

ALL API KEYS NEEDED FOR FRAMECORE BFD

PRIVATE AND CONFIDENTIAL – Dec 9th 25 Edition

Use the GPU

A Nvidia RTX 3080 can accelerate the parsing of large Parquet files (billions of entities) through GPU-accelerated libraries like the [NVIDIA RAPIDS cuDF](#) library and the [RAPIDS Accelerator for Apache Spark](#), but it is not handled by the GPU natively.

How GPU Acceleration Works for Parquet Parsing

Traditional data parsing is a CPU-intensive task. However, the NVIDIA RAPIDS suite of software provides tools to offload this work to CUDA-enabled GPUs, such as the RTX 3080.

cuDF Library: cuDF is a Python DataFrame library for loading, joining, aggregating, and otherwise manipulating data using CUDA. It provides a pandas-like API and can significantly speed up data manipulation and analysis tasks by keeping data on the GPU, avoiding costly CPU-GPU memory transfers. You can use it to read Parquet files directly into a GPU DataFrame.

cuDF is designed for NVIDIA GPUs using the CUDA framework. While it can be run on Windows via the [Windows Subsystem for Linux 2 \(WSL2\)](#), alternatives exist if you are looking for non-GPU or cross-platform solutions.

Running cuDF on Windows

The officially supported way to use cuDF on a Windows machine is within a Linux environment using WSL2. This approach allows you to leverage your NVIDIA GPU from within Windows without requiring a dual-boot setup.

Apache Spark Integration: For large-scale, distributed processing, the RAPIDS Accelerator for Apache Spark leverages the GPU for I/O and processing, including efficient Parquet scans. This is critical for optimal performance when dealing with terabytes of data.

Apache Arrow: RAPIDS uses the Apache Arrow memory format, which allows for efficient data interchange between different systems and languages with minimal serialization overhead, further boosting performance.

Optimization Techniques: Libraries like fsspec.parquet optimize access to remote Parquet files using byte-caching and parallel data transfer, which can be combined with GPU

processing for better performance. Parquet's columnar storage and various compression algorithms (like Snappy) also aid in faster data retrieval by allowing specific columns and data pages to be skipped during a query.

*****All RAPIDAPI*****

ABSOLUTELY NEEDED FOR EVERY RAPID API KEY

default-application_11126086

2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f

SECTION 1

ALL ABSTRACT APIS

SECTION 2

TV AND MOVIE METADATA

SECTION 1

ALL ABSTRACT APIS

Government News & Calendar

You are receiving this email to confirm the creation of an API key. If you did not request this, please disregard this email. Your API key for roy@framecore.ai is:

M3tn4hRUXS3jcjhLsgbCVNUWMsCBvoUjzJv2mQ7i

You can start using this key to make web service requests by referring to the relevant agency's API documentation. This API key is for your use and should not be shared.

For additional support, please [contact us](#). When contacting us, please tell us what API you're accessing and provide the following account details so we can quickly find you:

Account Email: roy@framecore.ai

Account ID: 951266f7-f199-417d-b032-d3ce9e6930c5

api.data.gov acts as a layer above your existing APIs. It transparently adds extra functionality to your APIs and helps deal with some of the repetitive parts of managing APIs.

Open Source

api.data.gov is powered by the open source project [API Umbrella](#). You can contribute to the development of this platform, or setup your own instance and run all this same stack yourself.

