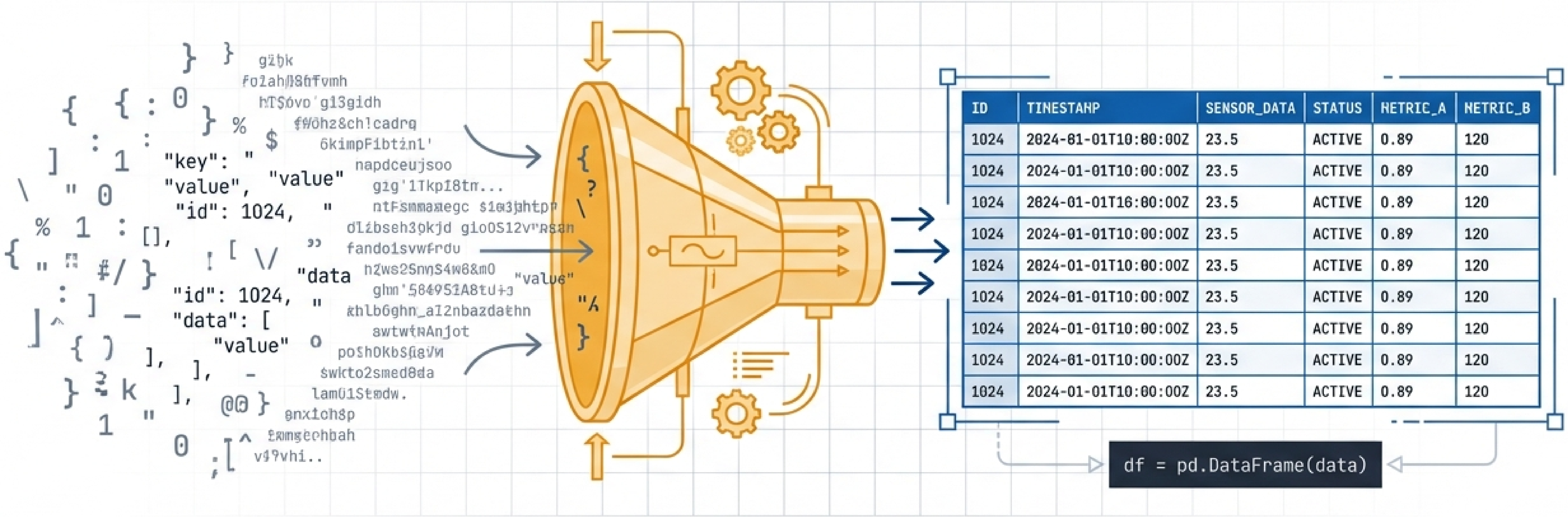
Harvesting the Web: Building Custom Datasets via APIs

From raw JSON responses to structured Pandas DataFrames.



A methodology for active data creation.

The Limits of Static Data

Stagnant Data (CSV/SQL)

- Pre-packaged and limited.
- Often outdated by the time of download.
- Passive consumption.

Dynamic Data (APIs)

- Real-time and vast.
- Direct access to the source.
- Active creation.

Insight: Real-world applications depend on live communication between software, not just pre-downloaded files.

