

API specification

Songyang Du, Shengyin Si, Yahan Xu, Yu Yang

April 2023

1. Route: /movies

- description: return all movies, ordered by year
- route parameters: None
- query parameters: None
- route handler: movies(req, res)
- return type: JSON Array
- return parameters: `{imdb_title_id (string), title (string), original_title (string), year (int), genre (string), duration (int), country (string), language (string), director (string), description (string), avg_vote (float), votes (int), poster_url (string), Oscar_nominated (boolean)}`

2. Route: /movies/:movie_id

- description: return return all the information about a movie with given imdb_title_id
- route parameters: movie_id(string)
- query parameters: None
- route handler: movies(req, res)
- return type: JSON Array
- return parameters: `{imdb_name_id (string), name (string), birth_name (string), date_of_birth (string), place_of_birth (string), date_of_death (string), height float, bio (string) photo_url (string)}, ...}`

3. Route: /search_movies

- description: return all movies with given criteria
- route parameters: None
- query parameters: title(string), yearlow(int), yearhigh(int), country(string), language(string), genre(string), isOscard(boolean), page(int), pageSize(int)(default: 10)
- route handler: search_movies(req, res)

- return type: JSON Array
 - return parameters: `{imdb_title_id (string), title (string), original_title (string), year (int), genre (string), duration (int), country (string), language (string), director (string), description (string), avg_vote (float), votes (int), poster_url (string), Oscar_nominated (boolean)}, ...}`
4. Route: `/movie_people/:movie_id`
- description: return a list actors in the given movies
 - route parameters: `movie_id`
 - query parameters: `page(int)`, `pageSize(int)(default: 10)`
 - route handler: `movie_people(req, res)`
 - return type: JSON Array
5. Route: `/people/:person_id`
- description: return all movies with given criteria
 - route parameters: `person_id`
 - query parameters: None
 - route handler: `people(req, res)`
 - return type: JSON Array
 - return parameters `{imdb_name_id (string), name (string), birth_name (string), date_of_birth (string), place_of_birth (string), date_of_death (string), height (float), bio (string), photo_url (string)}`
6. Route: `/people`
- description: return all movies with given criteria
 - route parameters: None
 - query parameters: `page(int)`, `pageSize(int)(default: 10)`
 - route handler: `people(req, res)`
 - return type: JSON Array
 - return parameters `{imdb_name_id (string), name (string), birth_name (string), date_of_birth (string), place_of_birth (string), date_of_death (string), height (float), bio (string), photo_url (string)}`
7. Route: `/search_people`
- description: return all movies with given criteria
 - route parameters: None
 - query parameters: `name(string)`, `page(int)`, `pageSize(int)(default: 10)`
 - route handler: `search_people(req, res)`

- return type: JSON Array
- return parameters {imdb_name_id (string), name (string), birth_name (string), date_of_birth (string), place_of_birth (string), date_of_death (string), height (float), bio (string), photo_url (string)}

8. Route 1: /top10_rated_oscar_movies

- description: return the top 10 highly rated Oscar movies
- route parameters: None
- query parameters: None
- route handler: top10_rated_oscar_movies(req, res)
- route parameters: {year_film(int), year_ceremony(int), ceremony(int), category(string), name(string), winner(boolean), imdb_title_id (string), movie_title (string)}
- return type: JSON Array

9. Route: /related_actors/:name

- description: a co-actor of a given actor X is an actor who has acted in the same movie as X. This query will return the name and id of all actors who are either: co-actors of X, or co-actors of some actor who is a co-actor of X, or co-actors of some actor who is a co-actor of some actor who is a co-actor of X. Do not include X in the result, and interpret “actor” as gender neutral.
- route parameters: name
- query parameters: page(int), pageSize(int)(default: 10)
- route handler: related_actors(req,res)
- return parameters: {name(string), id(string)}
- return type: JSON Array

10. Route: /average_vote_person/:person_id

- description: return the average vote of the given actor’s movies.
- route parameters: person_id
- query parameters: None
- route handler: average_vote_person(req, res)
- return parameters{avg_rating(float)}
- return type: JSON Object

11. Route: /oscar_movies

- description: return all movies with given criteria
- route parameters: None

- query parameters: page(int), pageSize(int)(default: 10)
 - route handler: search_movies(req, res)
 - return type: JSON Array
12. Route: /search_oscar_winner
- description: return all movies with given criteria
 - route parameters: None
 - query parameters: genre(string)
 - route handler: search_oscar_winner(req, res)
 - return type: JSON Array
13. Route: /top20_movies
- description: return all movies with given criteria
 - route parameters: None
 - query parameters: genre(string)
 - route handler: search_movies(req, res)
 - return type: JSON Array
14. Route: /oscar_recommend
- description: return a random Oscar winning movie with the movie title, release year, duration, poster, description, name of the director, and number of oscar nominations
 - route parameters: None
 - query parameters: None
 - route handler: oscarMovieRecommended(req,res)
 - return type: JSON Array
 - return Parameters:{title (string), year (int), duration (int), director_name (string), description (string), poster_url (string), num_nominations (int)}
15. Route: /recent10genre/:genre
- description: return the most recent 10 movies with their title, poster and description in the given genre
 - route parameters: genre(string)
 - query parameters: None
 - route handler: recent10genre(req,res)
 - return type: JSON Array
 - return Parameters:{title (string), description (string), poster_url (string)}

16. Route: `/top10language/:language`
 - description: return the top 10 rated movies with their title, poster and description in the given language
 - route parameters: `language(string)`
 - query parameters: `None`
 - route handler: `top10language(req,res)`
 - return type: `JSON Array`
 - return Parameters: `{title (string), description (string), poster_url (string)}`
17. Route: `/search_oscar_people`
 - description: all actresses in the best picture nominated movies
 - route parameters: `None`
 - query parameters: `page(int), pageSize(int)(default: 10)`
 - route handler: `search_oscar_people(req, res)`
 - return type: `JSON Array`
 - page: `people`
18. Route: `/search_oscar_filter`
 - description: return all movies, ordered by year
 - route parameters: `None`
 - query parameters: `title(string), language(string), durationHigh(int), durationLow(int)`
 - route handler: `search_oscar_filter(req, res)`
 - return type: `JSON Array`
 - page: `Oscar page`
19. Route: `/search_won`
 - description: return a list of movie with specific language/country/year etc and won any Oscar awards
 - route parameters: `None`
 - query parameters: `None`
 - route handler: `search_won(req, res)`
 - return type: `JSON Array`
 - page: `Oscar page`
20. Route: `/search_year/:year`
 - description: return all movies's title/country/language that were released in a particular year X

- route parameters: year(int)
- query parameters: page(int), pageSize(int)(default: 10)
- route handler: search_year(req, res)
- return type: JSON Array
- page: Oscar page

21. Route: /topDirector

- description: Find the top 10 directors who have directed the most movies that were nominated for an Oscar, along with the number of nominations their movies have received, and the number of movies they directed overall
- route parameters: None
- query parameters: None
- route handler: top_oscar_director(req, res)
- return type: JSON Array
- page: Oscar page

22. Route: /stats

- description: Find the actors who have appeared in the most movies that have been nominated for an Oscar in each decade, along with the number of nominations their movies have received, and the decade in which the movies were released
- route parameters: None
- query parameters: page(int), pageSize(int)(default: 10)
- route handler: oscar_decade(req, res)
- return type: JSON Array
- page: Oscar stats (static page)