SOCIAL SENTIMENT: DataForSEO Search Volume Key

App

default-application_11126086

X-RapidAPI-Key

2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f

Request URL

rapidapi.com

SOCIAL – ZEITGEIST

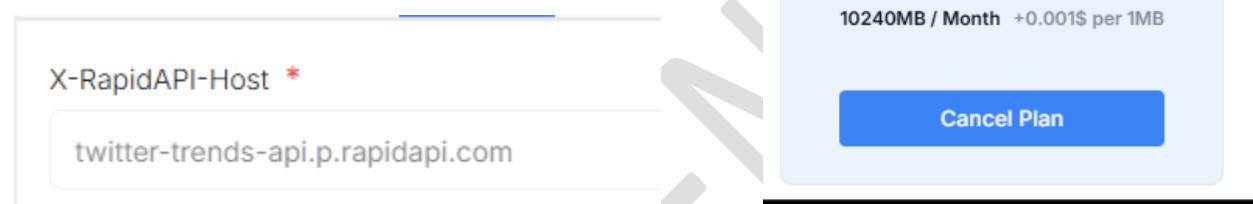
****Twitter Trends****

Query Params

woeid*

String

A valid WOEID Complete list can be found using /places endpoint



App

default-application_11126086

X-RapidAPI-Key

2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f

Request URL

rapidapi.com

node,JS

FOLLOWS

```
import http.client
conn = http.client.HTTPSConnection("twitter-trends-api.p.rapidapi.com")
headers = {
    'x-rapidapi-key':
```

```
2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f",
'x-rapidapi-host': "twitter-trends-api.p.rapidapi.com"
}
conn.request("GET", "/places", headers=headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

CalendarIndex Holiday Data

We provide a comprehensive API that returns holidays from over 200 countries. Tap into our calendar database and use it to display the dates of public holidays and observances in your application. The database covers over 200+ countries, states, and regions worldwide, and it is constantly being updated and extended.

<https://www.calendarindex.com/api-documentation> <https://www.calendarindex.com/supported-countries>

App

default-application_11126086

X-RapidAPI-Key

2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f

Request URL

rapidapi.com

TRENDING POPULARITY FOR TITLES, ACTORS, SENTIMENT, ETC ****Instagram Statistics API****

Instagram, Twitter, TikTok, YouTube, Facebook, Telegram. Followers, Audience demographics, Fake Followers, Mentions, Quality Score, hashtags, IGTV, ads, Audience quality, Engagement Rate. Business and Influencers audience demographics, interest categories. Search accounts by country, demographics, category and more metrics. [Detailed information about API and methods](#)

One integration with all social networks! Universal API social media.

Obtaining the distribution of subscribers by age groups. Analysis of the most active age categories. The ratio of men to women among subscribers. Determining the most active gender among the audience.

The image shows two plans for the Instagram Statistics API. The left plan is 'Pro' at \$75.00/mo with 10,000 requests/month. The right plan is 'Ultra' at \$189.00/mo with 50,000 requests/month. Both plans include features like General information, Content statistics, Demography, Content feed, Catalog, and All networks. They also have a bandwidth platform fee of 10240MB/month plus \$0.001 per 1MB. A 'Cancel Plan' button is at the bottom of the Pro plan, and an 'Upgrade Plan' button is at the bottom of the Ultra plan.

Plan	Price	Requests /mo	Features	Rate Limit	Bandwidth Platform Fee
Pro	\$75.00	10,000 / Month	General information, Content statistics, Demography, Content feed, Catalog, All networks	-	10240MB / Month +\$0.001\$ per 1MB
Ultra	\$189.00	50,000 / Month	General information, Content statistics, Demography, Content feed, Catalog, All networks	-	10240MB / Month +\$0.001\$ per 1MB

GET Profile by URL	(i)
GET Profile by ID	(i)
GET Feed	(i)
GET Retrospective	(i)
GET Activity	(i)
GET Search	(i)
GET Tags	(i)
GET Post by URL	(i)

Distribution of subscribers by countries and cities. Analysis of the geographical concentration of the audience. Determining the main languages spoken by subscribers. Analysis of the audience's language preferences. Determining the peak activity time of subscribers (e.g., by hours or days of the week). Analysis of interactions (likes, comments) based on demographic characteristics. Get historical data. Get Instagram audience demographics. Influencers API. Instagram followers history.

Query Params

q *(optional)*

String

page *

Number

Default: 1

perPage *

Number

Default: 10

sort *(optional)*

String

-score - by Relevance -usersCount - by Followers -avgViews - by Views -avgER - by Engagement Rate -qualityScore

- by Quality Score

tags *(optional)*

String

from Tags

locations *(optional)*

String

Account location (country or city) from Tags

genders *(optional)*

String

Account gender (for influencer) m - Male f - Female

ages *(optional)*

String

Account age (for influencer) 0_18 18_21 21_24 24_27 27_30 30_35 35_45 45_100

minAge *(optional)*

String

maxAge *(optional)*

String

socialTypes *(optional)*

many others available

App **default-application_11126086**

X-RapidAPI-Key

2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f

Request URL

rapidapi.com

Node.js

```
const http = require('https');

const options = {
    method: 'GET',
    hostname: 'instagram-statistics-api.p.rapidapi.com',
    port: null,
    path: '/search?page=1&perPage=10&sort=-score&locations=united-
states&socialTypes=INST%2CFB&trackTotal=true',
    headers: {
        'x-rapidapi-key': '2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f',
        'x-rapidapi-host': 'instagram-statistics-api.p.rapidapi.com'
    }
};

const req = http.request(options, function (res) {
    const chunks = [];

    res.on('data', function (chunk) {
        chunks.push(chunk);
    });

    res.on('end', function () {
```

```
        const body = Buffer.concat(chunks);
        console.log(body.toString());
    });

});

req.end();

import http.client

conn = http.client.HTTPSConnection("instagram-statistics-api.p.rapidapi.com")

headers = {

    'x-rapidapi-key': "2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f",

    'x-rapidapi-host': "instagram-statistics-api.p.rapidapi.com"

}

conn.request("GET", "/statistics/activity?cid=INST%3A17841400005463628",
headers=headers)

res = conn.getresponse()

data = res.read()