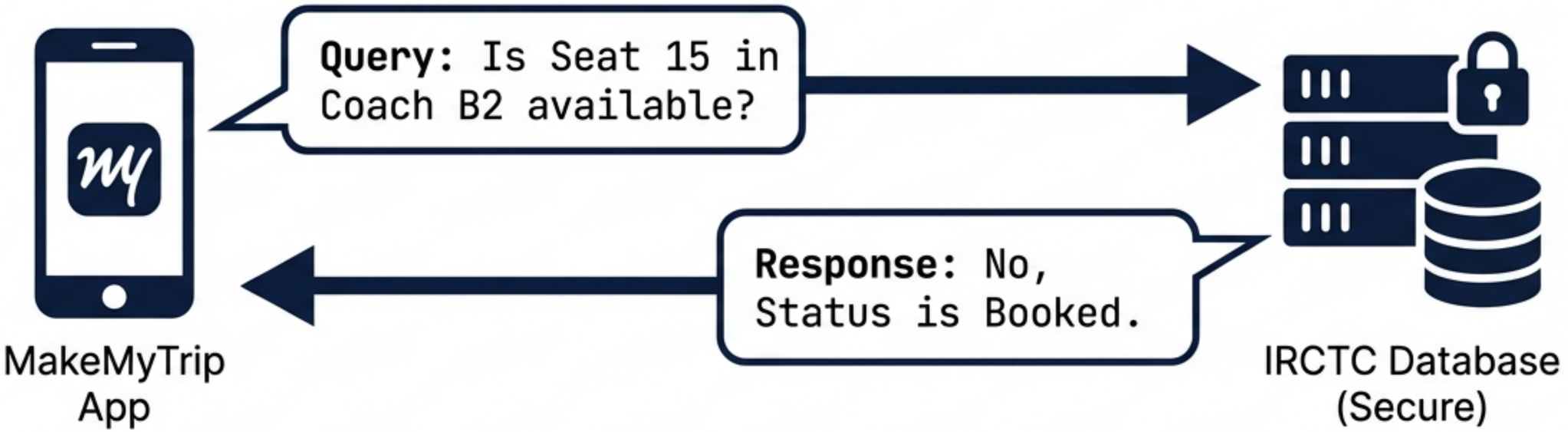
The Messenger: What is an API?

`API = Application Programming Interface`

An **API** acts as a **data pipeline**, allowing two isolated software systems to communicate. It takes a request from Point A and delivers a response from Point B.

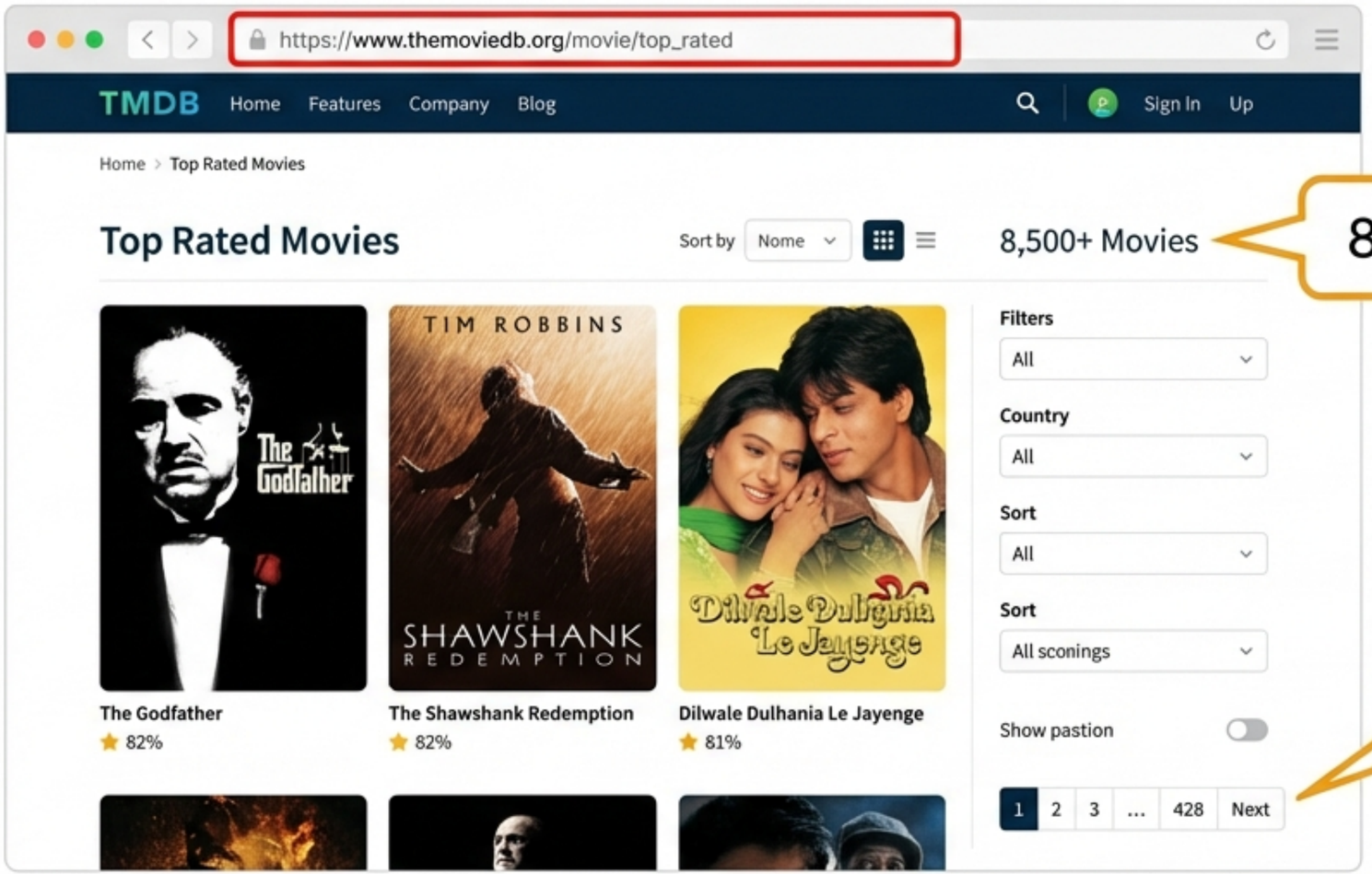
The 'Handshake' in Action: IRCTC & MakeMyTrip

