DHUM25A43 - 2
Investigating with AI

The Web

Feb 4th, 2025

What we saw last time

- Course
- Projects
- State of AI
- Tools: Colab
- ~~Demo: data analysis using Colab and Gemini~~

Today

- Guided practice: analysis IMDB in Colab
- Data sources
- APIs and the web
- Hands on: wikipedia API to build a dataset

- Project reviews (45mn)

At the end of this class

You

- understand what an **API** is
- know the 4 operations in the **REST** protocol
- can edit a **JSON** file
- can query a simple API (wikipedia,)
- understand best practice in terms of cap and throttling

Demo

Let's analyze IMDB

The dataset is available in the shared data folder in google drive (csv file, google spreadsheet)

Download it to your laptop

Movie Analysis

It contains information on 1000 movies

- title
- description
- ranking (imdb, Meta)
- duration
- genres
- actors, director
- revenue

imdb_top_1000

File Edit View Insert Format Data Tools Extensions Help

M30 | Lawrence A. Bonney

	B	C	D	E	F	G
1	Series_Title	Released_Year	Certificate	Runtime	Genre	IMDB_Rating
2	The Shawshank Redemption	1994	A	142 min	Drama	9.3
3	The Godfather	1972	A	175 min	Crime, Drama	9.2
4	The Dark Knight	2008	UA	152 min	Action, Crime, D	9
5	The Godfather: Part II	1974	A	202 min	Crime, Drama	9
6	12 Angry Men	1957	U	96 min	Crime, Drama	9
7	The Lord of the Rings: The Return of the King	2003	U	201 min	Action, Adventur	8.9
8	Pulp Fiction	1994	A	154 min	Crime, Drama	8.9
9	Schindler's List	1993	A	195 min	Biography, Dram	8.9
10	Inception	2010	UA	148 min	Action, Adventur	8.8
11	Fight Club	1999	A	139 min	Drama	8.8
12	The Lord of the Rings: The Fellowship of the Ring	2001	U	178 min	Action, Adventur	8.8
13	Forrest Gump	1994	UA	142 min	Drama, Romanc	8.8

Notebook and follow along

This is the final Colab notebook: demo_imdb_02
The prompts are available here

We'll start with a new colab notebook

First: upload the csv data file to the notebook

Demo

1. create a new notebook
2. upload the csv file to the notebook
3. ask Gemini to
   a. load the data
   b. analyze the data
   c. suggest & explain
   d. extract information from the movie description
   e. save the new data
4. share the notebook

Open source ?

Open source vs closed source

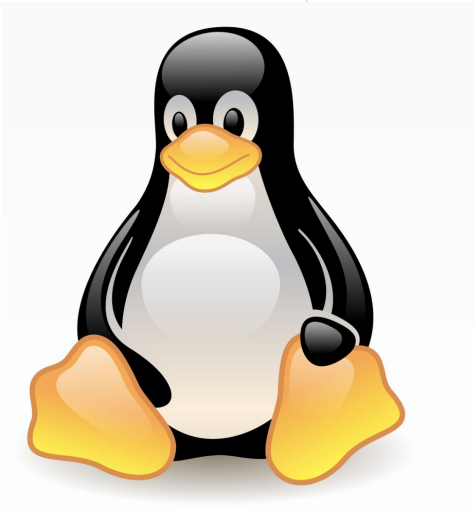
A distinction relevant for all software including AI models

Open source:

- The code is public.
- Linux, OpenOffice, Firefox, Chromium, Python, major databases,
- Can be copied and modified by anyone

Closed source:

- The code is proprietary.
- Windows, Word, Chrome, Edge, Oracle
- needs a license to use, black box



Open source is good for:

Since the code is public:

- **Innovation & Flexibility**: Users modify and enhance the software independently. Community-driven development, faster bug fixes, customization of code
- **Security & Transparency**: Security verify there's no virus.
  Issues are identified and patched quickly by the community.
- **Cost-Effective**: for users: free to use and modify, reduced licensing costs; no vendor lock-in. and <u>for creators</u>: community driven intelligence.

