

# **Introduction to digital trace data: Quality, ethics, and analysis**

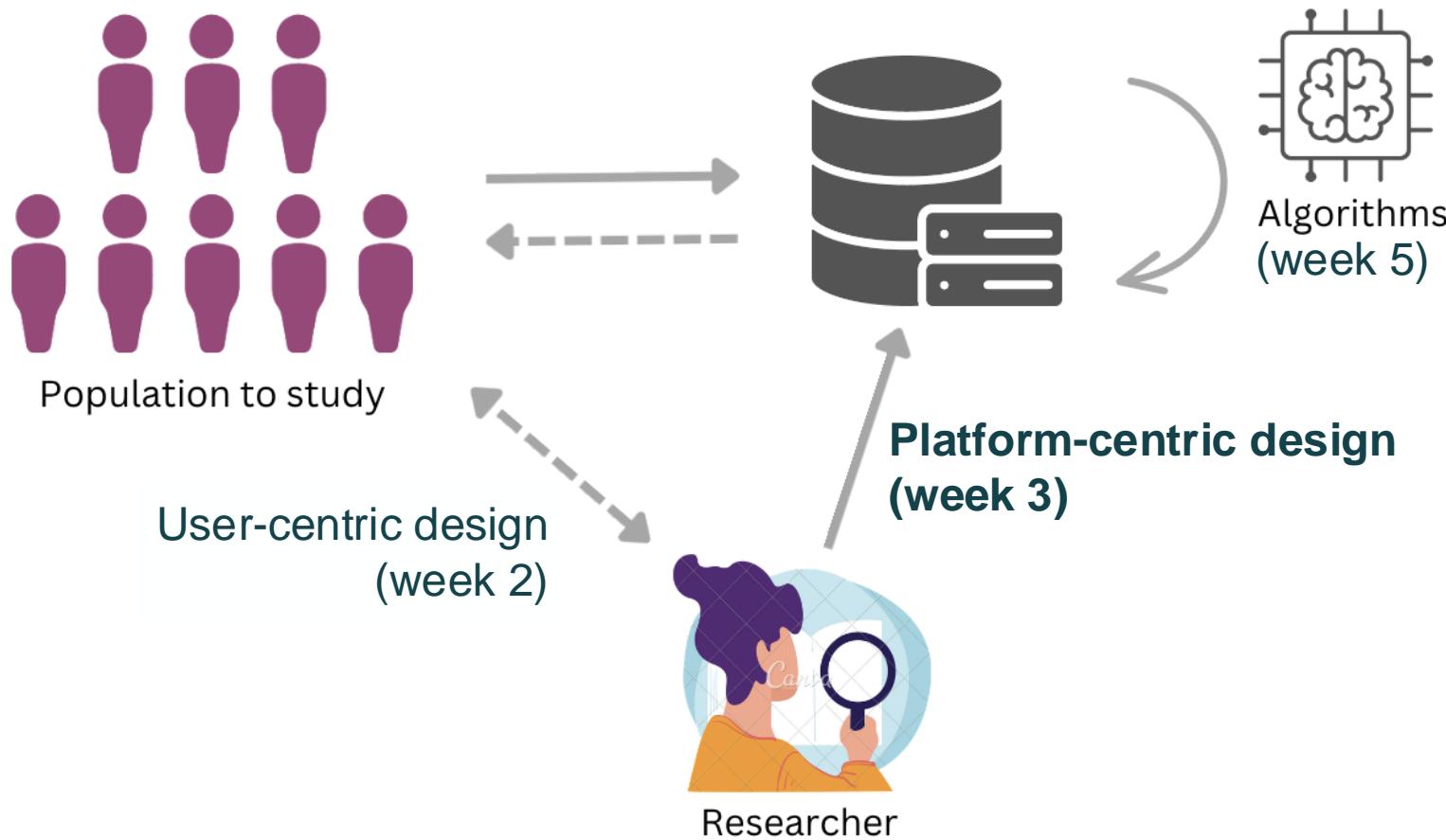
Lecture 3: Collecting platform-centric data

**Javier Garcia-Bernardo**

Assistant Professor

Department of Methodology and Statistics

# Where are we?



Week 4: Errors in DTD  
Week 6: Ethics and Legislation  
Week 7: Designed big data  
Week 8: Beyond DTD and Q&A

# Platform-centric design

## Collaboration with the organization holding the data

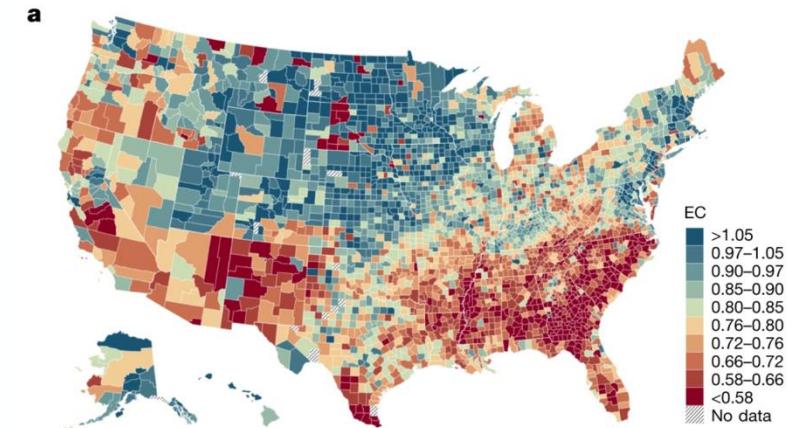
- The organization provides the data privately
- Public and private data

## APIs (Application Programming Interfaces)

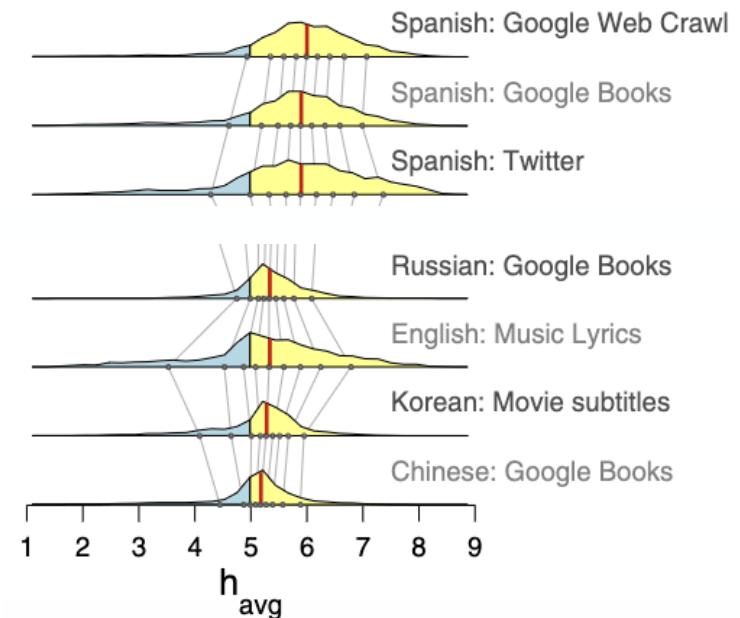
- The organization provide direct access to (parts) of the data.
- Public data only

## Web-Scraping

- Download the information as shown in the website
- Public data only (at least legally)



*Chetty et al, 2022, Nature*



*Dodds et al., 2015, PNAS*

# TODAY

## Lecture

Explain what APIs and web scraping are (in your own words).

Understand how HTML code is structured

Distinguish between robots.txt, Terms of Service and GDPR protections.

Understand the main advantages, challenges and legal considerations of APIs and Web Scraping.

## Lab

Use APIs to extract data:

- Wikimedia
- TheGuardian

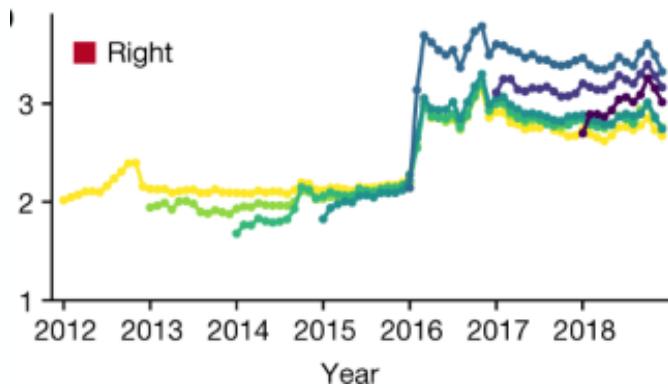
Web scraping data from NU.nl

Saving data to your hard drive

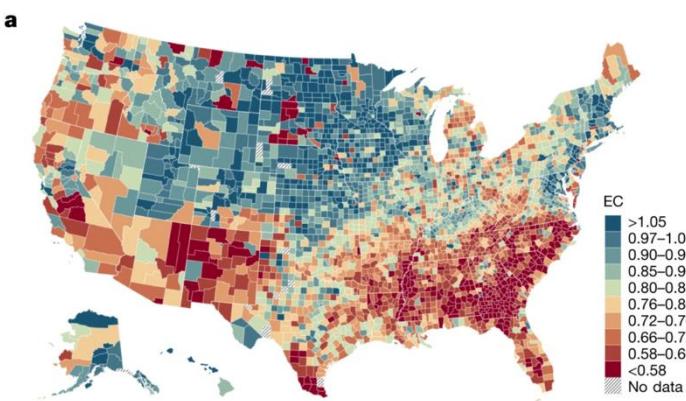
# Why take a platform-centric approach?

# Studying aggregated effects

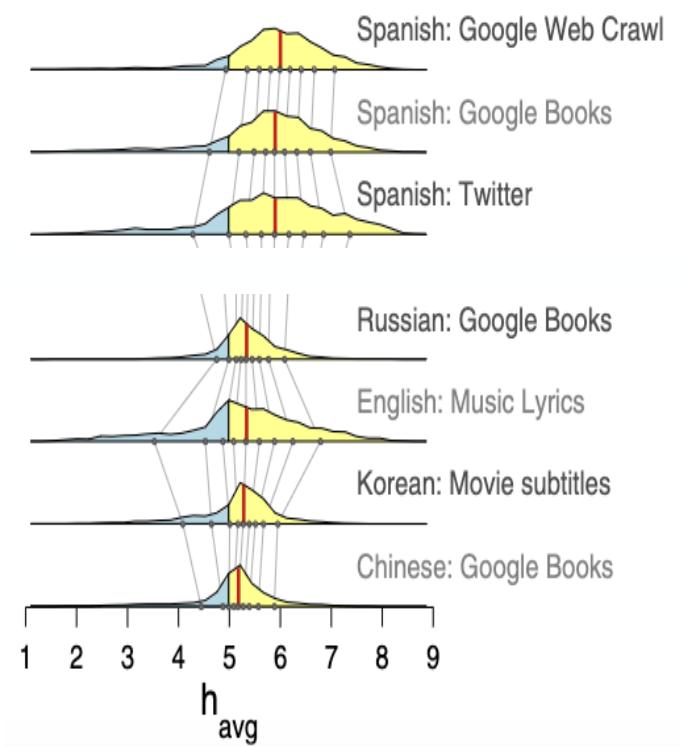
- Platform-specific effects
- How does X affects Y in general?
  - Big assumption here! the data is representative of the target population