print(data.decode("utf-8"))
```

REAL TIME WEATHER – THE WORSE IT IS THE MORE WE WATCH TV **meteostat**

The Meteostat JSON API is a web service which provides weather observations, historical statistics and long-term climate data for thousands of weather stations and places worldwide.

Hourly station data, many others available

App

default-application_11126086

X-RapidAPI-Key

2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f

Request URL

rapidapi.com

CORRELATE PAST GOOD AND BAD WEATHER FOR IMPACT ON TV VIEWING **Global Weather History**

Get accurate historical weather from over 104,000 locations around the world. Daily weather measurements from 1763! Weather information includes: minimum & maximum daily temperature, snow fall, precipitation, evaporation and more!

App

default-application_11126086

X-RapidAPI-Key

2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f

Request URL

rapidapi.com

CORRELATE ALGO ACCURACY FOR FORECASTING AGAINST AN INDEPENDENT SOURCE

****Ai Movie Recommender****

The AI Movie Recommender is a web application that uses AI to help you discover your next favorite movie. The application is built with Next.js, React, and Tailwind CSS. It uses the TMDB API to fetch movie data and the OpenAI API to generate movie recommendations based on your preferences.

To get started with the application, simply type in the search bar and press enter. The application will then display a list of movies that match your search query. You can click on a movie to view its details, including its title, release year, genres, and a short summary.

Others available

App

default-application_11126086

X-RapidAPI-Key

2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f

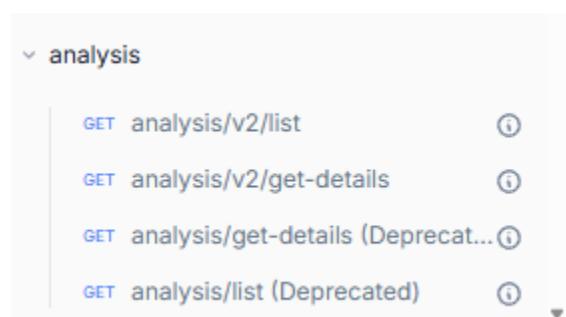
Request URL

rapidapi.com

FINANCIAL RESULTS (STREAMER SUBSCRIBERS, GEOGRAPHICAL BUSINESS SPLITS, IMPACT OF POLITICAL CHANGE AND THE ECONOMY

NEXT MOST IMPORTANT

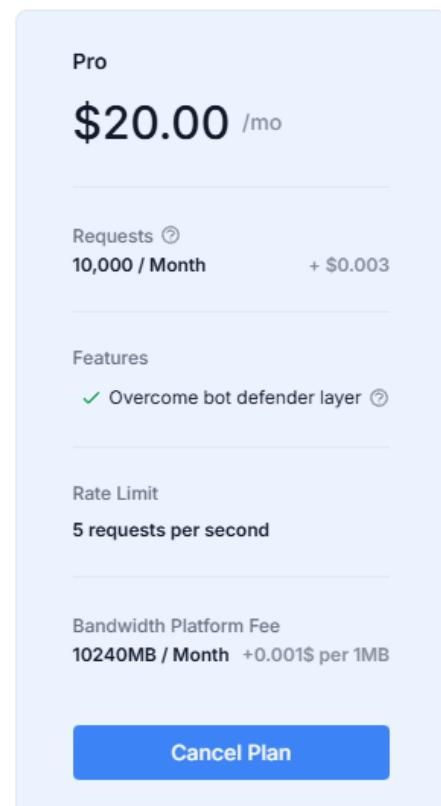
Seeking Alpha



The screenshot shows a section of the Seeking Alpha API documentation under the 'analysis' endpoint. It lists four GET requests:

- GET /analysis/v2/list
- GET /analysis/v2/get-details
- GET /analysis/get-details (Deprecated)
- GET /analysis/list (Deprecated)

Important additional source of news, especially company reports, filings, etc. is Seeking Alpha, vital and important abstract data, the news of events , for example netflix quaterly earnings which gives us subscriptions, regional performance we can use to disambiguate their global views etc. is vital, there is also a lively and up to the second news and discussion groups, etc., being able to deeply interrogate seeking alpha is a gold mine.



The screenshot shows a pricing plan for the 'Pro' plan. The price is \$20.00/mo. The plan includes:

- Requests: 10,000 / Month + \$0.003
- Features: Overcome bot defender layer
- Rate Limit: 5 requests per second
- Bandwidth Platform Fee: 10240MB / Month + 0.001\$ per 1MB

A blue 'Cancel Plan' button is at the bottom.

X-RapidAPI-Host *

seeking-alpha.p.rapidapi.com

accessToken (optional)

String

The value returned in .../accounts/get-access-token endpoint. This header is not required, used to access PREMIUM data with your SA subscribed accounts if provided.

App

default-application_11126086

X-RapidAPI-Key

2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f

Request URL

rapidapi.com

Node.js

```
const http = require('https');

const options = {
    method: 'GET',
    hostname: 'seeking-alpha.p.rapidapi.com',
    port: null,
    path: '/v2/auto-complete?query=apple&type=people%2Csymbols%2Cpages&size=5',
    headers: {
        'x-rapidapi-key': '2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f',
        'x-rapidapi-host': 'seeking-alpha.p.rapidapi.com'
    }
};

const req = http.request(options, function (res) {
    const chunks = [];
    res.on('data', function (chunk) {
        chunks.push(chunk);
    });
    res.on('end', function () {
        const body = Buffer.concat(chunks);
        console.log(body.toString());
    });
});

req.end();
```

Reuters Business and Financial News

Get articles by date, others available

App

default-application_11126086

X-RapidAPI-Key

2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f

Request URL

rapidapi.com

CORRELATE MAJOR EVENTS IMPACT ON TV VIEWING.

World Historical Events API

The World Historical Events API provides a comprehensive database of significant historical events from around the globe. This API allows users to discover notable occurrences that happened on any given day or in a specific year. Ideal for educational purposes, trivia, or enhancing applications with historical context, the API offers robust

X-RapidAPI-Host *

world-historical-events-api.p.rapidapi.com

endpoints for retrieving events based on the current date, a specified date, or a combination of date and year. The responses are structured for easy

integration, ensuring a smooth and informative user experience.

App

default-application_11126086

X-RapidAPI-Key

2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f

Request URL

rapidapi.com

Node.js

```
const http = require('https');

const options = {
    method: 'GET',
    hostname: 'world-historical-events-api.p.rapidapi.com',
    port: null,
    path: '/api/events',
    headers: {
        'x-rapidapi-key': '2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f',
        'x-rapidapi-host': 'world-historical-events-api.p.rapidapi.com'
    }
};

const req = http.request(options, function (res) {
    const chunks = [];
    res.on('data', function (chunk) {
        chunks.push(chunk);
    });

    res.on('end', function () {
        const body = Buffer.concat(chunks);
        console.log(body.toString());
    });
});

req.end();
```

CHECK CORRELATION BETWEEN SPORTS, SPORTS EVENTS AND TIME SPENT WATCHING STREAMING TV BY COUNTRY WORLD WIDE **Sports Highlights API**

App

default-application_11126086

X-RapidAPI-Key

2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f

Request URL

rapidapi.com

