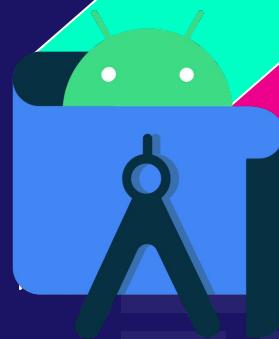


#10

Android Studio



Android Studio

" I have this awesome API i want to use,
How do i consume it? "



INNEHÅLLSFÖRTECKNING

01

Översikt

02

Setup,
WS / API,
Requests, URL,
Status koder

03

API structure,
Fetching &
Consuming

04

Uppgifter
&
Övningar



01

ÖVERSIKT

API, JSON, RETROFIT & INTERFACES



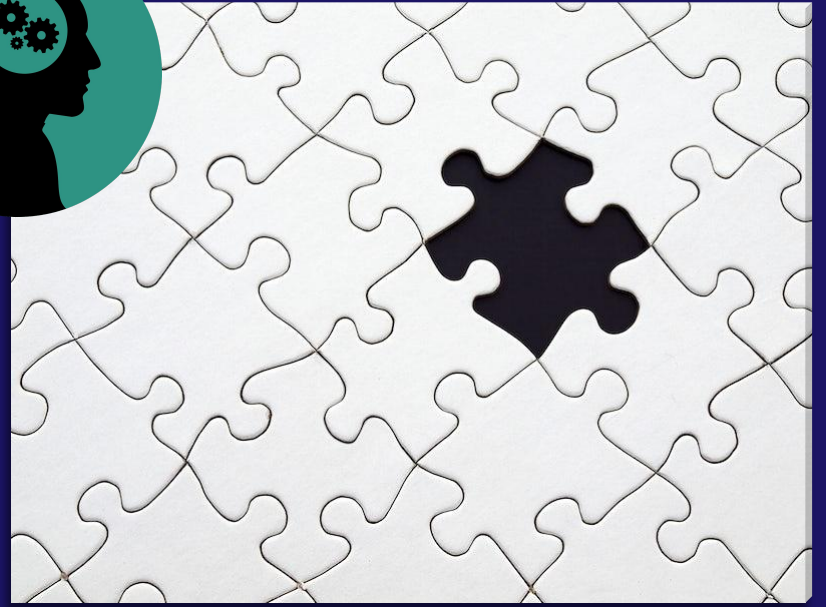
API

Innan vi börjar behöver vi ha förståelse över hur det hela fungerar.

- Skapa ett Request mot en URL
- Hämta datan
- Konvertera datan (JSON) till ett Objekt
- Arbeta med objektet



Frågor?





02

Setup, WS/API, Requests
URL; Status Codes

Retrofit





**“Retrofit manages the process of
receiving, sending, and
creating HTTP requests and
responses.”**

- Retrofit

[retrofit source](#)

Retrofit Dependency

2.9.0 is the latest as of 07-03-2023

GRADLE

```
implementation 'com.squareup.retrofit2:retrofit:(insert latest version)'
```

Retrofit requires at minimum Java 8+ or Android API 21+.

```
implementation 'com.squareup.retrofit2:retrofit:2.9.0'
```

<https://square.github.io/retrofit/>

Data Converter

CONVERTERS

By default, Retrofit can only deserialize HTTP bodies into OkHttp's `ResponseBody` type and it can only accept its `RequestBody` type for `@Body`.

Converters can be added to support other types. Six sibling modules adapt popular serialization libraries for your convenience.

- **Gson**: `com.squareup.retrofit2:converter-gson`
- **Jackson**: `com.squareup.retrofit2:converter-jackson`
- **Moshi**: `com.squareup.retrofit2:converter-moshi`
- **Protobuf**: `com.squareup.retrofit2:converter-protobuf`
- **Wire**: `com.squareup.retrofit2:converter-wire`
- **Simple XML**: `com.squareup.retrofit2:converter-simplexml`
- **JAXB**: `com.squareup.retrofit2:converter-jaxb`
- **Scalars** (primitives, boxed, and String): `com.squareup.retrofit2:converter-scalars`

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

Data Converter

GSON
(Converter)

Kotlin Object

```
1 {  
2   "endereco": {  
3     "cep": "31270901",  
4     "city": "Belo Horizonte",  
5     "neighborhood": "Pampulha",  
6     "service": "correios",  
7     "state": "MG",  
8     "street": "Av. Presidente Antônio Carlos, 6627"  
9   }  
10 }
```



Problem

Vad behöver vi för att arbeta med 'Fetch' och 'Requests' med JSON + Kotlin inom Android Studio!

