Movie Inspire Project Proposal

Jung Eun Choi & Sharon Wong

Introduction

The Movie Inspire Application is a quick movie search web application to inspire users on what movie to watch. We chose to build a web application because it can be accessed through users' desktop or mobile devices and it is where people generally go to when searching for movies. Specifically, this application shows users a list of movies based on their preferences. The filters include the genre, year released, director, cast members (all optional but at least one input is required). The App especially helps indecisive users to narrow down their options. They can find an adequate amount of movie information on one single platform. The intended audience is those who do not have a clear movie in mind. Nevertheless, this application also considers those who have an idea of the movie they plan to watch; all they need to do is indicate the movie title of their interest, which then provides further information about that particular movie.

The reason why this application adds value is because users can find a movie to watch, based on their preferences, on one platform. We will provide the most relevant information about each movie to narrow their choices down. This way, users do not have to perform further searches about the movies on external sites and open multiple tabs.

Data Source and Planned Processing

Our project will use three APIs: (the underlined APIs are linked to the document pages)

Although the Movie Database API has sufficient movie information that we need, we want to make the resource more diverse to avoid biased information from a fixed group of people (reviewer, etc.). That's why we chose to get certain information from the OMDb API and the NYT Movie Review API.

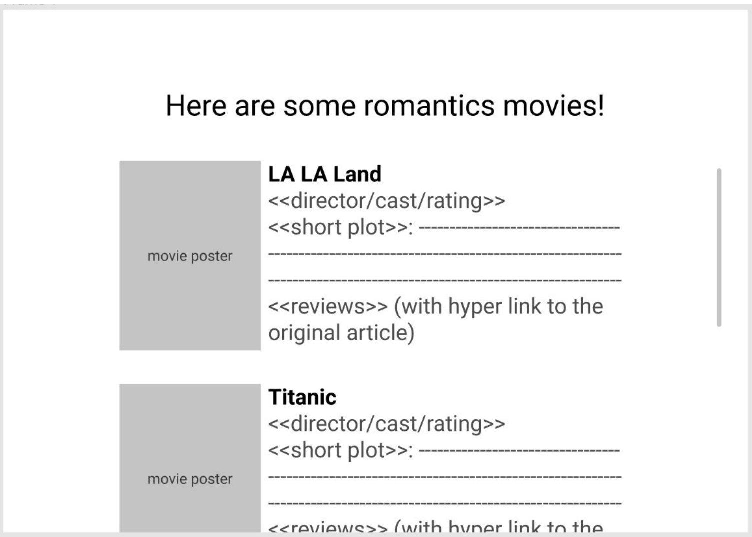
1. The Movie Database API
   a. Parameters: genre id
   b. Output: movie information (movie title)
2. The OMDb API
   a. Parameters: movie title, "short"
   b. Output: movie poster, plot description
3. The New York Times Movie Review API
   a. Parameters: movie title
   b. Output: movie review (with author and link to the review)

Full Description

The Movie Database API lets users discover movies by filtering through different types of data corresponding to each movie, such as genre and rating. For our project, we will use it to search for a list of movies mainly by the genre that is specified by the users. We will then send the movie titles to the OMDb API and the New York Times Movie Review APIs, as a keyword, to get the movie information and reviews respectively. OMDb is a RESTful web service that provides users with movie information, given a movie title. We will specifically get the movie poster and short plot description from OMDb API. The New York Times Movie Review API plays a similar role, where information of a certain movie is provided based on the data request. With the data given from this API, we will get the movie reviews, with additional information about the author and the url link to the article review.

Below is a sample scenario:

If a user would like to look for "romantic" movies, we will send this keyword to the Movie Database API to get a list of romantic movie titles. Then, we use this list (or one random movie from the list) as a parameter, to request the information we want from the OMDb API and the New York Times Movie Review APIs. Figure 1 below is a low fidelity prototype of what our HTML output may look like.



Here are some romantics movies!

LA LA Land
<<director/cast/rating>>
<<short plot>>: --------------------------------

<<reviews>> (with hyper link to the original article)

Titanic
<<director/cast/rating>>
<<short plot>>: --------------------------------

<<reviews>> (with hyper link to the

Figure 1. Low fidelity prototype of our HTML output.