And

- **Knowledge Sharing**: Developers learn from existing code, accelerating skill development and innovation, shared knowledge and best practices.
- **Long-term Viability**: the community keeps on improving the software long after the initial devs have left the project

Open source LLMs

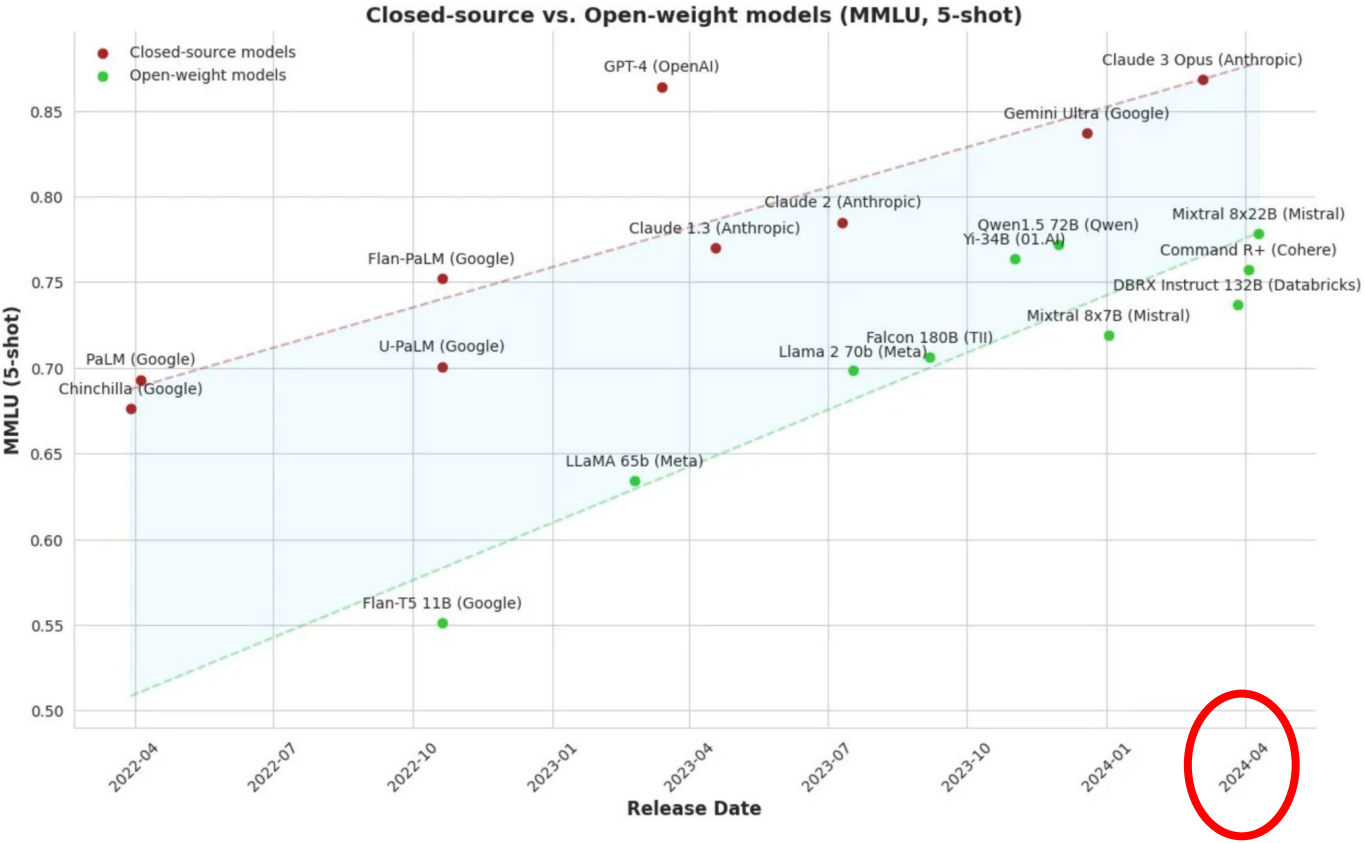
Different levels of openness:

- **model** : you can download the model and use it as is
- **code**: the code to create the model
- **training data**: the data used in training the model

Some models are fully open (DeepSeek), partially open (LLama, Mistral 7B), or closed (OpenAI o1, Claude Sonnet, Gemini)

Open source - Performance

deepseek-R1



Datasets & datasources

Datasets & datasources

News and Media

- **Examples:** Newspaper archives (NYT, Factiva, Europress etc.), Youtube (captions, comments, metadata), websites

Scholarly and Scientific Articles

- **Examples:** Academic journals, repositories like PubMed, arXiv, JSTOR.

Encyclopedic and Knowledge Bases

- **Examples:** Wikipedia / Wikidata, Encyclopedia Of Science Fiction

Entertainment and Cultural Datasets

- **Examples:** IMDb for film and television data, GoodReads

Government and Legislative Data

- **Examples:** Parliamentary records, government publications, election results

Consumer Reviews

- **Examples:** Amazon Reviews, CellarTracker

Kaggle datasets

Obtain and analyze an existing CSV dataset for the project.

Find Open Datasets and Machine Learning Projects | Kaggle

NYT Articles: 2.1M+ (2000-Present) Daily Updated
NYT Articles: 2.1M+ (2000-Present) Daily Updated
NYT Articles: Small Processed 500k Version

IMDB dataset
IMDb Dataset
IMDB Dataset of 50K Movie Reviews
IMDB Dataset of 50K Movie Reviews

arXiv Dataset
arXiv Dataset

…

Hugging Face

Hugging Face – The AI community building the future.

Datasets 295,389 | Filter by name | Full-text search | ↑↓ Sort: Trending

- open-thoughts/OpenThoughts-114k — Viewer • Updated 4 days ago • 114k • 11.4k • 189
- fka/awesome-chatgpt-prompts — Viewer • Updated 28 days ago • 203 • 8.91k • 7.27k
- cognitivecomputations/dolphin-r1 — Viewer • Updated 3 days ago • 814k • 436 • 133
- ServiceNow-AI/R1-Distill-SFT — Viewer • Updated 5 days ago • 1.85M • 1.16k • 115
- bespokelabs/Bespoke-Stratos-17k — Viewer • Updated 3 days ago • 16.7k • 21.3k • 176
- cais/hle — Viewer • Updated 10 days ago • 3k • 1.97k • 140
- OnDeviceMedNotes/synthetic-medical-conver… — Viewer • Updated 4 days ago • 143k • 139 • 27
- wikimedia/wikipedia — Viewer • Updated Jan 9, 2024 • 61.6M • 122k • 723
- open-r1/OpenThoughts-114k-math — Viewer • Updated 3 days ago • 89.1k • 380 • 24
- PrimeIntellect/NuminaMath-QwQ-CoT-5M — Viewer • Updated 11 days ago • 5.14M • 1.75k • 35
- promptfoo/CCP-sensitive-prompts — Viewer • Updated 5 days ago • 1.36k • 146 • 23
- speechbrain/LargeScaleASR — Viewer • Updated 6 days ago • 14.7M • 1.87k • 26
- Magpie-Align/Magpie-Reasoning-V2-250K-CoT… — Viewer • Updated 6 days ago • 250k • 863 • 22
- HumanLLMs/Human-Like-DPO-Dataset — Viewer • Updated 21 days ago • 10.9k • 2.73k • 184

Literature

GoodReads 100k books

GoodReads Best Books

GoodReads 100k books

Encyclopedia Of Science Fiction

Encyclopedia Of Science Fiction : Free Download, Borrow, and Streaming : Internet Archive

Awesome Sci-Fi

GitHub - sindresorhus/awesome-scifi: Sci-Fi worth consuming

Awesome-fantasy

GitHub - RichardLitt/awesome-fantasy: :european_castle: Fantasy literature worth reading

Scraping

Minet: GitHub - medialab/minet: A webmining CLI tool & library for python.