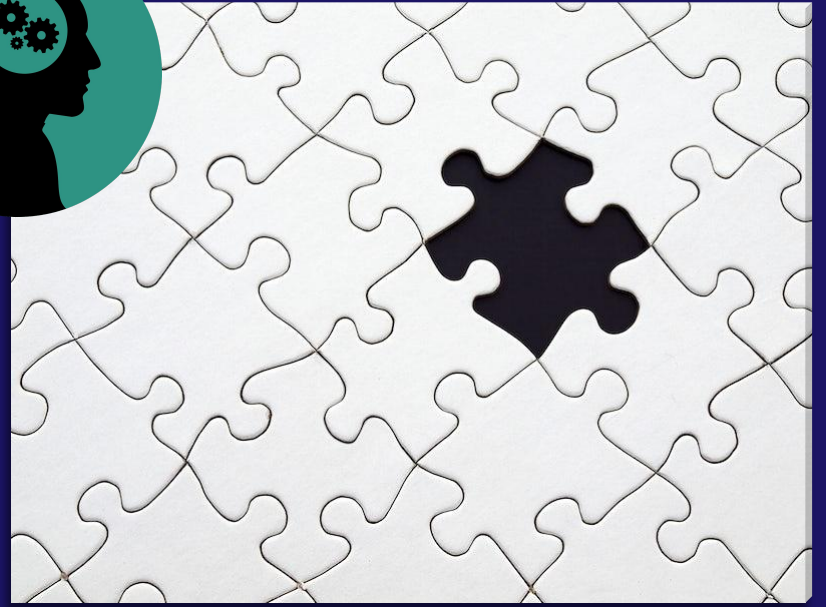


Solution

Retrofit + GSON, glöm inte att inkludera dessa inom build.gradle!



Frågor?

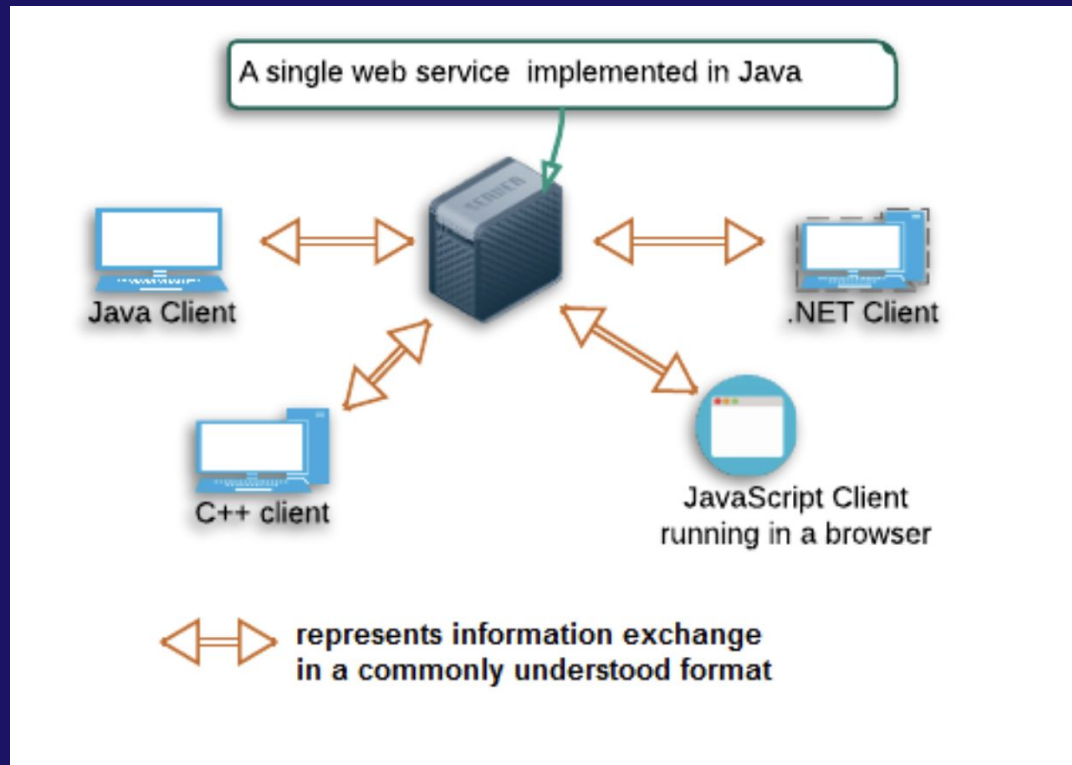


Web Service & API

What are they?

Requests...?





Överblick

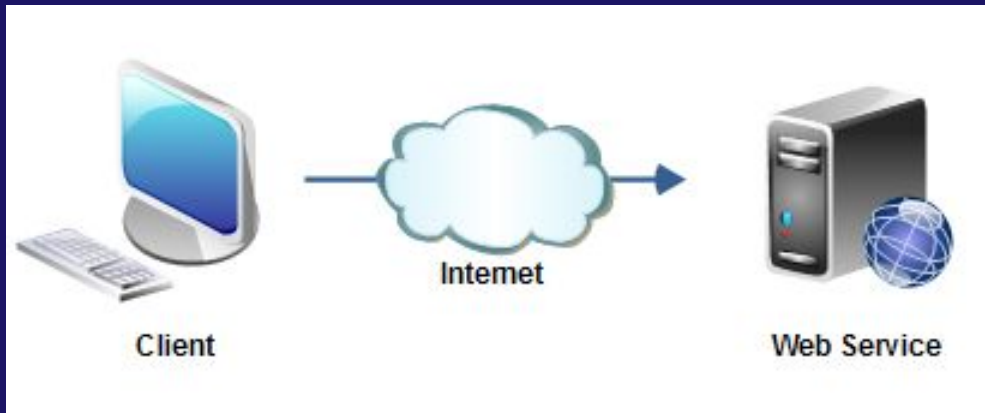
OS?