Waller and Anderson, 2021, Nature



Chetty et al, 2022, Nature



Dodds et al., 2015, PNAS

# Application Programming Interfaces

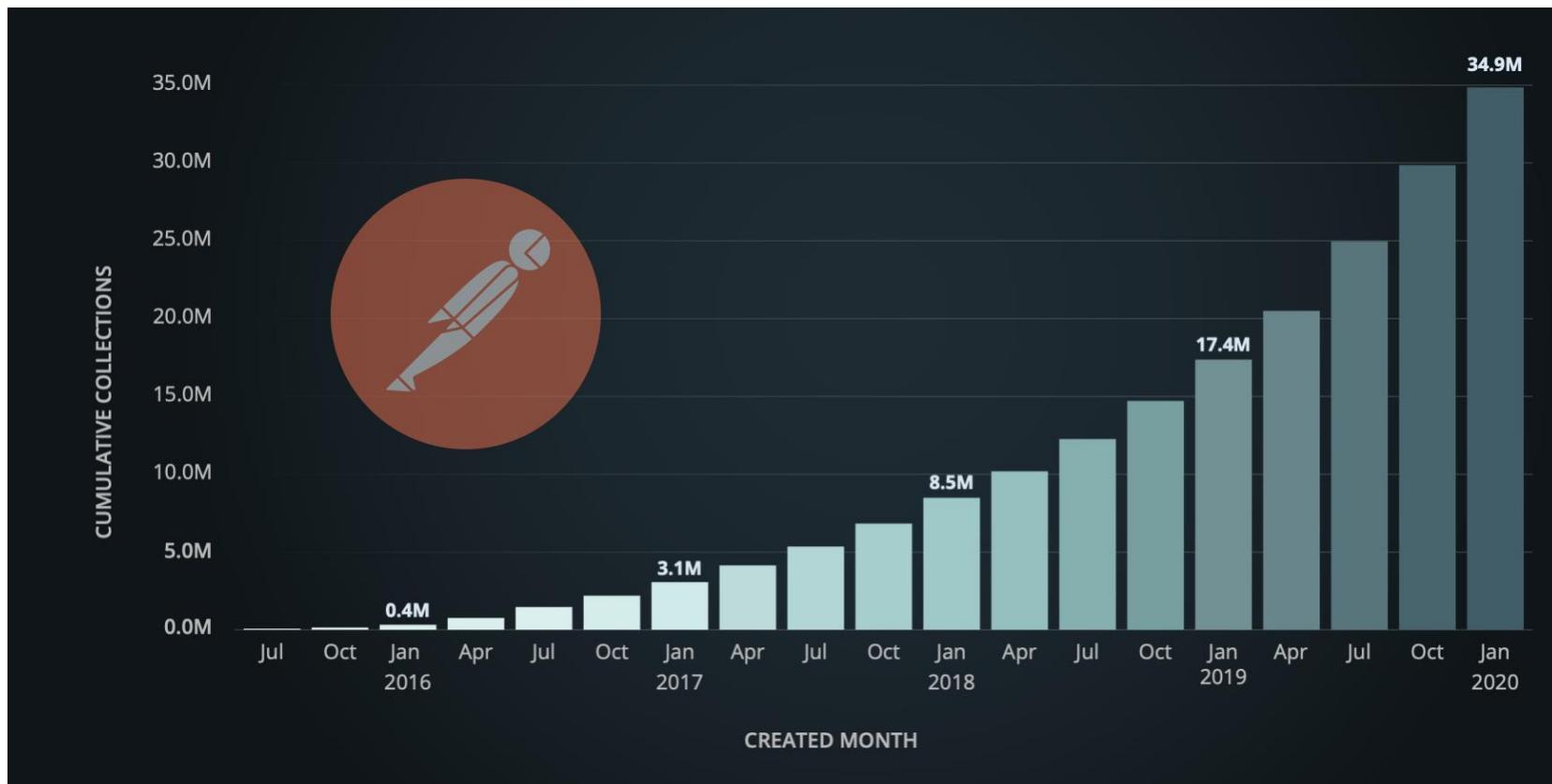
APIs

# What are Application Programming Interfaces (APIs)?

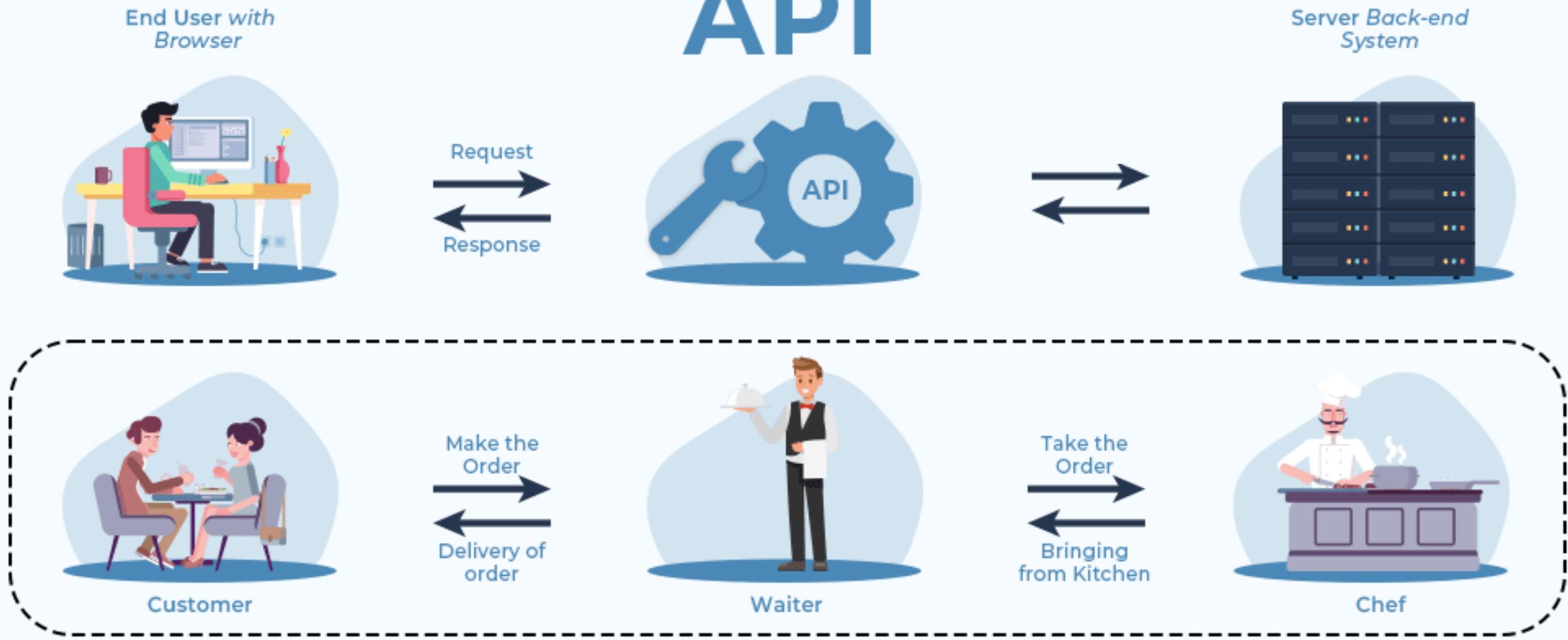
A way for computers (client/server) to talk between them. It provides:

- Communication protocols—i.e., rules to communicate
- Security—i.e., different users have access to different data

They are mostly created for developers, not researchers.



# API



You just need to know how to ask properly.

<https://www.geeksforgeeks.org/what-is-an-api/>

# Example use case

Het Parool wants to encourage readers to become members, it shows a banner at the bottom, but:

- The price should be up-to-date
- You want an independent company to take care of the memberships/payment

Solution: Use an online company providing an API



When you load the site, your browser calls the API:

<https://feeds.pexi.nl/api/feeds/bi6674262e89bda?brand=HP&articles=false&prices=false&pricetype=digitaal>

The API returns the data (JSON)

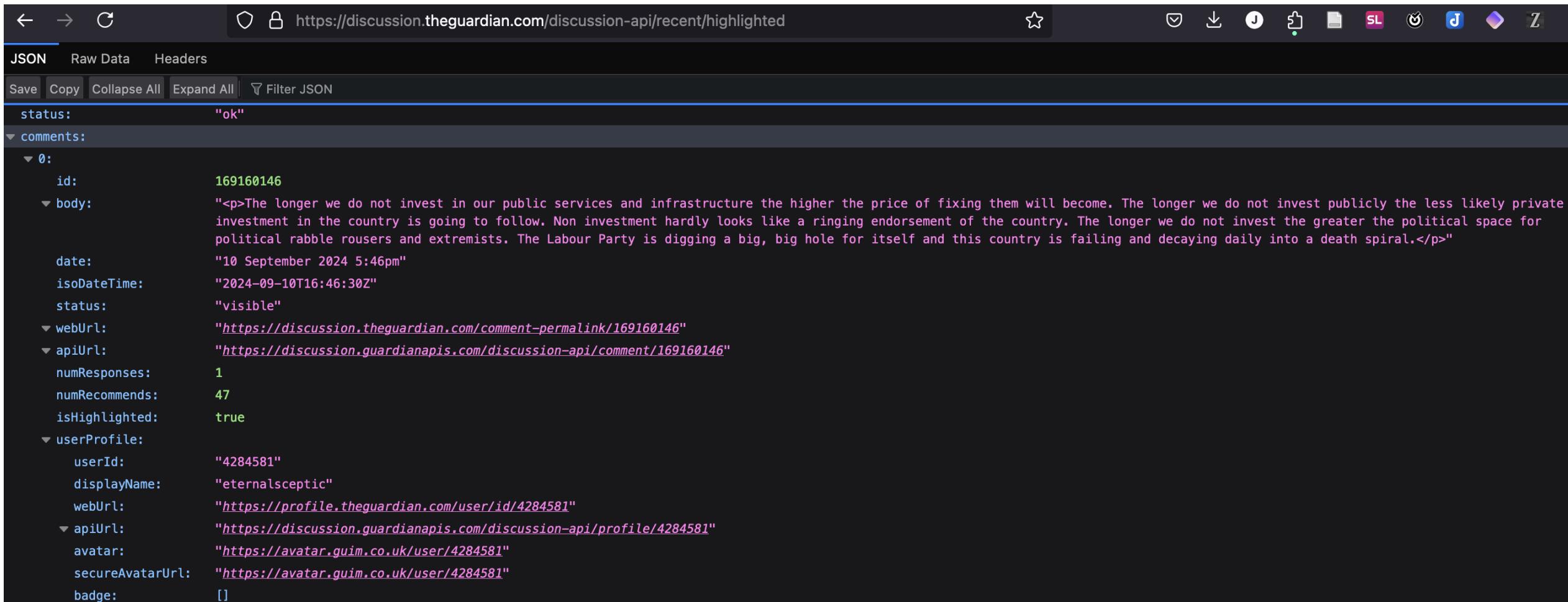
Your browser sets up the correct price

```
https://feeds.pexi.nl/api/feeds/bi6674262e89bda?brand=HP&articles=false&prices=true&pricetype=digitaal

[{"brand": "hp", "brand_title": "Parool", "brand_lopende_zin": "Het Parool", "brand_begin_zin": "Het Parool", "brand_logo": "https://static.pexi.nl/dpg-mediamagazines/styles/parool/logos/logo-parool.svg", "brand_packshot": "https://static.pexi.nl/dpg/packshots/Parool.png", "brand_cover": "https://static.pexi.nl/dpg-mediamagazines/kiosk_krant_covers/krant-hp.jpeg", "brand_url": "parool.nl", "brand_kleur": "#D72236", "brand_kleur2": "#0A0A95", "brand_kleur3": "#FFB600", "priceFrequency": "week", "priceBeforeDiscount": "5,86", "priceAfterDiscount": "3,75", "priceEuroBeforeDiscount": 5, "priceCentBeforeDiscount": 86, "priceEuroAfterDiscount": 3, "priceCentAfterDiscount": 75, "priceDiscount": "36%", "priceDuration": 36, "priceType": "digitaal", "priceDurationUnit": "weken"}]
```

# Example use case in social science

For example, to extract the comments from TheGuardian using their API

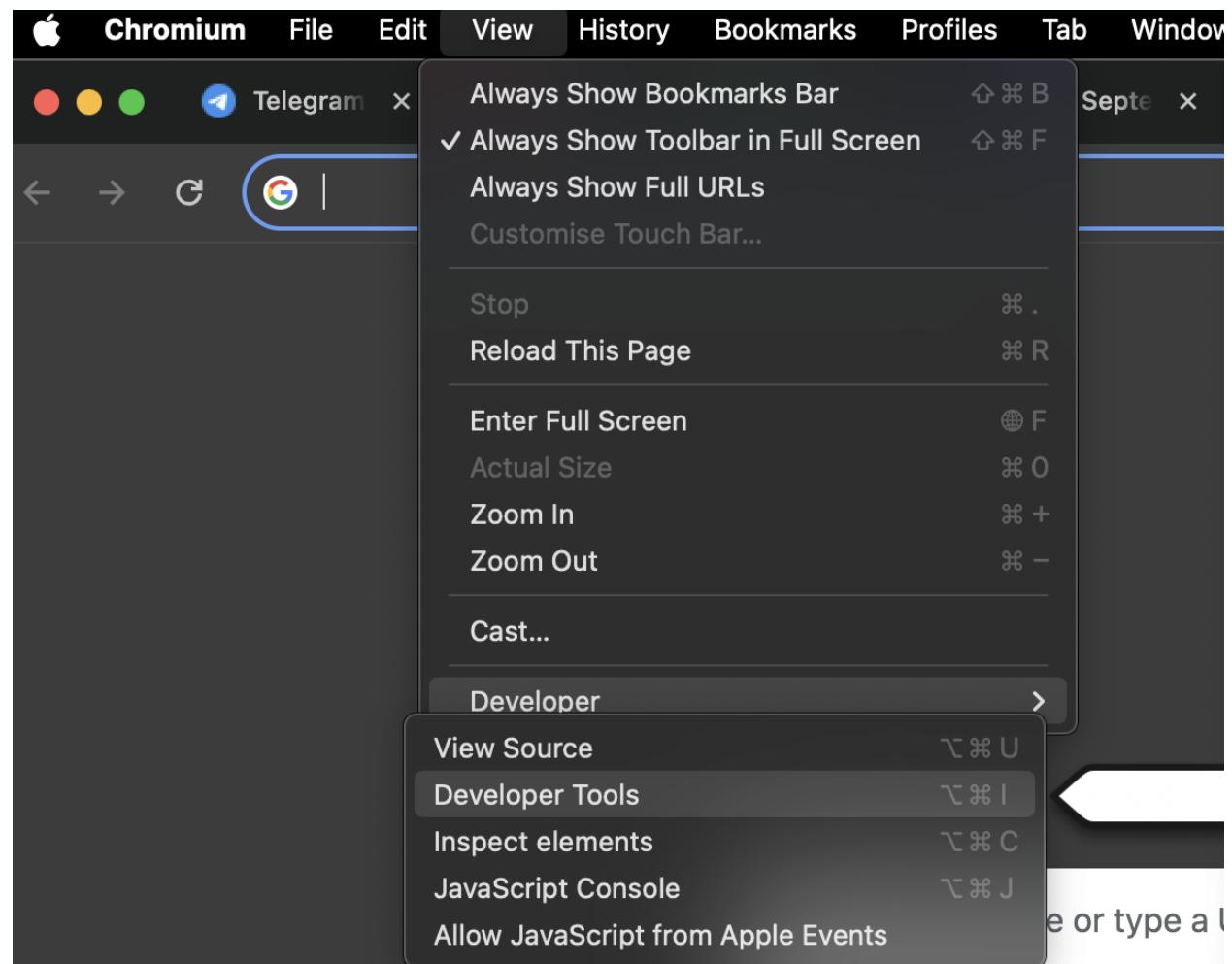
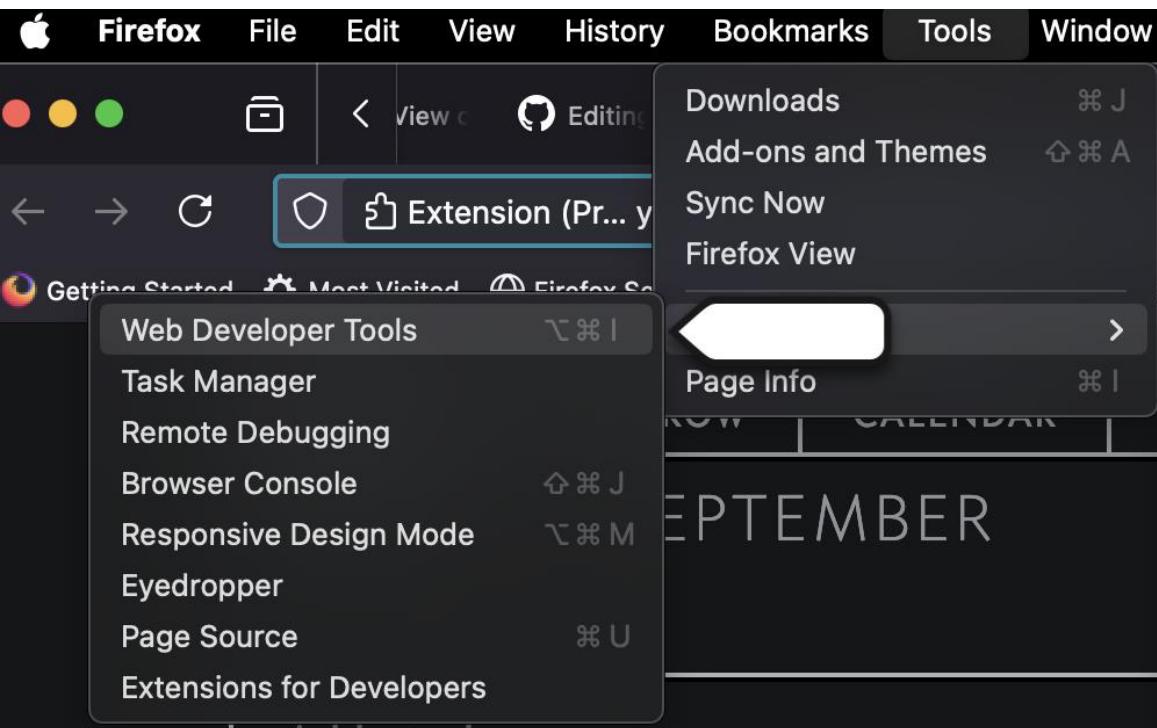


