdates(provider, 0, 0, this)
}

override fun onPause() { ← // ak aktivita prestane byť aktívna
    lm.removeUpdates(this)    // patrí sa to vypnúť
}

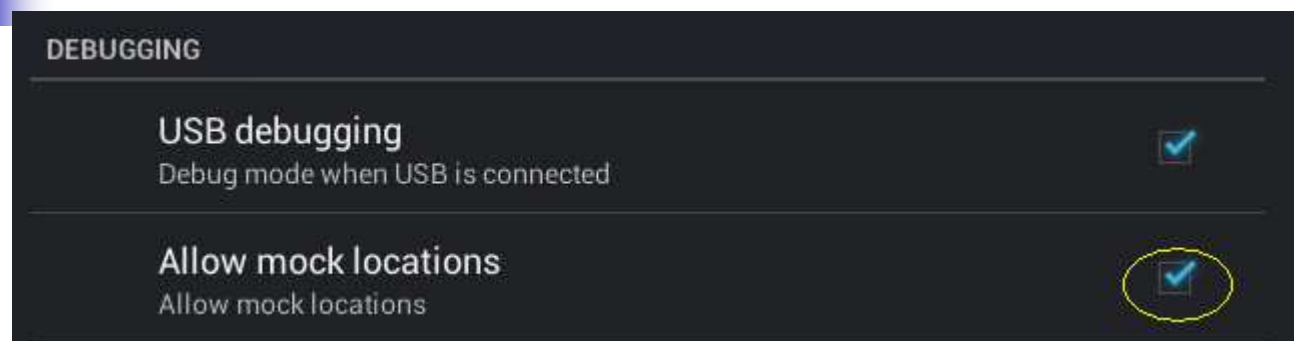
override fun onLocationChanged(Location arg0) { ←
    Log.d("MyMap", "onLocationChanged:" ...
    updateLocation(arg0)        // zobraz na display zmenu polohy
}
```

Na obežnej dráhe Zeme sa nachádza vždy najmenej 24 operačných GPAS satelitov. Satelity prevádzkované americkým ministerstvom obrany obiehajú s periódou 12 hodín (dve obežné dráhy za deň), vo výške asi 21 000 km a ich rýchlosť je asi 3.9 km/s (14 000 km/h)

Ako debugovať, ak prší...

ak máme mobil, tak napr. si nainštalujeme nejaký FakeGPS



MapActivity
Accur:1.0
Lat:-22.967033
Long:-43.180698
Altitude:65.0
Provider:gps
Speed:0.0
Bearing:0.0
Time:1353694738829 ...Nov 23, 2012 7:18:58 PM



Simulovať polohu možno aj v emulátore (avd) – Android Device Monitor, DDMS

ProximityListener



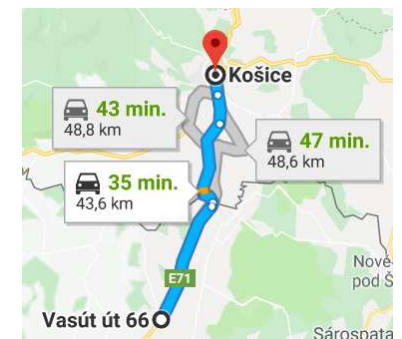
```
val pid = PendingIntent.getBroadcast(this, -1L,  
                                   Intent("blizko_KE"), 0)  
  
// ak sa priblížime k bodu (KE) na 50.000 metrov, odpáli sa pid  
lm.addProximityAlert(48.720297, 21.258333, 50000f, (long) (-1), pid)
```

// a kto to chytí ?

```
val bRec = object : BroadcastReceiver() {  
    override fun onReceive(Context context, Intent intent) {  
        if (intent.getBooleanExtra(  
            LocationManager.KEY_PROXIMITY_ENTERING, false))  
            Log.d("MyLoc", "už si v Košicoch (< 50 km)")  
    }  
}
```