- Crawl (using a declarative language to define a browsing behavior, and what to harvest)
- Mine or search:
  - *Mediacloud* (requires free API access)
  - *Twitter* (requires free API access)
  - *Wikipedia*
  - *Youtube* (requires free API access)
- Scrape (without requiring special access, often just a user account):
  - *Facebook*
  - *Instagram*
  - *Telegram*
  - *TikTok*
  - *Twitter*
  - *Google Drive* (spreadsheets etc.)

Scrape NYT

https://colab.research.google.com/drive/14HDOUMmkoijYktlZ9RtUcA8uuW7Q0gPP?usp=sharing

OpenAI

Research Products Safety Company

February 2, 2025

Introducing deep research

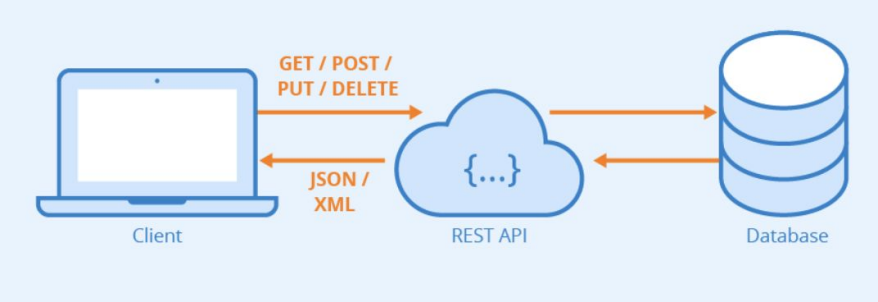
An agent that uses reasoning to synthesize large amounts of online information and complete multi-step research tasks for you. Available to Pro users today, Plus and Team next.

Try on ChatGPT ↗

API

Application Programming Interface

API



A generic definition

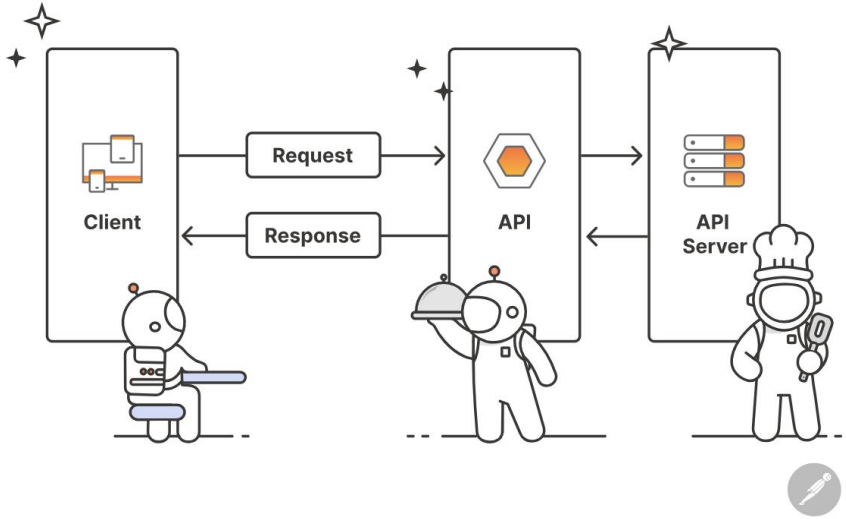
An API is a connection between computers or between computer programs.
It is a type of software interface, offering a service to other pieces of software.
(In contrast to a user interface, which connects a computer to a person)

A more practical definition

An API defines the methods and data formats
that applications can use
to request and exchange information.