The screenshot shows a browser window with the URL <https://discussion.theguardian.com/discussion-api/recent/highlighted>. The page title is "Recent highlighted comments". Below the title, there are tabs for "JSON", "Raw Data", and "Headers". The "JSON" tab is selected. At the top of the JSON viewer, there are buttons for "Save", "Copy", "Collapse All", "Expand All", and a "Filter JSON" input field. The JSON data is displayed in a hierarchical tree view:

```
status: "ok"
comments:
  0:
    id: 169160146
    body: "<p>The longer we do not invest in our public services and infrastructure the higher the price of fixing them will become. The longer we do not invest publicly the less likely private investment in the country is going to follow. Non investment hardly looks like a ringing endorsement of the country. The longer we do not invest the greater the political space for political rabble rousers and extremists. The Labour Party is digging a big, big hole for itself and this country is failing and decaying daily into a death spiral.</p>"
    date: "10 September 2024 5:46pm"
    isoDateTime: "2024-09-10T16:46:30Z"
    status: "visible"
    webUrl: "https://discussion.theguardian.com/comment-permalink/169160146"
    apiUrl: "https://discussion.guardianapis.com/discussion-api/comment/169160146"
    numResponses: 1
    numRecommends: 47
    isHighlighted: true
    userProfile:
      userId: "4284581"
      displayName: "eternalsceptic"
      webUrl: "https://profile.theguardian.com/user/id/4284581"
      apiUrl: "https://discussion.guardianapis.com/discussion-api/profile/4284581"
      avatar: "https://avatar.guim.co.uk/user/4284581"
      secureAvatarUrl: "https://avatar.guim.co.uk/user/4284581"
      badge: []
```

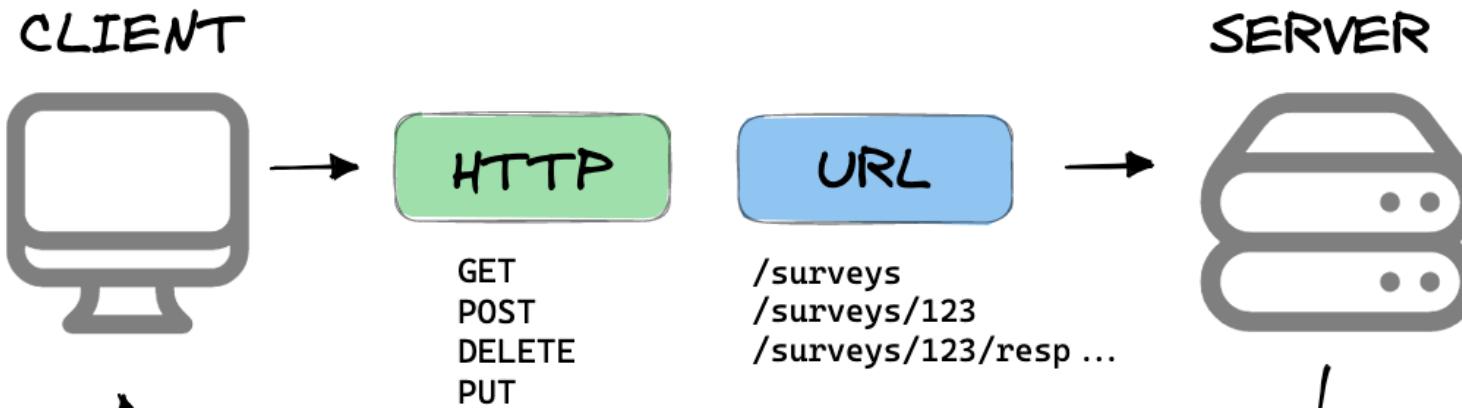
# Developer Tools in the browser

Inspect and analyze the HTML, CSS, and JavaScript of a webpage to understand its structure.  
Monitor network activity to see how data is requested and received.



# Web browsing works through HTTP(s) calls

- **HTTP** (Hypertext Transfer Protocol) calls = request a resource from a server
  - Can have *headers* and *body*, where extra information is sent to the server
- **URL** = location of the resource



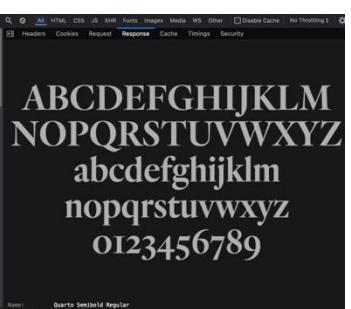
Nieuws

# Ook Ajax-FC Utrecht zondag afgelast vanwege politiestaking: 'Hiermee ontneem je supporters de kans om te laten zien dat het kan'

De Eredivisiewedstrijd tussen Ajax en FC Utrecht, die aankomende zondag in de Johan Cruijff Arena zou plaatsvinden, gaat niet door. Het is de tweede keer dat een

Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	www.parool.nl	/nieuws/ook-ajax-fc-utrecht-zondag-afgelast-vanwege-po	document	html	51,18 kB	22...
	POST	c.dpgmedia.net	b	68GC0udRiTWOX...			
0	GET	ingestion.smartoc...	r?p=0:m0w8d5zd:StNM679UFnv3c9bISQXgdu_X47ZI_kT~	ingestion.js:1 (xhr)	NS_BINDING_AB...		
200	GET	www.parool.nl	Quarto-Semibold.woff2	font	woff2	cached	17...
200	GET	www.parool.nl	Balto-Bold.woff2	font	woff2	cached	28...
200	GET	www.parool.nl	webpack-d539726d8629d14b.js	script	js	cached	6...
200	GET	www.parool.nl	5a2308db-197d117c075e5ad8.js	script	js	cached	17...
200	GET	www.parool.nl	8618-4e04cde1fe5c88d5.js	script	js	cached	12...
200	GET	www.parool.nl	main-app-23c7a5add3b63efc.js	script	js	cached	47...
	GET	www.parool.nl	7667-ddadb07973a8e84e.js	script	js	3 kB (raced)	8...
	GET	www.parool.nl	7851-a4b1a1f04c86042a.js	script	js	7,19 kB (raced)	21...
200	GET	www.parool.nl	114-90f0e964beab2111.js	script	js	cached	8...
200	GET	www.parool.nl	6571-1bf5426353e6e2a3.js	script	js	cached	13...
200	GET	www.parool.nl	5603-5007b131325ab046.js	script	js	cached	9...
200	GET	www.parool.nl	6016-04f32c51dcde3ca3.js	script	js	cached	9...
200	GET	www.parool.nl	1024-23c4cc7f3910878b.js	script	js	cached	10...
200	GET	www.parool.nl	2375-332aeee94059a7f20.js	script	js	cached	19...
200	GET	www.parool.nl	layout-3c573fd2c9e00c23.js	script	js	cached	29...
200	GET	www.parool.nl	4009-95f4238b40416690.js	script	js	cached	7...
200	GET	temptation.par...	temptation.js	script	js	cached	9...
200	GET	www.parool.nl	not-found-c5d982ab719aa3df.js	script	js	cached	26...
200	GET	www.parool.nl	error-a32b242946972f21.js	script	js	cached	6...
200	GET	www.parool.nl	1707-18e9e1c8b1885802.js	script	js	cached	27...

71 requests | 3,69 MB / 116,62 kB transferred | Finish: 3,93 s | DOMContentLoaded: 1,05 s | load: 1,25 s



# Ook Ajax-FC Utrecht zondag afgelast vanwege politiestaking: 'Hiermee ontneem je supporters de kans om te laten zien dat het kan'

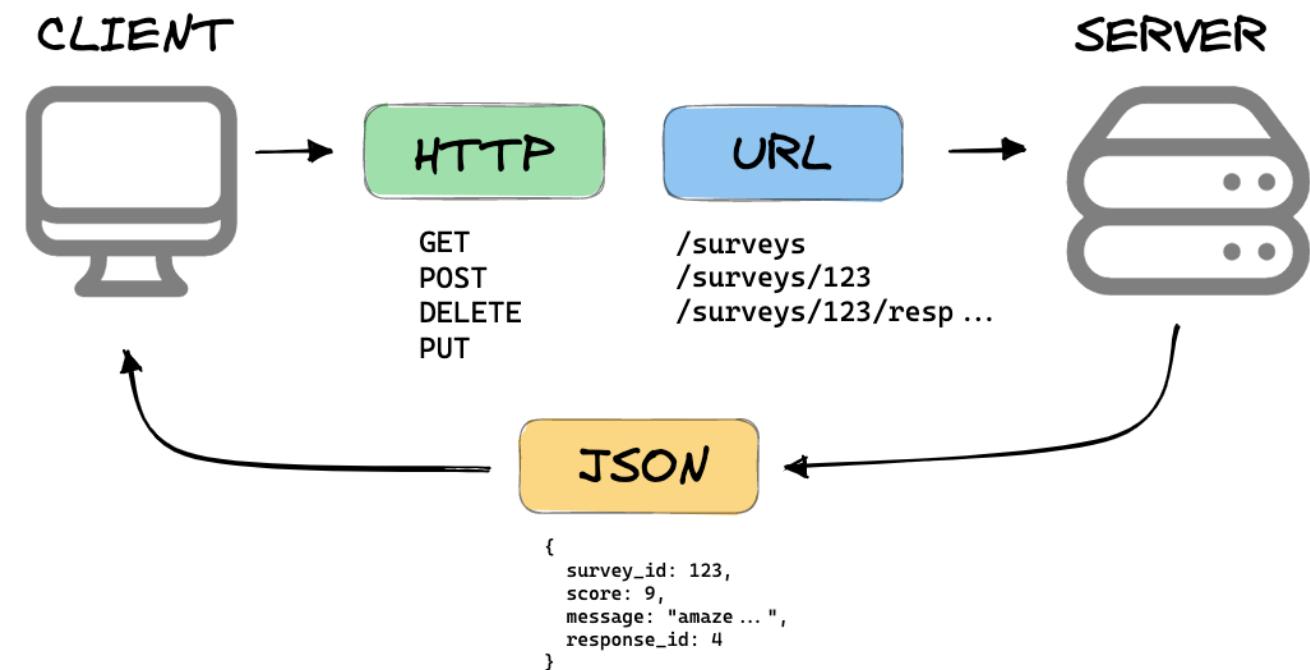
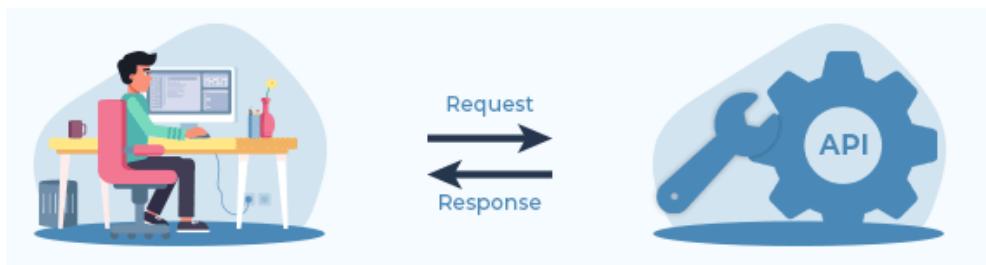
Data from their servers

Status	Method	Domain	File	Initiator	Type	Transferred
	POST	c.dpgmedia.net	b	<a href="#">68GC0udRITWOX0eR3pkv_v3.9.js:80</a> (...)		
200	GET	www.parool.nl	/api/_next-api/v1/auth/session/	<a href="#">7425-d5a90897e58d6a3c.js:1</a> (fetch)	json	1,40 kB
200	POST	profile-public-api.data.dpgmedia.clo...	dmoi	<a href="#">advert-xandr.js:1</a> (fetch)	json	386 B
200	OPTIONS	temptation.parool.nl	_resolve_from_website?trackingIdCookieName=TID_ID&jwtTokenCookieName=het-parool-oidc-access-tok	fetch	json	1,34 kB
200	GET	temptation.parool.nl	_resolve_from_website?trackingIdCookieName=TID_ID&jwtTokenCookieName=het-parool-oidc-access-tok	<a href="#">temptation.js:1</a> (fetch)	json	23,25 kB
200	GET	login.parool.nl	ssosession	<a href="#">main.js:1</a> (fetch)	json	763 B
200	GET	feeds.pexi.nl	bi6674262e89bda?brand=HP&articles=false&prices=true&pricetype=digitaal	<a href="#">temptation.js line 1 &gt; Function:49</a> (fetch)	json	1,08 kB
🚫	GET	ib.adnxs.com	getuidj	<a href="#">selfevokingxandr.js:2</a> (fetch)		Blocked By DuckDuckGo Privacy Essentials
200	GET	nmodpgendpoint.2cnt.net	?vendor=snowplow&cs_fpid=d241eb041f0c6b16fc4a92230608ddce0a463b77c6a181b1ab59d22ed2fb8c7	<a href="#">index.umd.min.js:7</a> (xhr)	plain	184 B
200	OPTIONS	c.dpgmedia.net	b	xhr	plain	349 B
200	GET	clientcdn.pushengage.com	3befc11084d1d0f40d7b419d4c128fa2?source=sdk&sdkv=3.0.44&swv=3.0.44	<a href="#">pushengage-subscription.js:2</a> (fetch)	json	cached
200	GET	web-sdk.pushengage.com	geo-details?sdkv=3.0.44&swv=3.0.44	<a href="#">pushengage-subscription.js:2</a> (fetch)	json	cached
200	GET	api.smartocto.com	tentacles?i=z7e3z8rzmqy2tcfecz7kp2n4dszsvqw8	<a href="#">tentacle.js:1</a> (xhr)	json	1,33 kB
204	GET	ingestion.smartocto.com	t?p=0:m0w8d5zd:StNM67 https://api.smartocto.com/api/brands/tentacles?	:1 (xhr)	xml	164 B

Ads, tracking, analytics, experiments (A/B testing), services

# Using RESTful APIs

- RESTful APIs work using **HTTP calls** to **URLs**
  - **HTTP** (Hypertext Transfer Protocol) calls = request a resource from a server (can be webpage/image)
    - Can have *headers* and *body*, where extra information is sent to the server
  - **URL endpoint** = location of the API resource
- Returns data (typically **JSON**)



# Examples of RESTful APIs



<https://api.wikimedia.org/feed/v1/wikipedia/en/featured/2024/09/20>

<https://api.crossref.org/works?query.author=“Javier Garcia-Bernardo”&filter=from-pub-date:2024-01-01,until-pub-date:2021-01-01>

<https://discussion.theguardian.com/discussion-api/discussion/p/3htd7/topcomments?pageSize=50&page=1&orderBy=newest>

[https://maps.googleapis.com/maps/api/geocode/json?address=Utrecht&key=YOUR\\_API\\_KEY](https://maps.googleapis.com/maps/api/geocode/json?address=Utrecht&key=YOUR_API_KEY)

How to read the url:

- Base API url = where to find the API
- Endpoints = requested resource, it can include required parameters
- Query (starts with "?") = parameters, usually optional, e.g., to filter/sort data
  - Parameters are concatenated with the symbol "&"

# Output of API (status)

Status code	Meaning
200 OK	Request was successful.
301 Moved Permanently	For SEO purposes when a page has been moved and all link equity should be passed through.
401 Unauthorized	Server requires authentication.
403 Forbidden	Client authenticated but does not have permissions to view resource.
404 Not Found	Page not found because no search results or may be out of stock.
500 Internal Server Error	Server side error. Usually due to bugs and exceptions thrown on the server side code.
503 Server Unavailable	Server side error. Usually due to a platform hosting, overload and maintenance issue.

