

Android - komunikácia



Peter Borovanský
KAI, I-18

MS-Teams: [2sf3ph4](#), [List](#), [github](#)
borovan 'at' ii.fmph.uniba.sk

- **Retrofit Rest API**
- http(s) GET, POST
 - static Google Maps V2
 - Google Directions
- **formáty json (gson) a xml**



Aký klient...

dnes to bude viac o org.apache ako o androide

<http://hc.apache.org/httpcomponents-client-ga/tutorial/html/>

Klient koho, resp. kto je server ?

- server je len náš (ale **nepoužívame** http protokol na komunikáciu s ním):
 - môže to byť aj socket-socket komunikácia, ale vyvoláva to veľa otázok ...
ako napr. bezpečnosť, robustnosť, multi-vlákno pre obsluhu viacerých klientov, ...
 - priamy prístup do databázy, napr. cez jdbc, iný komunikačný protokol
- server nie je náš, ale máme tam neadminovský účet (davinci.fmph.uniba):
 - najčastejšie provider poskytne rozhrania (okrem webservera/Apache) php, mysql, ...
 - používanie http protokol - najčastejšie, jediný otvorený port je http/https,
 - najčastejšie vznikne tzv. AMP riešenie (Apache-MySQL-Php/Perl/Python) – minulé dekáda
- server vôbec nie je náš
 - môžeme odtiaľ čítať,
 - resp. máme špeciálne API na prístup k dátam (Parse fy.Facebook, Firebase fy.Google)

Čo so ním chceme robiť (so serverom):

- download všeobecne prístupných, resp. zaheslovaných dát,
- upload (malé resp. veľké dáta)
- run/exec (RPC – Remote Procedure Call)

Obsah prednášky



Download:

- HTTP GET – primárne chceme dotiahnúť (veľké) dáta zo servera
 - malé dáta – *všetko jedno* ...
príklad: select/update mojej gps pozície (lat, long) z databázy na serveri
 - veľké dáta – potrebujeme *extra vlákno*, aby sa *nehryzla* apka,
 - *AsyncTask*
 - *corutina*
 - *Retrofit async-callback*
 - autorizácia (Basic Authorization)

Upload – primárne chceme poslať (veľké dáta na server)

- HTTP POST
 - veľké dáta (max. veľkosť HTTP GET requestu ~8kB, podľa http web-servera)
problém, ak chceme uploadovať napr. snímka z kamery, video, ...

REST API

- nie je protokol
- nie je štandard
- je klient-server **stateless** architektúra nad HTTP
- json, xml, html...

Interpretované dáta:

- Bitmapa - Google Static Maps – príde statická bitmapa
- JSON – REST API, JSON parser (com.google.gson, alternatíva: org.json.JSON)
 - LocationApi.org príklad už bol minule
 - Google Directions – získanie cesty-navigácie od služby Google



Retrofit

(bolo v Cvičení-A)

DefaultHttpClient

extends `AbstractHttpClient`

```
java.lang.Object
↳ org.apache.http.impl.client.AbstractHttpClient
↳ org.apache.http.impl.client.DefaultHttpClient
```

This class was deprecated in API level 22.

Please use `openConnection()` instead. Please visit [this webpage](#) for further details.



- Retrofit je REST klient pre Android
- zjednodušuje download & upload JSON (cez HTTP GET/POST)
- používa napr. Gson, resp. SimpleXML, tikXML converter
- build.gradle treba doplniť o

```
implementation 'com.squareup.retrofit2:retrofit:2.9.0'
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'
```

- data class zodpovedajúci JSONu (mapovanie na json tagy):

```
data class Stat (
    @SerializedName("name")          /* -> */ val countryName: String?,
    @SerializedName("capital")       /* -> */ val capital: String?,
    @SerializedName("flagPNG")       /* -> */ val flag: String?,
    @SerializedName("latlng")        /* -> */ val latlng: Array<Float>?,
    @SerializedName("borders")       /* -> */ val borders: List<String>?,
    @SerializedName("alpha3Code")    /* -> */ val code: String?
)
```

- REST API pre Retrofit

```
interface StatInterface {
    @GET("vlajky/staty.json")
    suspend fun get(): Response<List<Stat>>
}
```

<https://dai.fmph.uniba.sk/courses/VMA/vlajky/staty.json>



Coroutines+MVVM+Retrofit

(model)

<https://dai.fmph.uniba.sk/courses/VMA/vlajky/staty.json>

```
data class Stat(  
    @SerializedName("name")           /* -> */ val countryName: String?,  
    @SerializedName("capital")        /* -> */ val capital: String?,  
    @SerializedName("flagPNG")         /* -> */ val flag: String?,  
    @SerializedName("latlng")          /* -> */ val latlng: Array<Float>?,  
    @SerializedName("borders")         /* -> */ val borders: List<String>?,  
    @SerializedName("alpha3Code")      /* -> */ val code: String?  
)
```

```
/*  
{  
  "alpha2Code": "SK",  
  "alpha3Code": "SVK",  
  "altSpellings": [  
    "SK",  
    "Slovak Republic",  
    "Slovensk\u00e1 republika"  
  ],  
  "area": 49037,  
  "borders": [  
    "AUT",  
    "CZE",  
    "HUN",  
    "POL",  
    "UKR"  
  ],  
  "callingCodes": [  
    "421"  
  ],  
  "capital": "Bratislava",  
  "currencies": [  
    {  
      "code": "EUR",  
      "name": "Euro",  
      "symbol": "\u20ac"  
    }  
  ],  
  "demonym": "Slovak",  
  "flagPNG":  
    "https://dai.fmph.uniba.sk/courses/VMA/vlajky/svk.png",  
  "gini": 26.0,  
  "languages": [  
    {  
      "iso639_1": "sk",  
      "iso639_2": "slk",  
      "name": "Slovak",  
      "nativeName": "sloven\u010dina"  
    }  
  ],  
  "latlng": [  
    48.6666666666,  
    19.5  
  ],  
  "name": "Slovakia",  
  "nativeName": "Slovensko",  
  "numericCode": "703",  
  "population": 5426252,  
  "region": "Europe",  
  "regionalBlocs": [  
    {  
      "acronym": "EU",  
      "name": "European Union"  
    }  
  ],  
  "subregion": "Eastern Europe",  
  "timezones": [  
    "UTC+01:00"  
  ],  
}
```

Coroutines+MVVM+Retrofit

(REST API - model)

```
interface StatInterface {  
    @GET("vlajky/staty.json")  
    suspend fun get(): Response<List<Stat>>  
}  
  
object StatService {  
    private val BASE_URL = "httpS://dai.fmph.uniba.sk/courses/VMA/"  
  
    fun get(): StatInterface =  
        Retrofit.Builder()  
            .baseUrl(BASE_URL)  
            .addConverterFactory(GsonConverterFactory.create())  
            .build()  
            .create(StatInterface::class.java)  
}  
}
```

Coroutines+MVVM+Retrofit

(viewmodel)

```
class ListViewModel: ViewModel() {  
    val service = StatService.get() ←  
    lateinit var job: Job  
    val staty = MutableLiveData<List<Stat>>() →  
  
    fun fetch() { ←  
        job = CoroutineScope(Dispatchers.IO)  
            .launch { ←  
                val response = service.get() // : Response<List<Stat>>  
                withContext(Dispatchers.Main) {  
                    if (response.isSuccessful)  
                        staty.value = response.body() ←  
                    else  
                        Log.d("MODEL", "Error: ${response.message()}")  
                }  
            }  
        }  
    }  
    override fun onCleared() {  
        super.onCleared()  
        job.cancel()  
    }  
}
```

Coroutines+MVVM+Retrofit

(view)

```
class MainActivity : AppCompatActivity() {
    lateinit var viewModel: ListViewModel
    private val listAdapter = ListAdapter(arrayListOf())
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        viewModel = ViewModelProviders.of(this).get(ListViewModel::class.java)
        ← viewModel.fetch()
        listView.apply {
            layoutManager = LinearLayoutManager(context)
            adapter = listAdapter
        }
        observeViewModel()
    }
    fun observeViewModel() {
        ← viewModel.staty.observe(this, Observer { staty ->
        → staty?.let { // if staty != null ...
            countriesList.visibility = View.VISIBLE
            listAdapter.updateCountries(it)
        }
    })
} }
```




Glide

(knížnica na sťahovanie obrázkov)

```
Glide.with(this)
    .load(IMAGE_URL)
    .into(imageView)
```

- List adaptér používa Glide na čítanie obrázkov z URL
- <https://medium.com/@vlonjatgashi/using-glide-with-kotlin-5e345b557547>
- build.gradle:

```
apply plugin: 'kotlin-kapt' // kotlin anotation processing tool

dependencies {
    implementation 'com.github.bumptech.glide:glide:4.4.0'
    kapt 'com.github.bumptech.glide:compiler:4.4.0'
}
```

- kód:

```
import com.bumptech.glide.Glide

val options=RequestOptions().error(R.mipmap.ic_launcher_round)
Glide.with(this)
    .setDefaultRequestOptions(options)
    .load(country.flag)
    .into(imageView)
```

Glide

(spracovanie obrázku)

```
Glide.with(this)
    .load(IMAGE_URL)
    .into(imageView)
```

jednoduché demo:

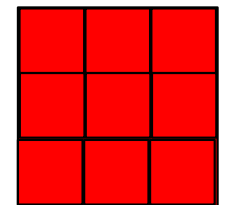
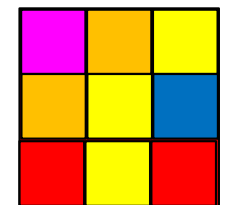
```
CoroutineScope(Dispatchers.Main).launch {
    Glide.with(this@MainActivity)
        .asBitmap()
        .error(R.mipmap.ic_launcher_round)
        .load(IMAGE_URL)
        .into(object : CustomTarget<Bitmap>() {
            override fun onResourceReady(
                resource: Bitmap,
                transition: Transition<in Bitmap>? ) {
                val filteredBitmap = toBlackAndWhite(resource)
                progressBar.visibility = View.GONE
                imageView.setImageBitmap(filteredBitmap)
                imageView.visibility = View.VISIBLE
            }
            override fun onLoadCleared(placeholder: Drawable?) { }
        })
}
```

Pixelizácia

úloha z cvičenia A

Implementácia bez väčšej logiky, celý štvorec má farbu bodu v dolnom spodnom rožku štvorca – vylepšovať je čo...

```
fun pixelize(pixelSize : Int, source: Bitmap):  
    Bitmap {  
    val w = source.width  
    val h = source.height  
    val bitmapOut = Bitmap.createBitmap(w, h, Bitmap.Config.RGB_565)  
    (0 until h step pixelSize).forEach { y->  
        (0 until w step pixelSize).forEach { x->  
            val color = source.getColor(x, y).toArgb()  
            (0 until pixelSize).forEach { dx->  
                (0 until pixelSize).forEach { dy ->  
                    val xx = Math.max(0, x - dx)  
                    val yy = Math.max(0, y - dy)  
                    bitmapOut.setPixel(xx,yy, color)  
                }  
            }  
        }  
    }  
    }  
    return bitmapOut
```



GSM-Retrofit

GSM lokalizácia - z cvičenia B

<https://eu1.unwiredlabs.com/v2/process.php>

```
{
  "token": "95b2941777892d",
  "mcc": 231,
  "mnc": 2,
  "cells": [{
    "lac": 1,
    "cid": 31441
  }],
  "address": 1
}
```

```
{
  "status": "ok",
  "balance": 97,
  "lat": 48.14875,
  "lon": 17.06679,
  "accuracy": 837,
  "address": "Botanická, Švédske domky, Bratislava, Karlova Ves, Bratislava, Region of Bratislava, 841 04, Slovakia"
}
```

V prednáške o polohe sme narazili na problem ako GSM súradnice (mcc,mnc,lac,cid) prekožiť do latitude-longitude volaním externého servisu

- potrebujeme mu ([process.php](https://eu1.unwiredlabs.com/v2/process.php)) poslať,prečítať json-dáta, cez HTTP-POST
 - ak zavrhneme riešenie, že *"lepíme reťazce" do JSON a vyhládávame v ňom podstringy, ...*
 - riešenie založené na json knižnici `android.util.JsonReader/JsonWriter` (ukážeme si)
 - riešenie založené na Gson knižnici (konvertuje json do objektu cez Java reflection model)
 - Gson converter priamo v Retrofit
- nesmieme to robiť v hlavnom vlákne, lebo to môže trvať...
 - riešenie pomocou AsyncTask (old-school – môžete nájsť v MyGSMLocation.zip)
 - riešenie pomocou coroutin (new-wave)

ako z JSON-vzorky dát to vyrobiť Kotlin Class ?

- podporuje to priamo AS plugin
Json to Kotlin Class

```
data class GSMRequest(
  val address: Int,
  val cells: List<Cell>,
  val mcc: Int,
  val mnc: Int,
  val token: String)
```

```
data class Cell(
  val cid: Int,
  val lac: Int)
```

```
data class GSMResponse(
  val accuracy: Int,
  val address: String,
  val balance: Int,
  val lat: Double,
  val lon: Double,
  val status: String
)
```

GSMRetrofit



mcc,mnc,
lac,cid



latitude
longitude

LocationAPI.org

```
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 dostanete kľúč (token),
- 95b2941777892d (keď toto čítate, ešte platí 😊)

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

Request: 1 cell | 3 cells | 7 cells

```
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 }
```

Response:

```
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>

```
val request = GSMRequest(
  token = "95b2941777892d",
  mcc = mcc,
  mnc = mnc,
  cells = listOf(Cell(lac = lac, cid = cid)),
  address = 1
)
```

LocationAPI z aplikácie

- potrebujeme urobiť http-POST request na <http://locationapi.org/v2/process.php>
- keďže to niečo trvá, nesmieme to robiť v hlavnom vlákne – coroutine
- do tela dotazu (requestu) potrebujeme zakódovať (cellID, lac, mcc, mnc + môj token) hoc jednoduchý, ale predsa-len JSON objekt
- z tela odpovede (responzu) potrebujeme dekodovať hoc jednoduchý, ale JSON objekt, t.j. prečítať latitude-longitude

Request: 1 cell | 3 cells | 7 cells

```
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 }
```

Response:

```
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 }
```

GSM-Retrofit

<https://eu1.unwiredlabs.com/v2/process.php>

```
{
  "token": "95b2941777892d",
  "mcc": 231,
  "mnc": 2,
  "cells": [{
    "lac": 1,
    "cid": 31441
  }],
  "address": 1
}
```

GSMRequest

```
{
  "status": "ok",
  "balance": 97,
  "lat": 48.14875,
  "lon": 17.06679,
  "accuracy": 837,
  "address": "Botanická,
Švédske domky, Bratislava,
Karlova Ves, Bratislava,
Region of Bratislava, 841 04,
Slovakia"
}
```

GSMResponse

- JSON to Kotlin Class
- build.gradle

```
implementation 'com.google.code.gson:gson:2.8.5'
implementation 'com.squareup.retrofit2:retrofit:2.6.2'
implementation 'com.squareup.retrofit2:converter-gson:2.6.2'
```

- toto si dáme vygenerovať pluginom JSON to Kotlin Class

```
data class Cell(
  val cid: Int,
  val lac: Int
)
```

ak interné mená zodpovedajú JSON tagom,
tak neriešime `@SerializedName`

```
data class GSMRequest(
  val address: Int,
  val cells: List<Cell>,
  val mcc: Int,
  val mnc: Int,
  val token: String
)
```

```
data class GSMResponse(
  val accuracy: Int,
  val address: String,
  val balance: Int,
  val lat: Double,
  val lon: Double,
  val status: String
)
```

GSMRetrofit



Rest API

retrofit callback

```
interface RestApiInterface {  
    @Headers("Content-Type: application/json")  
    @POST("process.php")  
    fun gsm2latlong(@Body gsmRequest: GSMRequest): Call<GSMResponse>  
}
```

```
class RestApiService {  
    suspend  
    fun gsm2latlong(gsmRequest: GSMRequest, onResult: (GSMResponse?) -> Unit){  
        val retrofit = ServiceBuilder.get()  
        retrofit.gsm2latlong(gsmRequest).enqueue(  
            object : Callback<GSMResponse> {  
                override fun onFailure(call: Call<GSMResponse>, t: Throwable) {  
                    onResult(null) ← null  
                }  
                override fun onResponse(call: Call<GSMResponse>,  
                    response: Response<GSMResponse>) {  
                    val resp = response.body() ← !=null  
                    onResult(resp)  
                }  
            } ) } } }
```


Service Builder



```
object ServiceBuilder {  
    private val client = OkHttpClient.Builder().build()  
  
    suspend  
    fun get(): RestApiInterface =  
        Retrofit.Builder()  
            .baseUrl("httpS://eu1.unwiredlabs.com/v2/")  
            .addConverterFactory(GsonConverterFactory.create())  
            .client(client)  
            .build()  
            .create(RestApiInterface::class.java)  
}
```


Volanie - bez corutiny

```
data class Cell(  
    val cid: Int,  
    val lac: Int  
)
```

```
data class GSMRequest(  
    val address: Int,  
    val cells: List<Cell>,  
    val mcc: Int,  
    val mnc: Int,  
    val token: String  
)
```

```
val request = GSMRequest(  
    token = "95b2941777892d",  
    mcc = mcc,  
    mnc = mnc,  
    cells = listOf(Cell(lac = lac, cid = cid)),  
    address = 1  
)
```

```
class RestApiService {  
    suspend  
    fun gsm2latlong(gsmRequest: GSMRequest,  
        onResult: (GSMResponse?) -> Unit)
```

```
    val apiService = RestApiService()  
    val response = apiService.gsm2latlong(request) {  
        response ->  // toto je onResult  
        if (response != null) {  
            Log.d(TAG, "${response.lat}, ${response.lon}")  
            latTV.text = response.lat.toString()  
            longTV.text = response.lon.toString()  
        } else  
            Log.d(TAG, "response is null")  
    }
```

```
data class GSMResponse(  
    val accuracy: Int,  
    val address: String,  
    val balance: Int,  
    val lat: Double,  
    val lon: Double,  
    val status: String  
)
```