```
{  
  "mcpServers": {  
    "RapidAPI Hub - Sport Highlights API": {  
      "command": "npx",  
      "args": [  
        "mcp-remote",  
        "https://mcp.rapidapi.com",  
        "--header",  
        "x-api-host: sport-highlights-api.p.rapidapi.com",  
        "--header",  
        "x-api-key: 2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f"  
      ]  
    }  
  }  
}
```

API Overview

Get ready for real time Sport live scores, player statistics, match lineups & predictions, live events, standings, highlights, game recaps, fouls, and post-match interviews on a global

scale. Integrate a wide spectrum of pre-match and live game odds into your projects for a complete sports data solution.

Platform migration

Highlightly just launched its own platform. By using the API through our official platform you can get up to a 40% discount on your plan purchases. To access our APIs through Highlightly please create an account here <https://highlightly.net/login>

> Note: RapidAPI will not be deprecated. Accounts do not sync cross platforms.

Coverage

When new sports are added in the future, Sport Highlights API will have immediate access for free.

But before using the API you should know that the BASIC tier has certain coverage restrictions in-place (PRO plan or higher removes these restrictions completely).

Sports which support the odds endpoint are available only to users who have paid subscriptions, meaning that the BASIC tier has no access to bookmaker odds and associated markets/selections.

Highlights geo restriction endpoint is available only to users with a paid subscription, meaning that the BASIC tier has no access to information about potential video restrictions.

Additional BASIC/FREE tier restrictions:

Football: FREE tier highlights are returned from following leagues:

Romania - Liga I,

Sweden - Superettan,

Poland - I Liga,

Slovenia - 1. SNL,

Kazakhstan - Premier League,

Lithuania - A Lyga,

Latvia - Virsliga,

Bulgaria - First League,

South Africa - Premier Soccer League,

Singapore - Premier League,
Uruguay - Primera División - Apertura,
Chile - Primera División,
Peru - Primera División,
Finland - Veikkausliiga,
Denmark - Superliga,
England - Championship,
USA - USL Championship,
USA - USL League One,
USA - Major League Soccer,
Netherland - Eredivisie,
Turkey - Süper Lig,
Turkey - 1. Lig,
Turkey - 2. Lig,
Turkey - Super Cup,
Turkey - Cup.

Basketball: highlights are excluded from the following leagues:

USA - NBA,
Europe - Euroleague,
Spain - ACB,
Russia - VTB United League Promo-Cup,
Russia - VTB Super Cup,
DE - BBL,
France - LNB.

Hockey: highlights are excluded from the following leagues:

USA - NHL,

Canada - OHL,

USA - NCAA,

USA - AHL,

Czechia - Extraliga,

Sweden - SHL,

Russia - KHL.

American Football: highlights excluded from NFL when using BASIC plan,

Rugby: includes highlights from all leagues,

Volleyball: includes highlights from all leagues,

Handball: includes highlights from all leagues,

Cricket: includes highlights from all leagues,

Baseball: highlights are excluded for MLB when using BASIC plan,

NBA, NCAA: highlights are excluded for NBA when using BASIC plan,

NHL, NCAA: highlights are excluded for NHL when using BASIC plan.

Reuters Biz and Financial News API

API Overview

This API helps to query for all articles as on reuters.com.

The API is delivered through several endpoints in order to satisfy different request types.

** REAL-TIME updated ** * CUSTOM endpoints *** PERSONALIZED pricing plans ***
DEDICATED support **

Feel free to contact us for more information.

Subscribe now to get access.

The Reuters News API is a comprehensive and accurate data source for real-time news content from Reuters. This API enables users to retrieve the latest news articles and updates across a wide range of categories, including business, politics, technology, health, entertainment, world events, and more. Users can filter news based on keywords, location, and date range, making it highly adaptable to diverse applications. Whether for media monitoring, news aggregation, research, or integration into apps and platforms, this API delivers timely and trusted news content from one of the world's most respected sources.

App

default-application_11126086

X-RapidAPI-Key

2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f

Request URL

rapidapi.com

```
import http.client
conn = http.client.HTTPSConnection("reuters-business-and-financial-news.p.rapidapi.com")
headers = {
    'x-rapidapi-key': "2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f",
    'x-rapidapi-host': "reuters-business-and-financial-news.p.rapidapi.com"
}
```

```
conn.request("GET", "/category-id/240/article-date/2024-01-01/0/20", headers=headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

CONFIDENTIAL

SECTION 2 TV AND MOVIE METADATA

FlixPatrol Professional Key

CORRECTED

Authentication

The FlixPatrol API uses API keys to authenticate requests. You can view your API key in [your profile](#).

Authentication to the API is performed via [HTTP Basic Auth](#). Provide your API key as the basic auth username value. You do not need to provide a password.

Your username (API key) for HTTP Auth:

aku_4bXKmMWPSCaKxwn2tVzTXmcg

Example of raw HTTP header usage:

Authorization: Basic YWt1XzRiWEttTVdQU0NhS3h3bjJ0VnpUWG1jZzo=

WARNING.

Flixpatrol is very finickity. Always use this script as a template
"C:\Users\RoyT6\Downloads\PERFECT FLIXPATROL CONNECT KEEP
SAFE.PY"

Or PYTHON SCRIPT TEMPLATE

```
#!/usr/bin/env python3
"""

FlixPatrol API v2 Connection Test

Using correct endpoint: api.flixpatrol.com/v2
"""

import requests
from requests.auth import HTTPBasicAuth

API_KEY = "aku_4bXKmMWPSCaKxwn2tVzTXmcg"
BASE = "https://api.flixpatrol.com/v2"
```

```
print("=" * 50)
print("FLIXPATROL API v2 CONNECTION TEST")
print("=" * 50)

# Test 1: OPTIONS (Help endpoint)
print("\n[1] Testing OPTIONS /titles (help)...")
try:
    r = requests.options(f"{BASE}/titles", timeout=15)
    print(f"  Response: {r.status_code}")
    if r.status_code == 200:
        print(f"  OK - Help available")
except Exception as e:
    print(f"  FAILED: {type(e).__name__}: {e}")

# Test 2: HEAD (count items)
print("\n[2] Testing HEAD /titles (count)...")
try:
    r = requests.head(
        f"{BASE}/titles",
        auth=HTTPBasicAuth(API_KEY, ""),
        timeout=15
    )
    print(f"  Response: {r.status_code}")
    if "x-total-count" in r.headers:
        print(f"  OK - Total count: {r.headers['x-total-count']}")
except Exception as e:
    print(f"  FAILED: {type(e).__name__}: {e}")

# Test 3: GET titles list (with auth)
print("\n[3] Testing GET /titles (list)...")
```

```
try:
    r = requests.get(
        f"{BASE}/titles",
        auth=HTTPBasicAuth(API_KEY, ""),
        timeout=15
    )
    print(f"  Response: {r.status_code}")
    if r.status_code == 200:
        data = r.json()
        print(f"    OK - Got data: {str(data)[:300]}...")
    elif r.status_code == 401:
        print(f"    AUTH FAILED - Check API key format")
        print(f"    Response: {r.text[:200]}")
    else:
        print(f"    Response body: {r.text[:200]}")
except Exception as e:
    print(f"    FAILED: {type(e).__name__}: {e}")

# Test 4: GET single title
print("\n[4] Testing GET /titles/:id...")
try:
    r = requests.get(
        f"{BASE}/titles/ttl_bHyGTvopBHPVtIKhR2CF68WD",
        auth=HTTPBasicAuth(API_KEY, ""),
        timeout=15
    )
    print(f"  Response: {r.status_code}")
    if r.status_code == 200:
        data = r.json()
        title = data.get("data", {}).get("title", "Unknown")
        print(f"    OK - Title: {title}")

