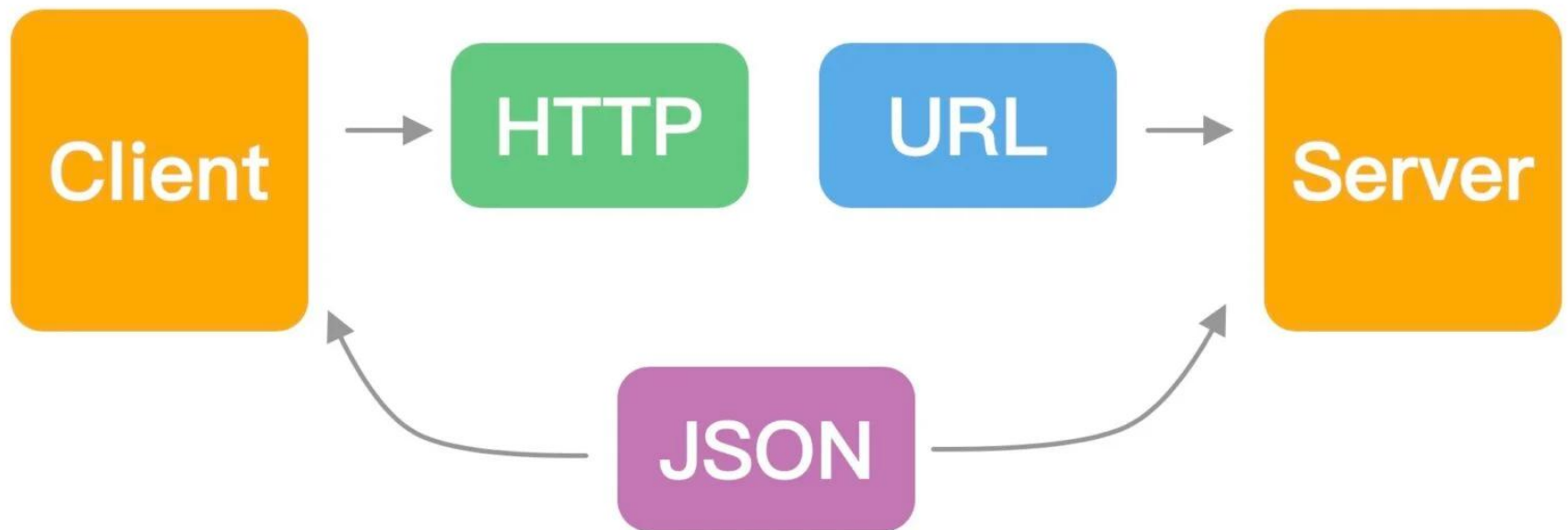


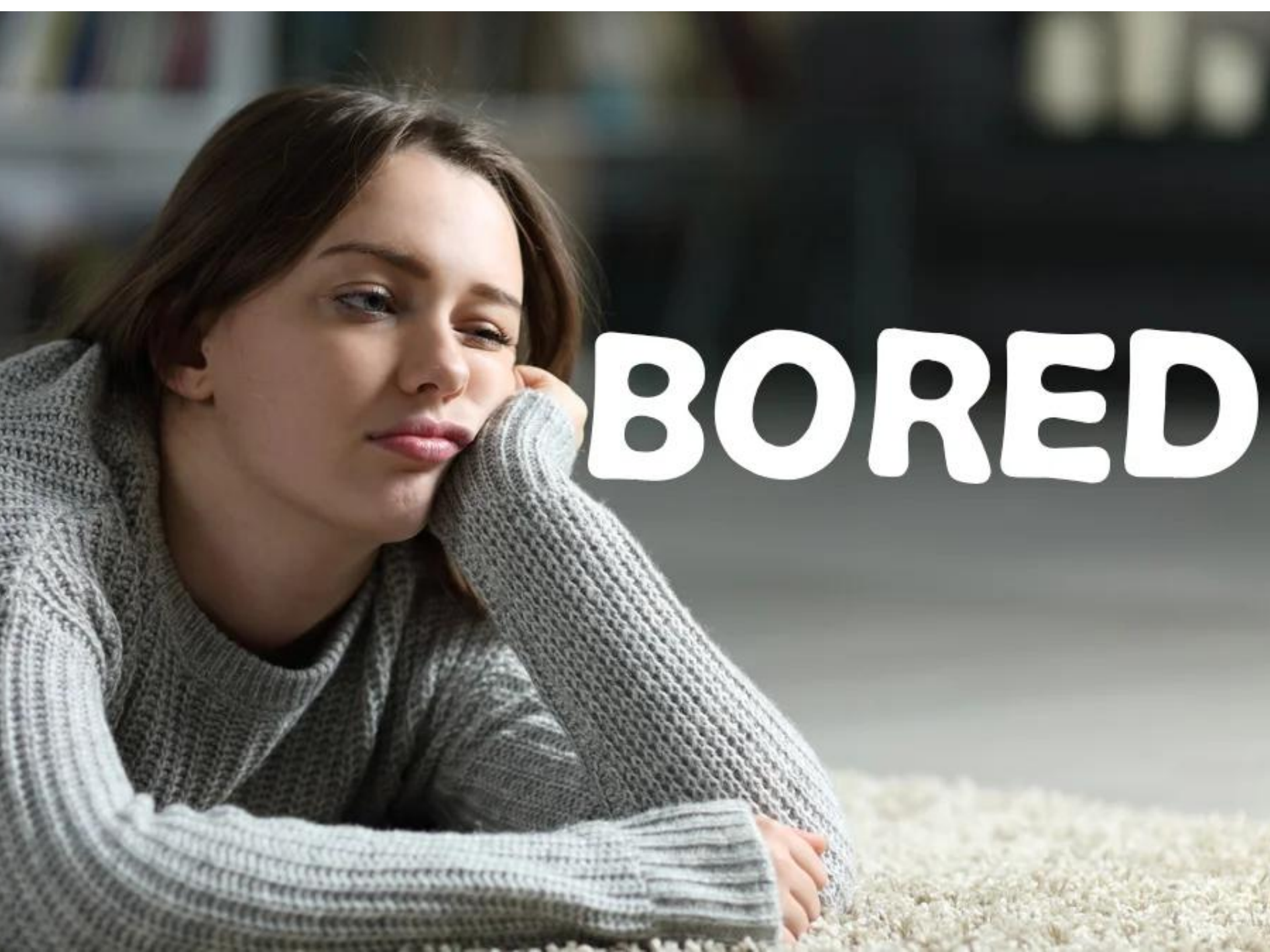
Workshop: Eksterne API'er

Et API er en service du tilgår via **HTTP** netværksprotokol (du kender fra din browser)