Volanie – s corutinou

```
val request = GSMRequest(
    token = "95b2941777892d",
    mcc = mcc,
    mnc = mnc,
    cells = listOf(Cell(lac = lac, cid = cid)),
    address = 1
)
CoroutineScope(Dispatchers.IO).Launch {
    val apiService = RestApiService()
    val response = apiService.gsm2latlong(request) {
        response -> // toto je onResponse
        if (response != null) {
            Log.d(TAG, "${response.lat}, ${response.lon}")
            latTV.text = response.lat.toString()
            longTV.text = response.lon.toString()
        } else
            Log.d(TAG, "response is null")
    }
}
```

```
class RestApiService {
    suspend
    fun gsm2latlong(gsmRequest: GSMRequest,
        onResponse: (GSMResponse?) -> Unit)
```

GUI len ako Dispatchers.Main

```
val request = GSMRequest(
    token = "95b2941777892d",
    mcc = mcc,
    mnc = mnc,
    cells = listOf(Cell(lac = lac, cid = cid)),
    address = 1
)
CoroutineScope(Dispatchers.IO).Launch {
    val apiService = RestApiService()
    val response = apiService.gsm2latlong(request) {
        response -> // toto je onResult
        if (response != null) {
            Log.d(TAG, "${response.lat}, ${response.lon}")
            CoroutineScope(Dispatchers.Main).Launch {
                latTV.text = response.lat.toString()
                longTV.text = response.lon.toString()
            }
        } else
            Log.d(TAG, "response is null")
    }
}
```

WifiLocation

(cvičenie-B)

V skratke: TelephonyManager nahrad' WifiManagerom a uprav jsony



AP-mac
address



latitude
longitude

Projekt GSMRetrofit získava od TelephonyManagera informácie polohe vo forme štvorice cid, lac, mcc, mnc. Prerobte tento projekt tak, aby ste zisťovali informáciu o polohe od WifiManagera (nahrad'te Telephony za Wifi Managera, kód sme mali pred 2 týždňami). Vytvorte zodpovedajúci json, ktorý obsahuje mac-adresu, napr. najsilnejšieho access pointu. Tento json otočte cez uvedený servis, a získajte tak polohu zariadenia. Keď tu budete testovať, overte si, napr. ručne, že MAC-ADDRESS šeho access-pointu je v databáze. Inak nebudete dostávať priaznivú odpoveď, a teda ani polohu. Keď si takéto APčko nájdete, urobte drobné video o funkčnosti vašej appky, aby sme pri hodnotení nemuseli hľadať vhodný Access point. Ak také APčko nevidíte, asi úloha nie je vhodná pre vás.

V projekte musíte zameniť TelephonyManagera za WifiManagera, predefinovať data class zodpovedajúci GSMRequestu na niečo ako WifiRequest, a vygenerovať si vlastný API-key u služby <https://eu1.unwiredlabs.com/v2/process.php>.

Vytvorenie (malého) JSON objektu

knižnica JsonWriter/JsonReader

```
val sw = StringWriter()
```

```
val jw = JsonWriter(sw)
```

```
try {
```

```
    jw.beginObject() -- {
```

```
        jw.name("token").value(token_locationAPIORG)
```

```
        jw.name("mcc").value(mcc)
```

```
        jw.name("mnc").value(mnc)
```

```
        jw.name("cells")
```

```
        jw.beginArray() -- [
```

```
            .beginObject() -- {
```

```
                jw.name("cid").value(cid)
```

```
                jw.name("lac").value(lac)
```

```
                jw.name("signal").value(-60)
```

```
                jw.name("tA").value(13)
```

```
            jw.endObject().endArray().endObject().close() -- } ] }
```

```
import android.util.JsonWriter
```

Request: 1 cell | 3 cells | 7 cells

```
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 }
```

Project:MyGSMLocation.zip

Dekódovanie (malého) JSON

knižnica JsonWriter/JsonReader

```
import android.util.JsonReader
```

```
val sr = StringReader(result)
```

```
val jr = JsonReader(sr)
```

```
jr.beginObject() -- {
```

```
    jr.nextName() -- skip: "status"
```

```
    jr.nextString()      -- skip: "ok"
```

```
    jr.nextName() -- skip: "balance"
```

```
    jr.nextInt()  -- skip: 45
```

```
    jr.nextName() -- skip: "lat"
```

```
    lat = jr.nextDouble()
```

```
    jr.nextName() -- skip: "lon"
```

```
    lng = jr.nextDouble()
```

```
    jr.nextName() -- skip: "accuracy"
```

```
    accur = jr.nextInt()
```

Response:

```
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 }
```

GSON

(fromJson)

```
{
  "id": "1547257485",
  "name": "Peter Borovansky",
  "first_name": "Peter",
  "last_name": "Borovansky",
  "link": "http://www.facebook.com/
         peter.borovansky",
  "username": "peter.borovansky",
  "gender": "male",
  "locale": "cs_CZ"
}
```

Idea: k JSON objektu definujeme zodpovedajúcu (1:1) java triedu

Obmedzenia (viac <https://github.com/google/gson/blob/master/UserGuide.md>):

- mená JSON tagov sa musia zhodovať s java menami polí v triede

```
class FBHeader {
    public String id = "";
    public String name = "";
    public String first_name = "";
    public String last_name = "";
    public String link = "";
    public String username = "";
    public String gender = "";
    public String locale = "";
}
```

```
import com.google.gson
```

```
Gson gson = new GsonBuilder().create();
```

```
FBHeader header = gson.fromJson(jsonstring, FBHeader.class);
```




FB Friends

(fromJson)

```
{ "data":  
  [ { "name": "Zuzka B...", "id": "582749468" },  
    { "name": "Lubica K...", "id": "583024903" },  
    { "name": "Barbora F...", "id": "632007063" },  
  ],  
  "paging": { "next": "https://graph.facebook.com/15..." }
```

```
class FBFriends { // dvojica  
    public FBPairs[] data = null;  
    public FB Paging paging = null; }  
class FB Pairs {    // dvojica  
    public String name = "";  
    public String id = ""; }  
class FB Paging {    // singleton  
    public String next = ""; }
```

```
import com.google.gson
```

```
Gson gson = new GsonBuilder().create();  
FBFriends friends = gson.fromJson(result, FBFriends.class);  
if (friends != null) {  
    if (friends.data != null)  
        for (int i = 0; i < friends.data.length; i++)  
            if (friends.data[i] != null)  
                tv.append(friends.data[i].name + ",");  
}
```

GSON – ako to funguje ?

Reflexivita

Ukázali sme

- `fromJson` (do Javy)
- ale analogicky funguje
- `toJson` (z Javy)

`org.json`

vs.

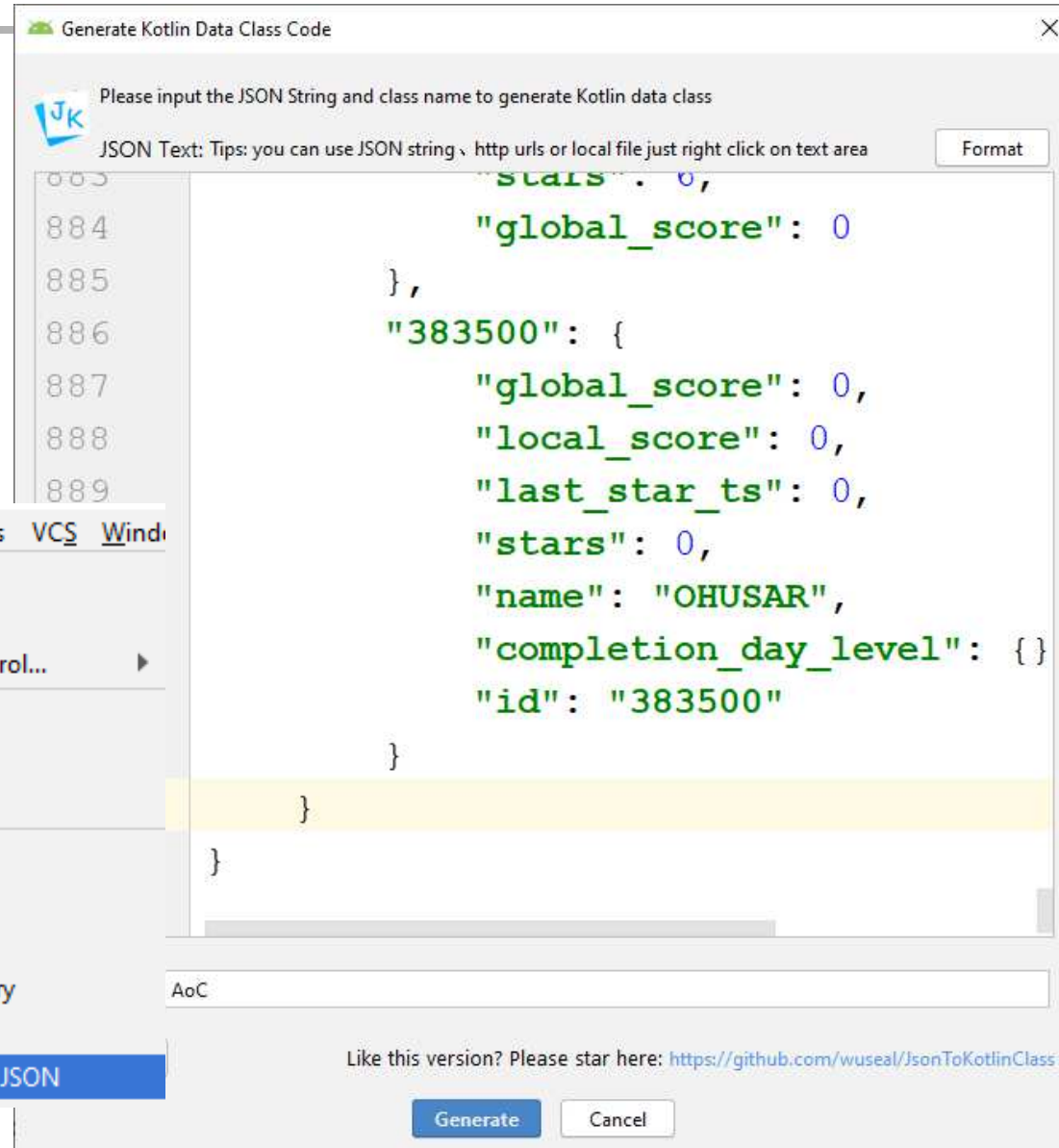
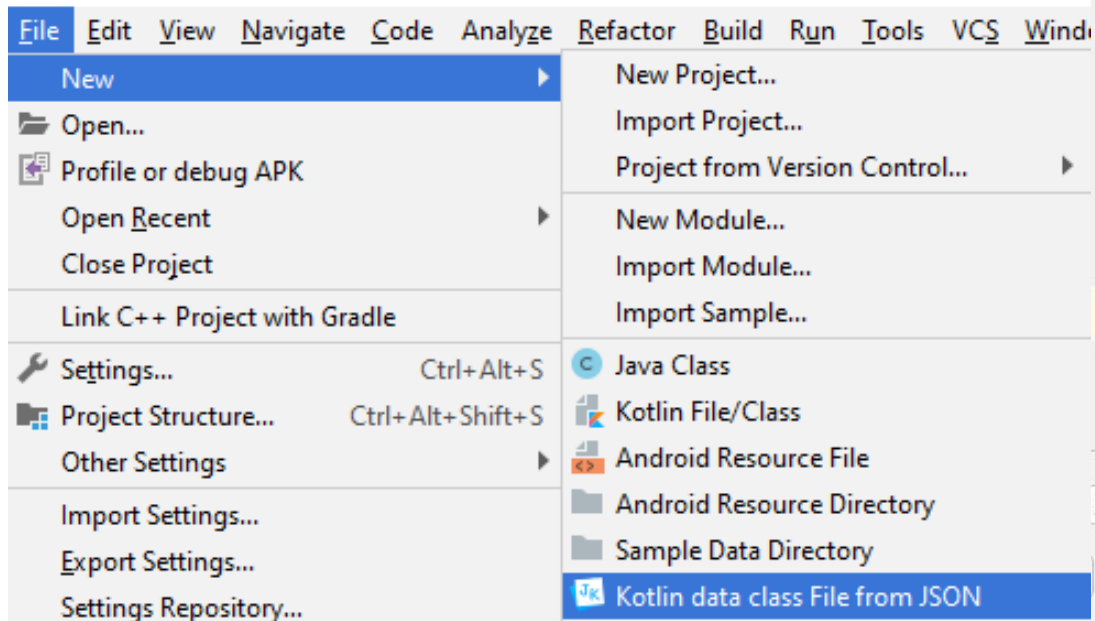
`com.google.gson`



