

Movies API Documentation

Base URL

<https://ancient-caverns-16784.herokuapp.com/>

Routes

Get All Movies

Auth: no auth needed

Method: GET

Path: /movies[?field=value][&take=t&skip=s]

Pagination

The results are paginated. By default, the first 10 results in the database are returned. Two query parameters are used to define: the number of results: *take* and the starting position of the array of movies: *skip*. They can be used to override the defaults (i.e., *take=10&skip=0*).

The response has the following format:

```
{
  pagination: {
    numberOfPages: 3,  ---> the total number of pages, computed with respect
                        to the take parameter sent by the client
    currentPage: 2,    ---> the current page, computed with respect to the
                        take and skip parameters sent by the client
    links: {
      self: originalUrl&take=t&skip=s,  ---> the link used to retrieve the items
                                         on the current page
      prev: originalUrl&take=t&skip=s1, ---> the link used to retrieve the items
                                         on the previous page
                                         (null for the first page)
      next: originalUrl&take=t&skip=s2, ---> the link used to retrieve the items
                                         on the next page
                                         (null for last page)
    }
  },
  results: [{...}, ... , {...}]  ---> the resulting array of movies
}
```

Search

The same route can be used to perform a search. The query string can contain any number of searchable* fields. The movies that fulfill **all** conditions will be returned.

The searchable fields for a movie are:

- Title
- Year
- Runtime
- Genre
- Language
- Country
- Poster
- imdbRating
- imdbVotes
- imdbID
- Type

For “Title” and “Genre”, the results will be retrieved if the movie title **contains** the value specified in the query. Some Regular Expressions are also allowed, and the search is case insensitive.

Examples:

- /movies - returns the first 10 movies
- /movies?Title=a - returns the first 10 movies that contain ‘a’ anywhere in the Title
- /movies?Title=^a - returns the first 10 movies whose Title starts with ‘a’ or ‘A’
- /movies?Title=a\$ - returns the first 10 movies whose Title ends in ‘a’ or ‘A’
- /movies?Title=^a&Year=2000 - returns the first 10 movies whose Title starts with ‘a’ or ‘A’ and have Year=2000
- /movies?Title=^a&Genre=Action - returns the first 10 movies whose Title starts with ‘a’ or ‘A’ and are action movies
- /movies?Title=^a&Genre=Action&take=5&skip=10 - returns five movies from indexed 11-15 whose Title starts with ‘a’ or ‘A’ and are action movies

Response:

- 200 - the array of movies
- 400 - an error stating that “Field’ does not exist for a movie or is not a searchable field’ or ‘An invalid regular expression was supplied”

Add Movie

Auth:required

Headers:

X-Auth-Token: <accessToken received in login/register response>

Method: POST

Path: /movies

Payload:{

"Title": "Angel-A",

"Year": "2005",

"imdbID": "tt0473753",

"Type": "movie",

"Poster":

"https://images-na.ssl-images-amazon.com/images/M/MV5BMTIwNTQ2OTgxOV5BMTI5BanBnXkFtZTcwODAwNDI0MQ@@._V1_SX300.jpg"

}

Response:

- 200 - the newly inserted movie object
- 403 - {message: "You need to be authenticated to be able to create a movie"}

Get Movie

Auth: no auth needed

Method: GET

Path: /movies/:id

Response:

- 200 - movie object

Edit Movie

Auth:required

Headers:

x-auth-token:<accessToken received in login/register response>

Method:PUT

Path: /movies/:id

Payload: {Year: 2009}

Response:

- 200 - the updated movie object
- 403 - {message: "You need to be authenticated to be able to update a movie"}
- 400 - {'message': 'Nothing to update'}

DELETE Movie

Auth:required

Headers:

x-auth-token:<accessToken received in login/register response>

Method: DELETE

Path: /movies/:id

Response:

- 200 - 'The movie with id {id} was deleted'
- 403 - {message: "You need to be authenticated to be able to delete a movie"}