```

```
except Exception as e:  
    print(f"  FAILED: {type(e).__name__}: {e}")  
  
    print("\n" + "=" * 50)  
    print("TEST COMPLETE")  
    print("=" * 50)  
  
    input("\nPress Enter to close...")
```

OMDB Professional Key

2d8b9d62

Send all data requests to:

[&](http://www.omdbapi.com/?apikey=[2d8b9d62])

Poster API requests:

[&](http://img.omdbapi.com/?apikey=[2d8b9d62])

TMDB Professional Key

Bearer read access token

eyJhbGciOiJIUzI1NiJ9eyJhdWQiOiI4ZmU0ZDc4MjA1MmQ2YzZkY2U0Y2YyM2Y0YjAxYjNjNSIслm5iZil6MTc1OTk3NzE1Ny4zMTEsInN1Yil6ljY4ZTcxZWM1MjdjZjcxOTcyNDczZDc5NilsInNjb3Blcyl6WyJhcGlfcmVhZCJdLCJ2ZXJzaW9uljoxfQ._Q_5lA7npuYpFJzYOFAsvxdGjzzKZSoC0t-63BJ8m4U

API Key

8fe4d782052d6c6dce4cf23f4b01b3c5

Thetvdb.com

id: 6931FBBC3066E8.48167962

03fb627f-e08c-418c-a1b2-3b40869def52

CONFIDENTIAL

****MOST IMPORTANT****

Streaming Availability

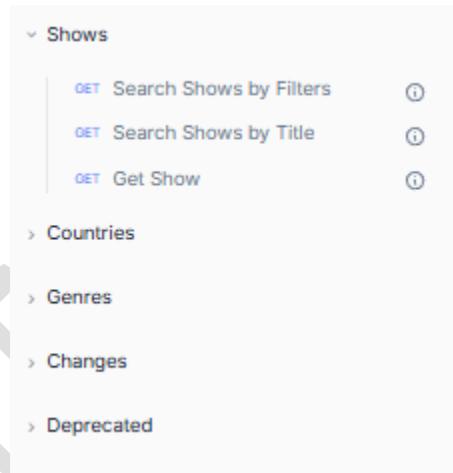
API Overview

Look up which shows are available on Netflix, Amazon Prime Video, Disney+, HBO Max, Hulu, Peacock, Paramount+, Starz, Showtime, Apple TV, Mubi, Stan, Now,

X-RapidAPI-Host *

streaming-availability.p.rapidapi.com

Wow, Crave,
All 4, BBC iPlayer, BritBox, Hotstar, Zee5 and many other
streaming services across 60 countries. Get deep links,
top shows and more!



Query Params

output_language *(optional)*

en

String

ISO 639-1 code of the output language. Determines in which language the output will be in.

Netflix, Prime, Disney+, HBO, Hulu, AppleTV and many others. Lookup which shows are available on streaming services across 60 countries with deep links.

Streaming Availability
API allows getting streaming availability information of movies and series; and querying the list of available shows on streaming services such as Netflix, Disney+, Apple TV, Max and Hulu across 60 countries!

Many others availability

App

default-application_11126086

X-RapidAPI-Key

2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f

Request URL

rapidapi.com

The screenshot displays two plan options side-by-side: 'Ultra' on the left and 'Mega' on the right. Both plans are monthly subscriptions.

Plan	Price	API Requests / Month	Rate Limit	Bandwidth Platform Fee
Ultra	\$49.00 /mo	100,000 / Month	150 requests per second	10240MB / Month +0.001\$ per 1MB
Mega	\$199.00 /mo	1,000,000 / Month	250 requests per second	10240MB / Month +0.001\$ per 1MB

Ultra Plan Details:

- API Request ⚡ 100,000 / Month Hard Limit
- Features
 - ✓ [V1 & V2] Leaving ⚡
 - ✓ [V1 & V2] Changes ⚡
 - ✓ [V1 & V2] Pro Features
 - ✓ [V1 & V2] Basic Features ⚡
 - ✓ [V1 & V2] Ultra Features ⚡
 - ✓ [V3 & V4] All Endpoints ⚡
 - ✓ [V1 & V2] Base Utilities ⚡
- Rate Limit: 150 requests per second
- Bandwidth Platform Fee: 10240MB / Month +0.001\$ per 1MB

Mega Plan Details:

- API Request ⚡ 1,000,000 / Month Hard Limit
- Features
 - ✓ [V1 & V2] Leaving ⚡
 - ✓ [V1 & V2] Changes ⚡
 - ✓ [V1 & V2] Pro Features
 - ✓ [V1 & V2] Basic Features ⚡
 - ✓ [V1 & V2] Ultra Features ⚡
 - ✓ [V3 & V4] All Endpoints ⚡
 - ✓ [V1 & V2] Base Utilities ⚡
- Rate Limit: 250 requests per second
- Bandwidth Platform Fee: 10240MB / Month +0.001\$ per 1MB

Buttons:

- Downgrade Plan (Ultra)
- Cancel Plan (Mega)

