

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



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, ...

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

REST API

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

Retrofit

(Cvičenie)

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": [  
    48.66666666,  
    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)
        }
    })
} }
```


GSM-Retrofit

GSM lokalizácia - cvičenie



mcc,mnc,
lac,cid



latitude
longitude

```
{
  "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 vyhl'adáваме 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
)
```

CoroutinesRetrofit2



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

```
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
)
```

```
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>.