Debrief on Our Output Format

The filtered search the user inputs will determine the displayed movies. Particularly, the webpage will illustrate the movie titles with its genre, release year, IMBd rating, poster image, short plot description, and an external link to a movie review article (the output of The New York Times

Movie Review API). These results are basic, relevant information that the users need to figure out which movies they are most interested in. We believe that the external movie review link will help validate their decision for determining whether to watch the movie associated with it. This information is also provided on one platform—our web application—to make it more quick and convenient to search for movies of users' interests without going through the trouble of opening multiple tabs or searching on different web platforms.

Our Tentative Development Plan

Stage One

- Create a GitHub repository so we can collaborate together to create our web application. Make a basic Python application that reads our APIs.
    - Put the API keys in a .gitignore file so others do not have access to our API keys.
    - Handle any errors due to HTTP or connection-related exceptions
- Learn about parameters, format, and basic outputs (i.e. the database outputs) of the three APIs.

Stage Two

- Create our main functions to get the movie information that we need.
    - Work on Movie Database API: when users input a genre, a function returns a list of movies within that genre.
    - Work on New York Times Movie Review API and OMdb API: functions that take in the movie title from Movie Database API, and return the reviews (including author and link), plots and posters.

Stage Three

- Display the application in the HTML format.
    - Work on templates or explore more options

If Time Permits

- Beautify our web application. Make it more visually appealing.
    - Use CSS.
- Randomize our movie results every time the user refreshes the page.
- Add extra web pages to display the movies.

Appendix

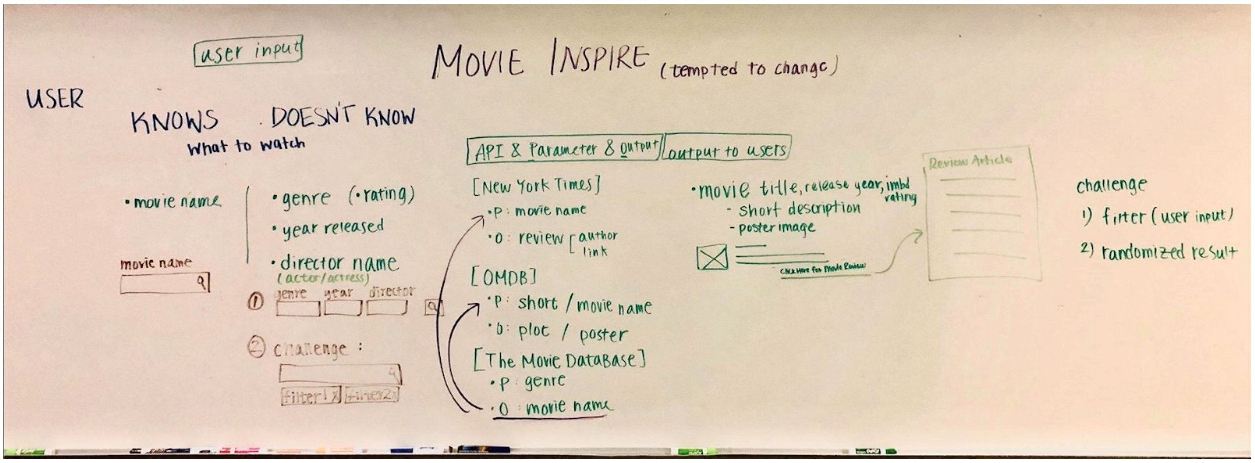
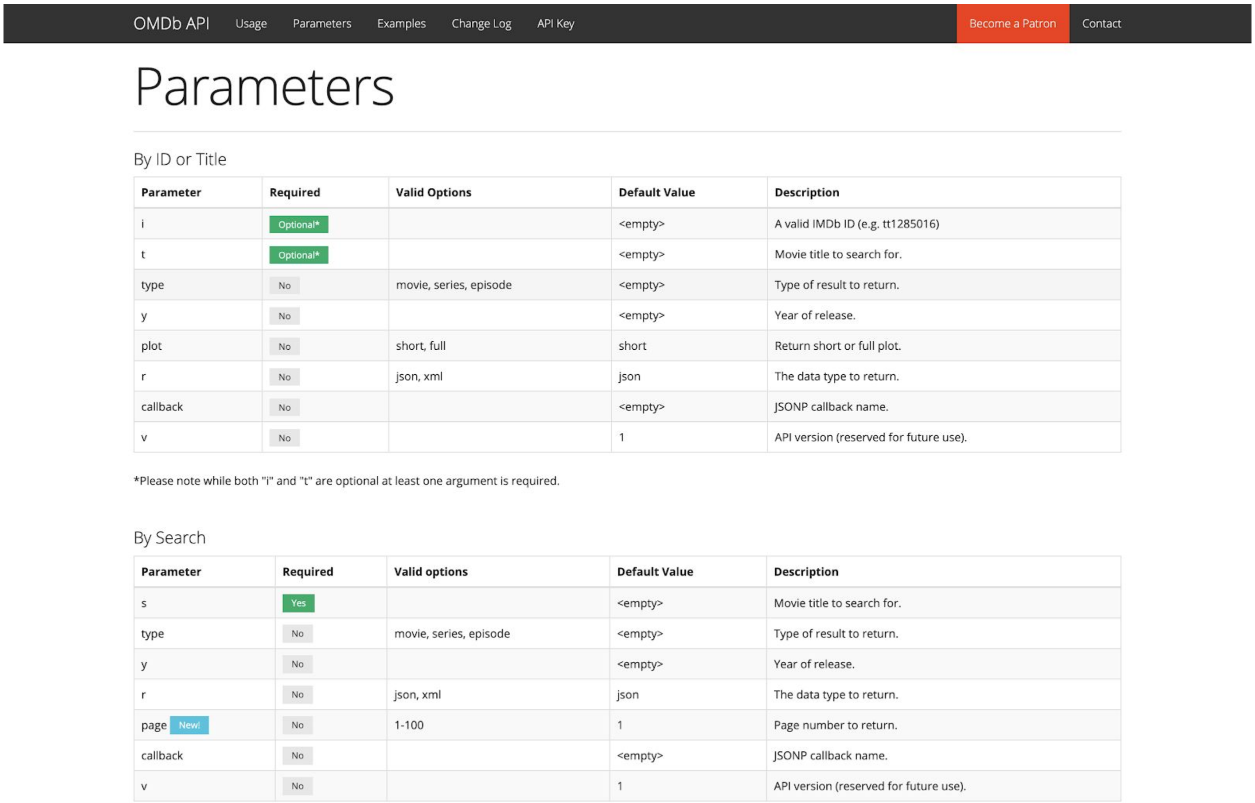