# Example documentation

<https://discussion.theguardian.com/discussion-api/discussion/p/3htd7/topcomments?pageSize=50&page=1&orderBy=newest>

Method	Path	Slug	Description	Examples
GET	/abuse/category	GetAllAbuseCategories	Return all abuse categories	/abuse/category
GET	/comment/:id	GetComment	Get a comment	/comment/12500137, /comment/12500137?displayResponses=true, /comment/12500137?displayResponses=true&displayThreaded=true
GET	/comment/:id/context	GetCommentContext	Returns the discussion url, key, and page number on which the comment appears in the discussion. Useful for permalinks.	/comment/12500137/context
POST	/comment/:id/highlight	PostHighlightToComment	Post a highlight to an existing unhighlighted comment	
GET	/comment/:id/permalink	GetCommentPermalink	Redirect to a comment permalink	/comment/1250013/permalink
POST	/comment/:id/reason	PostReasonToComment	Post a reason to an existing comment	

The screenshot shows a browser window with the URL <https://discussion.theguardian.com/discussion-api/discussion/p/3htd7/topcomments?pageSize=50&page=1&orderBy=newest>. The page title is "https://discussion.theguardian.com/discussion-api/discussion/p/3htd7/topcomments?pageSize=50&page=1&orderBy=newest". Below the title, there are tabs for "JSON", "Raw Data", and "Headers". Under the "JSON" tab, there are buttons for "Save", "Copy", "Collapse All", "Expand All", and "Filter JSON". The JSON response is displayed as follows:

```
status: "ok"
categories:
  0:
    id: 1
    name: "Personal abuse"
    description: "Personal abuse"
    reasonRequired: false
  1:
    id: 2
    name: "Off topic"
    description: "Off topic"
    reasonRequired: false
  2:
```

# Example documentation II

<https://api.wikimedia.org/feed/v1/wikipedia/en/featured/2024/09/20>

## Featured content

 Discussion  Updated 16 June 2023

**GET** /feed/v1/wikipedia/{language}/featured/{YYYY}/{MM}/{DD}

Returns featured content from Wikipedia for a given date. Depending on [language availability](#), the response can include the daily featured article, featured image or media file, list of most read articles, latest news stories, and events from that day in history.

## Examples

curl Python PHP JavaScript

```
# Get today's featured content from English Wikipedia
curl https://api.wikimedia.org/feed/v1/wikipedia/en/featured/2024/09/10
```

## Parameters

<code>language</code> required path	Language code. For example: ar (Arabic), en (English), es (Spanish). <a href="#">List supported languages</a> .
<code>YYYY</code> required path	Four-digit year
<code>MM</code> required path	Zero-padded month, 01 through 12
<code>DD</code> required path	Zero-padded day of the month, 01 through 31

## Responses

200	Success <a href="#">Example</a>	[Show]
400	Error: Invalid parameter <a href="#">Example</a>	[Show]

## Response schema

<code>tfa</code> object	Today's featured article (TFA) for the requested date. Available in <a href="#">10+ languages</a> . <a href="#">Properties</a>	[Show]
<code>mostread</code> object	Previous day's most read articles. Available in <a href="#">300+ languages</a> . <a href="#">Properties</a>	[Show]
<code>image</code> object	Daily featured image from <a href="#">Wikimedia Commons</a> . Available in English. <a href="#">Properties</a>	[Show]
<code>news</code> object	Stories from today's news. Available only for the current day in <a href="#">UTC</a> . Available in <a href="#">15+ languages</a> . <a href="#">Properties</a>	[Show]
<code>onthisday</code> object	Events that occurred on this day in history. Available in <a href="#">5+ languages</a> <a href="#">Properties</a>	[Hide]
<code>text</code> string	Short summary of the event in plain text	
<code>pages</code> array	Articles related to the event <a href="#">Properties</a>	[Hide]
<code>type</code> string	Type of article: <ul style="list-style-type: none"><li>standard : Encyclopedia article</li><li>disambiguation : Page that links to articles covering topics with similar titles</li><li>no-extract : Article without an extract</li><li>mainpage : A wiki's homepage</li></ul>	

# Exercise (in pairs)

Create an API call for Reddit with 100 links containing the query “Utrecht”. Sort hot results first.

Base URL = <https://api.reddit.com>

Endpoint = /search

GET [/r subreddit]/search read rss support #

Search links page.  
*This endpoint is a listing.*

after	fullname of a thing
before	fullname of a thing
category	a string no longer than 5 characters
count	a positive integer (default: 0)
include_facets	boolean value
limit	the maximum number of items desired (default: 25, maximum: 100)
q	a string no longer than 512 characters
restrict_sr	boolean value
show	(optional) the string all
sort	one of (relevance, hot, top, new, comments)
sr_detail	(optional) expand subreddits
t	one of (hour, day, week, month, year, all)
type	(optional) comma-delimited list of result types (sr, link, user)

# Authentication and rate limits

## Authentication

[Page](#) [Discussion](#)

[Read](#) [View source](#) [View history](#) [☆](#)

Apps using the Wikimedia API should authenticate their requests using [OAuth 2.0](#). This provides a secure process for accessing Wikimedia resources and applies an app-specific rate limit. For a streamlined experience for evaluation and prototyping, you can authenticate using a personal [API token](#).

### 1. Create credentials

[Log in](#) with your Wikimedia account, and visit the [API keys dashboard](#). To create credentials, select **Create key**, and choose the **server-side app** option. After creating the key, you'll be shown a client ID and secret. Make sure to store these credentials securely before exiting the dialog.

Allow for different users to have access to different data/services

## Rate limits

[Page](#) [Discussion](#)

[Read](#) [View source](#) [View history](#) [☆](#)

Rate limits restrict API calls to a set number of requests per hour based on the type of request. A 429 response code indicates that the applicable rate limit has been exceeded.

These limits only apply to APIs with `api.wikimedia.org` as the base URL. Rate limits may vary depending on the API; see the [API catalog](#) for the rate limits applicable to each API. For higher rate limits, check out [Wikimedia Enterprise](#).

Limit the number of requests per minute you can make

## Anonymous requests

API requests without an access token are limited to 500 requests per hour per IP address.

## Personal requests

API requests authenticated using a [personal API token](#) are limited to 5,000 requests per hour.



# Usually wrappers exist

e.g. API for OpenAI

```
curl https://api.openai.com/v1/chat/completions \
-H "Authorization: Bearer $OPENAI_API_KEY" \
-H "Content-Type: application/json" \
-d '{ "model": "gpt-3.5-turbo", "messages": [{"role": "user", "content": "Say this is a test"}] }'
```

```
import os
from openai import OpenAI

client = OpenAI(
    # This is the default and can be omitted
    api_key=os.environ.get("OPENAI_API_KEY"),
)

chat_completion = client.chat.completions.create(
    messages=[
        {
            "role": "user",
            "content": "Say this is a test",
        }
    ],
    model="gpt-3.5-turbo",
)
```

# Finding private APIs (be nice and respect robots.txt/ToS)

The screenshot shows a browser window for <https://www.uu.nl/medewerkers/Zoek?medewerker=jav>. The page title is "Medewerkers Organogram". The developer tools Network tab is active, displaying two requests:

- GET https://www.uu.nl/medewerkers/RestApi/Public/search?expression={"medewerker":"jav"}&t=7adc477ab8014ae5 (Status: 200, Type: json, Size: 6,65 kB)
- GET https://www.uu.nl/medewerkers/RestApi/Public/getFooterLinks?l=NL (Status: 200, Type: json, Size: 2,01 kB)

The response for the search query is expanded, showing 8 employees in JSON format:

```
Count: 8
Departments: []
Employees: [
    {
        "Email": "j.a.cousosalas@uu.nl",
        "Id": 37734,
        "Name": "prof. dr. Javier Couso Salas",
        ...
    },
    {
        "Id": 62122,
        "Name": "dr. Javanshir Fouladvand",
        "Url": "JFouladvand",
        ...
    },
    {
        "Bio": "Assistant professor at Utrecht University in the Social Data Science (SoDa) team.",
        "Email": "j.garciabernardo@uu.nl",
        "Id": 62185,
        ...
    },
    {
        "Bio": "PhD Candidate. PhD Candidate Researching the metropolitan mobility system through active sustainable mobility by bicycle in Barcelona, Seville and Valencia.",
        "Id": 80174,
        "Name": "Javier Iturriño Guerrero",
        ...
    },
    {
        "Bio": "PhD candidate",
        "Id": 71088,
        "Name": "Komar Javanmardi",
        ...
    },
    {
        "Id": 60449,
        "Name": "A. (Ali) Javed",
        "Url": "AJaved",
        ...
    },
    {
        "Bio": "Analytical chemist at the Chemical Biology and Drug Discovery group, Utrecht University",
        "Email": "j.sastretorano@uu.nl",
        "Id": 9430,
        ...
    },
    {
        "Email": "j.gomezsubils@uu.nl",
        "Id": 73591,
        "Name": "Javier Subils",
        ...
    }
]
```

Response Headers (296 B):

- cache-control: private
- content-length: 6358
- content-type: application/json; charset=utf-8
- date: Tue, 10 Sep 2024 13:44:50 GMT
- server: Microsoft-IIS/10.0
- strict-transport-security: max-age=31536000; includeSubDomains; preload
- X-Firefox-Spdy: h2
- x-frame-options: SAMEORIGIN

# Getting access to an API (e.g. Wikimedia)

## 1. Log in with your Wikimedia account

To log in to the API Portal, use the same account you use for Wikipedia and other Wikimedia projects. If you're new to Wikimedia (Welcome!), you can create a free account now.



## 2. Create a personal API token

Once you've logged in, visit [API keys](#) to create and manage your API credentials. Select **Create key**, and select the **Personal API token** option. This token is tied to your account. It should only be used by you and should not be published or shared. Remember to store your token in a secure place, like a password manager.



A screenshot of a success message from a web application. At the top right, there are 'Create API key' and 'Done' buttons. Below them is a green horizontal bar with a checkmark icon and the text 'API key created'. Underneath, there are three sections: 'Client ID' with the value '5be43a7d72ac8d8091c6eaec285bfc5f', 'Client secret' with the value '7cff841af8c82341d9b2df33fa61f0bd19c2c399', and 'Access token' followed by a very long, complex string of characters representing the token itself.

Client ID:  
5be43a7d72ac8d8091c6eaec285bfc5f

Client secret:  
7cff841af8c82341d9b2df33fa61f0bd19c2c399

Access token:  
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiIYmU0M2E3ZDcyYWM4ZDgwOTFjNmVhZWMyODViZmM1ZilsImp0aSI6ImRiZmE5OGExMDhjOGI1OTI2YTl4NjEzMnU5NzRlYTxYJlwODY1ZDk4YzRhMzAwZGE4M2jkMjRmMDQ4NmE2OTJzJzQzOThhOWlyMDg4NmE0liwaWF0ljoxNzEzOTUyNDc5LjEzNzE5NCwibmJmljoxNzEzOTUyNDc5LjEzNzE5OCwiZxhwlijozMzl3MDg2MTI3OS4xMzM2Niwc3VlijoiNzU1MDI3NDQiLCJpc3MiOjodHRwcovL21IdGEud2IraW1ZGhLm9yZylslnJhdGVsaW1pdCl6eyJyZXF1ZXN0c19wZXJfdW5pdCl6NTAwMCwidW5pdCl6IkhPVVifSwic2NvcGVzJpbImJhc2ijI19.CkCXA65R9oObvuQQaC1tG71Z25RQUdnA-SIOQDMKmActEcn9ADwb0WfAiTkYg6t0HMOLRR1-hOSWo3Jtcm-f5ZGtQXB1EFXgqNKPO\_0H5Hala\_50rbYY79AxmtT1A9JWIQq64RhizAj92Fzbo-U5-jxNw1aXLoSEUFbzrR-2-i2-M16DTDNW-ccAQwulMmchul1bbdrCoziIDAlJhCzb\_KC2zgkPz-BjOC-SCQ-Qadz8

### 3. Get today's featured article

For your first request, call the [featured content endpoint](#) to get today's featured article from English Wikipedia. Use your API token to authenticate the request.

curl

Python

PHP

JavaScript

```
# Python 3
# Get today's featured article from English Wikipedia

import datetime
import requests

today = datetime.datetime.now()
date = today.strftime('%Y/%m/%d')

url = 'https://api.wikimedia.org/feed/v1/wikipedia/en/featured/' + date

headers = {
    'Authorization': 'Bearer YOUR_ACCESS_TOKEN',
    'User-Agent': 'YOUR_APP_NAME (YOUR_EMAIL_OR_CONTACT_PAGE)'
}

response = requests.get(url, headers=headers)
data = response.json()
print(data)
```

# API world (before ~2020)



*Offering to Bacchus, 1720 (Michel-Ange Houasse)*

# APIcalypse

(Bruns, 2021)

2015: LinkedIn

Reason: Privacy/competition

2015/8: Facebook/Instagram

Reason: Improve privacy

2023: Twitter

Reason: Monetize data access (Musk)

2023: Reddit

Reason: Monetize data access



*The Raft of the Medusa, 1818–19 (Théodore Géricault)*

# Digital Services Act (DSA)

19 October 2022

## Article 40, Data access and scrutiny

12. Providers of very large online platforms or of very large online search engines **shall give access without undue delay to data**, including, where technically possible, to real-time data, **provided that the data is publicly accessible in their online interface by researchers**, including those affiliated to not for profit bodies, organisations and associations, who comply with the conditions set out in paragraph 8, points (b), (c), (d) and (e), and who use the data solely for performing research that contributes to the **detection, identification and understanding of systemic risks** in the Union pursuant to Article 34(1).



