

CinemaSeek

An amazing movies API





Overview

Our motivation and what technology we use





OVERVIEW

Motivation

We love movies, and we know a substantial number of people do so as well. Which is why we want to provide basic to advanced information about movies that are easily accessible and comprehensive to the mass. Be it a simple query, or graphically plotting a statistic.



OVERVIEW

Pain Points

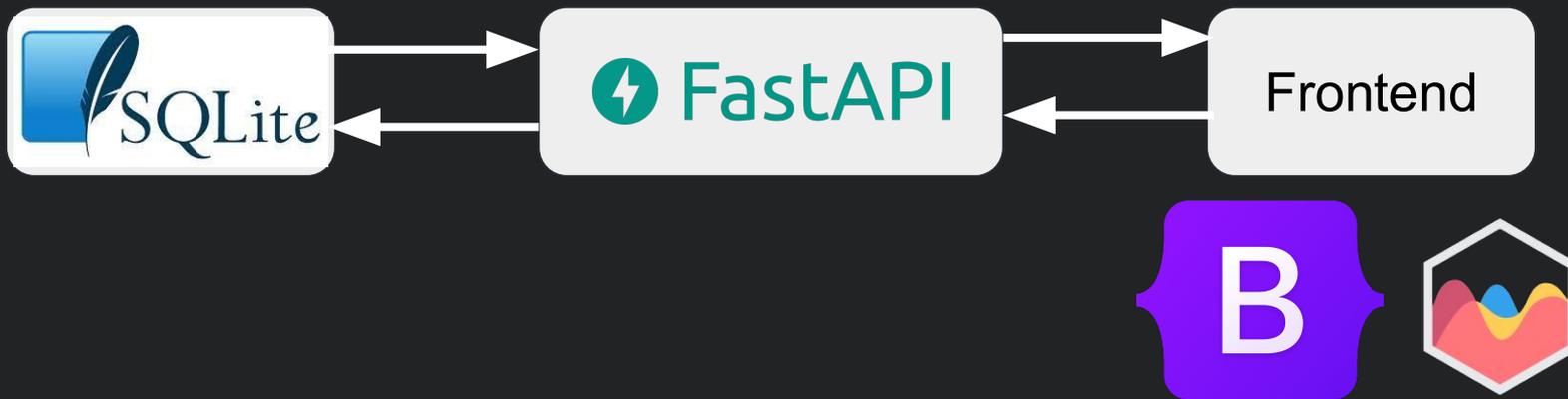
We are certain that our service can combat these pain points:

- Intimidating query services. As in the services are hard to comprehend and use.
- Messy API responses.
- Difficulty in checking a movie's availability on a streaming platform.
- Insufficient data visualization.



OVERVIEW

Overall architecture





OVERVIEW

API Documentation

Swagger documentation available at '/docs'

The screenshot shows the Swagger UI for the CinemaSeek API. At the top, it displays the logo 'CinemaSeek' with version '1.0.0' and 'OAS3' in a green badge. Below the logo is the file path '/openapi.json' and a brief description: 'An API about movie data.' The main content is organized into two sections: 'default' and 'Schemas'. The 'default' section contains a list of seven GET endpoints, each with a blue 'GET' button, the endpoint path, and a brief description. The 'Schemas' section is partially visible, showing two expandable items: 'ActorHighGrossing' and 'DirectorAsActor'.

CinemaSeek 1.0.0 OAS3
/openapi.json
An API about movie data.

default ^

- GET /api/actor/high-grossing-movies Get Actor High Grossing Movies
- GET /api/actor/feature-percent Get Feature Percent
- GET /api/movie/director-as-actor Get Director As Actor
- GET /api/movie/studio-movie-list Get Studio Movie List
- GET /api/rating/high-gross-low-rate Get High Gross Low Rate
- GET /api/rating/yearly-averages Get Yearly Averages
- GET /api/platform/flix-and-theatre Get Flix And Theatre

