

R2D2Project

July 4, 2019

A web application where an user can search for a movie name and see information like movie name, rating , Votes etc.

```
In [1]: import urllib.request, urllib.parse, urllib.error
import json
```

```
In [ ]: conda install psycpg2
```

```
WARNING conda.base.context:use_only_tar_bz2(632): Conda is constrained to only using the old .tar.bz2 file format
Collecting package metadata (repodata.json): done
Solving environment: /
```

The above is for Package for postgresql

<p><h1>Most Viewed Movies</h1></p>

Get Secret API key from OMDb website and use that, 1000 daily limit from a JSON file, stored in the same folder

```
In [2]: with open('DFG.json') as f:
        keys = json.load(f)
        omdbapi = keys[0]['OMDBapi']
```

```
In [3]: serviceurl = 'http://www.omdbapi.com/?'
        apikey = '&apikey='+omdbapi
```

```
In [4]: !pip install psycpg2
```

Requirement already satisfied: psycpg2 in /home/jupyterlab/conda/lib/python3.6/site-packages (2.7.6.1)

Function for printing a JSON dataset

```
In [5]: def print_json(json_data):
    list_keys=['Title', 'Year', 'Rated', 'Released', 'Runtime', 'Genre', 'Director', 'Writer',
               'Actors', 'Plot', 'Language', 'Country', 'Awards', 'Ratings',
               'Metascore', 'imdbRating', 'imdbVotes', 'imdbID']
    print("-"*50)
    for k in list_keys:
        if k in list(json_data.keys()):
            print(f"{k}: {json_data[k]}")
    print("-"*50)
```

Function to create/update the local movie database with the data retrieved from the web Saves the movie data (Title, Year, Runtime, Country, Metascore, and IMDB rating) into a local SQLite database called 'movieinfo.sqlite'

```
In [6]: def save_in_database(json_data):

    filename = input("Please enter a name for the database (extension not needed, it will be added automa
    filename = filename+'.sqlite'
    import psycopg2 as pg
    import sqlite3
    conn = sqlite3.connect(str(filename))
    cur=conn.cursor()

    title = json_data['Title']

    if json_data['Year']!='N/A':
        year = int(json_data['Year'])
    if json_data['Runtime']!='N/A':
        runtime = int(json_data['Runtime'].split()[0])
    if json_data['Country']!='N/A':
        country = json_data['Country']
    if json_data['Metascore']!='N/A':
        metascore = float(json_data['Metascore'])
    else:
        metascore=-1
    if json_data['imdbRating']!='N/A':
        imdb_rating = float(json_data['imdbRating'])
    else:
        imdb_rating=-1

    cur.execute("""CREATE TABLE IF NOT EXISTS MovieInformation
    (Title TEXT, Year INTEGER, Runtime INTEGER, Country TEXT, Metascore REAL, IMDBRating R

    cur.execute('SELECT Title FROM MovieInformation WHERE Title = ? ', (title,))
    row = cur.fetchone()

    if row is None:
```

```

        cur.execute("INSERT INTO MovieInformation (Title, Year, Runtime, Country, Metascore, IMDBRa
                    VALUES (?, ?, ?, ?, ?, ?)", (title, year, runtime, country, metascore, imdb_rating))
    else:
        print("Record already found. No update made.")

    conn.commit()
    conn.close()

```

Function to print contents of the local database

In [7]: `def print_database(database):`

```

    import sqlite3
    conn = sqlite3.connect(str(database))
    cur=conn.cursor()

    for row in cur.execute('SELECT * FROM MovieInformation'):
        print(row)
    conn.close()

```

Function to save the database content in an Excel file

In [8]: `def save_in_excel(filename, database):`

```

    if filename.split('.')[-1]!='xls' and filename.split('.')[-1]!='xlsx':
        print ("Filename does not have correct extension. Please try again")
        return None

    import pandas as pd
    import sqlite3
    import psycopg2 as pg
    conn = sqlite3.connect(str(database))

    df=pd.read_sql_query("SELECT * FROM MovieInformation", conn)
    conn.close()

    df.to_excel(filename,sheet_name='Movie Information')

```

Function to search for information about a movie

In [9]: `def search_movie(title):`

```

    if len(title) < 1 or title=='quit':
        print("Goodbye now...")
        return None

    try:
        url = serviceurl + urllib.parse.urlencode({'t': title})+apikey
        print(f'Retrieving the data of "{title}" now... ')

```

```

uh = urllib.request.urlopen(url)
data = uh.read()
json_data=json.loads(data)

if json_data['Response']=='True':
    print_json(json_data)
    # If you want to save the movie information in a local database then type yes otherwise no
    save_database_yes_no=input ('To Save the movie info in a local database? Enter "yes" or "no":')
    if save_database_yes_no=='yes':
        save_in_database(json_data)
else:
    print("Error encountered: ",json_data['Error'])

except urllib.error.URLError as e:
    print(f"ERROR: {e.reason}")

```

Searching a Desired Movie

```

In [10]: title = input('\nEnter the name of a movie (enter \'quit\' or hit ENTER to quit): ')
if len(title) < 1 or title=='quit':
    print("Ended...")
else:
    search_movie(title)

```

Enter the name of a movie (enter 'quit' or hit ENTER to quit): Avengers

Retrieving the data of "Avengers" now...

Title: The Avengers

Year: 2012

Rated: PG-13

Released: 04 May 2012

Runtime: 143 min

Genre: Action, Adventure, Sci-Fi

Director: Joss Whedon

Writer: Joss Whedon (screenplay), Zak Penn (story), Joss Whedon (story)

Actors: Robert Downey Jr., Chris Evans, Mark Ruffalo, Chris Hemsworth

Plot: Earth's mightiest heroes must come together and learn to fight as a team if they are going to stop the misch

Language: English, Russian, Hindi

Country: USA

Awards: Nominated for 1 Oscar. Another 38 wins & 79 nominations.

Ratings: [{'Source': 'Internet Movie Database', 'Value': '8.1/10'}, {'Source': 'Rotten Tomatoes', 'Value': '92%'}, {'

Metascore: 69

imdbRating: 8.1

imdbVotes: 1,186,132

imdbID: tt0848228

To Save the movie info in a local database? Enter "yes" or "no": yes
Please enter a name for the database (extension not needed, it will be added automatically): movies

Record already found. No update made.

Excel file

```
In [11]: print_database('movies.sqlite')
```

```
('The Avengers', 2012, 143, 'USA', 69.0, 8.1)
```

```
In [12]: save_in_excel('test.xlsx', 'movies.sqlite')
```

```
In [13]: import pandas as pd  
df=pd.read_excel('test.xlsx')  
df
```

```
Out[13]: Unnamed: 0      Title  Year  Runtime  Country  Metascore  IMDBRating  
0         0  The Avengers  2012     143     USA         69         8.1
```