### Researcher access

Last updated: 1 year ago

---

As of August 2023, we offer a [Beta] Researcher Access Program in order to meet our legal requirements under the European Union's Digital Services Act (DSA). Specifically, Article 40(12) of the DSA requires that we make available to qualified researchers, upon application, data that is publicly accessible on LinkedIn, so long as the following conditions are satisfied:

- Researchers must be independent from commercial interests.
- The researcher's application must disclose the funding of the research

# Digital Services Act (DSA)

19 October 2022



EN English

Search

Search

[Home](#) > [Press corner](#) > [Commission opens formal proceedings under DSA](#)

Available languages: English ▾

PRESS RELEASE | 30 April 2024 | Brussels | 6 min read

## Commission opens formal proceedings against Facebook and Instagram under the Digital Services Act

PRESS RELEASE | 19 February 2024 | Brussels | 3 min read

## Commission opens formal proceedings against TikTok under the Digital Services Act

PRESS RELEASE | 18 December 2023 | Brussels | 4 min read

## Commission opens formal proceedings against X under the Digital Services Act

PRESS RELEASE | 14 March 2024 | Brussels | 3 min read

## Commission opens formal proceedings against AliExpress under the Digital Services Act

Mechanism Description	Interface	Who has access?	Application required	Data Dictionary /Docu...
Apple Ad Repository and API	Searchable web interface and API	Public	No	Yes
Booking Ads Repository and API	Searchable web interface and API	Public	No	Yes
Google Ads Transparency Center	Searchable web interface and API	Public	No	Yes
LinkedIn Ad Library	Searchable web interface and API	Public	Yes	Yes
Meta Ad Library	Searchable web interface and API	Public	No	Yes
Microsoft Ad Library	Searchable web interface and API	Public	No	Yes
Pinterest Ads Repository	Searchable web interface and API	Public	No	No
Snap Ads Gallery	Searchable web interface	Public	No	No
Snap Political Ads Library	CSV Files	Public	No	Yes
TikTok Ad Library	Searchable web interface and API	Public	Yes	Yes
X (formerly Twitter) Ads Repository	Searchable web interface and API	Public	No	Yes

Mechanism Description	Interface	Who has access?	Application required	Data Dictionary /Docu...
AliExpress Open Research & Transparency	The mechanism description URL suggest...	Academic and civil society researchers as describ...	Yes	No
Booking.com Scraping Provision1	API "as applicable" (suggesting other met...	Anyone scraping for non-commercial purposes	No	No
Bing Qualified Researcher Program	API "as applicable" (suggesting other met...	Academic and civil society researchers as describ...	Yes	No
Google Request Records	Varies, see below:	Researchers affiliated with EU-based organizations	Yes	No
LinkedIn Researcher Access	API "as applicable" (suggesting other met...	Academic and civil society researchers as describ...	Yes	No
Meta Content Library and API	Searchable user interface and API provid...	To be eligible for product access, researchers mu...	Yes	Yes
Pinterest Researchers Intake		"If you're a researcher"	Yes	No
Reddit Researcher Access Request	Commercial API	Researchers accessing data for non- commercial ...	Yes	Yes
Snap Researcher Data Access		Requests are "in accordance with the Digital Servi...	Yes	No
TikTok Research API	API	Researchers from US and Europe	Yes	Yes
Wikipedia Tools	Scraping and a set of APIs	Public	No	Yes
X (formerly Twitter) API	Commercial API	Different levels of access on the basis of fees and...	Yes	Yes
YouTube Researcher Program	Commercial API	Must be affiliated with an "eligible academic instit...	Yes	Yes

# Advantages and Disadvantages of APIs

Advantages for the company:

- Track the usage of their data
- Safety and authentication (give access to specific endpoints to specific users)
- Rate limits (a maximum number of requests per minute)

Advantages for you:

- Easy(ish)
- No legal issues---but you usually have to accept the Terms of Service to get an API key.

Disadvantages for you:

- Only public data, or data where individuals cannot be identified
- The data you want may not be available (e.g. Facebook) or be very expensive (e.g. Twitter)

# Collaboration with companies

# Meta/Facebook case

## SOCIAL SCIENCE ONE

Hosted by Harvard's Institute for Quantitative Social Science

BLOG CONTACT US

### Building Industry-Academic Partnerships



Projects must be focused on the effects of social media on democracy and elections.

### RFP for URL Shares

This is a codebook for data on the demographics of people who viewed, shared, and otherwise interacted with web pages (URLs) shared on Facebook. The data has about 68 million URLs, over 3.1 trillion rows, and over 71 trillion cell values. It results from a collaboration between Facebook and Social Science One (at IQSS at Harvard), originally prepared for Social Science One grantees and describes the "full" URLs dataset, including its scope, structure, and fields. This is version 10 of the codebook and data (released 4/13/2023), first described by Gary King and Nathaniel Persily at <https://socialscience.one/blog/update-social-science-one> (2023).

Aggregated data available directly

#### FACEBOOK

Accessible data includes posts shared to and information about Pages, groups and events, as well as a subset of public profiles that have a [verified badge](#) or 25,000 or more followers.

#### GEOGRAPHICAL DATA

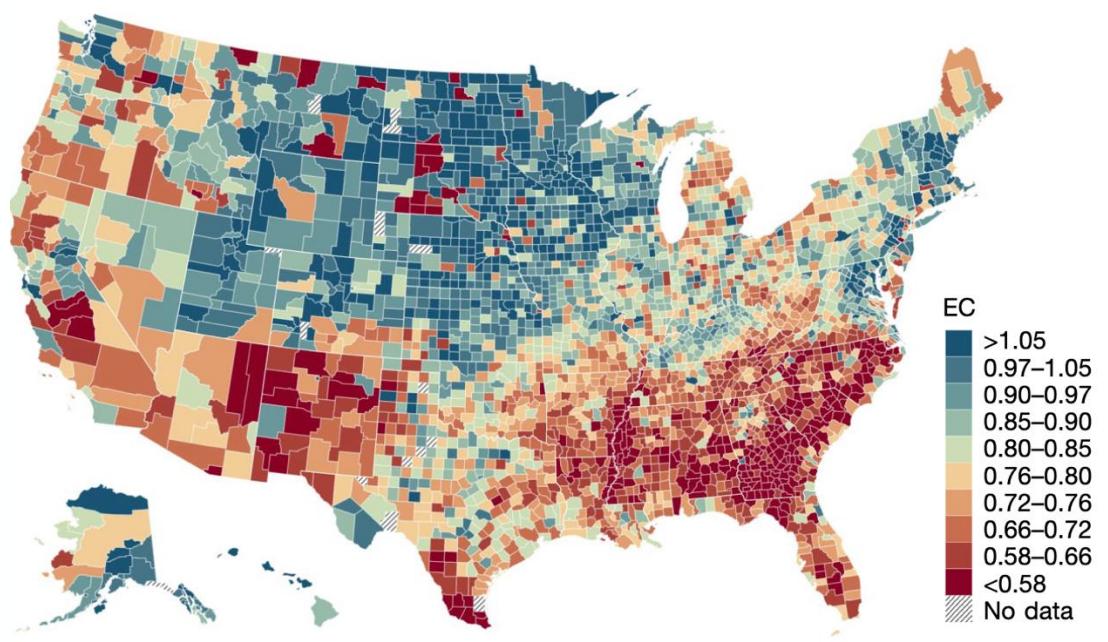
Data is available for most countries and territories but excluded from countries where Meta is still evaluating legal and compliance requirements.

#### INSTAGRAM

Accessible data includes posts shared by and information about business and creator accounts, as well as a subset of personal accounts that have been [set to public](#) and have a [verified badge](#) or 25,000 or more followers.

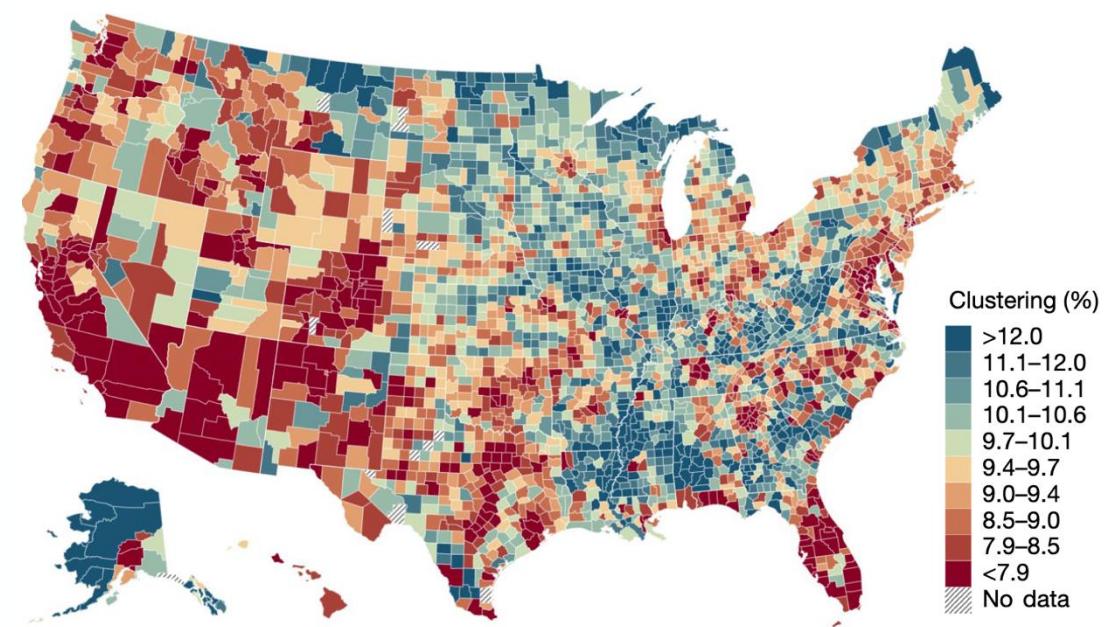
#### POST VIEW DATA

The post views metric indicates the number of times a post or reel was displayed on screen.



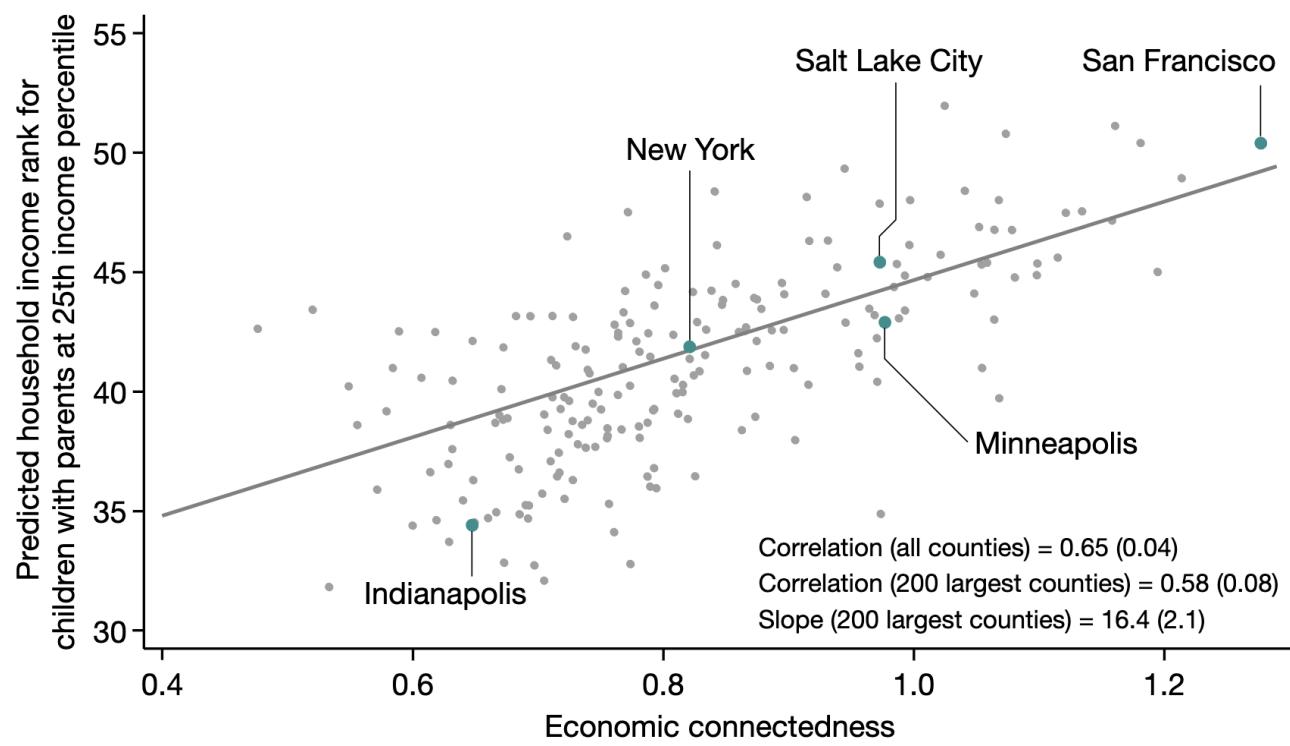
EC

- >1.05
- 0.97–1.05
- 0.90–0.97
- 0.85–0.90
- 0.80–0.85
- 0.76–0.80
- 0.72–0.76
- 0.66–0.72
- 0.58–0.66
- <0.58
- No data



Clustering (%)

- >12.0
- 11.1–12.0
- 10.6–11.1
- 10.1–10.6
- 9.7–10.1
- 9.4–9.7
- 9.0–9.4
- 8.5–9.0
- 8.5–9.0
- 7.9–8.5
- <7.9
- No data