Schemas ^

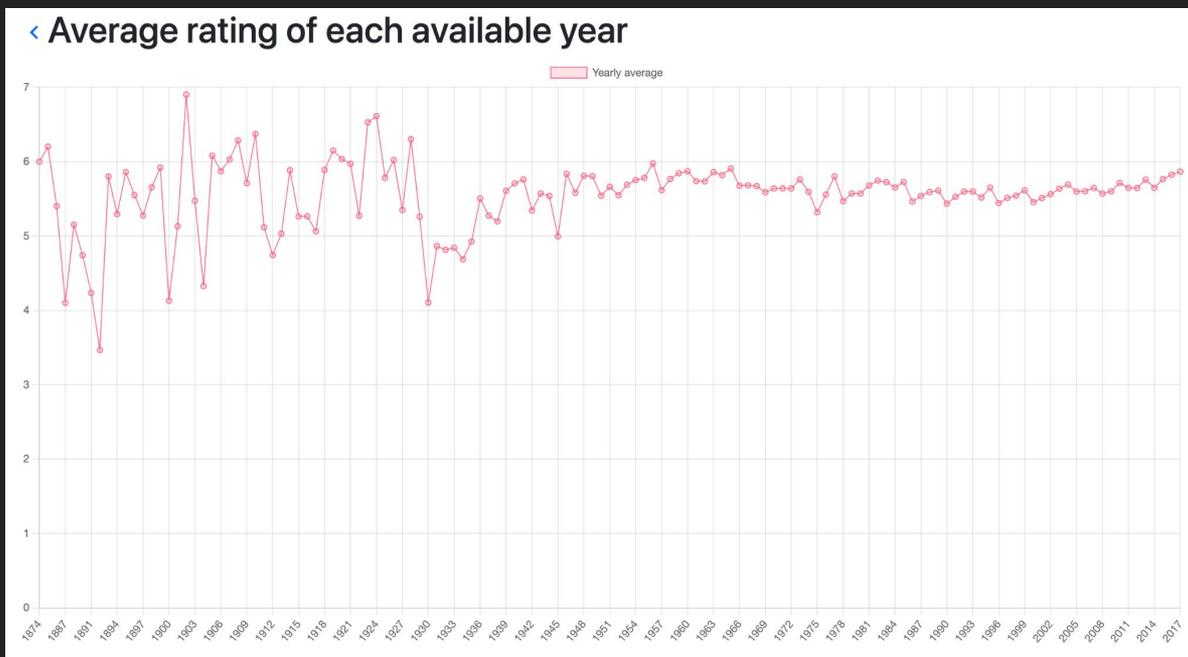
- ActorHighGrossing >
- DirectorAsActor >



OVERVIEW

Data Visualization

Use Chart.js's line and bar chart for visualization





Data Sources

Where we get the data from





DATA SOURCE

Primary Datasource

<https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset>

The screenshot shows the Kaggle interface for the 'The Movies Dataset'. On the left is a navigation sidebar with options like 'Create', 'Home', 'Competitions', 'Datasets', 'Code', 'Discussions', 'Learn', and 'More'. The main content area features a search bar, 'Sign In', and 'Register' buttons. Below these are buttons for '3035', 'New Notebook', and 'Download (239 MB)'. The dataset title 'The Movies Dataset' is prominently displayed, followed by a subtitle: 'Metadata on over 45,000 movies. 26 million ratings from over 270,000 users.' A small image of a woman is shown to the right. Below the title, there are tabs for 'Data', 'Code (374)', and 'Discussion (31)'. The 'About Dataset' section includes a 'Context' subsection with text about the dataset's origin and content, and a 'Content' subsection listing the files. On the right side, there are sections for 'Usability' (8.24), 'License' (CC0: Public Domain), and 'Expected update frequency' (Not specified).



DATA SOURCE

Support Datasource

<https://www.kaggle.com/datasets/sanjeetsinghnaik/top-1000-highest-grossing-movies> (Top 1000 Highest Grossing Movies)

The screenshot shows the Kaggle dataset page for 'Top 1000 Highest Grossing Movies'. The page layout includes a left sidebar with navigation options like 'Create', 'Home', 'Competitions', 'Datasets', 'Code', 'Discussions', 'Learn', and 'More'. The main content area features a search bar, a 'Sign In' button, and a 'Register' button. Below these, there's a user profile for 'SANJEET SINGH NAIK' with a '191' badge and a 'New Notebook' button. The dataset title 'Top 1000 Highest Grossing Movies' is prominently displayed, along with a 'Download (108 KB)' button and a grid of movie posters. The 'About Dataset' section provides context, acknowledgements, and an update note. On the right, the 'Usability' is 10.00, the license is 'CC0: Public Domain', and the expected update frequency is 'Annually'. At the bottom, there are tags for 'Movies and TV Shows', 'Beginner', 'Data Visualization', 'Exploratory Data Analysis', and 'Intermediate'.



DATA SOURCE

Supporting Datasource

<https://www.kaggle.com/datasets/harshitshankhdhar/imdb-dataset-of-top-1000-movies-and-tv-shows> (Unused)

The screenshot shows the Kaggle dataset page for 'IMDB Movies Dataset' by Harshit Shankhdhar. The page includes a search bar, navigation links, and dataset details. The 'About Dataset' section provides context and content information.

IMDB Movies Dataset
Top 1000 Movies by IMDB Rating

About Dataset

Context

IMDB Dataset of top 1000 movies and tv shows.
You can find the EDA Process on - <https://www.kaggle.com/harshitshankhdhar/eda-on-imdb-movies-dataset>
Please consider UPVOTE if you found it useful.