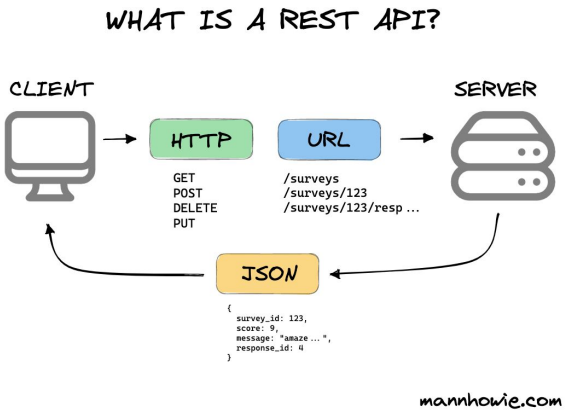
API

1. You send a request to a web address : a url
2. The server answers the request
3. You get back some data



Web = One BIG API + GET requests

1. on your browser you go to a url. This is the initial GET request
2. that request triggers a call to the server.
3. the server sends you back the html page as the response



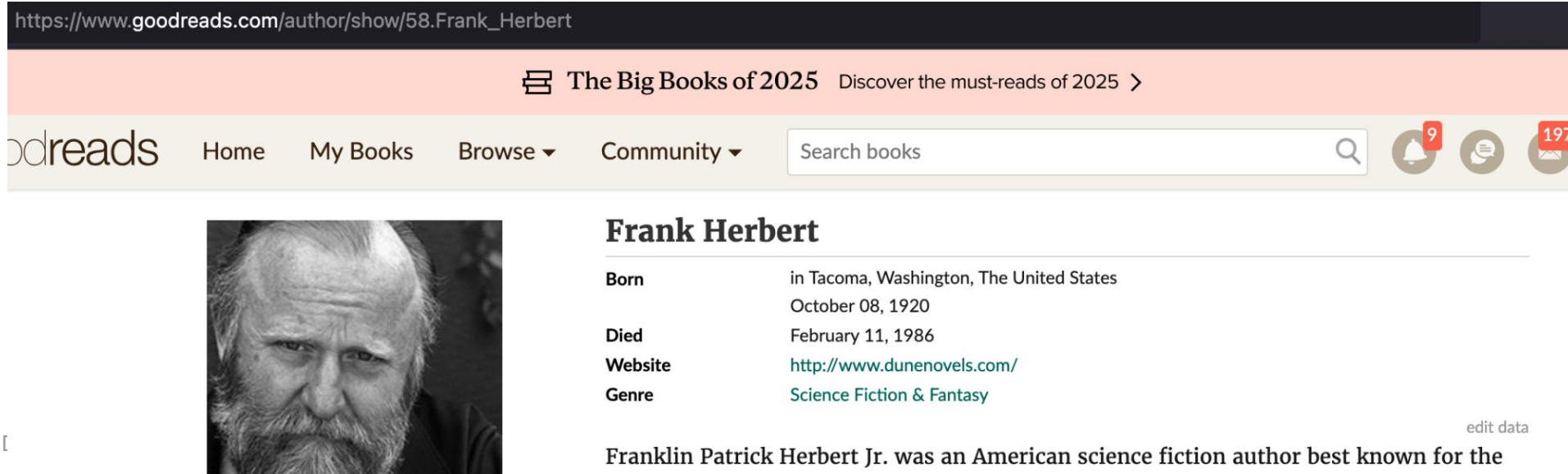
The Web is an API
Let's illustrate

The web is one gigantic API
It uses URLs to send requests to a server
The server sends the html page back

- Go on https://goodreads.com
- Search for **Dune**
- Click on the author's name Frank Herbert

You should end up on this URL:

https://www.goodreads.com/author/**show**/58.Frank_Herbert



What's a URL ?