*Chetty et al, 2022, Nature*

# Experimental evidence of massive-scale emotional contagion through social networks

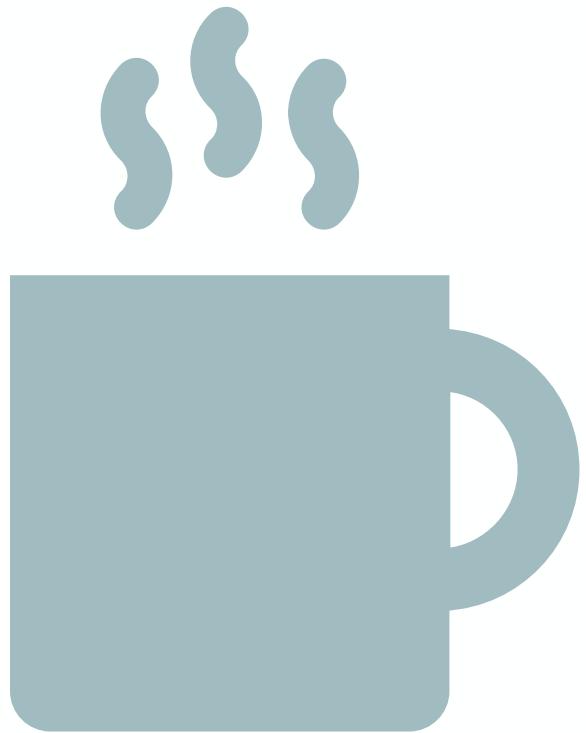
Adam D. I. Kramer , Jamie E. Guillory, and Jeffrey T. Hancock [Authors Info & Affiliations](#)

Edited by Susan T. Fiske, Princeton University, Princeton, NJ, and approved March 25, 2014 (received for review October 23, 2013)

June 2, 2014 | 111 (24) 8788-8790 | <https://doi.org/10.1073/pnas.1320040111>

## When positive expressions were reduced, people produced fewer positive posts and more negative posts

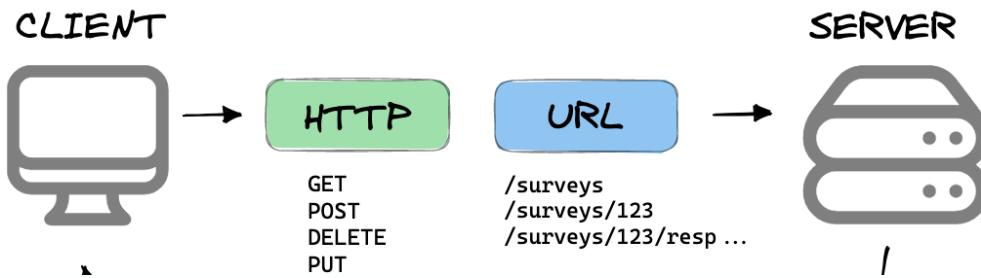
The authors noted in their paper, “[The work] was consistent with Facebook’s Data Use Policy, to which all users agree prior to creating an account on Facebook, constituting informed consent for this research.”



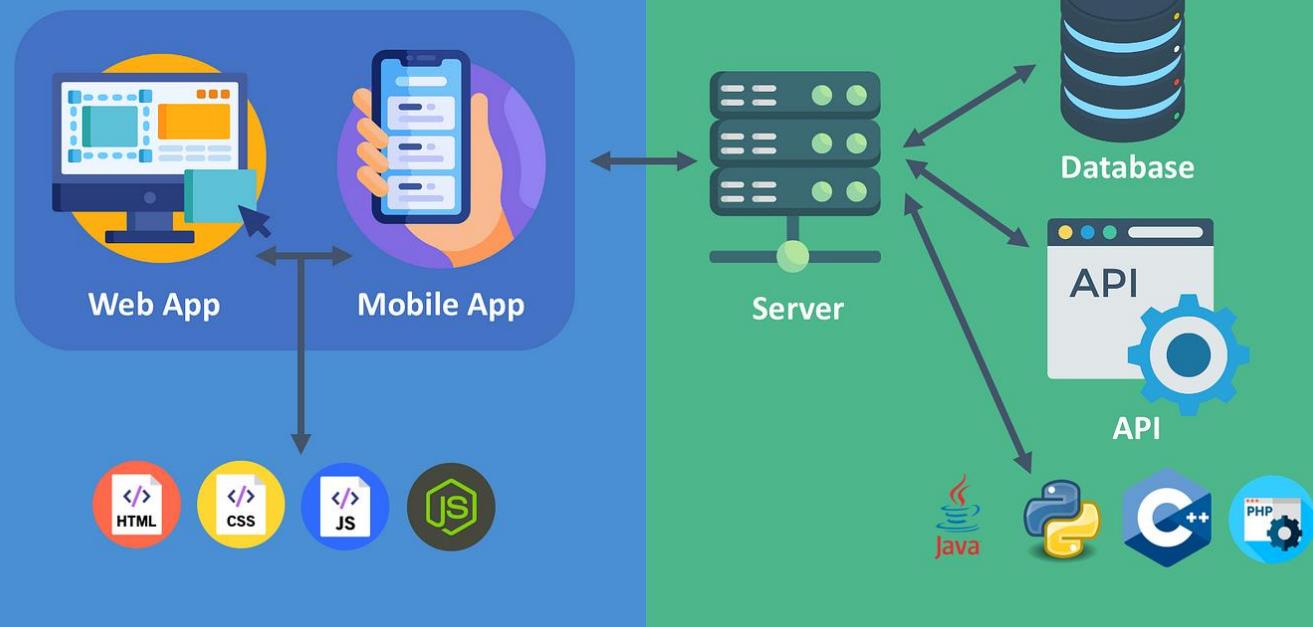
**15 min break**

# Web Scraping

# Web pages



FRONT-END      BACK-END



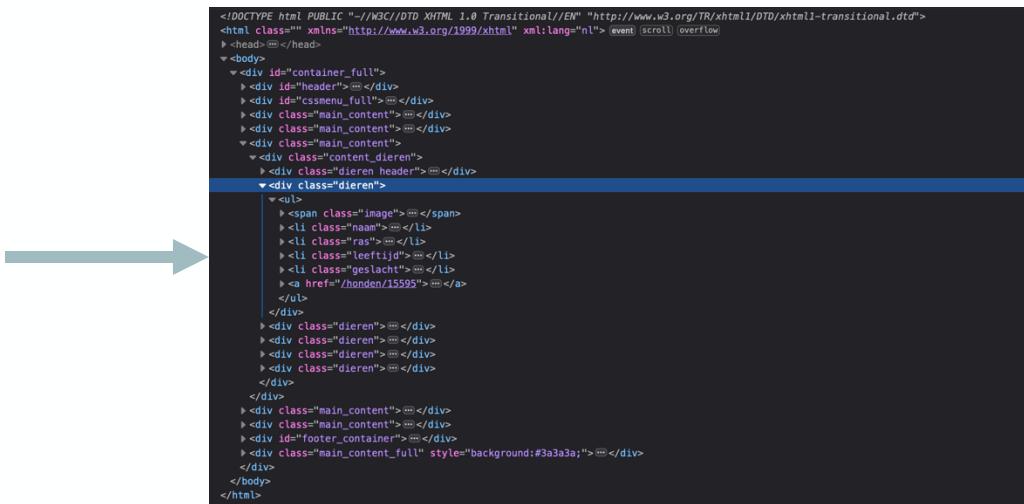
- **HTML** (HyperText Markup Language): The skeleton of a website, containing the sections and text.
- **CSS** (Cascading Style Sheets): Makes it pretty by adding styles and colors.
- **JS** (JavaScript): Brings complex interactivity and logic to the website

# What is web scraping?

Extracting data from the web in an automatic manner

Naam	Ras	Leeftijd	Geslacht	
	Tibbe	Kruising doodle	Volwassen	Reu
				<button>Beschikbaar</button>
	Pepper	Boerboel	Volwassen	Reu
				<button>Beschikbaar</button>
	Pepper	Amerikaanse Bulldog	Volwassen	Teef
				<button>Beschikbaar</button>
	Rocky	jack russel terriër	Volwassen	Reu
				<button>Geplaatst</button>
	Dexx	Kruising	Volwassen	Reu
				<button>Beschikbaar</button>
	Poppy	Kruising	Volwassen	Teef
				<button>Beschikbaar</button>

Start with a website you can legally scrape



The diagram illustrates the process of web scraping. It starts with a screenshot of a website displaying a list of dogs with their names, breeds, ages, genders, and availability status ('Beschikbaar' or 'Geplaatst'). An arrow points from this table to a block of raw HTML code representing the website's structure. Another arrow points from the HTML code to a large blue icon of a house with the word 'CSV' below it, representing the final output format.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="nl"> event scroll overflow
<head> ... </head>
<body>
  <div id="container_full">
    <div id="header"> ... </div>
    <div id="cssmenu_full"> ... </div>
    <div class="main_content"> ... </div>
    <div class="main_content"> ... </div>
    <div class="main_content">
      <div class="content_dieren">
        <div class="dieren_header"> ... </div>
        <div class="dieren">
          <ul>
            <li><span class="image"> ... </span>
            <li class="naam"> ... </li>
            <li class="ras"> ... </li>
            <li class="leeftijd"> ... </li>
            <li class="geslacht"> ... </li>
            <a href="/honden/15595"> ... </a>
          </ul>
        </div>
        <div class="dieren"> ... </div>
        <div class="dieren"> ... </div>
        <div class="dieren"> ... </div>
        <div class="dieren"> ... </div>
      </div>
    </div>
    <div class="main_content"> ... </div>
    <div class="main_content"> ... </div>
    <div id="footer_container"> ... </div>
    <div class="main_content_full" style="background:#3a3a3a;"> ... </div>
  </div>
</body>
</html>
```

1. Download the HTML code of the website

2. Parse the HTML

# How Censorship in China Allows Government Criticism but Silences Collective Expression

GARY KING *Harvard University*

JENNIFER PAN *Harvard University*

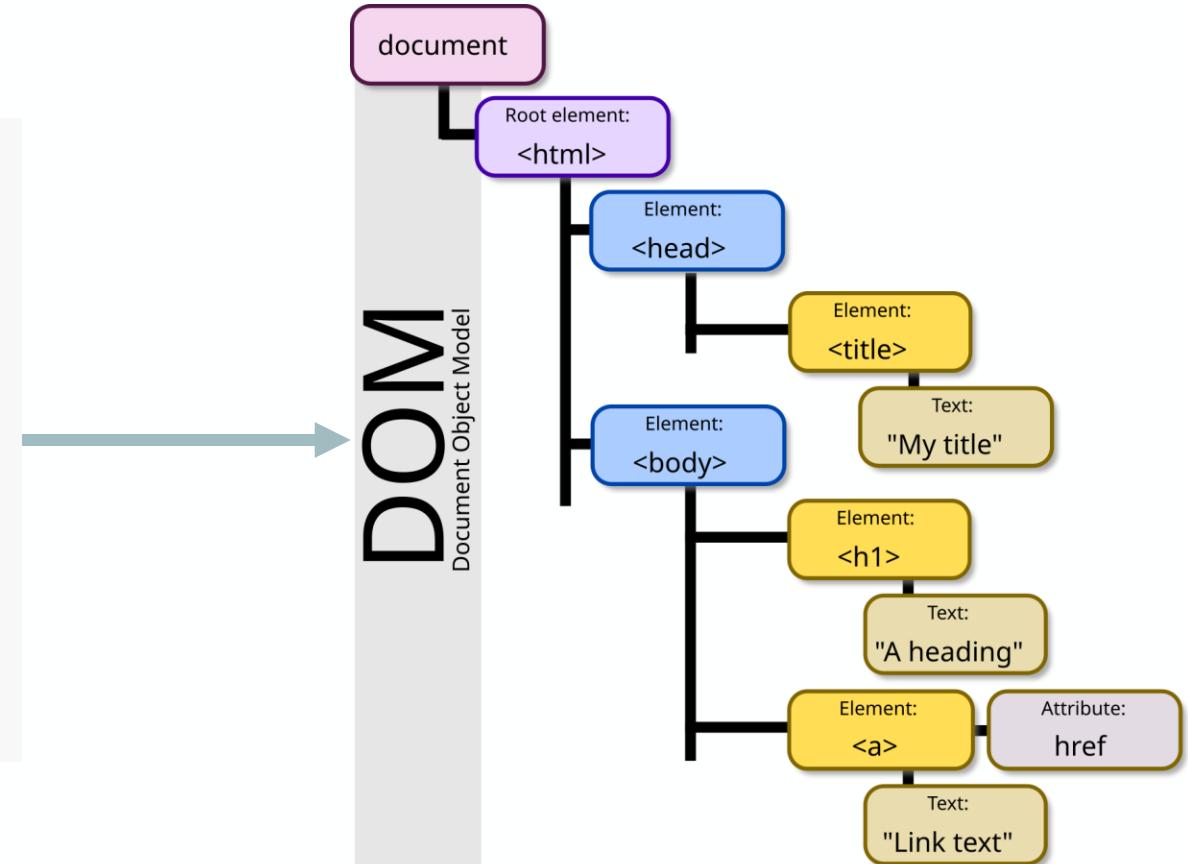
MARGARET E. ROBERTS *Harvard University*

We offer the first large scale, multiple source analysis of the outcome of what may be the most extensive effort to selectively censor human expression ever implemented. To do this, we have devised a system to locate, download, and analyze the content of millions of social media posts originating from nearly 1,400 different social media services all over China before the Chinese government is able to find, evaluate, and censor (i.e., remove from the Internet) the subset they deem objectionable. Using modern computer-assisted text analytic methods that we adapt to and validate in the Chinese language, we compare the substantive content of posts censored to those not censored over time in each of 85 topic areas. Contrary to previous understandings, posts with negative, even vitriolic, criticism of the state, its leaders, and its policies are not more likely to be censored. Instead, we show that the censorship program is aimed at curtailing collective action by silencing comments that represent, reinforce, or spur social mobilization, regardless of content. Censorship is oriented toward attempting to forestall collective activities that are occurring now or may occur in the future—and, as such, seem to clearly expose government intent.

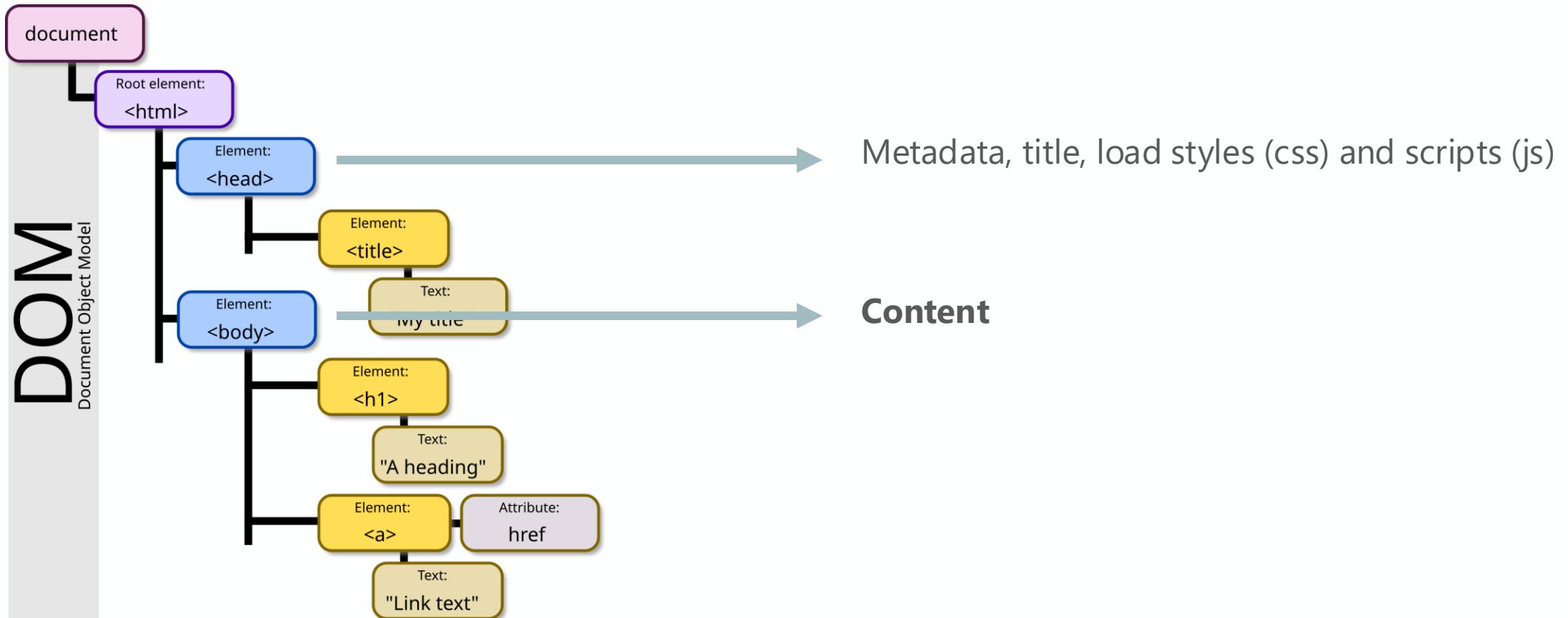
# Document Object Model (DOM)

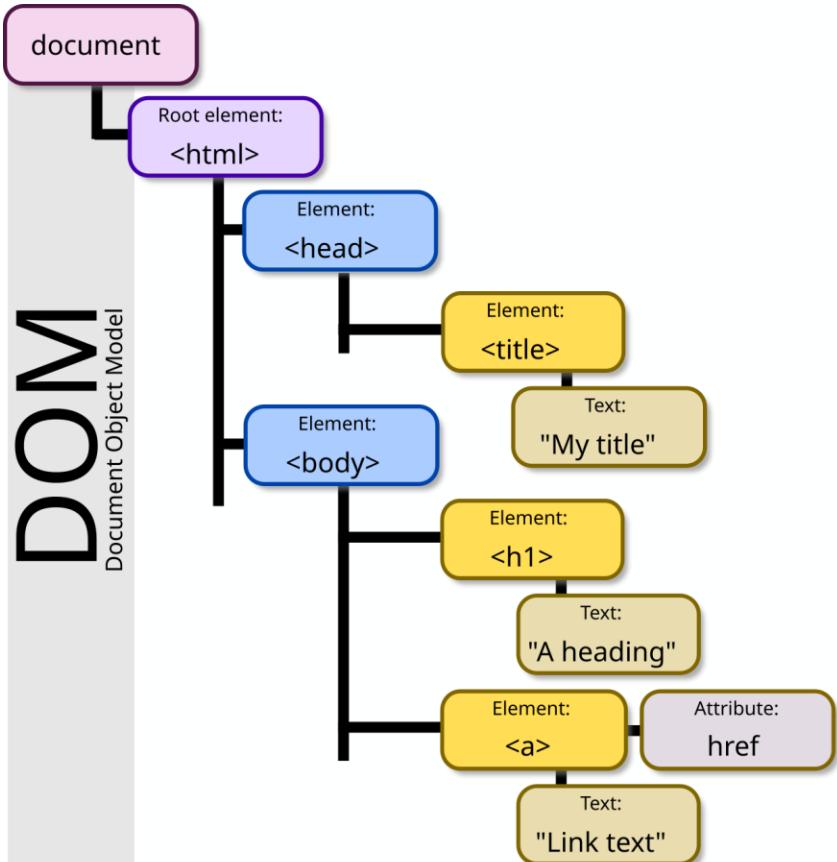
Represents the HTML code as a tree