Eksempel kode

https://github.com/Dat1Cphbusiness/fall2025_holdA/tree/main/src/workshop-api/src



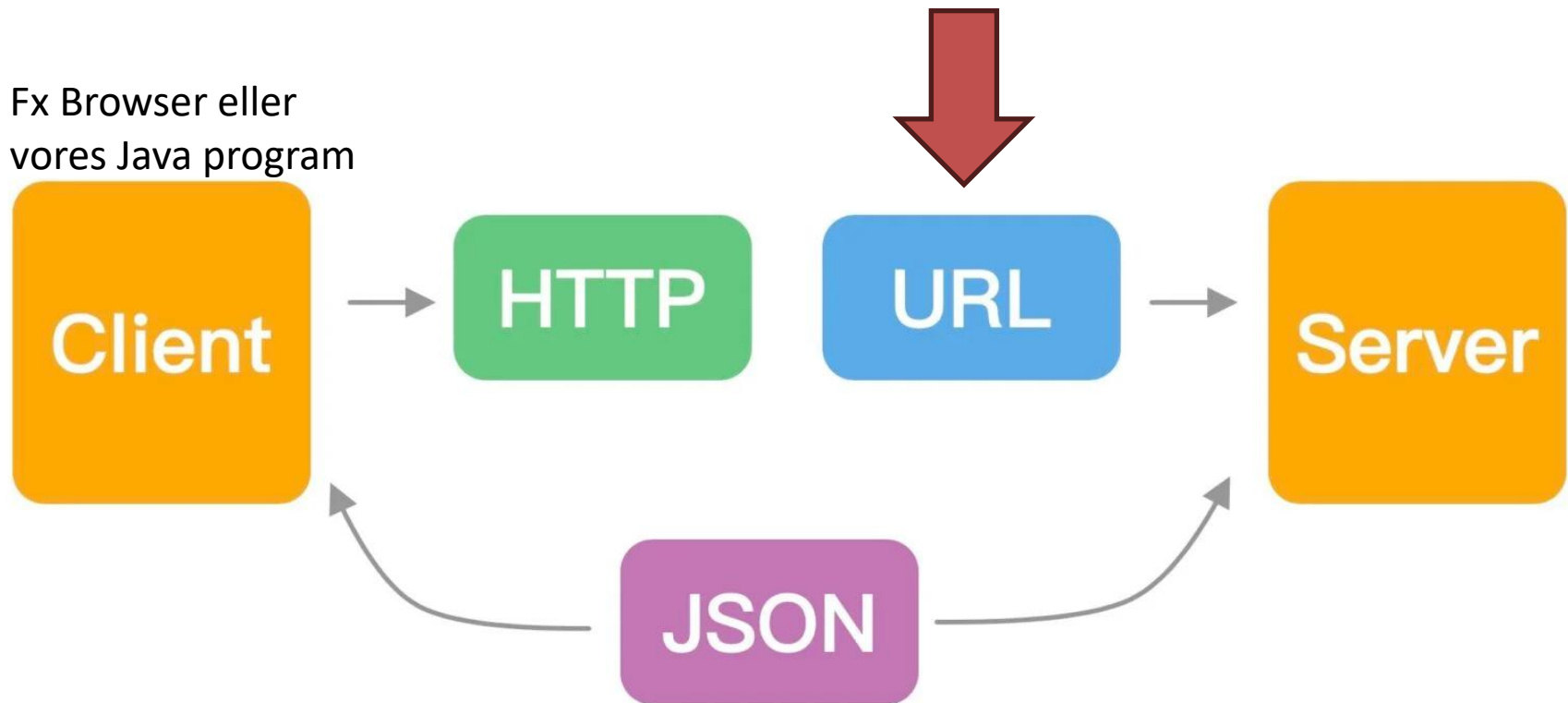
BORED