MakeMyTrip does not own the railway database. They must ask IRCTC for permission to view availability.



APIs allow proprietary databases to share specific answers without exposing their entire internal architecture.

The Target: Top Rated Movies on TMDB



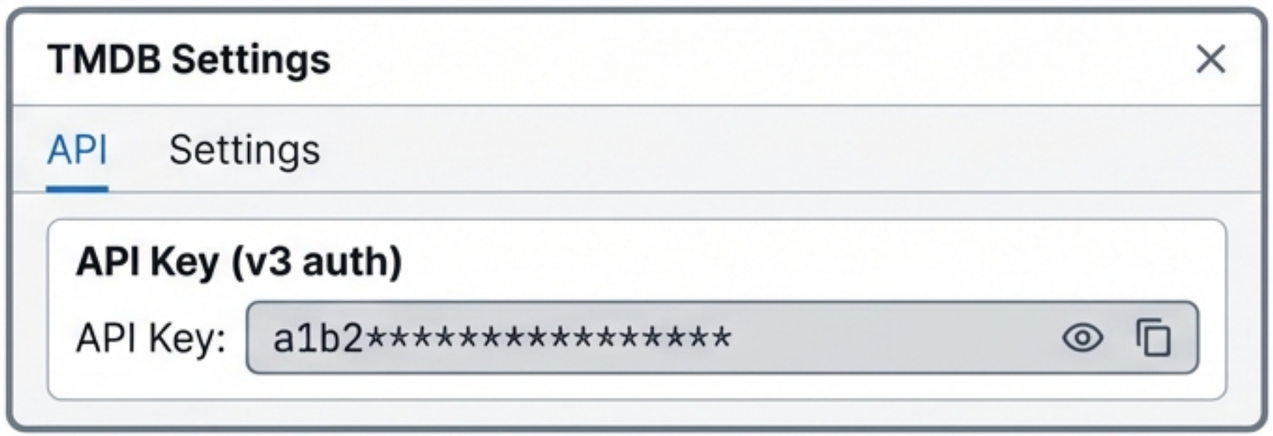
8,500+ Movies

428 Pages Available

Mission: Automate the extraction of data for every single movie in this list.

The Keys to the Kingdom

1. Login to TMDB
2. Navigate to Settings
3. Select API Section
4. Generate API Key



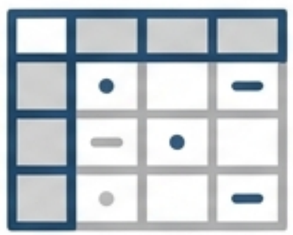
Security Warning:
Treat your API Key like a password. If shared publicy, others can consume your request quota.