Språk?

Client?



Liknar 'Client-server model'



Överblick

Vad är en 'Client'?

Internet... varför?

Web Service...?

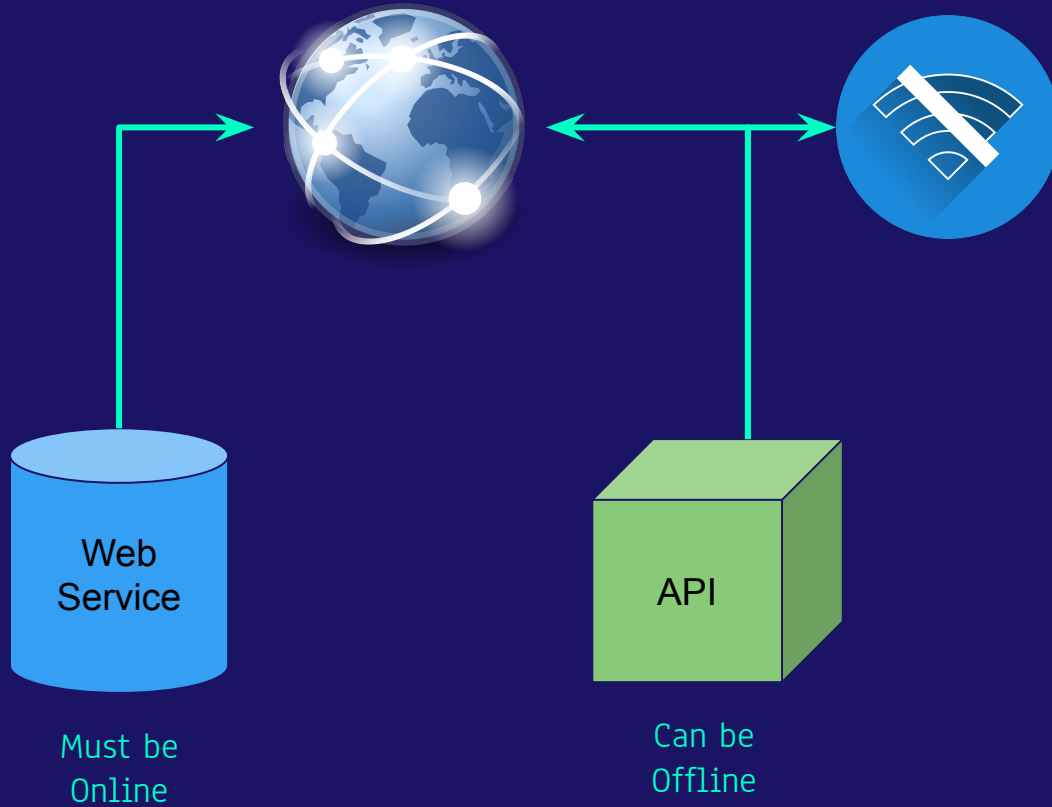
Databas, cloud.. Requests..?

Protokoll?

Web Service?

- Webbtjänster är designade för interaktion mellan maskin och applikation (eller applikation till applikation).
- Webbtjänster bör vara interoperabla - inte plattformsb beroende
- Webbtjänster bör tillåta kommunikation över ett nätverk