Service på ekstern server tilgås via **URL**

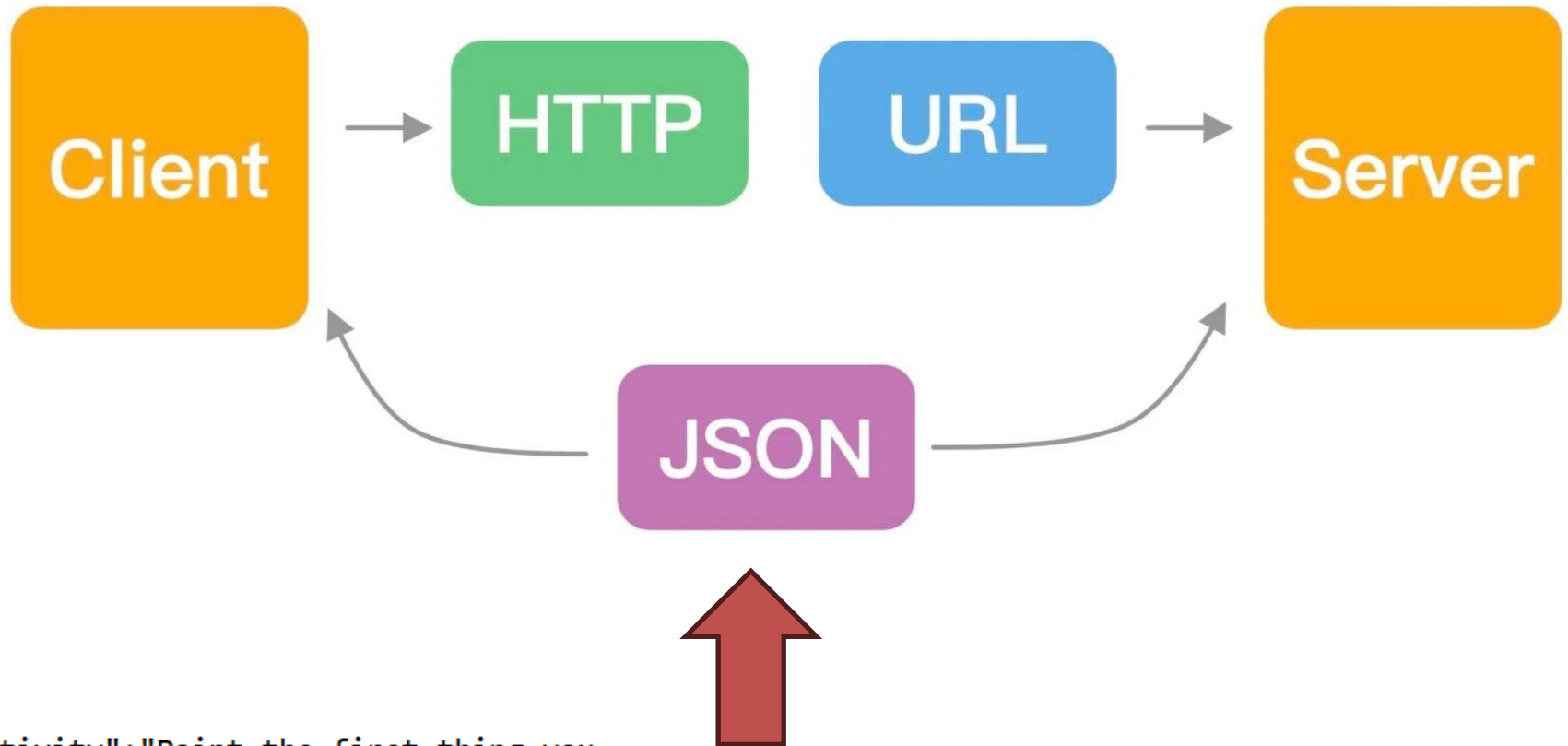
Dette API returnerer én tilfældig aktivitet, du kan lave, hvis du keder dig:

<https://bored-api.appbrewery.com/random>

Fx Browser eller
vores Java program



Vi får svar tilbage i **JSON** format



```
{"activity": "Paint the first thing you see", "availability": 0.2, "type": "recreational", "participants": 1, "price": 0.25, "accessibility": "Minor challenges", "duration": "minutes", "kidFriendly": true, "link": "", "key": "1162360"}
```

JSON formatering

Der findes værktøjer, der kan formatere og visualisere JSON, fx:

Formatering and validering <https://jsonlint.com/>

Fokus på det visuelle <https://jsonformatter.org/>

```
{  
  "activity": "Paint the first thing you see",  
  "availability": 0.2,  
  "type": "recreational",  
  "participants": 1,  
  "price": 0.25,  
  "accessibility": "Minor challenges",  
  "duration": "minutes",  
  "kidFriendly": true,  
  "link": "",  
  "key": "1162360"  
}
```

Java program kan kalde API vha. særlige netværksklasser

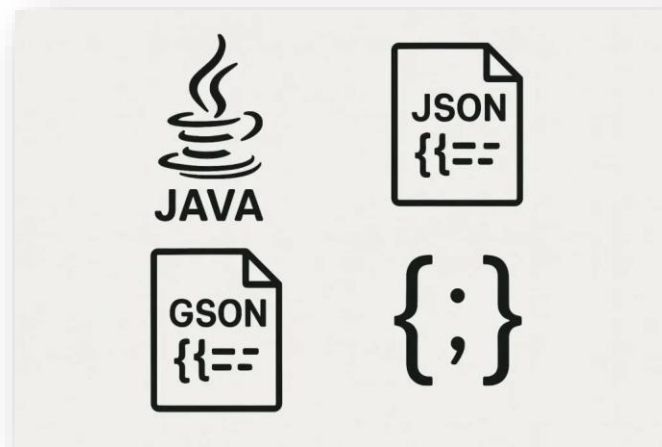
Eksempel: [fall2025 holdA/src/workshop-api/src/ApiDemo1.java](#)

```
public static void main(String[] args) throws Exception {  
  
    HttpClient client = HttpClient.newHttpClient();  
  
    String url = "https://bored-api.appbrewery.com/random";  
    HttpRequest request = HttpRequest.newBuilder()  
        .uri(URI.create(url))  
        .GET()  
        .build();  
  
    HttpResponse<String> response = client.send(request,  
        HttpResponse.BodyHandlers.ofString());  
    System.out.println(response.body());  
}
```

Parsing af JSON resultat

Java forstår ikke JSON (mest brugte dataformat)

Vi behøver hjælpebibliotek til at parse JSON resultat og omdanne til objekt, som Java kan forstå. Fx Gson udviklet af Google.