Taschen Books

Plugin JSON to Kotlin Class

- z daného JSON vytvorí definíciu Kotlin tried
- potom stačí zavolať fromJson prekonvertuje vám json-string do dátovej štruktúry



Kde zohnať zaujímavý json

(JSON je vraj čitateľnejší ako xml)

```
{ "id": "229344", "members": { "633850": { "stars": 2, "id": "633850", "completion_d...  
{"id": "50608", "completion_day_level": {"2": {"1": {"get_star_ts": "1575323440"}, "2":  
l_score": 0, "stars": 4}, "387689": { "completion_day_level": { "id": "387689", "last_star_  
Lachova", "last_star_ts": "1575317389", "global_score": 0, "id": "284148", "completion_d  
,"289473": { "stars": 0, "completion_day_level": { }, "id": "289473", "global_score": 0, "last  
Křuka", "global_score": 0, "local_score": 0, "stars": 0, "id": "133669", "stars": 6, "local_score":  
Onody", "id": "133669", "completion_day_level": { "2": {"2": {"get_star_ts": "1575333802"  
"1575400260"} } } }, "440747": { "name": "Ivan  
Martynovskiy", "last_star_ts": "1575401728", "global_score": 0, "local_score": 177, "comple  
{"2": {"get_star_ts": "1575401728"}, "1": {"get_star_ts": "1575400101"} } }, "id": "4407  
"1575185453"}, "1": {"get_star_ts": "1575184775"}, "2": {"2": {"get_star_ts": "15752  
Tomsanyi", "635829": { "completion_day_level": { "3": {"1": {"get_star_ts": "15753505  
s": "1575265098"} } } }, "id": "635829", "name": "Jozef Kubik", "last_star_ts": "1575350619  
Gergel", "last_star_ts": "1575359599", "global_score": 0, "local_score": 210, "completion_  
{"get_star_ts": "1575184467"}, "2": {"2": {"get_star_ts": "1575184888"} } }, "id": "425420", "  
:"432600", "last_star_ts": 0, "name": "Maroš  
Malý", "global_score": 0, "local_score": 0, "stars": 0, "id": "205219", "completion_  
_day_level": { "id": "413457", "398095": { "last_star_ts": 0, "name": "  
Kerak", "global_score": 0, "local_score": 143, "completion_day_level": { "2": {"2": {"get_s  
_day_level": { "1": {"2": {"get_star_ts": "1575379220"} } }, "id": "387446", "382973": { "completion_day_level": { }, "id": "382973", "last_star_ts": 0, "name": "Boris  
Silný", "global_score": 0, "local_score": 0, "stars": 0, "id": "633816", "stars": 6, "completion_day_level": { "3": {"2": {"get_star_ts": "1575355116"}, "1": {"get_star_  
_star_ts": "1575177501"}, "1": {"get_star_ts": "1575177229"} } }, "id": "633816", "last_star_ts": "1575355116", "global_score": 0, "name": null, "local_score": 230, "357988": { "last_star_ts": "1575384913", "name": null, "global_score": 0, "local_score": 181, "completion_  
_day_level": { "1": {"2": {"get_star_ts": "1575192085"}, "1": {"get_star_ts": "1575191610"}, "2": {"2": {"get_star_ts": "1575280368"}, "1": {"get_star_ts": "1  
75279807"}, "3": {"2": {"get_star_ts": "1575384913"}, "1": {"get_star_ts": "1575384157"} } }, "id": "357988", "s  
tars": 6, "442294": { "stars": 0, "local_score": 0, "last_star_ts": 0, "name": "mrkvost", "global_score": 0, "id": "442294", "completion_day_level": { }, "257369": {  
Drabecky", "id": "257369", "completion_day_level": { }, "stars": 0, "727545": { "completion_day_level": { }, "id": "727545", "last_star_ts": 0, "global_score": 0, "ame": null, "local_score": 0, "stars": 0, "406017": { "id": "406017", "completion_day_level": { "2": {"1": {"get_star_  
ts": "1575264301"}, "2": {"get_star_ts": "1575264561"}, "1": {"1": {"get_star_ts": "1575176687"}, "2": {"get_star_ts": "1575177031"}, "3": {"2": {"get_star_  
ts": "1575350314"}, "1": {"get_star_ts": "1575349931"} } }, "local_score": 249, "global_score": 0, "last_star_ts"  
:"1575350314", "name": "Erik  
Szalay", "stars": 6, "633833": { "stars": 0, "id": "633833", "completion_day_level": { }, "local_score": 0, "global_score": 0, "last_star_ts": 0, "name": null, "49729 : { "stars": 0, "completion_day_level": { }, "id": "49729", "name": "Seluwin", "last_star_ts": 0, "global_score": 0, "loc  
al_score": 0, "452327": { "stars": 4, "local_score": 89, "global_score": 0, "last_star_ts": "1575359071", "name": "Andrea  
Spišáková", "id": "452327", "completion_day_level": { "2": {"1": {"get_star_ts": "1575318972"}, "2": {"get_star_ts": "1575359071"}, "1": {"2": {"get_star_ts": "1575316042"}, "1": {"get_star_ts": "1575315442"} } }, "382474": { "completion_day_level": { "3": {"1": {"get_s  
tar_ts": "1575367655"}, "2": {"get_star_ts": "1575380238"}, "2": {"1": {"get_star_ts": "1575296554"}, "2": {"get_star_ts": "1575297391"}, "1": {"2": {"get_star_ts": "1575294741"}, "1": {"get_star_ts": "1575290505"} } }, "id": "382474", "last_star_ts": "1575380238", "gl  
bal_score": 0, "name": "linda Jurkasova", "local_score": 170, "stars": 6, "230440": { "local_score": 196, "last_star_ts": "1575355079", "name": "Tomáš  
Bočinec", "global_score": 0, "id": "230440", "completion_day_level": { "3": {"2": {"get_star_ts": "1575355079"}, "1": {"get_star_ts": "1575355063"}, "1": {"1": {"get_star_ts": "1575222029"}, "2": {"get_star_ts": "1575222656"}, "2": {"2": {"get_star_ts": "1575275021"},  
"1": {"get_star_ts": "1575274812"}, "stars": 6, "241618": { "id": "241618", "completion_day_level": { "2": {"1": {"get_star_ts": "1575272186"}, "2": {"get_star_ts": "1575279106"}, "1": {"1": {"get_star_ts": "1575184259"}, "2": {"get_star_ts": "1575184876"}, "3": {"1": {"  
get_star_ts": "1575369676"}, "2": {"get_star_ts": "1575375683"}, "local_score": 207, "last_star_ts": "1575375683", "global_score": 0, "name": null, "stars": 6, "289854": { "last_star_ts": "1575375697", "global_score": 0, "name": "Miki  
Hermann", "local_score": 206, "completion_day_level": { "2": {"2": {"get_star_ts": "1575267648"}, "1": {"get_star_ts": "1575265752"}, "1": {"2": {"get_star_ts": "1575185929"}, "1": {"get_star_ts": "1575185448"}, "3": {"1": {"get_star_ts": "1575369838"}, "2": {"get_st  
ar_ts": "1575375697"} } }, "id": "289854", "stars": 6, "677144": { "local_score": 196, "last_star_ts": "1575367533", "name": "Vsetky  
Borufky", "local_score": 0, "id": "677144", "completion_day_level": { "3": {"2": {"get_star_ts": "1575367533"}, "1": {"get_star_ts": "1575357449"}, "1": {"2": {"get_star_ts": "1575241741"}, "1": {"get_star_ts": "1575241062"}, "2": {"1": {"get_star_ts": "1575264300"},  
"2": {"get_star_ts": "1575267832"}, "stars": 6, "340516": { "stars": 6, "id": "340516", "completion_day_level": { "3": {"1": {"get_star_ts": "1575351837"}, "2": {"get_star_ts": "1575376586"}, "2": {"1": {"get_star_ts": "1575263547"}, "2": {"get_star_ts": "1575263974"}  
_day_level": { "1": {"2": {"get_star_ts": "1575182013"}, "1": {"get_star_ts": "1575181806"}, "local_score": 233, "last_star_ts": "1575376586", "name": "Lukáš Gajdošeň", "global_score": 0, "440518": { "last_star_ts": 0, "global_score": 0, "name": "Marek  
Stachera", "local_score": 0, "completion_day_level": { }, "id": "440518", "394723": { "local_score": 0, "last_star_ts": 0, "global_score": 0, "name": null, "id": "394723", "completion_day_level": { }, "stars": 0, "229344": { "stars": 6, "id": "229344", "completion_day_le  
vel": { "3": {"2": {"get_star_ts": "1575351770"}, "1": {"get_star_ts": "1575350735"}, "1": {"1": {"get_star_ts": "1575176691"}, "2": {"get_star_ts": "1575177020"}, "2": {"2": {"get_star_ts": "1575264839"}, "1": {"get_star_ts": "1575264081"} } }, "local_score": 245, "glob  
al_score": 0, "last_star_ts": "1575351770", "name": "Peter BOROVANSKY", "133724": { "stars": 4, "local_score": 104, "last_star_ts": "1575328931", "name": "Alexka  
Nyitraiová", "global_score": 0, "id": "133724", "completion_day_level": { "1": {"1": {"get_star_ts": "1575210606"}, "2": {"get_star_ts": "1575211974"}, "2": {"2": {"get_star_ts": "1575328931"}, "1": {"get_star_ts": "1575327359"} } }, "644573": { "id": "644573", "completi  
on_day_level": { "1": {"1": {"get_star_ts": "1575176824"}, "2": {"get_star_ts": "1575177172"}, "2": {"1": {"get_star_ts": "1575263803"}, "2": {"get_star_ts": "1575264338"}, "3": {"2": {"get_star_ts": "1575351620"}, "1": {"get_star_ts": "1575351021"} } }, "local_score":  
245, "last_star_ts": "1575351620", "name": "Midisuba", "global_score": 0, "stars": 6, "541831": { "id": "541831", "completion_day_level": { }, "local_score": 0, "global_score": 0, "last_star_ts": 0, "name": "Michal  
Knor", "stars": 0, "253915": { "last_star_ts": "1575311956", "name": "Michal  
Winczer", "global_score": 0, "local_score": 100, "completion_day_level": { "2": {"2": {"get_star_ts": "157531
```