Content

Data:-

- Poster_Link - Link of the poster that imdb using
- Series_Title = Name of the movie
- Released_Year - Year at which that movie released
- Certificate - Certificate earned by that movie

Usability 10.00

License CC0: Public Domain

Expected update frequency Quarterly



DATA SOURCE

Supporting Datasource

<https://www.kaggle.com/datasets/victorsoeiro/netflix-tv-shows-and-movies> (Netflix)

The screenshot shows the Kaggle dataset page for 'Netflix TV Shows and Movies'. The page layout includes a left sidebar with navigation options like 'Home', 'Competitions', 'Datasets', 'Code', 'Discussions', 'Learn', and 'More'. The main content area features a search bar, a user profile for 'VICTOR SOEIRO', and a '579' badge. The dataset title 'Netflix TV Shows and Movies' is prominently displayed, along with a description: 'Movies and TV Shows listings on Netflix (July, 2022)'. Below the title, there are tabs for 'Data', 'Code (88)', and 'Discussion (4)'. The 'About Dataset' section provides details about the data set's creation and purpose. On the right side, there are buttons for 'Download (2 MB)', 'New Notebook', and 'Sign In'. The 'Usability' section shows a score of 10.00 and a 'credits.csv (3.82 MB)' file. The 'Expected update frequency' is listed as 'Annually'. The 'Content' section mentions two files: 'titles.csv' and 'credits.csv'.

Netflix TV Shows and Movies
Movies and TV Shows listings on Netflix (July, 2022)

About Dataset

Netflix - TV Shows and Movies

This data set was created to list all shows available on Netflix streaming, and analyze the data to find interesting facts. This data was acquired in July 2022 containing data available in the United States.

Content

This dataset has two files containing the titles (titles.csv) and the cast (credits.csv) for the title.

Usability
10.00
credits.csv (3.82 MB)
CC0: Public Domain

Expected update frequency
Annually



DATA SOURCE

Back-up API

<https://developers.themoviedb.org/> (Unused)

The screenshot shows the 'Getting Started' page of the The Movie Database API version 3. The page has a dark blue header with the logo and navigation links. A left sidebar contains a table of contents with categories like 'GETTING STARTED', 'ACCOUNT', 'AUTHENTICATION', etc. The main content area is titled 'Getting Started' and 'Introduction'. It includes a welcome message, instructions on how to register for an API key, and a list of useful tips.

THE MOVIE DB The Movie Database API 3 <https://api.themoviedb.org/3> OAS RAML Support

Select a different version
Filter sections...

GETTING STARTED
[Introduction](#)
Authentication
Daily File Exports
Languages
Images
Image Languages
Regions
External IDs
Popularity
Request Rate Limiting
JSON & JSONP
Append To Response
Search & Query For Details

ACCOUNT
AUTHENTICATION
CERTIFICATIONS
CHANGES
COLLECTIONS
COMPANIES
CONFIGURATION
CREDITS
DISCOVER

Getting Started

Introduction

Welcome to version 3 of The Movie Database (TMDB) API. Below you will find a current list of the available methods on our movie, tv, actor and image API. If you need help or support, please head over to our [API support forum](#).

To register for an API key, click the [API link](#) from within your account settings page. You can also view the screenshots below for help:

1. Click on your avatar or initials in the main navigation (screenshot)
2. Click the "Settings" link (screenshot)
3. Click the "API" link in the left sidebar (screenshot)
4. Click "Create" or "click here" on the API page (screenshot)

Please note that the API registration process is *not optimized* for mobile devices so you should access these pages on a desktop computer and browser.

Before being issued an API key you will have to agree to our terms of use. You can read that [here](#).

A few useful tips...

- The configuration methods are useful to get the static lists of data we use throughout the database. You can find things like the languages, countries, timezones and translations that we use. The configuration method also holds useful image information.
- Understanding the basics of our authentication is useful. You can read about this [here](#).
- We enforce rate limiting on the API. You can read about that [here](#).



DATA SOURCE

Collection Mechanism

- We gather various data from freely available data sources in CSV format. Mainly from Kaggle as seen in previous slides.
- The collected CSV data is then imported into an SQLite database table with accommodating schema.
- This lets our API utilise SQL queries.
- We first tried querying directly from CSV and realized that it is way too unstructured and leads to spaghetti code.



Database

We use it to not deal with 1GB of CSV manually :)





DATABASE

Database schema

We define the database schema based on the CSV data source, which is imported into our database. Here are some examples.

(Full schema available in 'table.sql')

Schema example

highest_grossing_movies

```
create table highest_grossing_movies
(
    id                INTEGER,
    Title             TEXT,
    "Movie Info"     TEXT,
    Distributor       TEXT,
    "Release Date"   TEXT,
    "Domestic Sales (in $)"  INTEGER,
    "International Sales (in $)"  INTEGER,
    "World Sales (in $)"     REAL,
    Genre             TEXT,
    "Movie Runtime"   TEXT,
    License           TEXT
);
```

Schema example

movies_credit

```
create table movies_credits
(
    cast TEXT,
    crew TEXT,
    id TEXT
);
```



Demonstration time!

Let's see how it works!