Brug af Gson

```
HttpResponse<String> response = client.send(request,  
    HttpResponse.BodyHandlers.ofString());
```

```
//Gson:
```

```
JsonObject json =
```

```
JsonParser.parseString(response.body()).getAsJsonObject();
```

```
System.out.println("Activity:" + json.get("activity").getString());
```

- ✓ `response.body()` er **JSON** som tekst
- ✓ **`JsonParser.parseString(...)`** laver teksten om til et JSON-træ
- ✓ **`getAsJsonObject()`** betyder: "Roden er et objekt {...}"
- ✓ Vi får et **JsonObject**, der fungerer lidt som en HashMap: vi kan slå værdier op ved hjælp af nøgler.

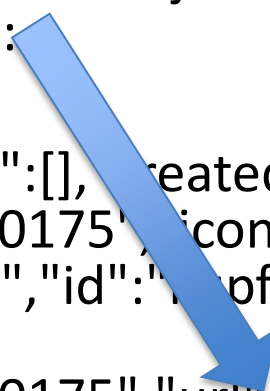
**WHEN GOOGLE HAS A QUESTION, THEY
"NORRIS" IT.**



Øvelse1 Chuck Norris Jokes

1. Tilføj Gson til et IntelliJ projekt, se hvordan her:
https://github.com/Dat1Cphbusiness/fall2025_holdA/blob/main/src/workshop-api/jar-guide.md
2. Kald denne url med Chuck Norris jokes via Java program:
<https://api.chucknorris.io/jokes/random>
3. Skriv ikke hele json svar, men kun jokens tekst ud på skærmen vha.

"value":



```
{"categories":[], "created_at":"2020-01-05  
13:42:23.240175", "icon_url":"https://api.chucknorris.io/img/avatar/chuc  
k-norris.png", "id":"kupfAt0zQ2yd4BwaS6bA2A", "updated_at":"2020-01-  
05  
13:42:23.240175", "url":"https://api.chucknorris.io/jokes/kupfAt0zQ2yd4B  
waS6bA2A", "value": "Chuck Norris has won the Publisher Clearing House  
1 million a year for live every year for the last ten years. But he has  
never entered their contest."}
```



Pause



Nogle API'er leverer forskellige typer resultat

<https://v2.jokeapi.dev/joke/Any> leverer one-liner ***eller*** joke med setup + punchline.

```
{  
  "type": "single",  
  "joke": "Why do Java developers wear glasses? Because they don't C#."  
}
```

Eller

```
{  
  "type": "twopart",  
  "setup": "What do you call a factory that makes okay products?",  
  "delivery": "A satisfactory."  
}
```

Vi vil lave JSON resultat til Java objekt

Gson kan parse JSON resultat til et Java objekt.

```
class Joke {  
    String type;  
    String joke; // bruges hvis type == "single"  
    String setup; // bruges hvis type == "twopart"  
    String delivery;  
}
```

Kodeeksempel:

[fall2025_holdA/src/workshop-api/src/JokeAPIDemo.java at main · Dat1Cphbusiness/fall2025_holdA](#)

Mere info om APlet: [JokeAPI project | by Riley Freier | Medium](#)

Øvelse2 Joke-maskine (10-15 min)