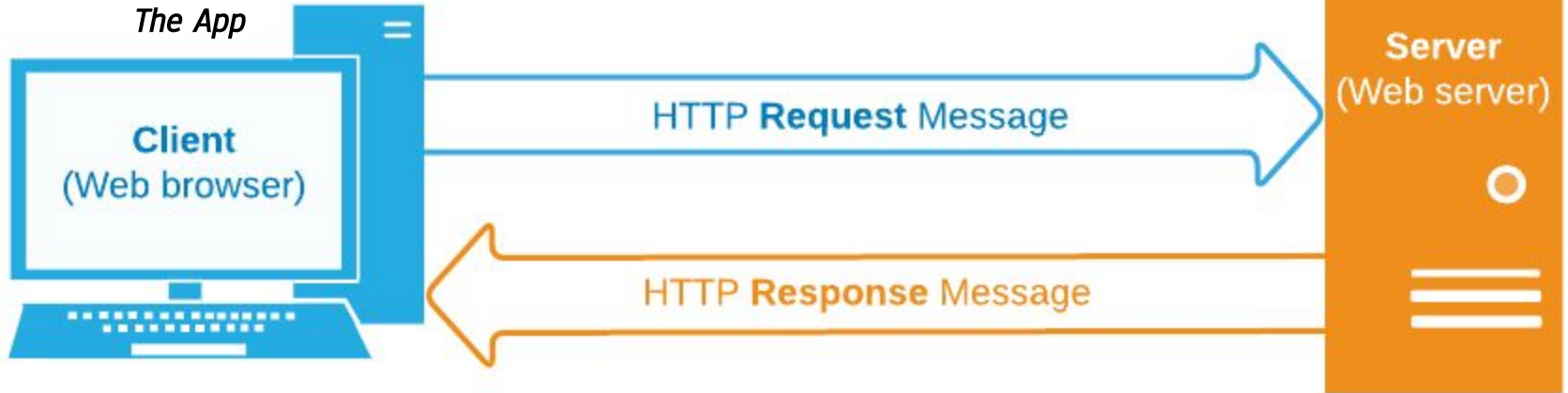
WS VS API

Web Service

Vad är skillnaden?



In our case
The App



Status Koder





Överblick

Sätt att kommunicera på,
via Web Services.

<https://www.iana.org/assignments/http-status-codes/http-status-codes.html#http-status-codes-1>

Annotationer AKA @



@Override

Annotation

Vad är en annotation?

Syntaktisk metadata...?

@Override?

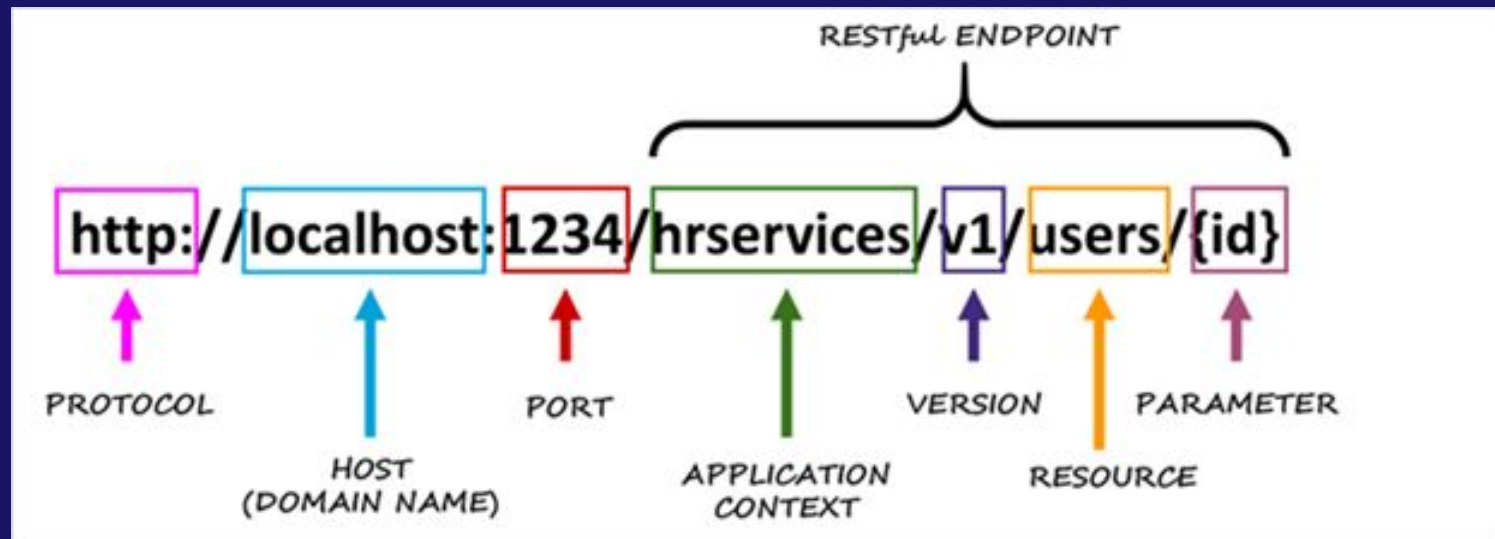
<https://www.geeksforgeeks.org/annotations-in-java/>



URL Structure



URL



Open Api Collection



Get Started!

<https://github.com/public-apis/public-apis>

Auth AKA Authentication OAuth, APIkey, No

HTTPS No, Yes

CORS No, Yes, Unknown

Api nycklar kan vara bra att försöka undvika - dessa är bra för övning, men om ni vill utöka applikationen i framtiden, så kommer de säkerligen krävas en prenumeration på deras 'service'.



Problem

Öppna apier?
Vart kan jag hitta dessa?



Solution

GitHub - bra källor!
Finns såklart också andra hemsidor!





03

API Structure, Fetching & Consuming

JSON Struktur



Struktur

Låt oss börja med ett väldigt enkelt
API

<https://randomfox.ca/floof/>

Webbläsare: Firefox

JSONRaw DataHeaders

SaveCopyCollapse AllExpand AllFilter JSON

image:

"https://randomfox.ca/images/75.jpg"

link:

"https://randomfox.ca/?i=75"

Struktur

Datotyp?