Node.js

```
const http = require('https');

const options = {
    method: 'GET',
    hostname: 'streaming-availability.p.rapidapi.com',
    port: null,
    path: '/countries?output_language=en',
    headers: {
        'x-rapidapi-key': '2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f',
        'x-rapidapi-host': 'streaming-availability.p.rapidapi.com'
    }
};

const req = http.request(options, function (res) {
    const chunks = [];

    res.on('data', function (chunk) {
        chunks.push(chunk);
    });

    res.on('end', function () {
        const body = Buffer.concat(chunks);
        console.log(body.toString());
    });
});

req.end();
```

We Care About These Streaming Platforms Supported

Service Id	Service Name	Supported Countries
netflix	Netflix	59 Countries
prime	Prime Video	56 Countries
disney	Disney+	46 Countries
hbo	Max	28 Countries
hulu	Hulu	United States
peacock	Peacock	United States
paramount	Paramount+	18 Countries
starz	Starz	United States
apple	Apple TV	52 Countries
mubi	Mubi	53 Countries
stan	Stan	Australia
now	Now	United Kingdom, Ireland, Italy
crave	Crave	Canada
all4	Channel 4	United Kingdom, Ireland
iplayer	BBC iPlayer	United Kingdom
britbox	BritBox	United States, Canada, Australia, South Africa
hotstar	Hotstar	India, Canada, United Kingdom, Singapore
zee5	Zee5	58 Countries
curiosity	Curiosity Stream	57 Countries
wow	Wow	Germany
discovery	Discovery+	United States, Canada, Ireland, Italy, United Kingdom, Germany, Austria
sonyliv	SonyLiv	India

Service Id	Service Name	Supported Countries
itvx	ITVX	United Kingdom
plutotv	Pluto TV	25 Countries
tubi	Tubi	Australia, Canada, New Zealand, Ecuador, Mexico, Panama, United States
blutv	BluTV	Turkey, Germany, Azerbaijan

We care about these countries and regions

	Country	Region
1	United States	USA
2	Germany	EU
3	India	ROW
4	Japan	ROW
5	United Kingdom	UK
6	France	EU
7	Italy	EU
8	Canada	USA
9	Brazil	ROW
10	Russia	ROW
		SE
11	South Korea	ASAI
		SE
12	Australia	ASAI
	New	SE
13	Zealand	ASAI
		SE
14	Singapore	ASAI
		SE
15	Hong Kong	ASAI
16	Mexico	ROW

17	Spain	EU
18	Netherlands	EU
19	Turkey	EU
20	Poland	EU
21	Sweden	EU
22	Belgium	EU
23	Argentina	ROW
24	Ireland	EU
25	Norway	EU
26	Denmark	EU
27	South Africa	ROW
28	Finland	EU
29	Portugal	EU

****IMDb API** AP17P**

The IMDb API provides access to detailed information about movies, TV shows, actors, directors, and more. With advanced search capabilities, ratings, metadata, and popular

Basic	Pro	Ultra	Mega
\$0.00 /mo	\$9.99 /mo	\$49.99 /mo	\$99.99 /mo
Requests 3 / Month Hard Limit	Requests 5,000 / Month Hard Limit	Requests 200,000 / Month Hard Limit	Requests 700,000 / Month Hard Limit
Rate Limit 1000 requests per hour	Rate Limit -	Rate Limit -	Rate Limit -
Bandwidth Platform Fee 10240MB / Month +0.001\$ per 1MB	Bandwidth Platform Fee 10240MB / Month +0.001\$ per 1MB	Bandwidth Platform Fee 10240MB / Month +0.001\$ per 1MB	Bandwidth Platform Fee 10240MB / Month +0.001\$ per 1MB
Downgrade Plan	Downgrade Plan	Cancel Plan	Upgrade Plan

content queries, this API is a powerful tool for integrating movie and TV data into your applications or websites.

App

default-application_11126086

X-RapidAPI-Key

2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f

Request URL

rapidapi.com

Country Code	Country Name
ae	United Emirates
ar	Argentina
at	Austria
au	Australia

Country Code	Country Name
az	Azerbaijan
be	Belgium
bg	Bulgaria
br	Brazil
ca	Canada
ch	Switzerland
cl	Chile
co	Colombia
cy	Cyprus
cz	Czech Republic
de	Germany
dk	Denmark
ec	Ecuador
ee	Estonia
es	Spain
fi	Finland
fr	France
gb	United Kingdom
gr	Greece
hk	Hong Kong
hr	Croatia
hu	Hungary
id	Indonesia

Country Code	Country Name
ie	Ireland
il	Israel
in	India
is	Iceland
it	Italy
jp	Japan
kr	South Korea
lt	Lithuania
md	Moldova
mk	North Macedonia
mx	Mexico
my	Malaysia
nl	Netherlands
no	Norway
nz	New Zealand
pa	Panama
pe	Peru
ph	Philippines
pl	Poland
pt	Portugal
ro	Romania
rs	Serbia
ru	Russia

Country Code	Country Name
se	Sweden
sg	Singapore
si	Slovenia
sk	Slovakia
th	Thailand
tr	Turkey
ua	Ukraine
us	United States
vn	Vietnam
za	South Africa

The IMDb API provides access to detailed information about movies, TV shows, actors, directors, and more. With advanced search capabilities,

X-RapidAPI-Host *

imdb-api17.p.rapidapi.com

applications or websites.

🚀 Features

- Advanced Search: Perform detailed searches with multiple filters for titles, names, and keywords.
- Search Suggestions: Get query-based search suggestions to assist with user queries.
- Title, Name, and Keyword Search: Search for movies, TV shows, people, and keywords.

ratings, metadata, and popular content queries, this API is a powerful tool for integrating movie and TV data into your

- GET Advanced search ⓘ
- GET Get search suggestions for a ... ⓘ
- GET Search for titles, names, and k... ⓘ
- GET Get all available title types ⓘ
- GET Get all available genres ⓘ
- GET Get most popular movies or T... ⓘ
- GET Get detailed information ⓘ
- GET Get rating information ⓘ
- GET Get Metacritic information ⓘ
- GET Get MetaScore ⓘ
- GET Get poster information ⓘ
- GET Get directors information ⓘ
- GET Get writers information ⓘ
- GET Get Cast Information ⓘ

- Available Title Types: Retrieve a list of all movie and TV show title types.
- Available Genres: Get a list of available genres for filtering content.

