

# SI 507 Final Project Checkpoint

## 1. Project code

[https://github.com/Phantom-eva/SI507\\_Final\\_Project](https://github.com/Phantom-eva/SI507_Final_Project)

## 2. Data Sources

(1) The Movie Database:

URL: <https://www.themoviedb.org/documentation/api>

Format: JSON

Type: Web API that requires an API key

Cache usage: yes

Summary of data: more than 700,000 movies available, 500 movies retrieved (estimate)

Description of records: movie list in different movie genres (action, comedy, crime, etc.)

Important attributes: genres (a filter), adult (a filter), imdb\_id (The ID of a movie in different movie databases), runtime (a filter)

(2) The Open Movie Database:

URL: <https://www.omdbapi.com/>

Format: JSON

Type: Web API that requires an API key

Cache usage: yes

Summary of data: more than 700,000 movies available, 500 movies retrieved (estimate)

Description of records: movie details containing ratings from different sources, rated, websites, etc.

Important attributes: Rated (a filter), Website (a link to watch this movie), imdb\_id (The ID of a movie in different movie databases), Ratings [Rotten Tomatoes, Metacritic] (viewers' comments on this movie)

(3) Google Showtimes Results:

URL: <https://serpapi.com/showtimes-results>

Format: JSON

Type: Web API that requires an API key

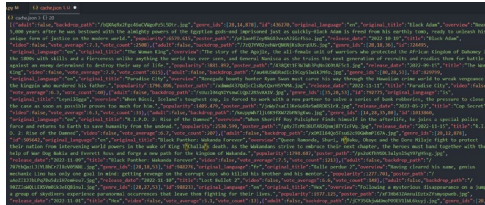
Cache usage: no

Summary of data: about 100 records available, about 100 records retrieved (estimate)

Description of records: Information about movies currently playing in local theaters

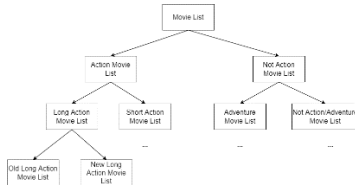
Important attributes: name (movie title), link (where to buy tickets), address (where to watch), showing time (when to watch)

I get the list of movies in different genres from the first database and get the ratings and other information about the movie in the second database through the same IMDB id. Also, I get information about movies currently playing in local theaters from the third database.



### 3. Data Structure

Movie data tree: Filter the raw data of the movie to reduce the total size of data, and organize all data obtained into a tree.



I get the list of movies in different genres from the first database. Then I judge the data based on the some conditions to get the next branches.

Question tree:

Which question is asked first? And what is the sequence of the questions?

```
class treeNode:
    def __init__(self, data):
        self.left = None
        self.right = None
        self.data = data
    def PrintTree(self):
        print(self.data)

def construct_movie_tree():
    movienode_1 = treeNode(Movie_list['all'])
    movienode_2 = treeNode(Movie_list['Action'])
    movienode_3 = treeNode(Movie_list['not Action'])
    movienode_4 = treeNode(Movie_list['Adventure'])
    movienode_1.left = movienode_2
    movienode_1.right = movienode_3
```

```
# construct question tree
def construct_question_tree():
    question_1 = treeNode("Do you want to watch an action movie?")
    question_2 = treeNode("Do you want to watch an adventure movie?")
    question_3 = treeNode("Do you want to watch an comedy movie?")
    question_4 = treeNode("Do you want to watch an horror movie?")
    question_5 = treeNode("Do you want to watch an science fiction movie?")
    question_6 = treeNode("Do you want to watch a movie longer than 120 minutes?")
    question_7 = treeNode("Do you want to watch a movie released after 2010?")
    question_1.right = question_2
    question_1.left = question_3
    question_2.right = question_4
    question_2.left = question_5
    question_3.right = question_6
    question_3.left = question_7
    question_4.right = question_5
    question_4.left = question_6
    question_5.right = question_6
    question_5.left = question_7
```

### 4. Interaction and Presentation Plans

Interaction question 1: Do you want to search for movies that are playing at local theaters or find movies by filters?

Yes: search for movies that are playing at local theaters

No: find movies by filters

Interaction question 2: Do you want to load movie data from the cache or update movie data?

Yes: load from cache

No: update data through API

Interaction question 3: Do you want a/an action/comedy/crime movie? Do you want a movie longer than 120 minutes? Do you want a movie released after 2018?...

Filter question.

After getting the filtered movie list, the user has the option to sort the movie list according to the rating or not.

Interactive and presentation technologies you plan to use: command line prompts, and Flask if time allows.