A URL (Uniform Resource Locator)
is the address of a unique resource on the internet.

https://www.goodreads.com/author/**show**/58.Frank_Herbert

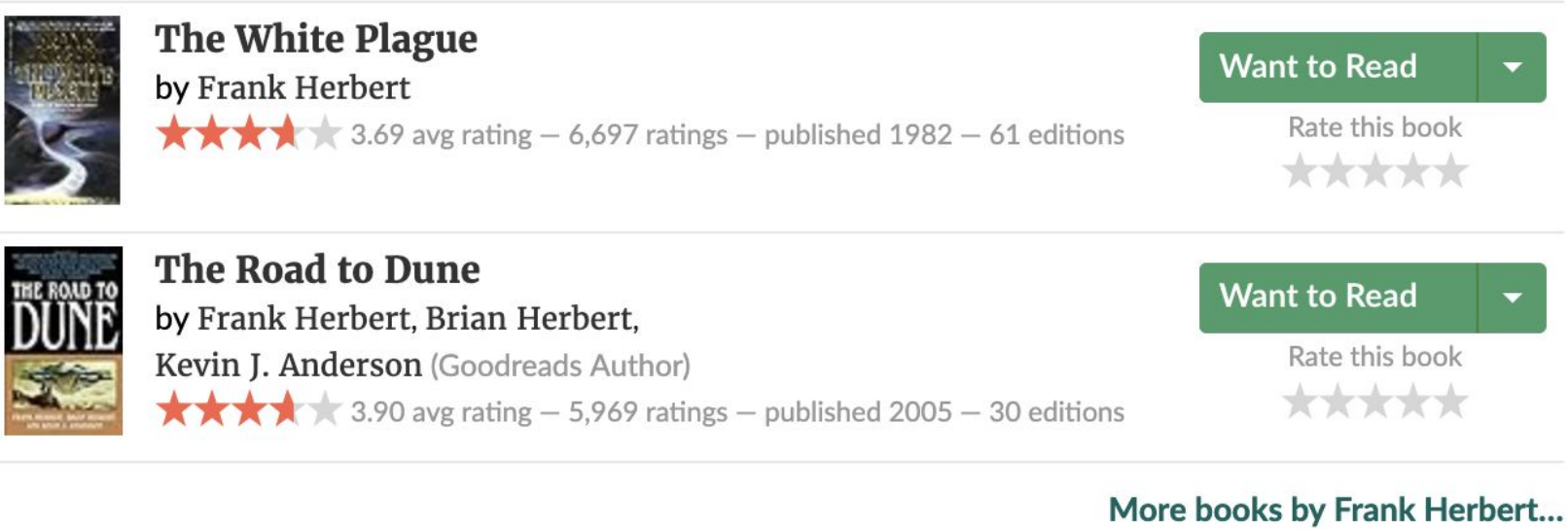
domain name

The data you request

- **show** an **author**,
- with label **58.frank.herbert**

/list instead of /show

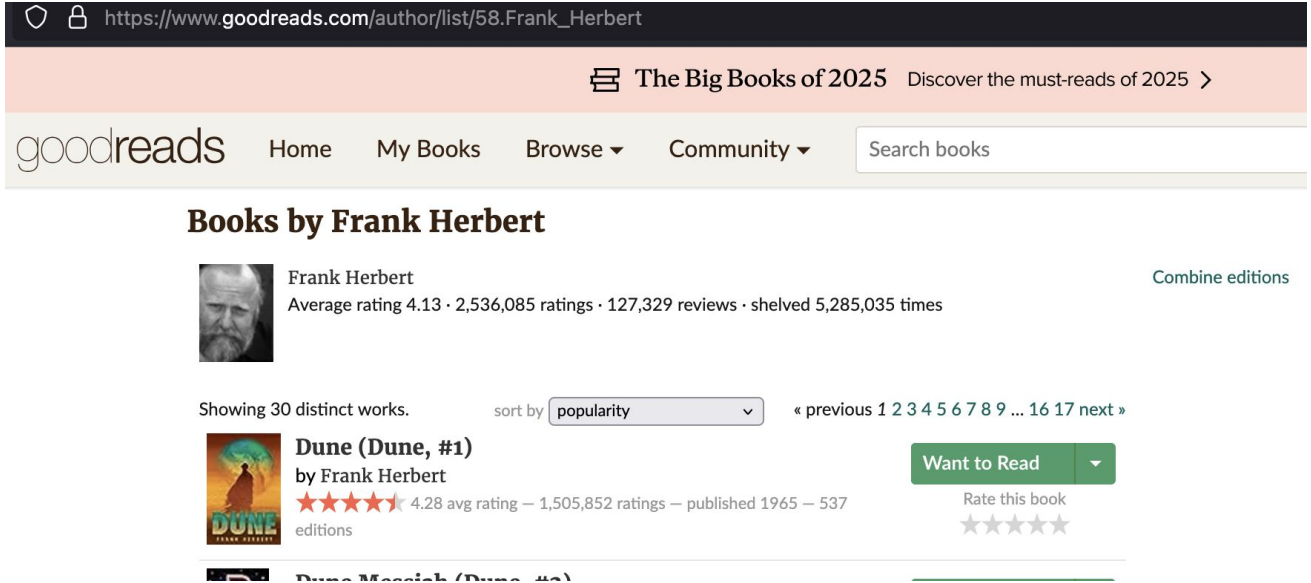
Now scroll down and click on "More Books by Frank Herbert"



/list instead of /show

The URL is now https://www.goodreads.com/author/**list**/58.Frank_Herbert

The verb "**/show**" is replaced with the verb "**/list**".



Parameters: ?page=2&per_page=30

Now click on page 2

The URL becomes
https://www.goodreads.com/author/list/58.Frank_Herbert?**page=2&per_page=30**

which reads