Query Params

query *(optional)*

String

sortOrder *(optional)*

ASC

Enum

limit *(optional)*

100

String

countries *(optional)*

IN

String

- Most Popular Movies/TV Series: Fetch the most popular movies or TV series based on ratings or user engagement.
- Detailed Title Information: Access detailed information such as plot, release year, cast, crew, etc.
- Rating Information: Get movie/TV ratings from IMDb and other sources like Rotten Tomatoes.
- Metacritic Information: Retrieve Metacritic scores and related metadata.
- MetaScore: Access the MetaScore of a title, helping to evaluate critical reception.
- Poster Information: Fetch movie posters or cover images for use in your application.

- Director Information: Get detailed information about directors for movies and TV shows.
- Writer Information: Retrieve detailed information about writers for a title.
- Cast Information: Get complete cast lists and biographies for titles.

⚙️ How It Works

1. Search: Use the search endpoints to find titles, people, or keywords from IMDb's massive database.
2. Filter: Narrow down results using title types, genres, or specific rating criteria.
3. Get Information: Fetch comprehensive details, including ratings, plot, cast, crew, posters, and more for movies and TV shows.
4. Use the Data: Easily integrate IMDb data into your website, app, or database for a rich, dynamic media experience.

Advanced search, many others available

CONFIDENTIAL

HBO Max Movies and TV Shows API by APIRobots

Access a wealth of entertainment content with our comprehensive HBO Max API. Explore thousands of movies and TV shows, filter results, and discover new favorites easily.

Key Features Comprehensive catalog of HBO Max titles
Filter results by name
Paginate results for efficient browsing
Get random title recommendations
Detailed metadata for each title
API Endpoints
Get HBO Max Titles GET /v1/hbo-max
Filter by name (optional)
Paginate results (default page 1)
Returns a list of HBO Max titles
Get Random Title GET /v1/hbo-max/random
Returns a randomly selected HBO Max title
Includes detailed metadata about the title
Response Format All responses are returned in JSON format.
Each title includes:

Title	Description	Release Year	Age Certification	Runtime	Genres	Production Countries
Number of Seasons	IMDB ID	IMDB Score	IMDB Votes	TMDB Popularity	TMDB Score	

Start exploring the world of HBO Max content today with our powerful API!

APIRobots APIs:

We provide and develop high-quality APIs. Our APIs have a 99.99% uptime.

Our aim is to achieve the lowest possible latency. Support: <https://www.apirobots.pro/>
App

default-application_11126086

X-RapidAPI-Key

2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f

Request URL

rapidapi.com

Disney

Experience the magic of Disney with our comprehensive Disney Plus Top Movies and TV Shows API!

Key Features

- Curated Disney Content: Access a handpicked selection of top Disney+, Hulu, and ESPN+ titles
- Flexible Filtering: Easily filter results by name, genre, or other criteria
- Pagination Support: Manage large result sets with ease
- Random Title Selection: Get instant excitement with random show or movie retrieval
- Comprehensive Episode Information: Retrieve extensive details about each title

Endpoints Our API offers two endpoints to cater to different needs:

/v1/disney-plus-top: Get a list of Disney Plus titles /v1/disney-plus-top/random: Get a random Disney Plus title

App

default-application_11126086

X-RapidAPI-Key

2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f

Request URL

rapidapi.com

curl --request GET

```
--url 'https://disney-plus-top-movies-and-tv-shows-api-by-
apirobots.p.rapidapi.com/v1/disney-plus-top?page=1'
```

```
--header 'x-rapidapi-host: disney-plus-top-movies-and-tv-shows-api-by-
apirobots.p.rapidapi.com'
```

```
--header 'x-rapidapi-key: 2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f'
```

Rotten Tomatoes

Rotten Tomatoes API Access real-time data on movies and TV shows powered by Rotten Tomatoes insights. This API is designed for developers integrating entertainment data into apps, websites, or services.

Endpoints Overview

General

GET /by_name Search by title to retrieve metadata for movies or TV shows

!!!! when using this endpoint to send correct film name is required you could use the movie query endpoint to fetch the name correctly and then try to provides that name in the params in this case you will get the most accurate results !!! there are two routes get by name one in movies and it get movies by name !!! second is for tv_shows in tv_shows collection

Movies

GET /query_movies Retrieve information of movies with pagination and sortby functions

TV Shows

GET /soon-in-theaters Explore upcoming TV shows and limited series expected to release soon.

GET /today-top100TVshows-netflix Get the daily top 100 trending TV shows on Netflix worldwide.

Note: More endpoints will be added regularly. Stay tuned for updates. For custom plans or enterprise access, contact us at: contact@g3oapi.com

Access Rotten Tomatoes movie and TV data instantly with our real-time API. Get movie reviews, critic and audience ratings, cast and crew details, trailers, posters, spoilers and high-quality images to power smarter entertainment apps. Stay updated with live Rotten Tomatoes scores and unlock personalized recommendations for films and shows. Perfect for building movie databases, streaming apps, or recommendation engines with trusted, up-to-date entertainment insights.

App

default-application_11126086

X-RapidAPI-Key

2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f

Request URL

rapidapi.com

curl --request GET

--url 'https://rottentomato.p.rapidapi.com/streaming-

tv?genre=action&sortby=critic_highest&after=Mjk%3D'

--header 'x-rapidapi-host: rottentomato.p.rapidapi.com'

--header 'x-rapidapi-key: 2e5f1ffd8dmsh65f83d24ed46b1cp18bd77jsn16c55b9c022f'

Ryzen 9 Optimizations

To optimize Python API scraping on a Ryzen 9 3950X, focus on asynchronous programming for the I/O-bound network requests and utilize multiprocessing for CPU-bound data parsing, leveraging your 16 physical cores effectively. The L3 cache is managed automatically by the CPU hardware, but efficient memory access patterns can improve its usage.