Alternatívne domáca úloha

pre AoC pozitívnych

- <https://adventofcode.com/>
- Leaderboardu 229344-861e5094.

Vašou úlohou je v aplikácii prečítať tento JSON a interpreterovať.

Čo sa chápe pod interpretáciou:

- v master view zobrazíte v jednom riadku ListView mená (ID ak name=null) s počtom hviezd,
- po kliknutí na konkrétny riadok (hráča) sa zobrazí detail view, ktorý obsahuje dni a časy, kedy tento hráč vyriešil ktorú úlohu. JSON obsahuje timestampy vo formate long (napr. 1575234400), tie samozrejme prevedíte na čitateľný Date-Time formát. Pre jednoduchosť, dni sú dni adventu 1..25 a úlohy sú v každom dni len 2, teda 1..2.
- kvôli autorizácii musíte pochopiť cookies _gid, _ga – viac na konci prezentácie
- cookies získate po autentifikácii (prihlásení napr. cez google/github account)
- Cookies vylovíte v developer tools vášho obľúbeného browsera
- tie následne použijete v kóde, ktorý dotahuje json
- rozparsujete json a vymaľujete do aplikácie

AoC cookies

ako poznať, že ich mám

Advent of Code [About] [Events] [Shop] [Settings] [Log Out] Peter BOROVANSKY (AoC++) 12*
0x0000|2022 [Calendar] [AoC++] [Sponsors] [Leaderboard] [Stats]

This is your private leaderboard for Advent of Code 2022. You can remove a user by clicking the [X] next to their name. You can use a different [Ordering], manage your [Private Leaderboards], use an [API], or switch to another [Event].

Gold indicates the user got both stars for that day, silver means just the first star, and gray means none.

```
1111111111222222
1234567890123456789012345
1) 923 ***** Erik Szalay [X]
2) 896 ***** -1010011010- [X]
3) 892 ***** Peter BOROVANSKY (AoC++)
4) 890 ***** Jozef Kubík (AoC++) [X]
5) 870 ***** Jakub Murin [X]
6) 866 ***** Marian Kravec [X]
7) 854 ***** Dominika Miháľová [X]
8) 826 ***** Michal Winczer [X]
```

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application Adblock Plus

Filter URLs

Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	adventofcode.com	229344	document	html	6.11 kB	118.42 kB
200	GET	www.google...	analytics.js	229344:199 (scri...	js	cached	0 B
200	POST	www.google...	collect?v=1&_v=j98&aip=1&a=330388368&t=pageview&_s=1&dl=https:	analytics.js:43 (x...	plain	620 B	4 B
200	GET	adventofcode.com	favicon.png	FaviconLoader.j...	png	cached	5.16 kB
200	POST	stats.g.doubl...	collect?t=dc&aip=1&_r=3&v=1&_v=j98&tid=UA-69522494-1&cid=80370	analytics.js:43 (x...	plain	690 B	1 B

Headers Cookies Request Response Timings Security

Filter Cookies

Request Cookies

- _ga: "GA1.2.803700850.166860"
- _gid: "GA1.2.563218653.16696"
- session: "53616c7465645f5f0ee90b2d44d595830531c9e4feaa5c8120ae0955abba5b5"

<http://dai.fmph.uniba.sk/courses/VMA/>
<http://dai.fmph.uniba.sk/courses/VMA/ISLAND.JPG>
<http://dai.fmph.uniba.sk/courses/VMA/ISLAND2.JPG>
<http://dai.fmph.uniba.sk/courses/VMA/android/03Http/KOZA.JPG>
<http://dai.fmph.uniba.sk/~borovan/rosnicka/>



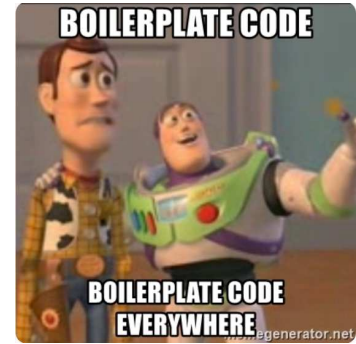
Retrofit Download Image

```
interface RetrofitInterface {  
    @GET  
    fun getImage(@Url url: String): Call<ResponseBody?>  
}
```

```
val retrofit = Retrofit.Builder()  
    .baseUrl("https://dai.fmph.uniba.sk/courses/VMA/")  
    .build()  
val retrofitInterface = retrofit.create(  
    RetrofitInterface::class.java)  
val request = retrofitInterface.getImage("ISLAND2.JPG")  
try {  
    val body: ResponseBody? = request.execute().body()  
    downloadImage(body)  
} catch (e : IOException) {  
    e.printStackTrace()  
    Toast.makeText(applicationContext, e.message,  
        Toast.LENGTH_SHORT).show()  
}
```

Retrofit Autentifikácia

Basic Authentication



```
class BasicAuthInterceptor(username:String, password:String): Interceptor {  
    private var credentials: String = Credentials.basic(username, password)  
  
    override fun intercept(chain: Interceptor.Chain): okhttp3.Response {  
        var request = chain.request()  
        request = request.newBuilder()  
            .header("Authorization", credentials).build()  
        return chain.proceed(request)  
    }  
}
```