Figure 2. White Board Brainstorming



Parameters

By ID or Title

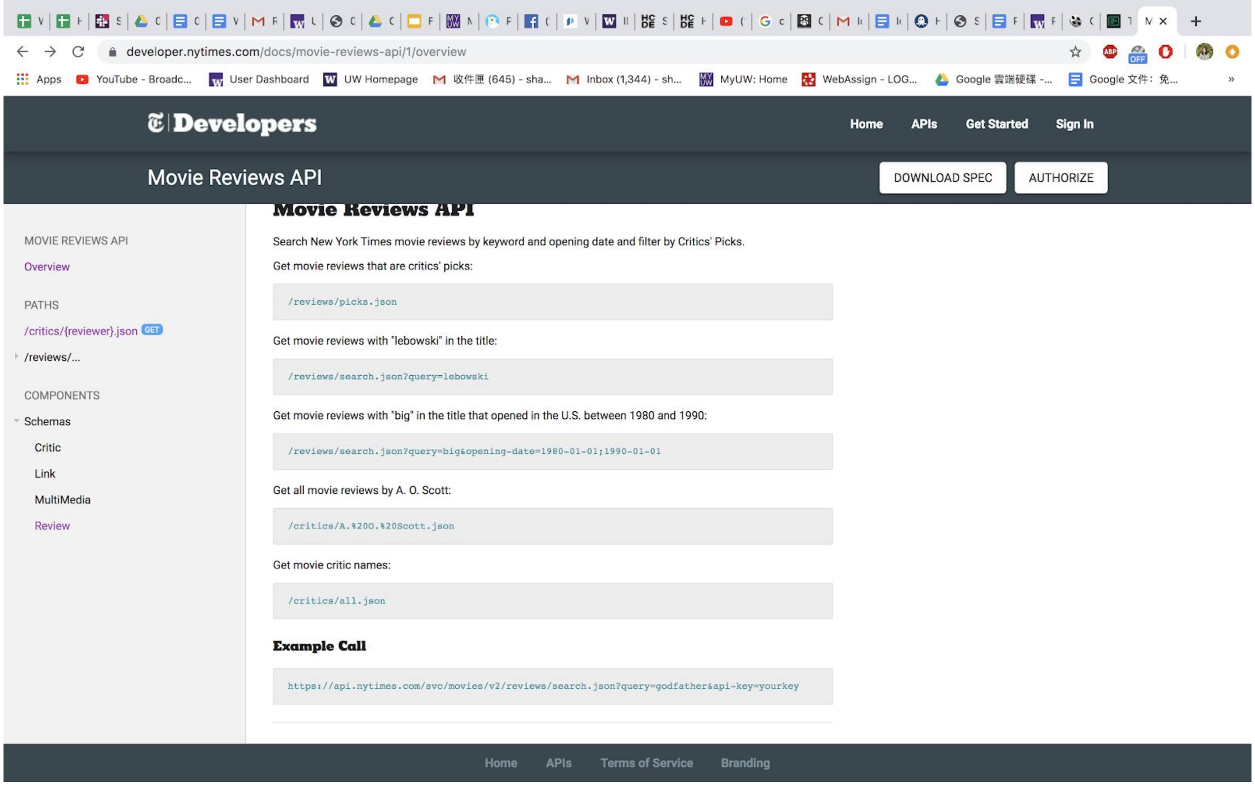
Parameter	Required	Valid Options	Default Value	Description
i	Optional*		<empty>	A valid IMDb ID (e.g. tt1285016)
t	Optional*		<empty>	Movie title to search for.
type	No	movie, series, episode	<empty>	Type of result to return.
y	No		<empty>	Year of release.
plot	No	short, full	short	Return short or full plot.
r	No	json, xml	json	The data type to return.
callback	No		<empty>	JSONP callback name.
v	No		1	API version (reserved for future use).

*Please note while both "i" and "t" are optional at least one argument is required.

By Search

Parameter	Required	Valid options	Default Value	Description
s	Yes		<empty>	Movie title to search for.
type	No	movie, series, episode	<empty>	Type of result to return.
y	No		<empty>	Year of release.
r	No	json, xml	json	The data type to return.
page New!	No	1-100	1	Page number to return.
callback	No		<empty>	JSONP callback name.
v	No		1	API version (reserved for future use).

Figure 3. OMDb parameter types. Use OMDb to get the movie title, year of release, poster, and basic movie descriptions.



Movie Reviews API

Search New York Times movie reviews by keyword and opening date and filter by Critics' Picks.

Get movie reviews that are critics' picks:

```
/reviews/picks.json
```

Get movie reviews with "lebowski" in the title:

```
/reviews/search.json?query=lebowski
```

Get movie reviews with "big" in the title that opened in the U.S. between 1980 and 1990:

```
/reviews/search.json?query=big&opening-date=1980-01-01;1990-01-01
```

Get all movie reviews by A. O. Scott:

```
/critics/A.%20O.%20Scott.json
```

Get movie critic names:

```
/critics/all.json
```

Example Call

```
https://api.nytimes.com/svc/movies/v2/reviews/search.json?query=godfather&api-key=yourkey
```

Figure 4. New York Times Review API (parameters: movie title, opening dates in the U.S., API key)

Genre List (and their ID)

Send request to get a list of genre:
https://api.themoviedb.org/3/genre/movie/list?api_key=<<mykey>>&language=en-US
Genre id as parameter to get the movie /discover/movie?with_genres=35
https://developers.themoviedb.org/3/discover/movie-discover

Example Genre List in Dictionary:

```
{
 "genres": [
  {
   "id": 28,
   "name": "Action"
  },
  {
   "id": 12,
   "name": "Adventure"
  },
.....
 ]
}
```

"id": 18, "name": "Drama" "id": 10751, "name": "Family" "id": 14, "name": "Fantasy" "id": 36, "name": "History" "id": 27, "name": "Horror"	"id": 10402, "name": "Music" "id": 9648, "name": "Mystery" "id": 10749, "name": "Romance" "id": 878, "name": "Science Fiction" "id": 10770, "name": "TV Movie"	"id": 53, "name": "Thriller" "id": 10752, "name": "War" "id": 37, "name": "Western"