The Architect's Toolkit

Requests

The Python library used to “make the phone call” to the server and fetch raw data.

Pandas



The powerhouse library for structuring data into rows and columns for analysis.

JSON Viewer

A browser tool to parse and visualize nested JSON responses into a readable tree.

Establishing Contact: The First Request

```python
import requests

# The Endpoint URL with API Key appended
url = 'https://api.themoviedb.org/3/movie/top_rated?api_key=YOUR_KEY'

# Making the GET request
response = requests.get(url)
```

200: Success
(The server answered)

404: Not Found
(Wrong URL)

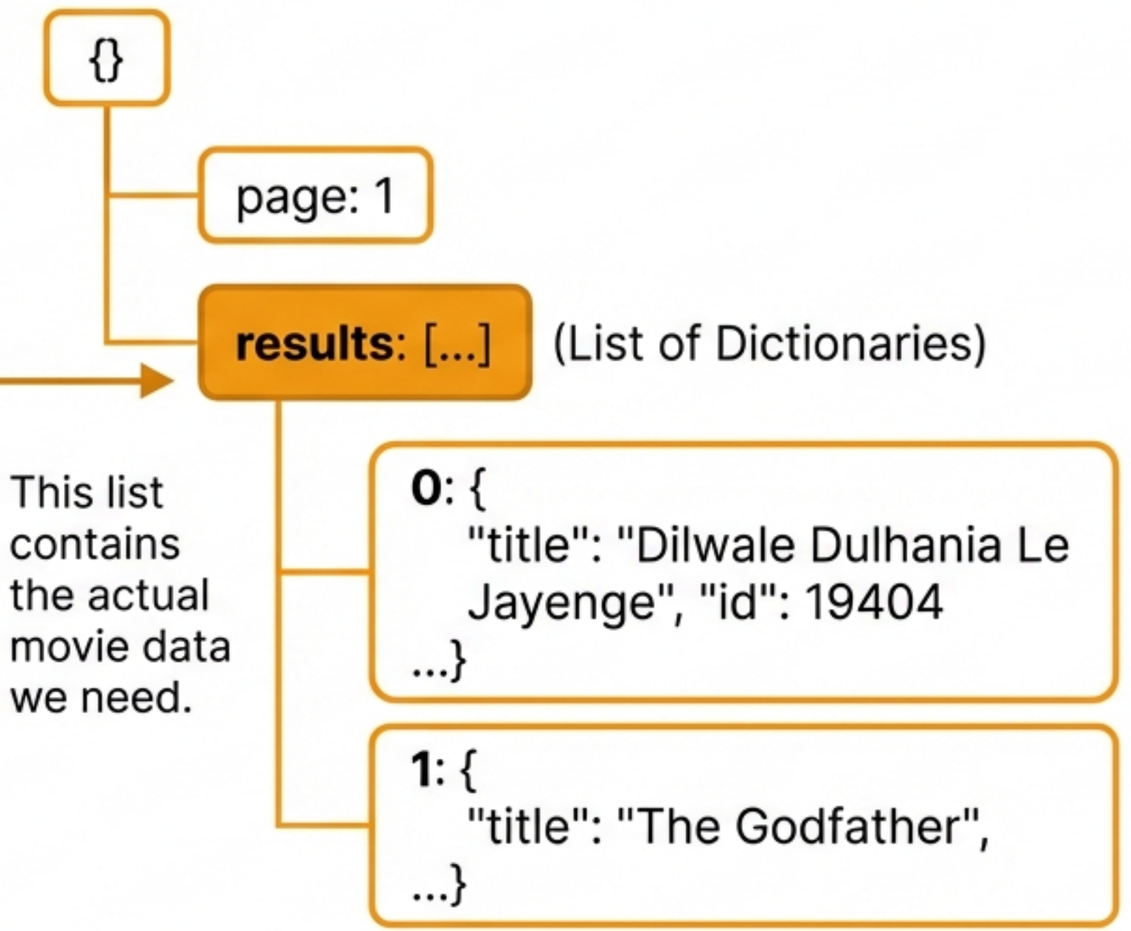
500: Server Error
(Internal issue)