Populate Movies Database

Auth: no auth needed

Method: POST

Path: /movies/all

Response:

- 200 - an array with the movies inserted in the database

Login

Auth: no auth needed

Method:POST

Path: /auth/login

Payload: {

username: test,
password: test

}

Response:

- 200 - {
"authenticated": true,
"accessToken": "5nyQ3lmfwzsbujW7vJPr2qY3mPJ-3Rs"
}
- 401 - {message: "user not found/wrong password"}

Register

Auth: no auth needed

Method:POST

Path: /auth/register

Payload: {
 username: test,
 password: test
}

Response:

- 200 - {
 "authenticated": true,
 "accessToken": "5nyQ3ImfwzsbufjW7vJPr2qY3mPJ-3Rs"
}
- 409 - {message: "Username already existing"}

Log out

Auth:required

Headers:

 x-auth-token:<accessToken received in login/register response>

Method: GET

Path: /auth/logout

Response:

- 200 - {
 message: "User logged out successfully"
}
- 403 - {message: "You have to be logged-in in order to log out"}

Data

Movie Details Example

```
{
  "Title":"Angel-A",
  "Year":"2005",
  "Rated":"R",
  "Released":"21 Dec 2005",
  "Runtime":"91 min",
  "Genre":"Comedy, Drama, Fantasy",
  "Director":"Luc Besson",
  "Writer":"Luc Besson",
  "Actors":"Jamel Debbouze, Rie Rasmussen, Gilbert Melki, Serge Riaboukine",
  "Plot":"A beautiful woman helps an inept scam artist get his game together.",
  "Language":"French, Spanish",
  "Country":"France",
  "Awards":"N/A",

  "Poster":"https://images-na.ssl-images-amazon.com/images/M/MV5BMTIwNTQ2OTgxOV5BMl5BanBnXkFtZTcwODAwNDI0MQ@@._V1_SX300.jpg",
  "Ratings":[
    {
      "Source":"Internet Movie Database",
      "Value":"7.1/10"
    },
    {
      "Source":"Rotten Tomatoes",
      "Value":"44%"
    },
    {
      "Source":"Metacritic",
      "Value":"48/100"
    }
  ],
  "Metascore":"48",
  "imdbRating":"7.1",
  "imdbVotes":"27,750",
  "imdbID":"tt0473753",
  "Type":"movie",
  "DVD":"20 Nov 2007",
```

```
"BoxOffice": "N/A",
"Production": "Sony Pictures Classics",
"Website": "http://www.angel-amovie.com",
"Response": "True"
}
```

Movies List Example

```
[
{
  "_id": "59d79f05b0b596040599aac8",
  "Title": "Star Wars: Episode IV - A New Hope",
  "Year": "1977",
  "Runtime": "121 min",
  "Genre": "Action, Adventure, Fantasy",
  "Language": "English",
  "Country": "USA",
  "Poster":
    "https://images-na.ssl-images-amazon.com/images/M/MV5BYTUwNTdiMzMtNTNmNS00ODUzLThlMDMtMTM5Y2JkNWJjOGQ2XkEyXkFqcGdeQXVyNzQ1ODk3MTQ@._V1_SX300.jpg",
  "imdbRating": "8.7",
  "imdbVotes": "999,345",
  "imdbID": "tt0076759",
  "Type": "movie"
},
{
  "_id": "59d79f05b0b596040599aac9",
  "Title": "A Beautiful Mind",
  "Year": "2001",
  "Runtime": "135 min",
  "Genre": "Biography, Drama",
  "Language": "English",
  "Country": "USA",
  "Poster":
    "https://images-na.ssl-images-amazon.com/images/M/MV5BMzcwYWFKYzktZjAzNC00OGY1LWI4YTgtNzc5MzVjMDVmNjY0XkEyXkFqcGdeQXVyMTQxNzMzNDI@._V1_SX300.jpg",
  "imdbRating": "8.2",
  "imdbVotes": "677,499",
  "imdbID": "tt0268978",
  "Type": "movie"
}
]
```