JSON

Raw Data

Headers

Save

Copy

Collapse All

Expand All

Filter JSON

image: `"https://randomfox.ca/images/75.jpg"`

link: `"https://randomfox.ca/?i=75"`

Struktur

Image : String

Link: String

Datotyp?

JSON	Raw Data	Headers
Save	Copy	Collapse All
Expand All	Filter JSON	
image:		"https://randomfox.ca/images/75.jpg"
link:		"https://randomfox.ca/?i=75"

Coding



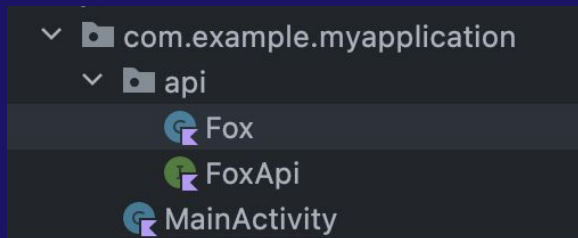
Project Structure

Project Hierarchy



Code Structure

Project Hierarchy



```
3  class Fox {  
4  
5      var link: String = ""  
6      var image: String = ""  
7  
8  }  
9
```

JSON comparison

JSON	Raw Data	Headers
Save	Copy	Collapse All Expand All Filter JSON
image:	"https://randomfox.ca/images/75.jpg"	
link:	"https://randomfox.ca/?i=75"	

Notera: Namnen MÅSTE matcha!

```
3  class Fox {  
4  
5      var link: String = ""  
6      var image: String = ""  
7  
8  }  
9
```

Your own variable name

```
5  class Fox {  
6  
7      @SerializedName("link")  
8      var someNewRandomName: String = ""  
9      var image: String = ""  
10  
11 }
```

Observera: Om du vill ha ett eget namn på din variabel, använd `@SerializedName()`

Navigate here



Interface API



```
6  interface FoxApi {  
7  
8      @GET("endpointGoesHere")  
9      fun getInfo(): Call<Fox>  
10  
11 }
```

*Observera: att vi arbetar med en annotation @GET här!
Vi arbetar också med Call<T>
Dessa kommer från Retrofit!*

Interface



```
) : Call  
Call (android.telecom)  
Call (okhttp3)  
Call<T> (retrofit2) ←
```

FoxAPI

FoxAPI Code

Detta är allt som behövs just nu

Vi saknar dock ett /endpoint

```
package com.example.myapplication.api

import retrofit2.Call
import retrofit2.http.GET

interface FoxApi {

    @GET("endpointGoesHere")
    fun getInfo(): Call<Fox>

}
```

Endpoint

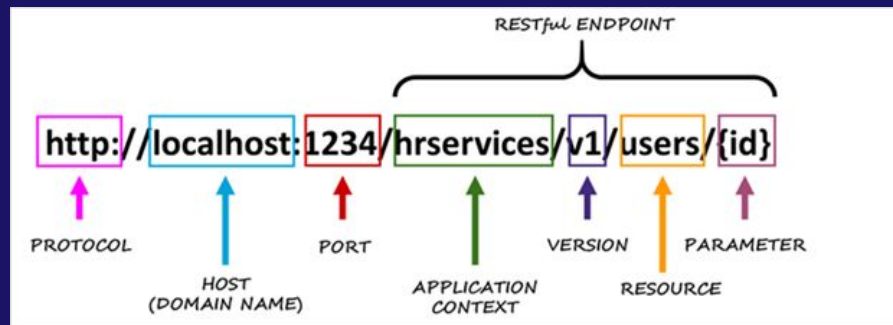


"Vad är ett Endpoint?"

Endpoint



"Vad är ett Endpoint?"



Our URL



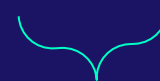
<https://randomfox.ca/floof/>

Vilken del är vårt endpoint?

Our URL



<https://randomfox.ca/floof/>



Endpoint

FoxAPI

FoxAPI Code

Detta är allt som behövs just nu

Vi saknar dock ett /endpoint

```
package com.example.myapplication.api

import retrofit2.Call
import retrofit2.http.GET

interface FoxApi {

    @GET("/floof")
    fun getInfo(): Call<Fox>

}
```


URL 's Differ

`www.test.com/?foodtype=someFood`

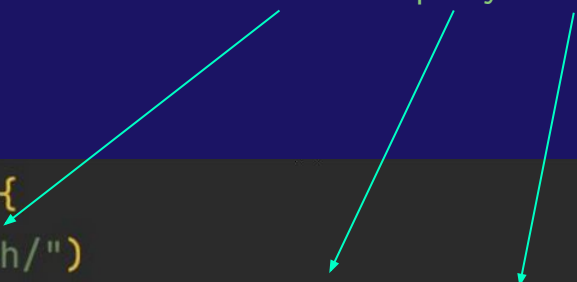
`www.test.com/?userName=benny`

URL'er framkommer väldigt olika. Här ovan ser vi ett exempel på en parameter!