- list all the works of author 58.Frank_Herbert
- show page 2
- and show only 30 works per page

REST is the building block of the internet

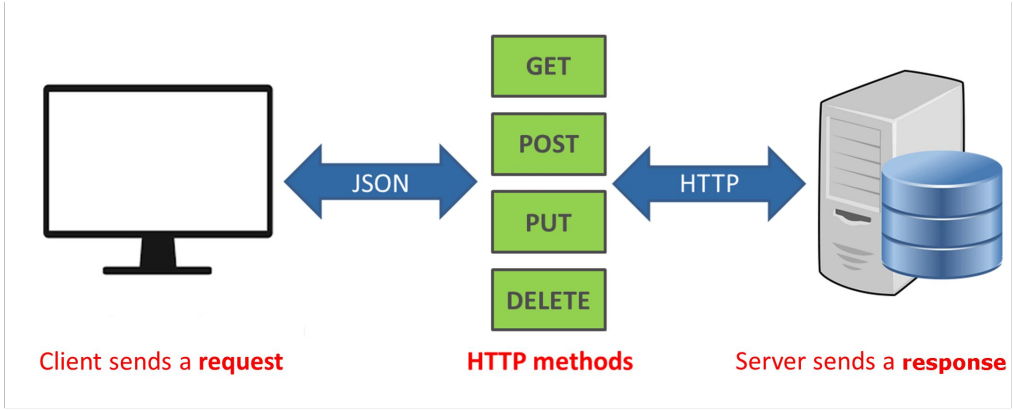
An endpoint: a URL + a verb + parameters

A method : here the method is to GET the page

The html page as the server response

By extending this concept of *method*, we define the 4 verbs of a REST API

- **read: GET**
- create: POST
- update: PUT
- delete: DELETE



Beyond html: JSON

There are multiple API standards and languages

And APIs can return all sorts of content: html, text, xml, pdfs,

APIs often return raw data formatted as JSON

JSON (JavaScript Object Notation) is a lightweight data format.

- easy for humans to read and write.
- easy for machines to parse and generate.