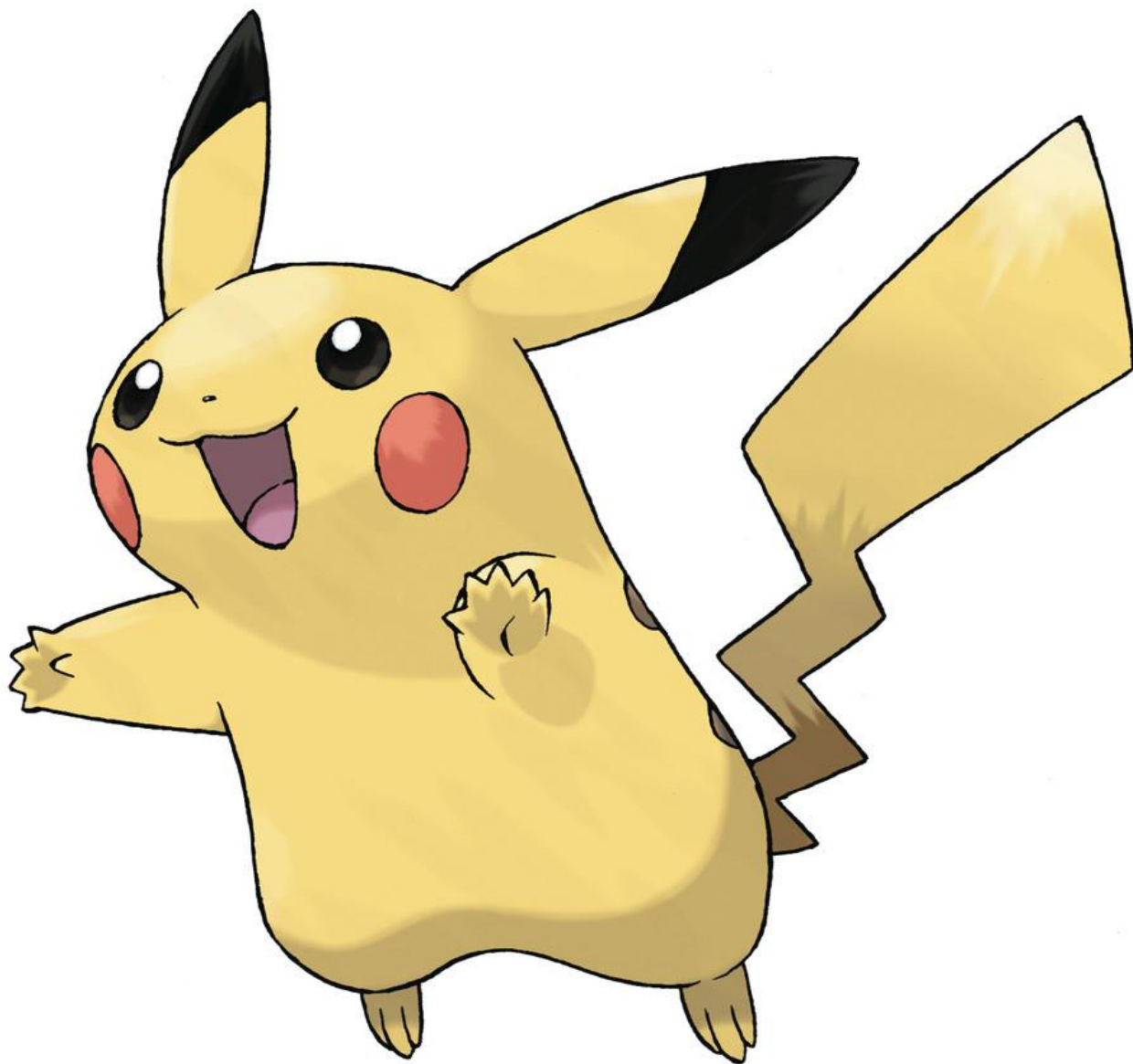
Konsol:

- Udskriv et random joke objekt
- Lav en “joke-maskine” med 5 jokes i træk
- Gem evt. jokes i en tekstfil

```
1. Ny joke  
2. 5 jokes i træk  
0. Afslut  
Valg:
```

Alternativt i JavaFX:

- Vis joke i et tekstfelt
- Lav “Next joke”-knap
- Vis setup → klik → punchline



Nogle API'er leverer et komplekst resultat tilbage (arrays + nested JSON)

Fx søge efter pokemon: <https://pokeapi.co/api/v2/pokemon/pikachu>

Uddrag af resultat:

```
{
  "name": "pikachu",
  "height": 4,
  "weight": 60,
  "types": [
    {
      "type": {
        "name": "electric"
      }
    }
  ],
  "sprites": {
    "front_default": "https://raw.githubusercontent.com/.../pikachu.png"
  }
}
```