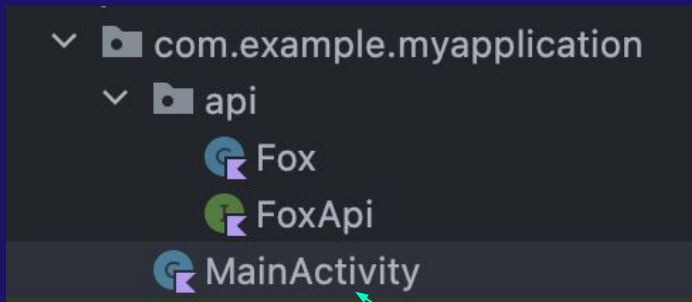
How to Query Params?

www.weather.com/search/?query=value

```
9 interface WeatherApi {  
10     @GET(value: "search/")  
11     fun getWeather(@Query(value: "query") query: String): Call<List<Weather>>  
12 }
```



Time to build Retrofit



Retrofit

Build Retrofit

baseUrl = API Url
converterFactory = konverterare

```
// Build Retrofit
val retrofit = Retrofit.Builder()
    .baseUrl("https://randomfox.ca/floof/")
    .addConverterFactory(GsonConverterFactory.create())
    .build()
```



getInfo()

FoxApi method

Här kallar vi på vår metod vi skrev inom FoxApi interface.

getInfo() är vår metod!

Detta är en @Get mot en endpoint

```
// GETTER  
val getFoxApi = retrofit.create<FoxApi>().getInfo()
```



enqueue()

Make the call!

Här så utför vi själva anropet!

Det tillkommer två lyssnare:

`onResponse` - Svaret

`onFailure` - Om det misslyckats

```
getFoxApi.enqueue(object : Callback<Fox> {  
  
    override fun onResponse(call: Call<Fox>, response: Response<Fox>)  
    {}  
  
    override fun onFailure(call: Call<Fox>, t: Throwable) {}  
  
})
```

```
// Make the call
getFoxApi.enqueue(object : Callback<Fox> {
    override fun onResponse(call: Call<Fox>, response: Response<Fox>) {

        // SUCCESS
        if (response.isSuccessful) {
            val fox = response.body()
            if (fox != null) {
                println("image: ${fox.image}, link: ${fox.myLink}")
            }
        } else {
            println(response.errorBody()) // ERROR 404 or 500 can still happen here
        }
    }

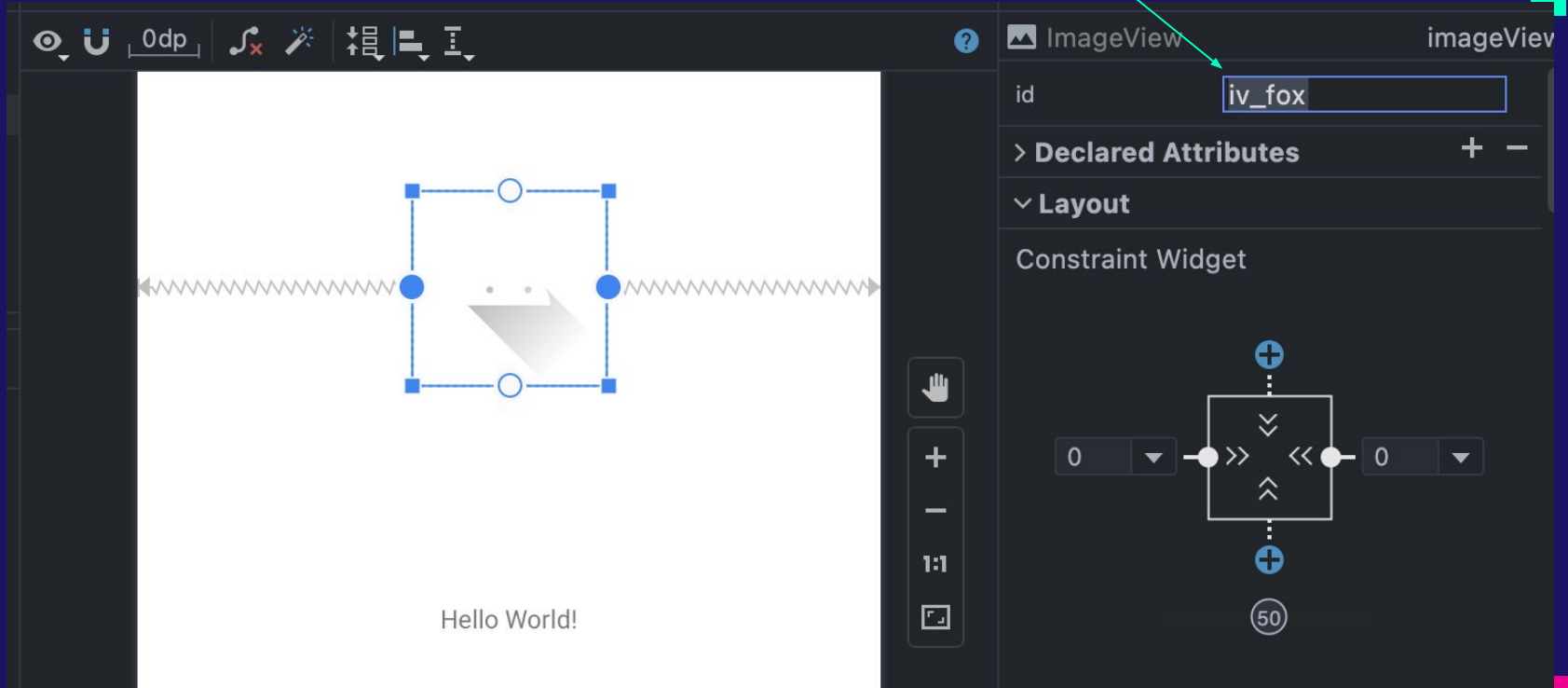
    // Error handling
    override fun onFailure(call: Call<Fox>, t: Throwable) {
        println(t.message) // ERROR MESSAGE
    }
})
```