```
registerReceiver(bRec, IntentFilter("blizko_KE"))
```

```
11-23 20:21:37.034: D/MyLoc(1100): Location:47.83066:20.736218333333333  
11-23 20:21:37.234: D/MyLoc(1100): už si v Košicoch (< 50 km)  
11-23 20:21:37.284: D/MyLoc(1100): onLocationChanged:47.83317:20.738178333333334
```



Project: MyLocation2.zip

Geocoder

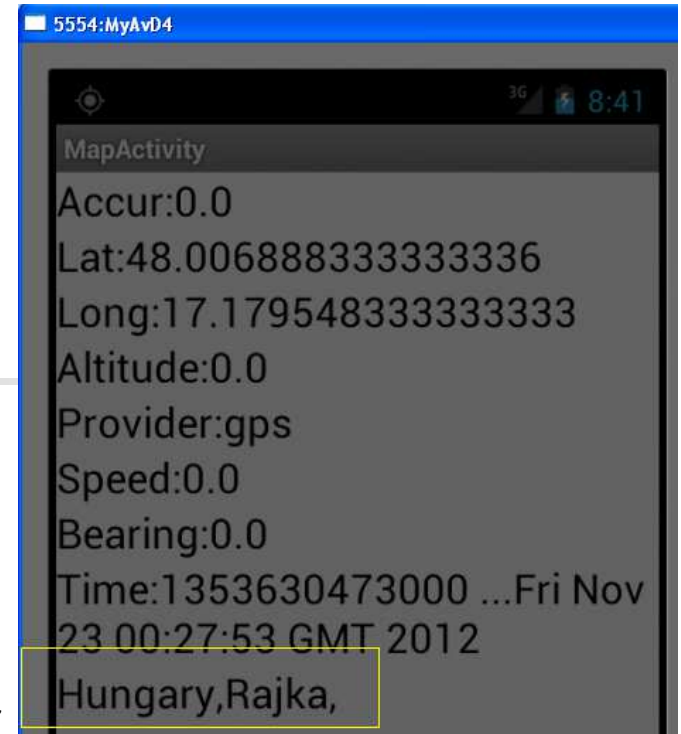
(getFromLocation)

Geocoder.getFromLocation transformuje
(Latitude;Longitude) na List<Address>,
ale potrebuje internet:

```
<uses-permission android:name=
```

```
android.permission.INTERNET"/>
```

```
if (Geocoder.isPresent()) {  
    val gc = Geocoder(baseContext(), Locale.getDefault());  
    try {  
        // if gc != null  
        for(adr in gc.getFromLocation( // max. 1 adresu  
            loc.getLatitude(), loc.getLongitude(), 1)) {  
            String str = ""  
            str += adr.getCountryName()+"", // if adr != null  
            for (i in 0 until adr.getMaxAddressLineIndex())  
                str += adr.getAddressLine(i)+"",  
            tvaddr.setText(str)  
        }  
    }  
}
```



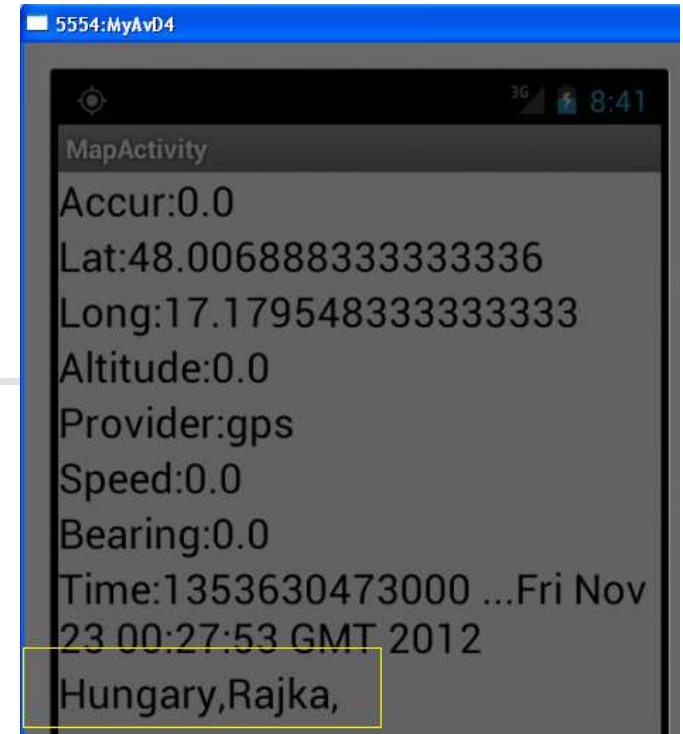
Geocoder

(getFromLocation)

- **Geocoder.getFromLocation** transformuje (Latitude;Longitude) na List<Address>
- Geocoder.getFromLocationName nájde miesto podľa popisu, vráti List<Address> ale potrebuje internet:

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

if (Geocoder.isPresent()) {
    Geocoder gc = new Geocoder(baseContext(), Locale.getDefault())
    try {
        for(ad in gc.getFromLocation(
            loc.getLatitude(),
            loc.getLongitude(), 1)) { // max. 1 adresu
            ....
        }
        for(ad in gc.getFromLocationName(
            "Univerzita Komenského, Mlynska dolina, Bratislava", 5);
            // max. 5 adresies
```



Geocoder

(getFromLocationName)

- **Geocoder.getFromName** transformuje String na List<Address>

```
val addr = gc.getFromLocationName(
    "Bratislava", 5); // max. 5 adres
for (adr : addr) { // addr != null
    locationBA = Location("Blava")
    locationBA.latitude = adr.latitude
    locationBA.longitude = adr.longitude
    var s = Math.round(loc.distanceTo(locationBA)/1000) + " km, "
        ... loc.bearingTo(locationBA)/1000 ... azimuth
    s += adr.countryName + ", " // if adr != null
    for (i = 0 until adr.getMaxAddressLineIndex())
        s += adr.getAddressLine(i) + ", "
    tvBA.text = s
}
```