Decoding the Matrix: The JSON Structure

Raw Data


```
{"page":1,"results":[
  [{"adult":false,"backdrop_path":"/tmU7…
    ","id":19404,"original_language":"hi",
    "title":"Dilwale Dulhania Le Jayenge",
    "id": 19404,
    "coude": 15,
    "language": "hi,
    "adult": 10,"backdrop_path":"/tmEP7…",
    "title": "Diwale Dulana Le Jayenge",
    "anina": 11,
    "id": 19404,
    "footbal",
    "title": "hi",
    "testing"
  }]...
}
```

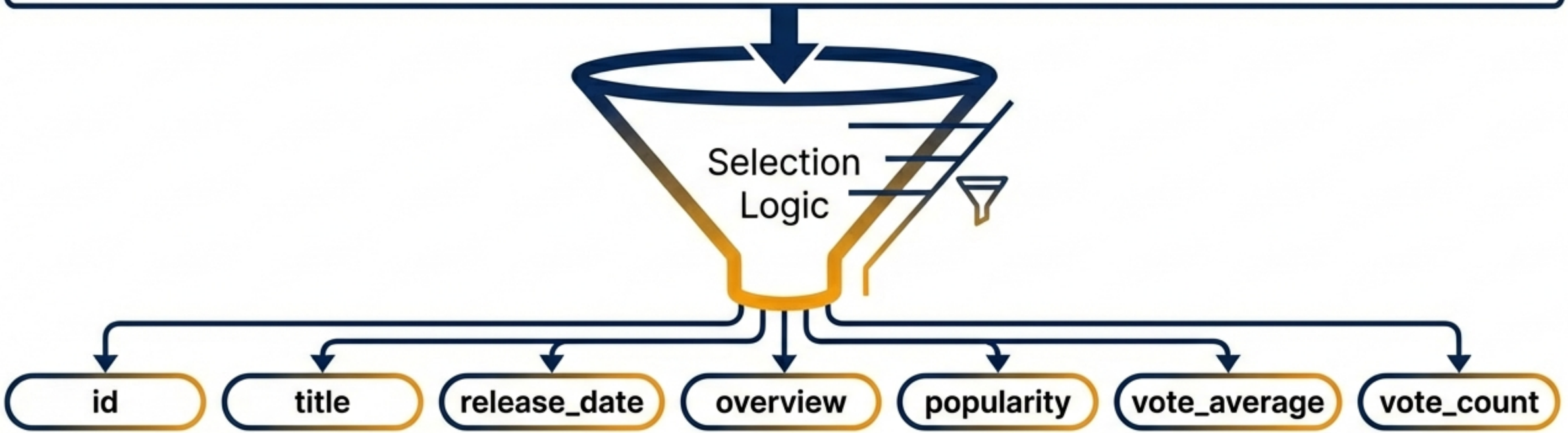

Parsed Tree



The Extraction Strategy

Raw JSON Object


```
{"adult": false,"backdrop_path":"/tmU7.....","id":19404,"original_language":"hi","genre_ids": [[23, 35, 42],
"id": 19404,"coude": 15,"tegion": 5,"ercoude": 11,"language": "hi","lavage": "atlingshmts","adult": false": 10,
"backdrop_path":" AGeha","backdrop_path":"/tmEP7…","genre_cods": 11,"title": "Diwale Dulana Le par","id": 19404,
"vote": "popularity","ote_average": ."hi","testing" } ...
```

We filter the noise and keep only the essential attributes for our dataset.

