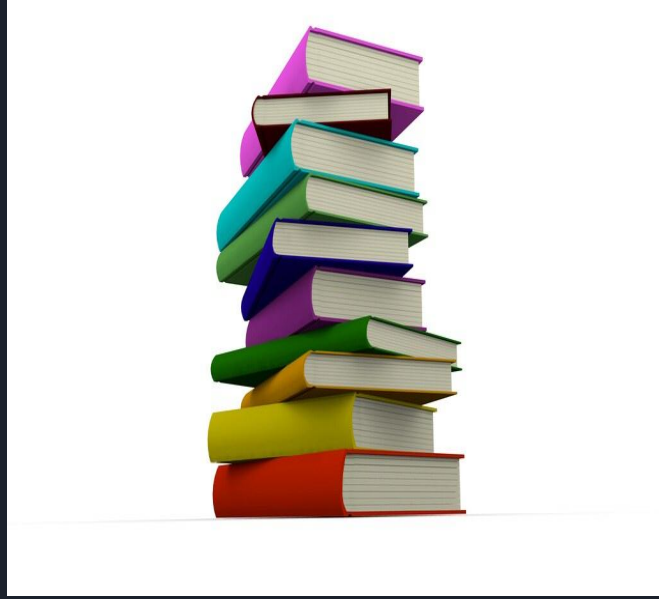
New York Times Books API ETL Pipeline Project



202: Object Oriented Programming
Final Project by Gausnary Decius

Project Description

Goal:
The “question” I want my ETL Pipeline to answer is “what are the current 15 best selling books in the United States?” The goal of my project is to create a list of the current New York Times Best selling list and compile it into a file that can later be used as a list that holds the light synopsis of each entry so that readers can skim to find current reading suggestions based on the most popular literature out. The endpoint I will be using is in the “Books Api” on developers.nytimes.com/apis. The actual endpoint I will be modifying in my ETL Pipeline is https://api.nytimes.com/svc/books/v3/lists/{date}/{list}.json. This specific endpoint holds all the data I need for my ETL project, however half of the information given by the API is not needed for this ETL pipeline . To transform the data I will see what key value pairs the api will generate and keep the key value pairs that I think are relevant to this pipeline.

The point of this project is to have the information on the list easily accessible and in a ready to use format in a database so that the information can be copy and pasted and posted on social media. In real world scenarios this would be perfect for blogs or social media influencers that have a big audience in the reading community and constantly have to post the updated list of the current best sellers list.

Project Description

Continued:

- Data Storage:
  I will store the data in a text file that will allow for the making spreading and spreading of information in this ETL Pipeline Project easier after it is processed. I will create a folder that holds the new data file in it that can later be added onto to hold each additional week's list that the code could make in one central spot, while also making a backup folder that will hold a copy of the best seller list data file as way. This is just a safety precaution just in case anything happens to my original copy.

The key values I felt were relevant to the pipeline were:

- "**rank**" >> the rank of the **book** this week
- "**description**" >> description of the book
- "**price**" >> price of the book
- "**title**" >> title of the book
- "**author**" >> author of the book
- "**contributor**" >> contributors of the book
- "**buy links**" >> links to buy the book

Code-Image: Endpoint Retrieval

- In this code image is the url of the endpoint I used, with the following code to be able to access it.
- Because the endpoint was a sequence of nested lists surrounding the information I needed, I created variables to hold the the values of the dict I needed and be able to quickly access and modify them at a later time
- Many of the variable used in this code image specifically were created for ease of access if I needed to make modifications in the future, making the big data from the api more manageable
- In this portion of the code I also created the text file that would hold my API data

```
url = 'https://api.nytimes.com/svc/books/v3/lists/current/hardcover-fiction.json' # url of nyt best
sellers list endpoint
API_KEY = 'PtvCMGfmR4QeiGGsiRsTuS3TNwjZGRtX' # api key
params = {'api-key' : API_KEY} # paramater variable to hold dict of my api key fot get method
response = requests.get(url,params=params) #response variable holding api information
seller_list= response.json() # seller_list variable that is the usable .json format
api_text = nyt_best_seller_list/"API.txt" # creating path for api text directory
api_text.touch() # creating api text file

books = seller_list['results'] # trimming endpoint information the values the pipeline needs for easier
use
ranks = books['books'] # trimming endpoint information the values the pipeline needs for easier use
```

Code-Image: Key Value Loop

- This is the loop I used to traverse the dictionary of information for the key and value pairs I needed to have useful readable human information.
- This dict held 15 entries so I made a nested for loop with a range of 15, so it would capture each list entry. This is also why I set n = ranks[i] so that as the inside loop is completed the outer loop runs again.
- The if statement is used to only drag the key components from the ranks dict that I needed to have human readable information
- The code is also where I append each human readable part of the endpoint to the text file that is holding the data data

```
for i in range(15): # for loop traversing the list of books
    text = open(api_text, 'a') # opening text file to be written to
    n = ranks[i]
    for key in n:
        if key in ('rank', 'description', 'price',
                   'title', 'author','contributor', 'buy_links'): # key values from end point pipeline
needs


            text.write(f'{key.upper()}: ') # writing key value in upper case
            text.write(f'{n[key]} \n') # writing the corresponging value of the key
        print('\n')
text.close() # closing text file
```

Code-Image: Backups

```
nyt_best_seller_list_backup = path/"NYT Best Seller List Backup" #creating file_backups directory
nyt_best_seller_list_backup.mkdir() # creatiing file_backups folder

api_text_backup = nyt_best_seller_list_backup/"API Backup.txt" # creating api text back up directory
shutil.copy( path/api_text , api_text_backup ) # copying api text back up file
```

- In this portion of code I create a backup folder and file copy of the exported data for safekeeping. I felt a project like this would not be complete without copies of the data your pipeline gathered and transformed

Project Summary & Potential Expansion

The “question” I set out to answer with my ETL Pipeline Projects was “what are the current 15 best selling books in the United States?”

The purpose of this project was to be able to take the best sellers lists services endpoint of the New York Times API to create a useful list out of the information on this webpage. To create this ETL Pipeline I chose the relevant components under the NYT API *Books API*, utilizing their Best Sellers Lists Services endpoint held what I thought humans would be able to use for book suggestions. The hope of the Pipeline was to put these components in a useful and readable language that could be easily accessible in text document form for later use. In its current form the information can be a useful archive for the weekly updates of the list. With the use of the NYT API and this specific endpoint I believe my project to be a success for the potential options it has set up for the future going forward. Now with a simple copy and paste I can have people updated on the 15 best selling books in the US and synopses of each entry.

Future Potential Project Expansion:

- ➔ A pipeline that allows users to search APIs for the names of authors on this list for other literature to read
- ➔ A Pipeline that compares the authors on this list to shared articles, cross referencing to see who is talking about the piece of literature for future analysis