GLIDE?

Om ni vill att er ImageView ska agera som HTML's src attribut så kan ni installera:

```
implementation 'com.github.bumptech.glide:glide:4.15.0' // GLIDE
annotationProcessor 'com.github.bumptech.glide:compiler:4.15.0' // GLIDE
```


GLIDE?

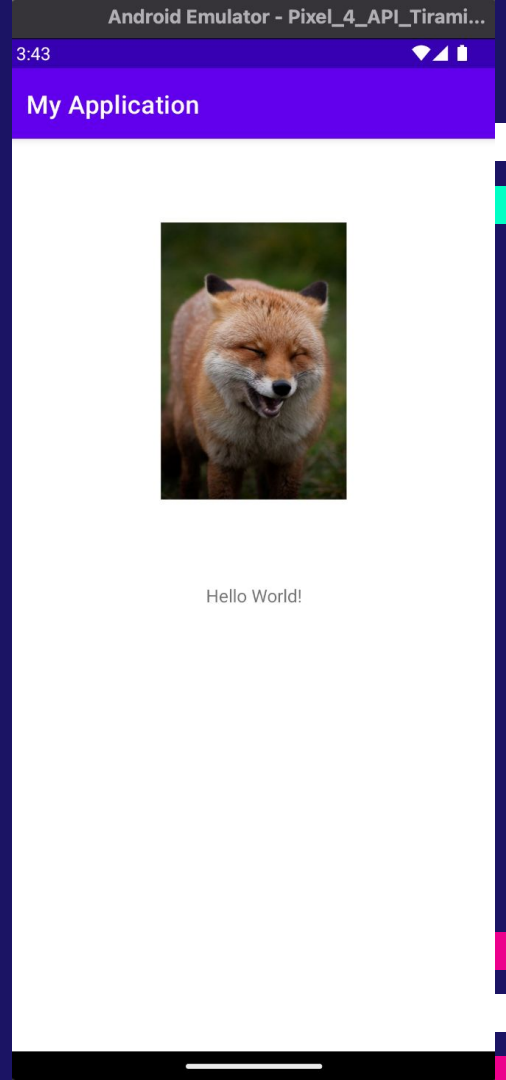


GLIDE?

```
// SUCCESS
if (response.isSuccessful) {
    val fox = response.body()
    if (fox != null) {
        println("image: ${fox.image}, link: ${fox.myLink}")

        // Glide
        Glide.with(this@MainActivity)
            .load(fox.image)
            .into(ivFox)
    }
} else {
    println(response.errorBody()) // ERROR
}
```

DONE





04

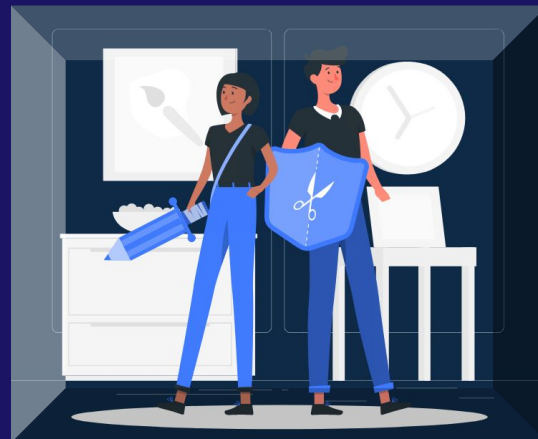
Uppgifter
&
Eget Arbete

Välkommen till första uppgiften!

Uppgifterna är till för att testa dina färdigheter och kunskaper för att både öva och repetera på det vi har arbetat med under föreläsningarna.

Dessa är **INTE** obligatoriska.
Men är starkt rekommenderat att arbeta med.