https://github.com/Dat1Cphbusiness/fall2025_holdA/blob/main/src/workshop-api/src/pokemon/PokeAPIDemo.java



```
← ↻ 🔒 https://pokeapi.co/api/v2/pokemon/pikachu
EK Bookmarks 1sem SOFT SP- Lyngby Chrome DevTools -... Importeret Team Cam
Flot udskrift ✓
{
  "abilities": [
    {
      "ability": {
        "name": "static",
        "url": "https://pokeapi.co/api/v2/ability/9/"
      },
      "is_hidden": false,
      "slot": 1
    },
    {
      "ability": {
        "name": "lightning-rod",
        "url": "https://pokeapi.co/api/v2/ability/31/"
      },
      "is_hidden": true,
      "slot": 3
    }
  ],
  "base_experience": 112,
  "cries": {
    "latest": "https://raw.githubusercontent.com/PokeAPI/cries/main/cries/pokemon/latest/25.ogg",
    "legacy": "https://raw.githubusercontent.com/PokeAPI/cries/main/cries/pokemon/legacy/25.ogg"
  },
  "forms": [
    {
      "name": "pikachu",
      "url": "https://pokeapi.co/api/v2/pokemon-form/25/"
    }
  ],
}
```

Vi behøver ikke vise ALLE data

https://github.com/Dat1Cphbusiness/fall2025_holdA/blob/main/src/workshop-api/src/pokemon/PokeAPIDemo2.java

Indtast Pokémon-navn: Navn: absol

Højde: 12

Vægt: 470

Type: dark

Sprite URL: <https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/359.png>

Øvelse3

Udskriv alle typer for én Pokémon
(loop over types):

```
for (TypeEntry entry : p.types) {  
    System.out.println(" - " + entry.type.name);  
}
```

Design din kode med en ekstra
klasse, der henter en pokemon via
denne metode:

```
public static Pokemon fetchPokemon(String name) throws  
IOException, InterruptedException
```

Ndex	MS	Pokémon	Type	
#0460		Abomasnow	Grass	Ice
#063		Abra	Psychic	
#0359		Absol	Dark	
#0617		Accelgor	Bug	
#0681		Aegislash	Steel	Ghost
#0142		Aerodactyl	Rock	Flying
#0306		Aggron	Steel	Rock
#0190		Aipom	Normal	
#0065		Alakazam	Psychic	
#0869		Alcremie	Fairy	
#0594		Alomomola	Water	
#0334		Altaria	Dragon	Flying
#0698		Amaura	Rock	Ice
#0424		Ambipom	Normal	
#0591		Amoonguss	Grass	Poison