```
<html>
  <head>
    <title>My Website</title>
  </head>
  <body>
    <h1>Welcome</h1>
    <p>This is my website.</p>
  </body>
</html>
```



# Document Object Model (DOM)





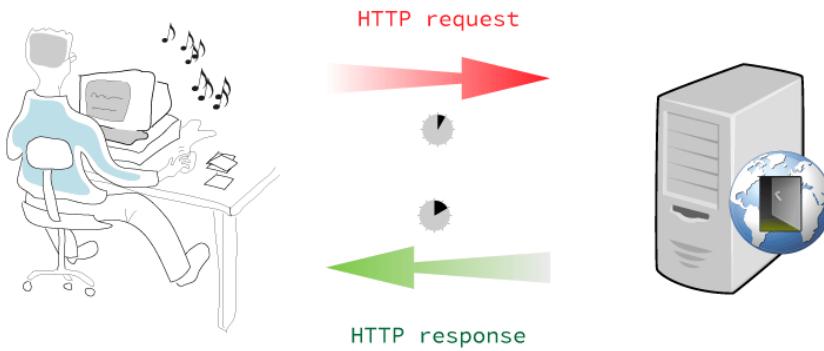
### List of elements (tags):

- div: basic container
- a: link to url
- p: paragraph
- h1-h6: titles
- img: image
- table: table

### List of common attributes:

- id: unique of the element
- class: set the style/interactivity for multiple elements
- href: url
- ...

# Scraping static websites



<https://about.gitlab.com/blog/2016/06/03/ssg-overview-gitlab-pages-part-1-dynamic-x-static/>

Naam	Ras	Leeftijd	Geslacht	
	Tibbe Kruising doodle	Volwassen	Reu	<button>Beschikbaar</button>
	Pepper Boerboel	Volwassen	Reu	<button>Beschikbaar</button>
	Pepper Amerikaanse Bulldog	Volwassen	Teef	<button>Beschikbaar</button>
	Rocky jack russel terrier	Volwassen	Reu	<button>Geplaatst</button>
	Dexx Kruising	Volwassen	Reu	<button>Beschikbaar</button>
	Poppy Kruising	Volwassen	Teef	<button>Beschikbaar</button>

<https://www.dierenasielutrecht.nl/honden/>

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html class="" xmlns="http://www.w3.org/1999/xhtml" xml:lang="nl"> [event] [scroll] [overflow]
```

```
► <head> [..] </head>
```

```
▼ <body>
```

```
  ▼ <div id="container_full">
```

```
    ► <div id="header"> [..] </div>
```

```
    ► <div id="cssmenu_full"> [..] </div>
```

```
    ► <div class="main_content"> [..] </div>
```

```
    ► <div class="main_content"> [..] </div>
```

```
  ▼ <div class="main_content">
```

```
    ▼ <div class="content_dieren">
```

```
      ► <div class="dieren header"> [..] </div>
```

```
      ▼ <div class="dieren">
```

```
        ▼ <ul>
```

```
          ► <span class="image"> [..] </span>
```

```
          ► <li class="naam"> [..] </li>
```

```
          ► <li class="ras"> [..] </li>
```

```
          ► <li class="leeftijd"> [..] </li>
```

```
          ► <li class="geslacht"> [..] </li>
```

```
          ► <a href="/honden/15595"> [..] </a>
```

```
        </ul>
```

```
      </div>
```

```
      ► <div class="dieren"> [..] </div>
```

```
    </div>
```

```
  </div>
```

```
  ► <div class="main_content"> [..] </div>
```

```
  ► <div class="main_content"> [..] </div>
```

```
  ► <div id="footer_container"> [..] </div>
```

```
  ► <div class="main_content_full" style="background:#3a3a3a;"> [..] </div>
```

```
  </div>
```

```
</body>
```

```
</html>
```

Naam	Ras	Leeftijd	Geslacht	
	Tibbe	Kruising doodle	Volwassen	Reu
	Pepper	Boerboel	Volwassen	Reu

```
▼ <div class="dieren">
```

```
  ▼ <ul>
```

```
    ► <span class="image"> [..] </span>
```

```
    ▼ <li class="naam">
```

```
      <span>Naam:</span>
```

```
      Tibbe
```

```
    </li>
```

```
    ▼ <li class="ras">
```

```
      <span>Ras:</span>
```

```
      Kruising doodle
```

```
    </li>
```

```
    ▼ <li class="leeftijd">
```

```
      <span>Leeftijd:</span>
```

```
      Volwassen
```

```
    </li>
```

```
    ▼ <li class="geslacht">
```

```
      <span>Geslacht:</span>
```

```
      Reu
```

```
    </li>
```

```
    ▼ <a href="/honden/15680">
```

```
      ► <li class="status beschikbaar"> [..] </li>
```

```
      </a>
```

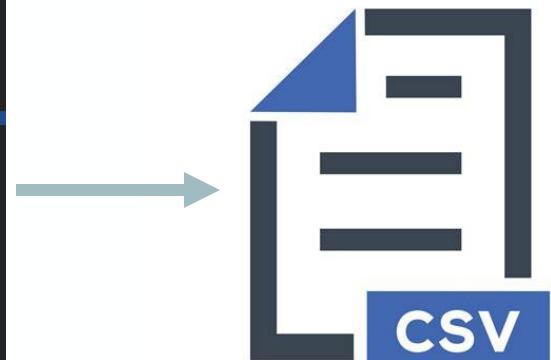
```
    </ul>
```

```
  </div>
```

Naam	Ras	Leeftijd	Geslacht	
Tibbe	Kruising doodle	Volwassen	Reu	Beschikbaar
Pepper	Boerboel	Volwassen	Reu	Beschikbaar
Pepper	Amerikaanse Bulldog	Volwassen	Teef	Beschikbaar
Rocky	jack russel terriér	Volwassen	Reu	Geplaatst
Dexx	Kruising	Volwassen	Reu	Beschikbaar
Poppy	Kruising	Volwassen	Teef	Beschikbaar



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html class="" xmlns="http://www.w3.org/1999/xhtml" xml:lang="nl"> event scroll() overflow
  <head> </head>
  <body>
    <div id="container_full">
      <div id="header"></div>
      <div id="cssmenu_full"></div>
      <div class="main_content"></div>
      <div class="main_content"></div>
      <div class="main_content"></div>
      <div class="content_dieren">
        <div class="dieren_header"></div>
      </div>
      <div class="dieren">
        <span class="image"></span>
        <ul class="naam">
          <li class="naam"></li>
          <li class="ras"></li>
          <li class="leeftijd"></li>
          <li class="geslacht"></li>
          <a href="/honden/15595"></a>
        </ul>
      </div>
      <div class="dieren"></div>
      <div class="dieren"></div>
      <div class="dieren"></div>
      <div class="dieren"></div>
    </div>
    <div class="main_content"></div>
    <div class="main_content"></div>
    <div id="footer_container"></div>
    <div class="main_content_full" style="background:#3a3a3a;"></div>
  </body>
</html>
```



Start with a website you can legally scrape

1. Download the HTML code of the website

2. Parse the HTML

1. Download the HTML code of the website (easy)

```
import requests
r = requests.get("https://www.dierenasielutrecht.nl/honden/")
print(r.status_code)
print(r.text)

200
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="nl">
<head>
  <title>Honden - Dierenasiel Utrecht</title>
  <meta name="author" content="Dierenasiel, Utrecht" />
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <meta http-equiv="content-script-type" content="text/javascript" />
  <meta http-equiv="content-style-type" content="text/css" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

2. Parse the HTML (can be very hard)

```
import bs4
html = bs4.BeautifulSoup(r.text)
html.find_all("li", attrs={"class": "naam"})

[<li class="naam">Naam</li>,
 <li class="naam"><span>Naam:</span>Tibbe</li>,
 <li class="naam"><span>Naam:</span>Pepper </li>,
 <li class="naam"><span>Naam:</span>Pepper</li>,
 <li class="naam"><span>Naam:</span>Dexx</li>,
 <li class="naam"><span>Naam:</span>Poppy</li>]
```

# Exercise (in pairs)

What HTML element contains the date when the dog was added?  
What element/attribute would you need to select to extract the image?



Milo

IJMUIDEN 18-09-2024

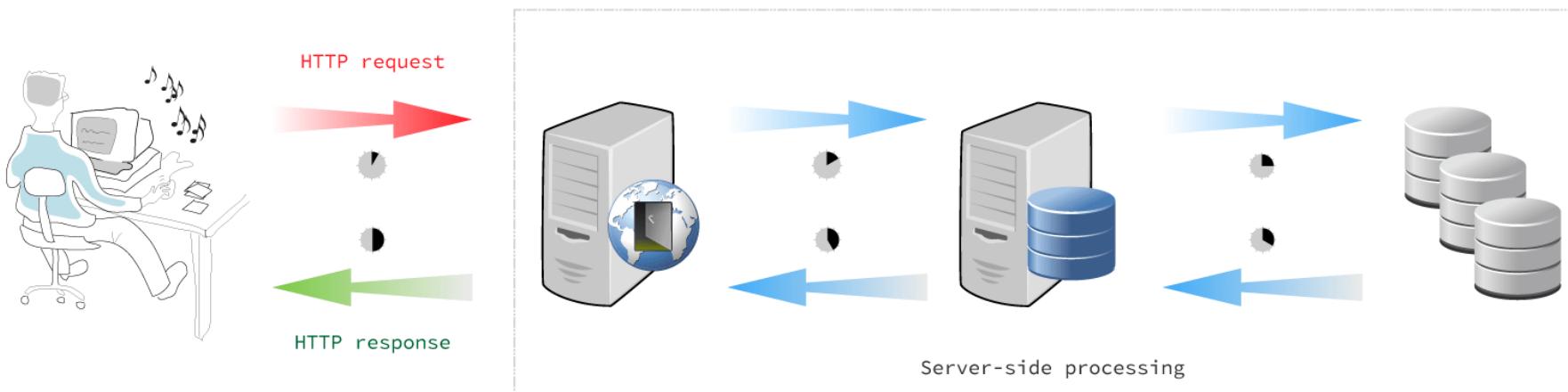
```
▼<a class="animal-card-grid" target="_blank" href="/plaatsprofiel/189022-adoptiehond-zoekt-baasje-griekse-berghondbeagle-ijmuiden/"> event grid
  ▼<div class="card-image">
    
      event
    </div>
  ▼<div>
    ▼<div class="card-header">
      ▼<div class="align-items-center">
        <h3 class="font-size-subtitle text-dark">Milo</h3>
      </div>
    </div>
    ▼<div class="card-body">
      ▼<div class="d-flex justify-content-between align-items-center"> flex
        ▼<div>
          <label class="font-family-system">IJmuiden</label>
          <span class="text-truncate text-gray-light">18-09-2024</span>
        </div>
        <div class="text-gray-light small"></div>
      </div>
    </div>
  </div>
```

<https://verhuisdieren.nl/honden/adoptiehonden/>

# Scraping dynamic Websites

Examples:

- Infinite scrolls (most social media these days)
- Websites that update parts of the website when the user interacts



<https://about.gitlab.com/blog/2016/06/03/ssg-overview-gitlab-pages-part-1-dynamic-x-static/>

A composite image featuring two separate scenes. On the left, a man with dark hair and a beard, wearing a light blue button-down shirt over a white t-shirt, stands with his hands clasped in front of him against a textured, light-colored wall. On the right, an orangutan wearing a bright red suit is sitting on a large, light-colored rock, facing towards the left. The background of the right scene shows a stone wall and some foliage.