OkHttpClient je implementácia HTTP & HTTP/2 klienta, umožňuje nám pridávať interceptory, napr. na autentifikáciu

```
val okclient = OkHttpClient.Builder()  
    .addInterceptor(BasicAuthInterceptor("java", "vaja"))  
    .build()  
  
val retrofit = Retrofit.Builder()  
    .baseUrl("http://dai.fmph.uniba.sk/courses/VMA/")  
    .client(okclient)  
    .build()  
  
val retrofitInterface = retrofit.create(RetrofitInterface::class.java)  
val request = retrofitInterface.getImage("ISLAND2.JPG")
```




Retrofit Download Image

Save response body to file

```
private fun downloadImage(body: ResponseBody?) {
    var count = 0
    val data = ByteArray(1024 * 4)
    val fileSize = body?.contentLength()
    val inputStream = BufferedInputStream(body?.byteStream(), 1024 * 8)
    val outputFile = File(Environment
        .getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLOADS),
        "downloaded.jpg")
    val outputStream = FileOutputStream(outputFile)
    var total = 0L
    var downloadComplete = false
    while (inputStream.read(data).also({ count = it }) != -1) {
        total += count.toLong()
        val progress = ((total*100).toDouble()/(fileSize?:100).toDouble()).toInt()
        updateNotification(progress)
        outputStream.write(data, 0, count)
        downloadComplete = true
    }
    onDownloadComplete(downloadComplete)
    outputStream.flush()
    outputStream.close()
    inputStream.close()
}
```

```
Android-10
<application
    android:requestLegacyExternalStorage="true"
```

Project:RetrofitDownloadImage.zip



Retrofit Download Image

Load image from downloaded file

```
val file = File(Environment.getExternalStoragePublicDirectory(  
    Environment.DIRECTORY_DOWNLOADS)  
    .getPath() + File.separator.toString() +  
    "downloaded.jpg")  
Picasso.get().load(file).into(imgView)
```

```
implementation 'com.squareup.picasso:picasso:2.71828'
```

```
Android-10  
<application  
    android:requestLegacyExternalStorage="true"
```

Project:RetrofitDownloadImage.zip

Static Google Maps

(život pred Google Maps API...)



Google Maps poskytujú API pre download statickej mapy, príklad.

<http://maps.googleapis.com//maps/api/staticmap?center=48.152177,17.07153&zoom=15&size=500x500&maptype=mobile&markers=48.152177,17.07153,ref/&key=AIzaSyCUUeYuihYGL8&sensor=false&>

Povinné polia v HTTP Requeste:

- center – stred mapy,
napr. “48.152177,17.07153”, resp. “Mlynska dolina, Bratislava”
- zoom – 0..21
- size – veľkosť obrázku
- formát – PNG (default), jpg, gif, ...
- key (25kAccess/day ☺) si vygenerujete tu: <https://console.developers.google.com/apis/https://developers.google.com/maps/documentation/staticmaps/https://developers.google.com/maps/documentation/maps-static/get-api-key>

iné, nepovinné polia:


- markers,
- path, ...

Alternativa

http://maps.googleapis.com/maps/api/staticmap?size=230x200&path=weight:3|color:blue|geodesic:true|Brisbane,Australia|Hong%20Kong|Moscow,Russia|London,UK|Reyjavik,Iceland|New%20York,USA|San%20Francisco,USA&key=AIzaSyCUU53EADiCbbSzFreLbD_FzEeYuihYGL8





Google Cloud Console


 Google Cloud Platform


My Project ▼


🔍

 Google Maps

 Overview

 APIs

 Metrics

 Support

APIs

In use APIs

Select an API to view details. Figures are for the last 30 days.

API ↑	Requests	Errors
Directions API	4	3
Distance Matrix API	0	0
Maps Embed API	0	0
Maps JavaScript API	0	0
Maps SDK for Android	0	0
Maps SDK for iOS	0	0
Maps Static API	139	127
Roads API	0	0



Reminder: To use the Maps Static API, you must enable billing. You can enable billing when you get your API key (see the [Quick guide](#)) or as a separate process (see [Usage and Billing](#)).



Disclaimer statement

(tento kurz neberie žiadnu zodpovednosť za straty na vašich účtoch)

! Requests per Day (QPD) limits have ended, effective June 11, 2018

For Existing Customers:

- If you are an existing customer (using the Google Maps Platform before June 11, 2018), you were billed under the previous plan until July 16, 2018.
- **Note:** Starting on June 11, 2018, Google began to roll out the removal of the default QPD limits on existing billing accounts. This process may take up to six weeks.
- If you rely on the default QPD limits to help you manage your cost of use, we recommend you **set your own daily limits** in the Google Cloud Platform Console, to override the default QPD limits set by Google. This will ensure that your preferred QPD limits remain in place.

For New Customers:

- If you are a new customer (using the Google Maps Platform starting on or after June 11, 2018), you were billed under the previous plan until July 16, 2018.
- **Note:** New customers began receiving unlimited QPD starting on June 11, 2018.
- To help you manage your cost of use, you can **set your own QPD limits** in the Google Cloud Platform Console.

Pricing



Maps Static API
Google

[OVERVIEW](#)

[PRICING](#)

[DOCUMENTATION](#)

[SUPPORT](#)

Category: [Maps](#)


Service name: static-maps-backend.go

About Google

Google's mission is to organize the world's information and make it universally accessible and useful. Through products and platforms like Search, Maps, Gmail, Android, Google Play, Chrome and YouTube, Google plays a meaningful role in the daily lives of billions of people.

Pricing

Flexible pricing that scale to fit your needs. Plus, get [USD200.00](#) in free usage for Maps, Routes, and Places every month.

	TIER 1	TIER 2
Static Maps 	USD2.00 /1K requests	USD1.60 /1K requests
	0 - 100K	100K+

Static Maps API v2

(V2 Upgrade Guide)

Tiež prišlo **API V2 Static Maps**

- <https://developers.google.com/maps/documentation/staticmaps/upgrade>

Static map creator:

- vygeneruje request <http://staticmapmaker.com/> , len si pridajte API Key:

- [http://maps.google.com/maps/api/staticmap?](http://maps.google.com/maps/api/staticmap?center=48.160020,17.075810&zoom=13&markers=mlynska+dolina,bratislava&size=400x400&sensor=TRUE_OR_FALSE)
[center=48.160020,17.075810&zoom=13&](http://maps.google.com/maps/api/staticmap?center=48.160020,17.075810&zoom=13&markers=mlynska+dolina,bratislava&size=400x400&sensor=TRUE_OR_FALSE)
[markers=mlynska+dolina,bratislava&size=400x400&](http://maps.google.com/maps/api/staticmap?center=48.160020,17.075810&zoom=13&markers=mlynska+dolina,bratislava&size=400x400&sensor=TRUE_OR_FALSE)
[sensor=TRUE OR FALSE](http://maps.google.com/maps/api/staticmap?center=48.160020,17.075810&zoom=13&markers=mlynska+dolina,bratislava&size=400x400&sensor=TRUE_OR_FALSE)





Retrofit request s parametrami

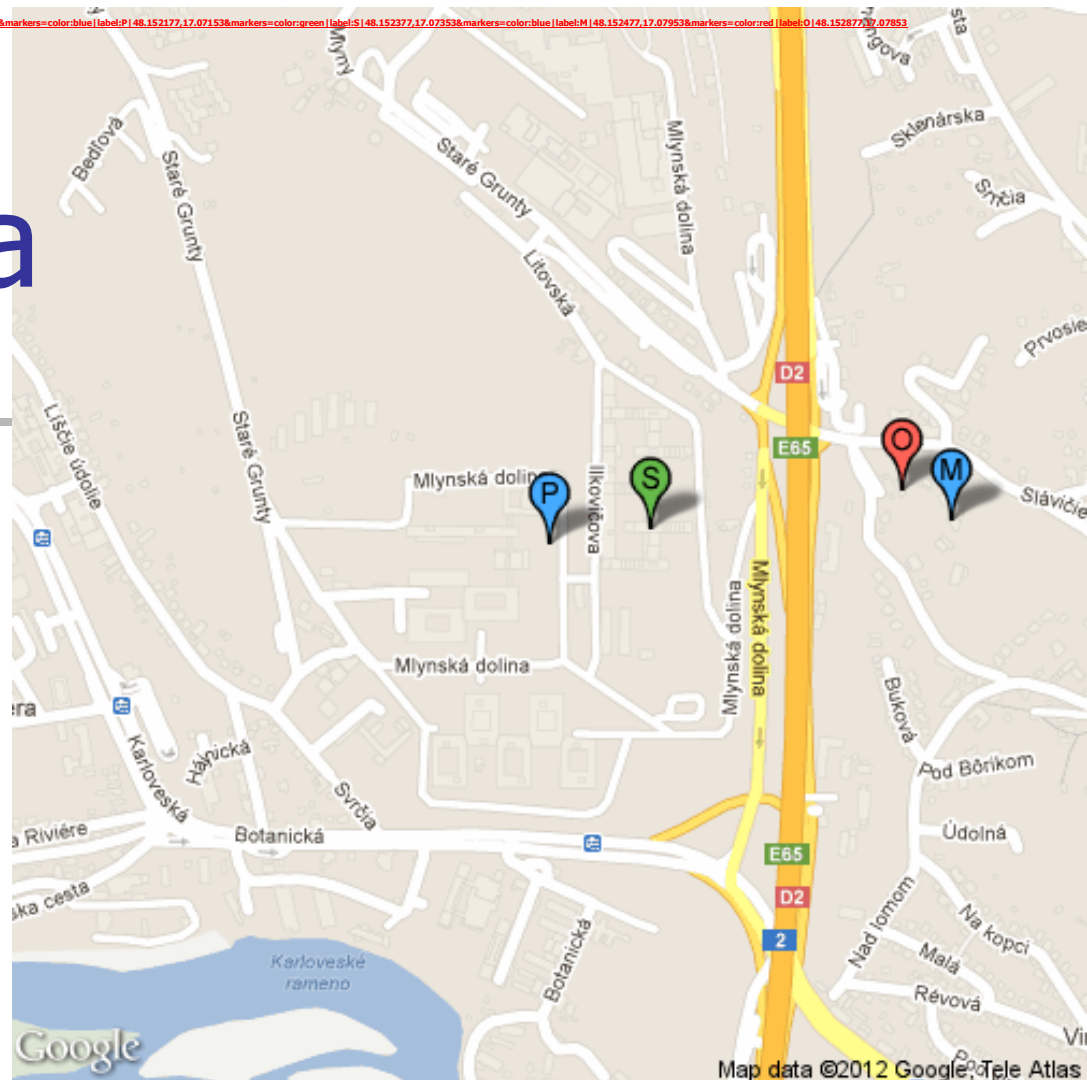
```
@GET
fun getStaticMap(@Url url: String,
                 @Query("center") lng : String,
                 @Query("zoom")    zoom : String,
                 @Query("size")    size : String,
                 @Query("markers") markers : String,
                 @Query("key")     key : String): Call<ResponseBody?>
```

```
val retrofit = Retrofit.Builder()
    .baseUrl("https://maps.google.com/")
    ...
val request = retrofitInterface.getStaticMap(
    "maps/api/staticmap",
    "48.160020,17.075810",
    "13",
    "480x480",
    "48.160020,17.075810",
    "AIzaS*****EeYuihYGL8"
)
```