Uppgifter



MINNS DU?

```
// Vad är ett API?  
Hur urskiljer sig ett API från en  
Web Service?
```

```
// Vad är ett request?  
// Vad är en endpoint?
```

```
// Vad innebär status koderna:  
• 200  
• 404  
• 500
```

```
// Vad är en Annotation?
```

MINNS DU?

```
// Vad är en API nyckel?
```

```
// Vad är en parameter / query inom en  
URL? Varför existerar detta?  
När är detta bra?
```

JSONRaw DataHeaders

SaveCopyCollapse AllExpand AllFilter JSON

latitude:52.52

longitude:13.419998

generationtime_ms:0.2180337905883789

utc_offset_seconds:0

timezone:"GMT"

timezone_abbreviation:"GMT"

elevation:38

▼ hourly_units:

time:"iso8601"

temperature_2m:"°C"

▼ hourly:

▼ time:

0:"2023-03-08T00:00"

1:"2023-03-08T01:00"

2:"2023-03-08T02:00"

3:"2023-03-08T03:00"

4:"2023-03-08T04:00"

5:"2023-03-08T05:00"

6:"2023-03-08T06:00"

7:"2023-03-08T07:00"

8:"2023-03-08T08:00"

9:"2023-03-08T09:00"

10:"2023-03-08T10:00"

11:"2023-03-08T11:00"

12:"2023-03-08T12:00"

13:"2023-03-08T13:00"

MINNS DU?

TOUGH NUT

Analysera bilden till vänster

Om latitude, longitude är `'double'`

Vad är `'timezone'` för datatyp?

Vad är `'hourly_units'` för datatyp?

Vad är `'hourly'` för datatyp?


```

1          // -Uppgift #1- //
2
3  /* INSTRUCTIONS
4
5      Skapa ett nytt projekt!
6
7      Döp projektet till: Lektion_10_uppgifter
8
9      Analysera följande API:
10     https://api.adviceslip.com/advice
11
12     Skapa sedan ett objekt som har samma
13     datatyper!
14
15     */
16
17     // HINT & Examples
18     hint(" Du kan se om det är ett nästlat objekt om
19     den har flertal måsvingar inom sig.
20     Exempelvist: { {} }
21
22     Om det är nästlat så måste du matcha datan:
23     Skapa en kotlin klass som heter: Advice & Slip
24     Advice innehåller Int + String
25     Slip innehåller Advice ")

```

Uppgift #1

Kom igång enkelt med uppgift #1

```
1          // -Uppgift #2- //
2
3  /* INSTRUCTIONS
4
5      Lägg till dependencies för Retrofit + GSON
6
7      Skapa API Interface klassen.
8      Tillför en @GET annotation med din endpoint
9
10     Sedan inom MainActivity, bygg Retrofit!
11
12 */
13
14 // HINT & Examples
15 hint("
16     Slide #10 & 11 (Retrofit + GSON)
17     Slide #43 (API interface)
18     Slide #52 (Build Retrofit)
19 ")
20
21
22
23
```

Uppgift #2

Glöm inte synkronisera!

*@GET annotationen är viktig, den
skall komma från Retrofit!*

```

1          // -Uppgift #3- //
2
3      /* INSTRUCTIONS
4
5          Gör nu kallelsen!
6
7          val getAdvice =
8      retrofit.create<AdviceAPI>().getInfo()
9
10     getAdvice.enqueue(object : Callback<Advice> {
11
12         override fun onResponse(call: Call<Advice>, response:
13         Response<Advice>) {}
14         override fun onFailure(call: Call<Advice>, t: Throwable)
15         {}
16     }
17
18     */
19
20     // HINT & Examples
21     hint(" Slide #53, 54, 55 går igenom dessa steg!")
22
23

```

Uppgift #3

*Advice i detta fall är
Huvud objektet.*

*Byt bara namnen med det
Du har döpt klasserna till!*

```

1          // -Uppgift #4- //
2          TOUGH NUT
3
4      /* INSTRUCTIONS
5
6          Gör nu om koden så att den istället för synkron
7          blir asynkron!
8
9          Ändra om interface:
10         'fun getInfo(): Deferred<Response<Fox>>'
11
12         Deferred är inne på att det är ett objekt som
13         kan avbrytas när som!
14
15         Inom activity Main kan man sedan skriva:
16         // Coroutine Scope
17         val coroutineScope = CoroutineScope(Dispatchers.Main)
18
19     */
20
21     // HINT & Examples
22     hint(" .await() inväntar svaret och kommer sitta
23     tills den får det asynkrona svaret tillbaka! ")

```

Uppgift #4

```

// Make the async call
coroutineScope.launch { this: CoroutineScope
    try {
        // GETTER
        val getFoxApi = retrofit.create<FoxApi>().getInfo()

        // Await the response
        val response = getFoxApi.await()

        // SUCCESS
        if (response.isSuccessful) {
            val fox = response.body()
            if (fox != null) {
                println("image: ${fox.image}, link: ${fox.myLink}")
            }
        }
    }
}

```

THANKS!

Do you have any questions?
kristoffer.johansson@sti.se

sti.learning.nu/

CREDITS: This presentation template was created by
Slidesgo, including icons by Flaticon, and
infographics & images by Freepik.

You can also contact me VIA Teams (quicker response)
Du kan också kontakta mig VIA Teams (Snabbare svar)