The First DataFrame (Page 1)

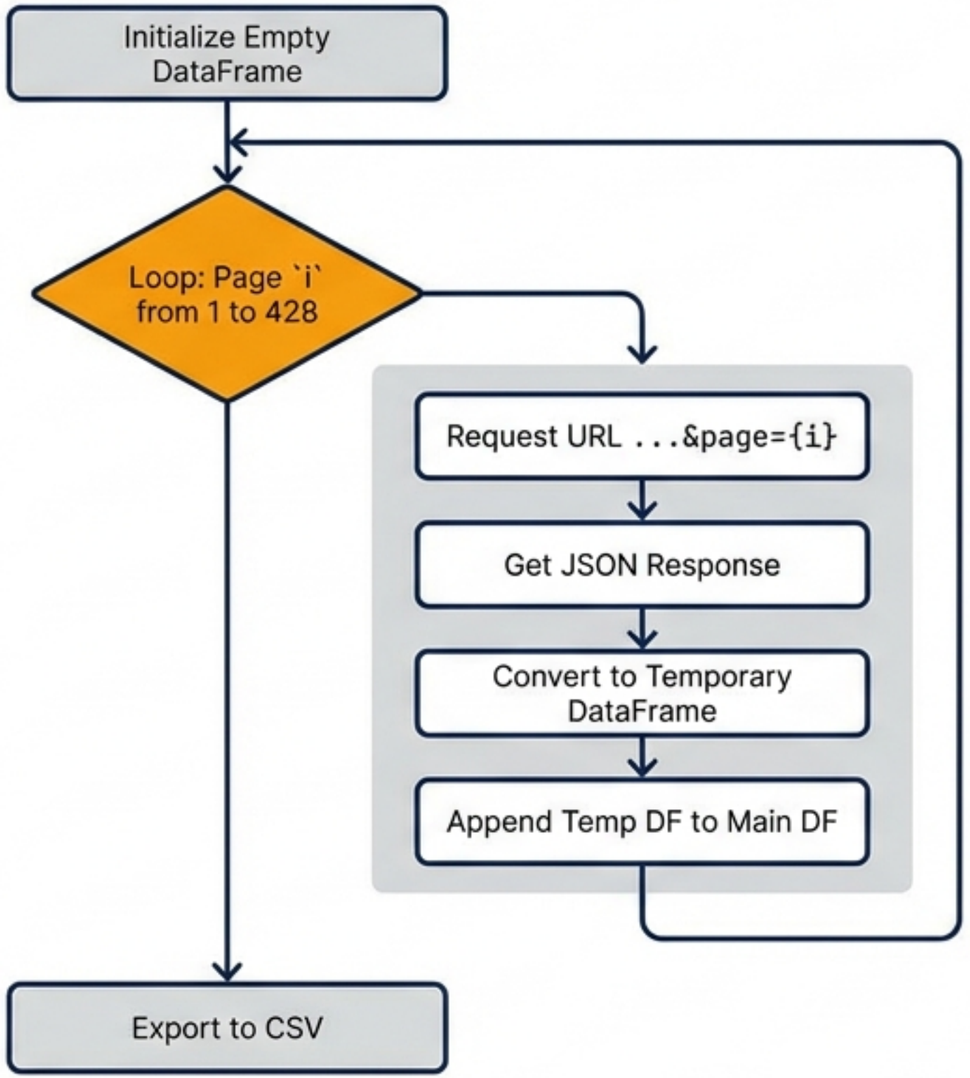
```
# Convert the 'results' list into a DataFrame
df = pd.DataFrame(response.json()['results'])
```

id	title	release_date	vote_average	...
19404	Dilwale Dulhania Le Jayenge	1995-10-20	8.7	
238	The Godfather	1972-03-14	8.7	
				...
				...
...	...	...	...	...

... (Rows fade out at Row 19)

Status: 20 Movies Collected. 8,500+ Remaining.