Domáca úloha

(kamaráti na mape)

- napíšete aplikáciu, ktorá po zapnutí uploaduje súradnice užívateľa (lat,long) na server pomocou priloženého php scriptu.
- tabuľka obsahuje ID, DateTime, Latitude, Longitude,
- zobrazte posledné záznamy všetkých aktívnych (max.60 min. starý záznam) užívateľov na statickej mape farebne odlíšených markermi s iniciálkou
- GMAPs alebo Static GMAPs



maps.googleapis.com/maps/api/staticmap?center=48.152177,17.07153&zoom=15&size=500x500&maptype=mobile&key=AIzaSyCUU53EADiCbbSzFreLbD_FzEeYuihYGL8&sensor=false&markers=color:blue|label:P|48.152177,17.07153&markers=color:green|label:S|48.152377,17.07353&markers=color:blue|label:M|48.152477,17.07953&markers=color:red|label:O|48.152877,17.07853

Potrebné detaily k DÚ

<http://dai.fmph.uniba.sk/courses/VMA/android/php/PositionUpdate.php?name=Peter&lat=48.152177&long=17.07153>

```
<?php
$dbhost = 'kempelen.dai.fmph.uniba.sk';
$dbuser = 'androids';
$dbpass = 'HotelDiamantDunakility';
$conn = mysqli_connect($dbhost, $dbuser, $dbpass, $dbuser);
if(! $conn ) {
    die('Could not connect: ' . mysqli_connect_error());
}
$sql = "insert into myfriends (name, time, lati, longi) values
      ('".$_GET['name']."', NOW() ,".$_GET['lat'].",$_GET['long'].")";
$retval = mysqli_query( $conn, $sql );
if(! $retval ) {
    die('Could not update data: ' . mysqli_error());
}
echo "Insert successfull\n";
mysqli_close($conn);
?>
```

<http://kempelen.ii.fmph.uniba.sk/phpmyadmin/>

Potrebné detaily k DÚ

<http://dai.fmph.uniba.sk/courses/VMA/android/php/PositionSelect.php>

```
<?php
```

```
...
```

```
$sql = "select * from myfriends where time > now() - INTERVAL 1 DAY;";
```

```
$retval = mysqli_query( $conn, $sql );
```

```
if(! $retval ) {
```

```
    die('Could not select data: ' . mysqli_error());
```

```
}
```

```
$rows = array();
```

```
while($r = mysqli_fetch_assoc($retval)) {
```

```
    $rows[] = $r;
```

```
}
```

```
print json_encode($rows);
```

```
mysqli_close($conn);
```

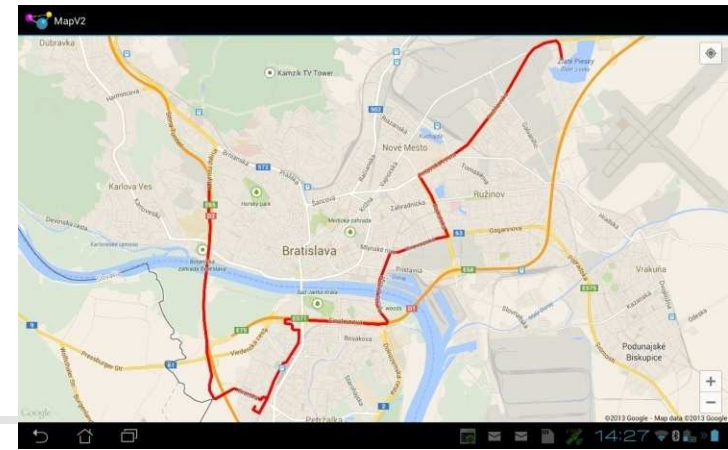
```
?>
```

```
[  
  {"name":"Peter","time":"2012-11-28  
15:41:50","lati":"48.3443","longi":"17.2322"},  
  {"name":"Peter","time":"2012-11-28  
19:27:20","lati":"48.1522","longi":"17.0715"},  
  {"name":"Silvia","time":"2012-11-28  
19:27:42","lati":"48.1522","longi":"17.0415"},  
  {"name":"Peter","time":"2012-11-28  
19:28:44","lati":"48.1522","longi":"17.0715"}  
]
```

<http://kempelen.ii.fmph.uniba.sk/phpmyadmin/>

GoogleDirections

(ako ide matfyzák na pláž)

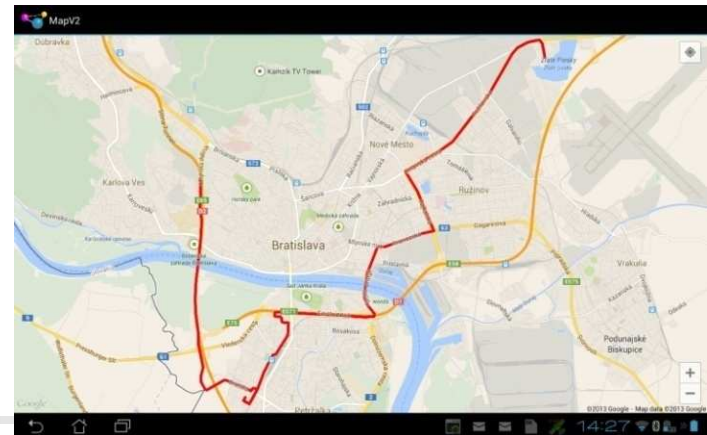


```
■ https://developers.google.com/maps/documentation/directions/
val startLocation = "Mlynska dolina, Bratislava"
                        .replace(" ", "+")
val endLocation = "Zlate piesky, Bratislava"
                        .replace(" ", "+")
val via = "AGEM, Kopcianska, Bratislava|
          Alza, Prievozska, Bratislava".replace(" ", "+")
val urlString =
    "http://maps.googleapis.com/maps/api/directions/json?" +
    "origin=" + startLocation +
    ",+&destination=" + endLocation +
    ",+&mode=bicycle,&avoid=highways,&waypoints=" + via
    +
    ",+&sensor=false"
```

Project:GMapDirections.zip

<http://maps.googleapis.com/maps/api/directions/json?origin=Mlynska+dolina,+Bratislava,+&destination=Zlate+piesky,+Bratislava,+&mode=bicycle,+&avoid=highways,+&waypoints=AGEM,+Kopcianska,+Bratislava|Alza,+Prievozska,+Bratislava,+&sensor=false>

Hľadaná trasa



vo formáte json, resp. xml, na vykreslenie trasy potrebujeme dekódovať
„overview_polyline“

```
{
  "maneuver" : "keep-right",
  "polyline" : {
    "points" : "urbeHyj{gBRAdDm@fGeA"
  },
  "start_location" : {
    "lat" : 48.187468,
    "lng" : 17.1846087
  },
  "travel_mode" : "DRIVING"
},
"via_waypoint" : []
],
"overview_polyline" : {
  "points" :
"cg}dHybfgBvCWbDEpQSn@B`B`@|AV|@Fb@AjAGdCg@lC_@bBQr@OzGH|I
O|BcA|FoAxKkBrASn@?t@Fr@PbDvAlA`@jBF~AQh@Sl@a@zCgCfA}@PUJS
AdA`DfCiCrJsC_CrC~BhCsJaDgCkAeA{AiAuB_BuDqCqE_CiJeEsAk@i@[
Wj@EZKnI_DGCMaEIOIK?MBsCHuGZsQb@s^PmNJ)Lb@yZ
\\oWVsMP{O@oCDwDA]?e@?s@Ca@UgAS]MM[M_@@[JUL{0x@cEpG]p@s@f@
iAmHaEcYoAiIqAcJc@oEaAyLGgAG{0MD]NWJWL}Aj@kAj@uBvAc@
\\CTkCtAgCpAyFtC_CjAgL`G}D|BiFIEkF`FyExDgC~BkCeJiAsDc@gB_C
gB{A_@a@q@}@_AoAk@iAq@uAmAaD}A}GUy@e@}Cc@kDY_DOyBO}FRILQjA
},
  "summary" : "Route 61",
  "warnings" : [],
  "waypoint_order" : [ 0, 1 ]
}
```

```
- <DirectionsResponse>
  <status>OK</status>
  - <route>
    <summary>Route 61</summary>
    + <leg></leg>
    + <leg></leg>
    + <leg></leg>
    <copyrights>Map data ©2013 Google</copyrights>
  - <overview_polyline>
    - <points>
      cg}dHybfgBvCWbDEpQSn@B`B`@|AV|@Fb@Aj
      \oWVsMP{O@oCDwDA]?e@?s@Ca@UgAS]MM
      \CTkCtAgCpAyFtC_CjAgL`G}D|BiFIEkF`FyExDg
    </points>
    <overview_polyline>
    <waypoint_index>0</waypoint_index>
    <waypoint_index>1</waypoint_index>
    + <bounds></bounds>
  </route>
</DirectionsResponse>
```

