Retrieving Western Movies with TMDB API

Ahmet Yiğit Doğan

14 Dec, 2022

Contents

1 Introduction

In this study, an API call will be sent to the The Movie Database (TMDB) servers to retrieve a list of Italian Western Movies. The resulting table will include th first 20 movies with the highest number of user votes.

One can access to the API details by clicking here. The API provides a variety of features to get movies and TV Shows related data such as cast, synopsis, user votes, and country based statistics.

The steps taken during the preparation of this report can be summarized as follows:

- Receiving an API Key by following the 4 steps addressed here.
- Setting up an R Markdown file and preparing the environment by installing the related packages.
- Performing test calls to check whether the environment is working.
- Calling the base URL's and extending them with additional queries to obtain more specific results.
- The main problem with the code was the readability due to the length of the URL's. Thus, the URL's are decomposed into three parts (body, query strings, and api key) by using the base R function "gsub" along with a custom function called "extend_URL". This way, the code became more clean and also reusable with different API keys and queries.

```
# Installing the required R packages
# install.packages("httr")
# install.packages("jsonlite")

# Library imports

library("httr")
library("jsonlite")

# Setting the API key

api_key <- "5169379406b250c1d86d329ce1fcd1c9"

# Defining a function to concatenate URL's and query strings easily

extend_URL <- function(URL, query_string){

    result <- paste(URL, "&", query_string, sep ="")
    return(result)

}
```

2 API Call for Genres

First, the list of official genre labels used in the website should be retrieved.

```
# URL template for the "genre" call

genre <- "https://api.themoviedb.org/3/genre/movie/list?api_key=<<api_key>>"

# Setting the language

genre <- extend_URL(genre, "language=en-US")

# Inserting the API key to the template

genre <- gsub("<<api_key>>", api_key, genre)

# Performing the call

genre_call <- httr::GET(genre)

# Checking the status of the call

genre_call$status_code
```

```
## [1] 200
```

The status code implies that the call was successful. The next step is finding out the genre ID of "western".

3 API Call for Genres - Results

```
# Converting the call results to an easily readable list

genre_char <- base::rawToChar(genre_call$content)

genre_json <- jsonlite::fromJSON(genre_char, flatten = TRUE)

knitr::kable(genre_json$genres)
```

id	name
28	Action
12	Adventure
16	Animation
35	Comedy
80	Crime
99	Documentary
18	Drama
10751	Family
14	Fantasy
36	History
27	Horror
10402	Music
9648	Mystery
10749	Romance
878	Science Fiction
10770	TV Movie
53	Thriller
10752	War
37	Western

4 API Call for Movie Discovery

```
# URL template for the "discover" call

discover <- "https://api.themoviedb.org/3/discover/movie?api_key=<<api_key>>"

# Setting the language

discover <- extend_URL(discover, "language=en-US")

# Retrieve the movies with highest number of votes

discover <- extend_URL(discover, "sort_by=vote_count.desc")

# Filtering by genre (Western's genre id is 37)

discover <- extend_URL(discover, "with_genres=37")

# Filtering by original language (Italian)

discover <- extend_URL(discover, "with_original_language=it")

# Inserting the API key to the template

discover <- gsub("<<api_key>>", api_key, discover)

# Performing the call

discover_call <- httr::GET(discover)

# Checking the status of the call

discover_call$status_code
```

```
## [1] 200
```

The call was successful.

5 API Call for Movie Discovery - Results

```
# Converting the call results to an easily readable list

discover_char <- base::rawToChar(discover_call$content)

discover_json <- jsonlite::fromJSON(discover_char, flatten = TRUE)

results <- discover_json$results

# Checking the retrieved field names

colnames(results)
```

```
##  [1] "adult"             "backdrop_path"     "genre_ids"
##  [4] "id"                "original_language" "original_title"
##  [7] "overview"          "popularity"        "poster_path"
## [10] "release_date"      "title"             "video"
## [13] "vote_average"      "vote_count"
```

6 Final Results

Finally, a clean summary table consisting of basic movie information can be generated as follows:

```
table <- results[c("title", "release_date", "vote_average", "vote_count")]

knitr::kable(table)
```

title	release_date	vote_average	vote_count
The Good, the Bad and the Ugly	1966-12-23	8.5	7156
Once Upon a Time in the West	1968-12-21	8.3	3646
A Fistful of Dollars	1964-01-18	7.9	3540
For a Few Dollars More	1965-12-18	8.0	3333
They Call Me Trinity	1970-12-22	7.6	1110
Duck, You Sucker	1971-10-20	7.7	877
My Name Is Nobody	1973-12-13	7.3	792
Trinity Is Still My Name	1971-10-21	7.4	762
Django	1966-04-06	7.2	739
The Great Silence	1968-02-21	7.5	311
God Forgives... I Don't!	1967-10-31	6.6	211
Troublemakers	1994-11-25	6.1	189
Ace High	1968-10-02	6.5	166
Death Rides a Horse	1967-08-31	7.0	155
A Genius, Two Friends, and an Idiot	1975-12-16	6.4	133
The Big Gundown	1966-11-29	7.3	132
Boot Hill	1969-12-20	6.0	132
Zorro	1975-03-06	6.4	117
Day of Anger	1967-12-19	7.0	115
Keoma	1976-11-25	6.9	115