BADMOMS



JOHN WICK

Keanu Reeves
Best Action
Performance



**GUARDIANS
OF THE
GALAXY**

MARVEL STUDIOS



MOVIE



WILL FERRELL
MARK WAHLBERG
**DADDY'S
HOME**



Kingsman
THE SECRET SERVICE



THE AVENGERS
Age of Ultron

MARVEL STUDIOS



SHREK



PITCH PERFECT

2



Disney PIXAR
Cars



THE AVENGERS



IRON MAN 3



THE CONJURING

2

Nogle API'er kræver en nøgle

Movie database: opret bruger <https://developer.themoviedb.org/>

Login

Hent gratis API-nøgle

- I venstre menu → **Settings** → **API**
- Kopiér din **API Key**

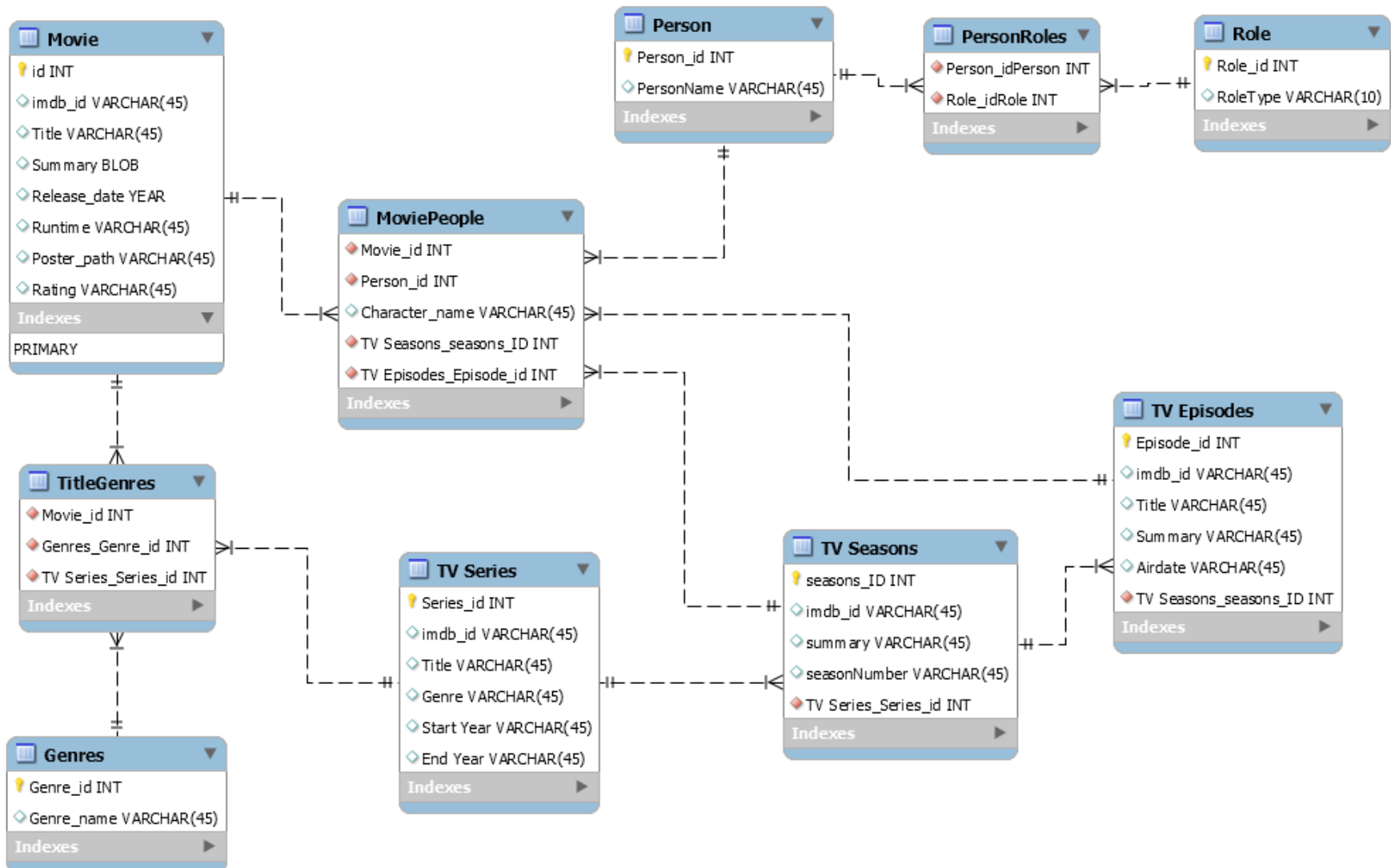
Hent film med din nøgle: https://api.themoviedb.org/3/movie/550?api_key=YOUR_KEY

Med id 550, får du Fight Club data som dette:

```
{  
  "title": "Fight Club",  
  "release_date": "1999-10-15",  
  "vote_average": 8.4,  
  "overview": "A ticking-time-bomb insomniac..."  
}
```

TMDB – dokumentation

<https://developer.themoviedb.org/docs/getting-started>



Øvelse4 – søg på titel

Et API kan have flere endpoints, fx er det muligt at søge på filmens navn i TMDB og evt. få et array af movies tilbage.

1. Søg efter filmtitel i browser (query skal URL-encodes (mellemrum → %20):

https://api.themoviedb.org/3/search/movie?query=fight%20club&api_key=YOUR_KEY

2. Søg derefter med app: [SearchMovieApp](#)

3. Søg efter **serie** i browser (fx Breaking Bad, The Office, Game of Thrones) vha. dette endpoint:
/search/tv

4. Find selv på flere søgninger, fx

Skuespillere: /search/person

Genrer: /genre/movie/list

Populære film: /movie/popular

Film på dansk - tilføj: &language=da-DK

{ "id": 28, "name": "Action" }
{ "id": 35, "name": "Comedy" }

5. Filtrer på (ikke search endpoint direkte, men via discover):

rating: /discover/movie?vote_average.gte=8

Udgivelsesdato /discover/movie?primary_release_date.gte=2025-01-01

gte = greater than or equal