Project:GMapDirections.zip

<http://maps.googleapis.com/maps/api/directions/xml?origin=Mlynska+dolina,+Bratislava,+&destination=Zlate+piesky,+Bratislava,+&mde=bicycle,+&avoid=highways,+&waypoints=AGEM,+Kopcianska,+Bratislava|Alza,+Prievozska,+Bratislava,+&sensor=false>


```
{
  "geocoded_waypoints" :
  [
    {
      "geocoder_status" : "OK",
      "place_id" : "ChIJpYu5e_OLbEcRAZcCF8QGikC",
      "types" :
      [
        "route"
      ]
    },
    {
      "geocoder_status" : "OK",
      "partial_match" : true,
      "place_id" : "ChIJY1a5tZCjbEcRAnZpsjcUYLA",
      "types" :
      [
        "electronics_store",
        "establishment",
        "point_of_interest",
        "store"
      ]
    },
    {
      "geocoder_status" : "OK",
      "partial_match" : true,
      "place_id" : "ChIJY1a5tZCjbEcRAnZpsjcUYLA",
      "types" :
      [
        "electronics_store",
        "establishment",
        "point_of_interest",
        "store"
      ]
    }
  ],
  "routes" :
  [
    {
      "summary" : "E65",
      "warnings" : [],
      "waypoint_order" :
      [
        0,
        1
      ]
    }
  ]
}
```

```
package com.example.gmapdirections

data class json(
    val geocoded_waypoints: List<GeocodedWaypoint>,
    val routes: List<Route>,
    val status: String
)

data class Bounds(
    val northeast: Northeast,
    val southwest: Southwest
)

data class Distance(
    val text: String,
    val value: Int
)

data class Duration(
    val text: String,
    val value: Int
)

data class EndLocation(
    val lat: Double,
    val lng: Double
)

data class GeocodedWaypoint(
    val geocoder_status: String,
    val partial_match: Boolean,
    val place_id: String,
    val types: List<String>
)

data class Leg(
    val distance: Distance,
    val duration: Duration,
    val end_address: String,
    val end_location: EndLocation,
    val start_address: String,
    val start_location: StartLocation,
    val steps: List<Step>,
    val traffic_speed_entry: List<Any>,
    val via_waypoint: List<Any>
)
```

```
data class Northeast (
    val lat: Double,
    val lng: Double
)

data class OverviewPolyline(
    val points: String
)

data class Polyline(
    val points: String
)

data class Route(
    val bounds: Bounds,
    val copyrights: String,
    val legs: List<Leg>,
    val overview_polyline: OverviewPolyline,
    val summary: String,
    val warnings: List<Any>,
    val waypoint_order: List<Int>
)

data class Southwest(
    val lat: Double,
    val lng: Double
)

data class StartLocation(
    val lat: Double,
    val lng: Double
)

data class Step(
    val distance: Distance,
    val duration: Duration,
    val end_location: EndLocation,
    val html_instructions: String,
    val maneuver: String,
    val polyline: Polyline,
    val start_location: StartLocation,
    val travel_mode: String
)
```

Project:GMapDirections.zip

Ako sa dostať k ceste

```
{
  "routes" : [
    {
      "copyrights" : "Map data ©20
      .....
      "overview_polyline" : {
        "points" : "cg}dHybCc@kDY
      },
    },
  ],
  "status" : "OK"
}
```

```
val jsonOutput = response.toString()
val jsonObject = JSONObject(jsonOutput)
val routesArray = // z routes berieme prvú alternatívu
    jsonObject.getJSONArray("routes")
val route = routesArray.getJSONObject(0)
val poly = // pod route je uzol overview_polyline
    route.getJSONObject("overview_polyline")
polyline = poly.getString("points")
decodePoly(polyline)
```

ako dekódovať reťazec na zoznam bodov cesty ???

```
private List<LatLng> decodePoly(String encoded)
```

je mágia mimo rozsahu tejto prednášky

<http://stackoverflow.com/questions/15924834/decoding-polyline-with-new-google-maps-api>

decodePoly Prémia



Encode
Decode
Base64

HttpClient – POST

Ak potrebujem uploadovať väčšie dáta (napríklad fotku), použijeme POST

- vytvoríme aplikáciu, ktorá zosníma obrázok z kamery,
- zobrazí na display, kde ju môžeme pomenovať,
- pomocou HTTP-POST pošleme na server
- tam ju pomocou malého php-scriptu ukladáme do „galérie“ (adresára),

<?php

<https://dai.fmph.uniba.sk/courses/VMA/galeria/>

```
$base=$_REQUEST['image'];
```

```
$iname=$_REQUEST['iname'];
```

```
if ($iname != "") {
```

```
    $binary=base64_decode($base); // dekoduje z MIME base64
```

```
    $file = fopen($iname.'.jpg', 'wb');
```

```
    fwrite($file, $binary);
```

```
    fclose($file);
```

```
    echo "... OK ...";
```

ShotUploaderPhp=

```
} else
```

["https://dai.fmph.uniba.sk/courses/VMA/galeria/upload.php"](https://dai.fmph.uniba.sk/courses/VMA/galeria/upload.php)

```
    echo "... NULL NAME ...";
```

```
?>
```

download



HttpClient – POST

RetrofitServiceObject

Encode
Decode
Base64

```
interface RetrofitInterface {
    @FormUrlEncoded
    @POST("galeria/upload.php")
    suspend fun post(@Query("iname") iname: String,
                     @Field("image") image: String)
}

val okHttpClientInterceptor: OkHttpClient = OkHttpClient.Builder()
    .addInterceptor(BasicAuthInterceptor("java", "vaja"))
    .build()

object RetrofitServiceObject {
    private val BASE_URL = "httpS://dai.fmph.uniba.sk/courses/VMA/"
    fun post(): RetrofitInterface {
        return Retrofit.Builder()
            .baseUrl(BASE_URL)
            .addConverterFactory(GsonConverterFactory.create())
            .client(okHttpClientInterceptor)
            .build()
            .create(RetrofitInterface::class.java)
    }
}
```



HttpClient - POST

Encode
Decode
Base64

```
val byteArrayOutputStream = ByteArrayOutputStream()
bitmap.compress(Bitmap.CompressFormat.JPEG, 90,
    byteArrayOutputStream)
val encodedPhoto: String = Base64.encodeToString(
    byteArrayOutputStream.toByteArray(),
    Base64.DEFAULT)
```

```
val service = RetrofitServiceObject
```

```
CoroutineScope(Dispatchers.IO).launch {
    service.post().post(filename, encodedPhoto)
    withContext(Dispatchers.Main) {
        makeToast(...)
    }
}
```



Prémia

(len krátko-trvajúca)

Napište malú jednoúčelovú androidovskú aplikáciu, hoc aj bez GUI,..., ale ktorá z adresára

<https://dai.fmph.uniba.sk/courses/VMA/android/crackme/>

- zmažte súbor crack.me
- vytvorí tam súbor VašeMeno.VašePriezvisko (kvôli bodom za prémii), ktorý ale obsahuje vašu hackerskú prezývku, teda žiaden php-kód...
- všetky stopy (napr. pomocné súbory) po sebe upracte.

Poznámka: zvyšok webu prosím nechajte tak 😊

Deadline: čím skôr...

bodovanie 3 body (zlato), 2 (striebro), 1.5 (bronz), 1 (za účasť)



CRACKER



Cookie store

```
val cookieStore = BasicCookieStore()
val bc1 = BasicClientCookie("_gid", "GA1.2.126736...")
bc1.domain = "adventofcode.com"
bc1.path = "/" bc1.isSecure = false
cookieStore.addCookie(bc1)

val bc2 = BasicClientCookie("_ga", "GA1.2.11794137...")
bc2.domain = "adventofcode.com"
bc2.path = "/" bc2.isSecure = false
cookieStore.addCookie(bc2)

val bc3 = BasicClientCookie("session", "53616c7...")
bc3.domain = "adventofcode.com"
bc3.path = "/" bc3.isSecure = false
cookieStore.addCookie(bc3)

val ctx = BasicHttpContext()
ctx.setAttribute(ClientContext.COOKIE_STORE, cookieStore)

val httpresponse = httpClient.execute(httpget, ctx)
```