Core and Thread Optimization

Web scraping is primarily an I/O-bound task, meaning the program spends most of its time waiting for network responses, not using the CPU. Therefore, standard Python multithreading or asynchronous I/O (asyncio) is highly effective as it allows concurrent network requests while waiting.

Asynchronous I/O (Asyncio): This is generally more efficient for high-concurrency network tasks than threading due to less overhead. Use libraries such as `httpx` for requests and `asyncio` for managing concurrency.

Multithreading: The `concurrent.futures.ThreadPoolExecutor` provides a straightforward way to implement multithreading for I/O operations.

Multiprocessing: The `multiprocessing` module is used for CPU-bound tasks to bypass Python's Global Interpreter Lock (GIL). Use it to offload data parsing (e.g., using `lxml` or `BeautifulSoup`) or complex data analysis to separate cores after the data has been fetched. The Ryzen 9 3950X has 16 physical cores, so a process pool size slightly less than or equal to this (e.g., 15) is a good starting point to leave one core free for the OS.

L3 Cache Optimization

You cannot program the L3 cache directly. The optimization is automatic based on memory access patterns. Key strategies to encourage efficient cache usage include:

Memory Locality: Structure your data and code to access data that is close together in memory frequently. For scraping, this means processing a batch of data in the same process to keep relevant data in its local cache before moving to the next batch.

Efficient Data Structures: Use optimized libraries like NumPy or Pandas for data processing tasks, as they are C-extensions with better memory management and access patterns that compilers can optimize for cache use.

Batch Processing: Scrape and process data in manageable batches rather than one item at a time. This reduces inter-process communication overhead and allows local processing to better leverage the L3 cache.

Recommended Python Libraries

Scrapy: A powerful, highly efficient framework designed for large-scale crawling. It handles asynchronous requests internally and is optimized for performance and scalability, making it an excellent choice for a high-core system.

Requests + BeautifulSoup / Lxml: A popular combination for smaller, more static scraping tasks.

Use lxml as a parser with BeautifulSoup for faster HTML/XML parsing.

Httpx / Aiohttp: Asynchronous HTTP clients that are well-suited for high-speed, concurrent API requests.

Summary of Best Practices

Separate Concerns: Scrape data asynchronously/multithreaded and then parse/process it using a multiprocessing.Pool.

Monitor Cores: Start with a process/thread count around your physical core count (16) and adjust based on performance monitoring to find the sweet spot, as too many concurrent tasks can increase overhead.

Minimize Communication Overhead: Limit the amount of data passed between processes, as this negates the benefits of multiprocessing.

Use Optimized Libraries: Leverage libraries written in lower-level languages (like C) where possible, such as lxml for parsing and NumPy for data handling.

Anti Throttling Guide

Managing API throttling in Python involves a combination of libraries for **asynchronous programming**, **rate limiting**, and **retries with exponential backoff**. The mentions of "scrappy" and "hash" likely refer to the web scraping framework Scrapy and the use of hashes for identifying clients (e.g., API keys), respectively.

1. Asynchronous Programming (asyncio)

Python's built-in asyncio library is the foundation for efficient, concurrent API calls, as it handles I/O operations without blocking the main event loop.

- **Key Concept:** Asynchronous programming allows your program to manage thousands of open network connections simultaneously, rather than waiting for each one to complete sequentially.
- **Implementation:** Use async/await syntax with an asynchronous HTTP library like aiohttp or frameworks like FastAPI.

2. Rate Limiting Libraries

Instead of manually implementing throttling logic, use existing libraries that provide robust mechanisms like the leaky bucket or token bucket algorithms.

- **limits / slowapi:** These libraries offer common APIs for both synchronous and asynchronous Python codebases and support various storage backends like Redis for distributed rate limiting across multiple processes.
- **pyrate-limiter:** A general-purpose rate-limiting library with optional persistence via SQLite or Redis, offering an async interface try_acquire_async that uses asyncio.sleep to prevent blocking the event loop.
- **requests-ratelimiter:** This library is useful if you are using the popular requests library, adding rate-limiting functionality directly to sessions and tracking limits per host.

3. Retries and Exponential Backoff (tenacity)

When a rate limit is hit (an HTTP 429 Too Many Requests error), a crucial strategy is to automatically retry the request after a delay.

- **tenacity:** This powerful library simplifies the task of adding retry behavior using a decorator. It implements **exponential backoff**, which means the delay between retries increases with each failure, reducing the load on the API and avoiding immediate re-hitting of the limit.
- **Implementation:**

```
python
```

```
from tenacity import retry, wait_exponential, stop_after_attempt
import requests

@retry(wait=wait_exponential(multiplier=1, min=2, max=60), stop=stop_after_attempt(5))
def fetch_data(url):
    response = requests.get(url)
    if response.status_code == 429:
        raise Exception("Rate limited!")
    return response.json()
```

This approach ensures retries are handled gracefully until the request succeeds or a maximum number of attempts is reached.

4. Scrapy Framework

If you are using Scrapy for web scraping, it has built-in mechanisms for managing crawling speed:

- **AutoThrottle Extension:** This extension dynamically adjusts the download delay to be "nicer to sites" and optimize crawling speed based on server response latency.
- **Settings:** You can configure DOWNLOAD_DELAY and AUTOTHROTTLE_TARGET_CONCURRENCY in your settings.py file to control request frequency and concurrency.

5. Using "Hash" for API Key Management

The term "hash" in this context likely refers to using a specific API key (which is often a hash or unique string) to identify a user for per-user rate limiting. You can use libraries like slowapi with a custom key_func to limit requests based on the API key provided in the request headers:

python

```
from slowapi.util import get_remote_address  
from slowapi.extension import Limiter  
  
def get_api_key(request: Request):  
    return request.headers.get("X-API-Key", get_remote_address(request))  
  
limiter = Limiter(key_func=get_api_key)
```

This ensures different API clients are throttled individually.