The Master Plan: Automation Loop



Implementing the Loop

```
# Loop through all 428 pages
for i in range(1, 429):
    # Dynamic URL with f-string for page number
    response = requests.get(f'https://api...&page={i}')

    # Create temporary dataframe for current page
    temp_df = pd.DataFrame(response.json()['results'])

    # Append to main dataframe
    df = df.append(temp_df, ignore_index=True)
```

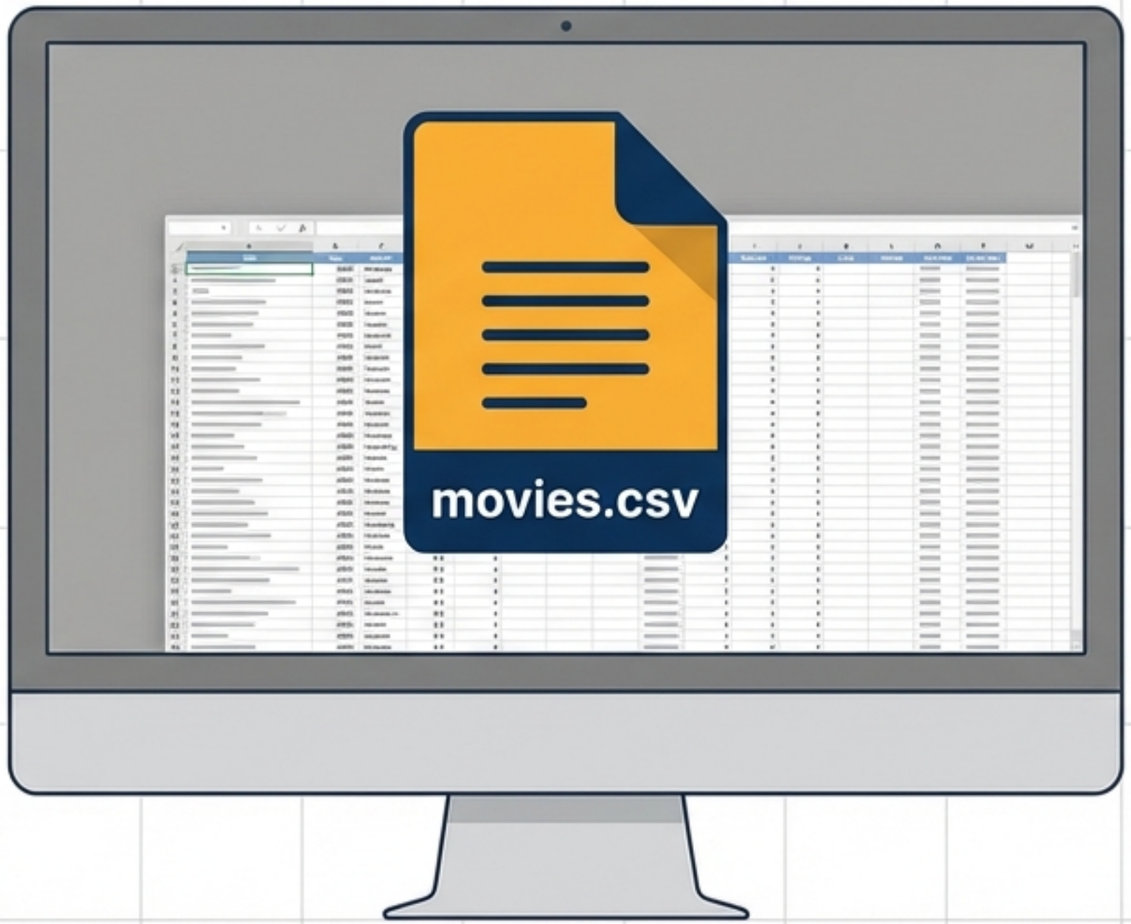
Critical: Prevents index duplication. Ensures a continuous index from 0 to 8550 instead of repeating 0-19 hundreds of times.

The Final Product

Final Shape: (8551, 7)

Execution Time: ~2 Minutes

```
# Export final DataFrame
df.to_csv('movies.csv')
```



From Coder to Contributor

Upload to Kaggle. Build a portfolio, gain medals, and demonstrate expertise to recruiters.

Explore RapidAPI. Find data for Football, Finance, Weather, or Crypto.

The internet is your database. Build your own library.