Here's an example

```
{
  "volume": "blaring",
  "current" : {
              "band": "rednex",
              "song": "cotton eye joe",
              "members":[
                      {"firstname":"Kent","lastname":"Olander"},
                      {"firstname":"Urban","lastname":"Landgren"},
                      {"firstname":"Jonas","lastname":"Lundstrom"},
                      {"firstname":"Tor","lastname":"Nilsson"}
              ]
             },
  "next" : {
          "band": "the dubliners",
          "song": "finnegan's wake",
          "members":[
                  {"firstname":"Ronnie","lastname":"Drew"},
                  {"firstname":"Luke","lastname":"Kelly"},
                  {"firstname":"Ciaran","lastname":"Bourke"},
                  {"firstname":"Barney","lastname":"McKenna"}
          ]
       }
}
```

How do you edit a JSON file ?

- do not use word of google docs. these are for real text
- use a code editor such as vscode (but it's overkill for just editing JSON)
- An online JSON formatter and editor is a good alternative
  - https://jsoneditoronline.org/
  - and many others

Wikipedia API

The wikipedia API

Check out :

- The API documentation https://www.mediawiki.org/wiki/API:Main_page
- The wikipedia python library: https://wikipedia.readthedocs.io/en/latest/code.html
- AI studio: https://aistudio.google.com or chatGPT as an alternative to generate code
- The sandbox : https://en.wikipedia.org/wiki/Special:ApiSandbox

Your turn

Constitute a dataset of content from wikipedia pages related to a given topic.

For instance :

- Extract all information related to immigration from European capitals pages
- Given a list of political parties en France, find their position on the environment

Think of

- page attributes
- generate list of pages beforehand
- tell the LLM to use the wikipedia library
- how you output the data

Steps

The method is important. It's a gradual iterative process. build on a series of small actions

- Using the wikipedia api library, find the page for Paris
- In the page of Paris, extract the section titles
- In the section titled Demographics or equivalent, extract the information on immigration

Prompts Precisions

The LLM has a tendency to recreate the whole code from scratch (installing etc …). To avoid that, state that the libraries are already installed, the page is already available etc

The code is often too complex. I specify : `simple code, not for production, no error handling, python beginner level`

Ask for potential methods first (and no code), then specify the one you want to generate the code.

Projects

Seed projects

Climate change, energy

Turan Kerem: “ two potential research questions focusing on climate change and diplomatic negotiations at COP meetings”

1 debate over legal and financial responsibility…

2 financial solution mechanisms for climate change…

Andrei: Innovations in Geoengineering (Stratospheric Aerosol Injection, Marine Cloud Brightening)

Controversies Surrounding Geoengineering, Technological Uncertainties, Risks…

Data: Press, scientific publications, news…

Interesting datasets:

10 Years of Climate Science Denial on RCGroups

Public Opinion on Climate Change (Updated Daily)

Earth Negotiation Bulletins for the COPs 1995-2016 - Harvard Dataverse

AI - robotics

Andrei: AI and the Future of Work

The Fear: "We'll Lose Our Jobs", Sectors at Risk, The Skills Gap, Emergence of AI-Related Roles

Data: Press, scientific publications, news…

Andrei: Artificial General Intelligence (AGI) and the technological singularity

Ability to predict or control it, Potential Benefits and Risks

Brain–computer interface

Andrei: Risks and Controversies, Neuroenhancement, Brain implants, Transhumanism, neurocybernetic augmentation

Data: press, web, scientific publications, sci-fi movies and literature

Other themes:

Carla: supermarket’s data to try to anticipate customers’ behavior

Projects

Let's review your projects

For next time

Build a web search dataset

- ask an LLM for web search APIs
- for each (google and bing search for instance)
  - how to get an API key
  - boilerplate code example
- or use the Duckduckgo_search python library
  - boilerplate code example

Watch out

The code generated by Gemini had trouble with capitalization
the right code for duckduckgo should be

```
from duckduckgo_search import DDGS

results = DDGS().text("Paris", max_results=5)
print(results)
```