Your scraper

# Advantages and Disadvantages of Scraping

Advantages for company:

- No need to set up a separate API

Disadvantage for company:

- No control about their data
- Typically not allowed in the Terms of Service

Advantages for you:

- All publicly available data can be scraped

Disadvantages for you:

- You may be interested in data that is not public
- It can be hard
  - Technically: Especially if the website actively tries to prevent scraping (e.g. captchas). Your code will tend to break.
  - To clean up the data: harmonizing data is prone to errors
- Complex legally, usually not allowed in the Terms of Service.

# How to do this practically?

## Programming:

- Static websites: *requests* + *beautifulsoup* (*bs4*)
- Dynamic websites: *selenium*

## Web browser extensions (practical):

- Click on elements
- The software figures it out by itself
- Not so flexible

# Legality of APIs/web scraping

# Different levels (specific for web scraping/APIs)

## **robots.txt**

- Used to communicate what parts of a website should or should not be accessed
- Not legally binding, but please respect them!

## **Terms of Service**

- Legal agreements that define how a user or a scraper are allowed to interact with a website
- Only enforceable if the user clicks "I agree" (e.g. by creating an account, or through a pop-up)
- Can have serious legal consequences, especially if the user bypasses authentication to download data (e.g. created fake account to access data, scraped data in bulk from an institutional account, etc).
- Some companies (LinkedIn, Meta) have already won legal battles against scraping for commercial purposes



Aaron Swartz

# Robots.txt

← → ⌂



https://nos.nl/robots.txt

```
# www.robotstxt.org/
# www.google.com/support/webmasters/bin/answer.py?hl=en&answer=156449

User-agent: GPTBot
Disallow: /

User-agent: *
Disallow: /hybrid/
Disallow: /humans.txt
Disallow: /api
Disallow: /zoeken

Sitemap: https://nos.nl/sitemap/index.xml
```

# Terms of Service

App/Service	Word Count	How many minutes to read? (240 wpm)
Microsoft	15,260	63.5
Spotify	8,600	35.8
Niantic (Pokemon Go)	8,466	35.2
TikTok	7,459	31.4
Apple (Media Services)	7,314	30.5
Zoom	6,891	28.7
Tinder	6,215	25.9
Slack	5,782	24.1
Uber	5,658	23.6
Twitter	5,633	23.5

<https://www.visualcapitalist.com/terms-of-service-visualizing-the-length-of-internet-agreements/>

# More regulations (EU)

## **Database Directive** (Directive 96/9/EC)

- Databases are protected products
- Non-commercial scientific research is exempted (with caveats)

## **Copyright law**

- The copyright of created materials can lie with the user (e.g. Reddit) or the platform (e.g TikTok)
- Personal data cannot be protected by copyright

## **Digital Copyright Directive** (Directive 2019/790)

- Exemptions for text and data mining: Research are allowed to mine public data (like Reddit posts) for non-commercial research purposes, even without permission from the rightsholder, but only if the data is lawfully accessed.

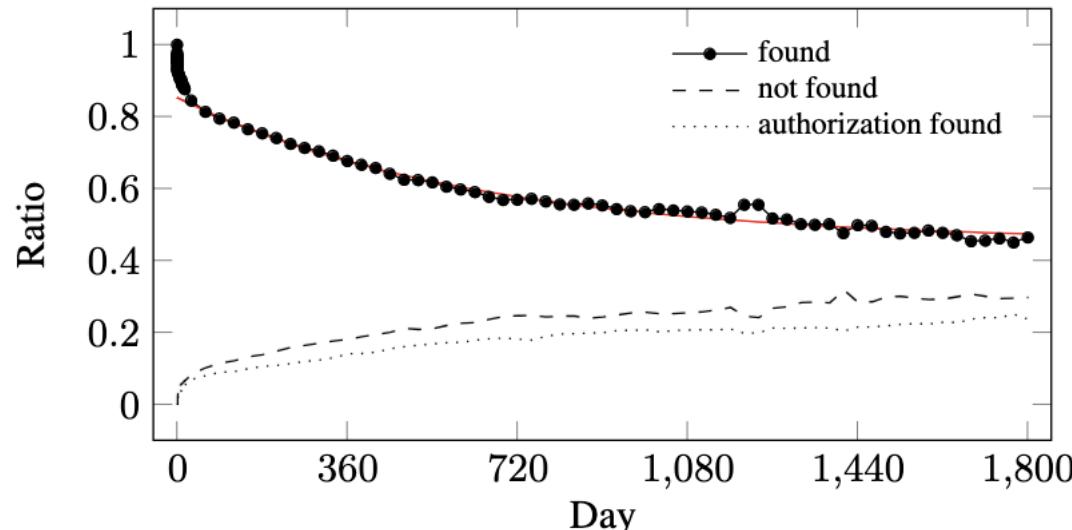
# Privacy regulations: GDPR

## Last week:

- GDPR enables data donation (rights of access and data portability)
- **Personal data:** Information relating to an identified or identifiable natural person.
- **Data controller:** The person or organization responsible for processing personal data.

If you collect personal data, you become a data controller!

- **Right to Rectification (Article 16):** Individuals can request that inaccurate or incomplete personal data be corrected.
- **Right to Erasure (Article 17):** Individuals can request the deletion of their data if it's no longer necessary, consent is withdrawn, or processing is unlawful.



# **GDPR Principles in Data Collection for Research**

## **Lawfulness, Fairness, and Transparency**

- Collect data with a legal basis (explicit informed consent, legitimate interest that outweighs any privacy risk to individuals)

## **Purpose limitation, data minimization and storage limitation**

- Data must be collected for specific, explicit and legitimate purposes
- Only collect data that is necessary
- Only keep personal data as long as necessary

## **Accuracy, security and accountability**

- Ensure data is accurate and up to date: provide data subjects the right to access, rectify, and delete data
- Protect personal data

# Anonymization and Pseudonymization

**Anonymization** means data is processed so individuals cannot be identified, even with additional information.

- *Example:* Removing names, birthdates, and any other identifiable information from a dataset, leaving only aggregated statistics.
- Anonymized data is not personal data and GDPR does not apply

**Pseudonymization** involves replacing identifying information (e.g., names) with fake identifiers, but the data can still be linked back to individuals with additional information.

- Pseudoanonymized data is considered personal data.
- You should still do it when full anonymization is not possible (it reduces risks)

It is your responsibility as data collector to handle the data appropriately

# Clearview AI gets another EU fine

Share

By Pieter Cranenbroek, Editor at LinkedIn News

Updated 6 days ago 

The Dutch Data Protection Authority has fined Clearview AI €30.5m for violating European privacy law. According to the Dutch authority, the facial recognition company **illegally maintains a database** of billions of photos, which were taken from the internet without the knowledge or consent of the people in question. If the US company does not change the way it operates, a penalty of up to €5m could be added on top of the fine. Clearview AI said the decision was "**unlawful**" as it does not have a place of business or customers in the EU. The data watchdog is also investigating whether the company's directors can be held **personally liable** for failing to take action despite previous fines from other authorities.

- Clearview AI has now been **fined a total of €90.5m in the EU** as authorities in France, Italy and Greece previously found the company in breach of data protection regulations.

# Ethics?

Not everything that is legal is ethical  
Not everything that is ethical is legal

## We'll talk in lecture 6!

Different definition of what is important (morality? consequences?)

Three guiding principles:

- **Respect for Persons:** Respect autonomy: receive informed consent if possible
- **Beneficence and respect for public interest:** Do not harm / maximize benefit/risk
- **Justice:** Ensure the risks and benefits of research are distributed fairly

# Exercise (in pairs)

We are interested in detecting how implicit racism/sexism/classism on LinkedIn leads to inequalities in job acquisition.

Below are the ToS.

- Could we create accounts to scrape photos and interactions between people?
- Could we scrape the information publicly available on Google?
- Could we use a plugin that downloads the data that we see?

What would we need to do to be compliant with GDPR?

Customer agrees that it will not:

- Except as expressly authorized by LinkedIn in writing, use any automated means or form of scraping or data extraction to access, modify, download, query or otherwise collect information from LinkedIn's websites; or

	API Data	Web Scraping	Data Donations	Tracking
User- vs. platform centrality				
Definition				
Time frame of collected data				
Types of data				
Robots.txt				
Terms of Service				
GDPR				
Consent of participants				
Potential for reactivity/ social desirability biases				
Level of gathered content				
Privacy risks in the collection of personally identifiable information				
Main advantages				
Main disadvantages				

# TODAY

## Lecture

Explain what APIs and web scraping are (in your own words).

Understand how HTML code is structured

Distinguish between robots.txt, Terms of Service and GDPR protections.

Understand the main advantages, challenges and legal considerations of APIs and Web Scraping.

## Lab

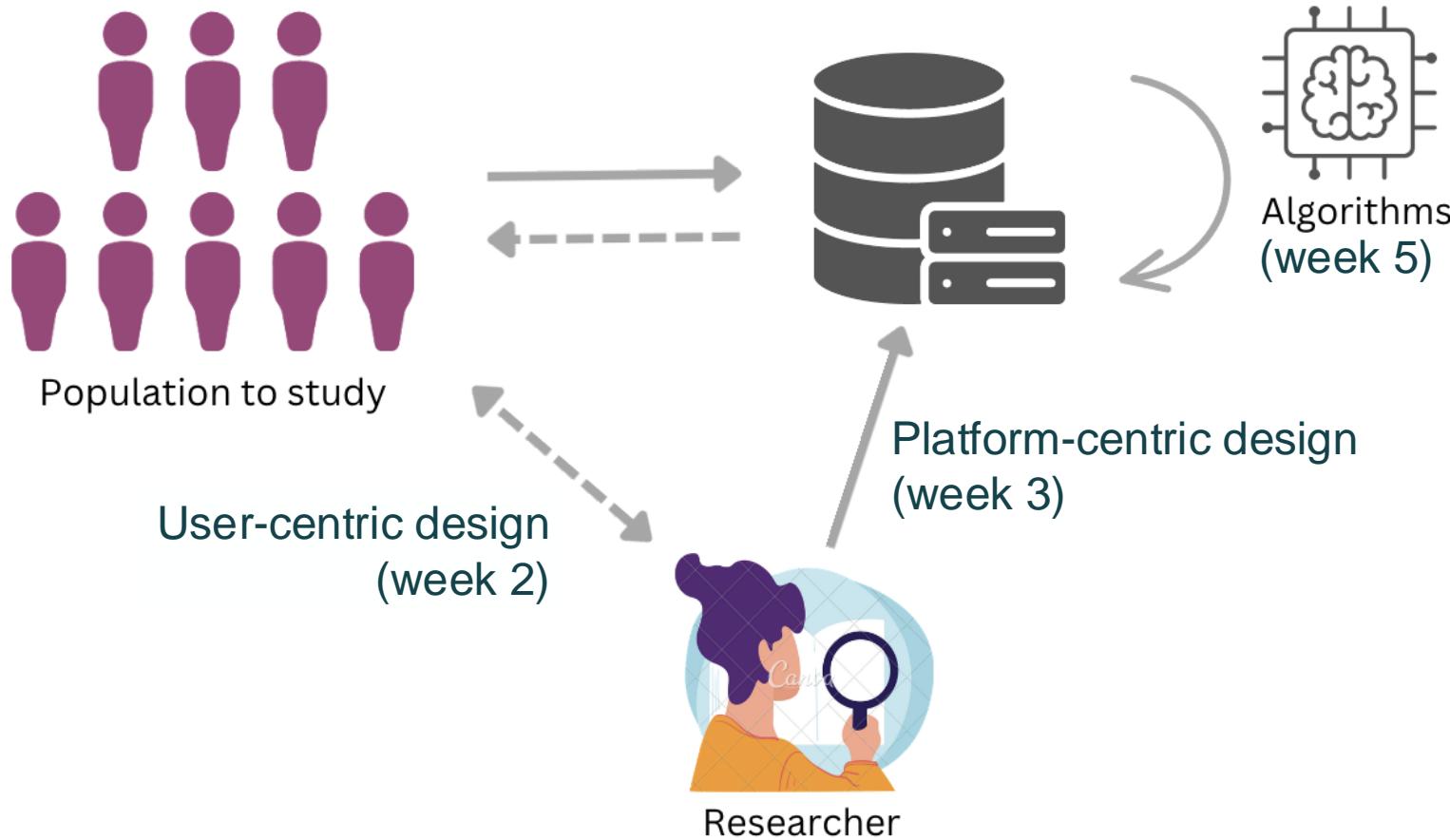
Use APIs to extract data:

- Wikimedia
- TheGuardian

Web scraping data from NU.nl

Saving data to your hard drive

# Summary of the course



**Week 4: Errors in DTD**  
Week 6: Ethics and Legislation  
Week 7: Designed big data  
Week 8: Beyond DTD and Q&A

# Announcements

Next Wednesday: second feedback moment

Finish collecting the data

Start discussing the errors (representation and measurement) → You will need to read the Bit By Bit book chapters 3.1—3.4

Please print/write the list of tasks