Movie information API 1.0.0 OAS3

This API provides access to publicly available data on a limited number of movies published from the year 1990 from the Internet Movie Database and other sources. The API endpoints and their usage are described in detail below.

Authorize

Movies

GET /movies/search

Returns a list of movie data. The list is arranged by imdbId, in ascending order.

Parameters Try it out

Name	Description
title string *(query)*	Text to search for in the primary title of the movie. title
year string *(query)*	The year of initial release of the movie year
page string *(query)*	Page number page

Responses

Code	Description	Links
200	An array of objects containing title, year, imdbID, imdbRating, rottenTomatoesRating, metacriticRating and classification properties. The results are limited to 100 per page. An example of one object in the array is shown below. Media type **application/json** Controls `Accept` header. **Example Value** Schema `{ "data": [ { "title": "Star Trek: First Contact", "year": 1996, "imdbID": "tt0117731", "imdbRating": 7.6, "rottenTomatoesRating": 92, "metacriticRating": 71, "classification": "PG-13" } ], "pagination": { "total": 6, "lastPage": 1, "perPage": 100, "currentPage": 1, "from": 0, "to": 6 } }`	*No links*
400	Invalid year query parameter. Click on 'Schema' below to see the possible error responses. Media type **application/json** **Example Value** Schema `{ "error": true, "message": "Invalid year format. Format must be yyyy." }`	*No links*
429	Rate limit exceeded. Media type	*No links*

Code	Description	Links

text/html

Example Value Schema

```
Too many requests, please try again later.
```

GET /movies/data/{imdbID}

Get data for a movie by imdbID

Parameters Try it out

Name	Description
imdbID * required string *(path)*	The imdbID of the movie imdbID

Responses

Code	Description	Links
200	An object containing the data for the movie. Media type application/json Controls `Accept` header. **Example Value** Schema	*No links*

```
{
  "title": "Star Trek: First Contact",
  "year": 1996,
  "runtime": 111,
  "genres": [
    "Action",
    "Adventure",
    "Drama"
  ],
```

Code	Links


```
  "country": "United States",
  "principals": [
    {
      "id": "nm0005772",
      "category": "cinematographer",
      "name": "Matthew F. Leonetti",
      "characters": []
    },
    {
      "id": "nm0001772",
      "category": "actor",
      "name": "Patrick Stewart",
      "characters": [
        "Picard"
      ]
    },
    {
      "id": "nm0000408",
```


400 — *No links*

Invalid query parameters. Click on 'Schema' to see possible er responses.

Media type

application/json

Example Value Schema


```
{
  "error": true,
  "message": "Invalid query parameters: year. Query parameters are n
ot permitted."
}
```


404 — *No links*

The requested movie could not be found

Media type

application/json

Example Value Schema


```
{
  "error": true,
  "message": "No record exists of a movie with this ID"
}
```


429 — *No links*

Rate limit exceeded.

Media type

text/html

Example Value Schema


```
Too many requests, please try again later.
```

People

GET /people/{id}

Get information about a person (actor, writer, director etc.) from their IMDB ID.

Parameters Try it out

Name	Description
id * required `string` *(path)*	The person's IMDB ID id

Responses

Code	Description	Links
200	An object containing data about that person Media type **application/json** Controls `Accept` header. **Example Value** Schema	*No links*


```
{
  "name": "Patrick Stewart",
  "birthYear": 1940,
  "deathYear": null,
  "roles": [
    {
      "movieName": "Star Trek: First Contact",
      "movieId": "tt0117731",
      "category": "actor",
      "characters": [
        "Picard"
      ],
      "imdbRating": 7.6
    }
  ]
}
```

Code	Description	Links
400	Invalid query parameters. Click on 'Schema' to see possible error responses. Media type **application/json** **Example Value** Schema `{ "error": true, "message": "Invalid query parameters: year. Query parameters are not permitted." }`	*No links*
401	Unauthorized. Click on 'Schema' below to see the possible error responses. Media type **application/json** **Example Value** Schema `{ "error": true, "message": "Authorization header ('Bearer token') not found" }`	*No links*
404	The requested person could not be found Media type **application/json** **Example Value** Schema `{ "error": true, "message": "No record exists of a person with this ID" }`	*No links*
429	Rate limit exceeded. Media type **text/html** **Example Value** Schema `Too many requests, please try again later.`	*No links*

Authentication

POST /user/register

Creates a new user account. A request body containing the user to be registered must be sent.

Parameters Try it out

No parameters

Request body required application/json

An object containing the email and password of the user to be registered.

Example Value Schema

```
{
  "email": "mike@gmail.com",
  "password": "password"
}
```

Responses

Code	Description	Links
201	User successfully created Media type application/json Controls `Accept` header. **Example Value** Schema `{ "message": "User created" }`	*No links*
400	Bad request Media type	*No links*

Code	Description	Links

application/json

Example Value Schema


```
{
  "error": true,
  "message": "Request body incomplete, both email and password are r
equired"
}
```


409

User already exists

No links

Media type

application/json

Example Value Schema


```
{
  "error": true,
  "message": "User already exists"
}
```


429

Rate limit exceeded.

No links

Media type

text/html

Example Value Schema


```
Too many requests, please try again later.
```


POST /user/login

Log in to an existing user account. A request body containing the user credentials must be sent. The longExpiry bool is a setting for development use only that makes both tokens expire after a year.

Parameters Try it out

No parameters

Request body required application/json

The credentials of the user to log in.

Example Value Schema


```
{
  "email": "mike@gmail.com",
  "password": "password",
  "longExpiry": false
}
```


Responses

Code	Description	Links
200	Log in successful Media type **application/json** Controls `Accept` header. **Example Value** Schema `{ "bearerToken": { "token": "ajsonwebtoken", "token_type": "Bearer", "expires_in": 600 }, "refreshToken": { "token": "ajsonwebtoken", "token_type": "Refresh", "expires_in": 86400 } }`	*No links*
400	Invalid log in request Media type **application/json** **Example Value** Schema `{ "error": true, "message": "Request body incomplete, both email and password are required" }`	*No links*
401	Log in failed Media type **application/json**	*No links*

Code	Description	Links

Example Value Schema

```
{
  "error": true,
  "message": "Incorrect email or password"
}
```

429

Rate limit exceeded.

No links

Media type

text/html

Example Value Schema

```
Too many requests, please try again later.
```

POST /user/refresh

Obtain a new bearer token by using a refresh token

Parameters Try it out

No parameters

Request body required **application/json**

The refresh token

Example Value Schema

```
{
  "refreshToken": "ajsonwebtoken"
}
```

Responses

Code	Description	Links
200	Token successfully refreshed Media type **application/json** Controls `Accept` header. **Example Value** Schema `{ "bearerToken": { "token": "ajsonwebtoken", "token_type": "Bearer", "expires_in": 600 }, "refreshToken": { "token": "ajsonwebtoken", "token_type": "Refresh", "expires_in": 86400 } }`	*No links*
400	Invalid refresh request Media type **application/json** **Example Value** Schema `{ "error": true, "message": "Request body incomplete, refresh token required" }`	*No links*
401	Unauthorized. Click on 'Schema' below to see the possible error responses. Media type **application/json** **Example Value** Schema `{ "error": true, "message": "JWT token has expired" }`	*No links*
429	Rate limit exceeded. Media type **text/html**	*No links*

Code	Description	Links

Example Value Schema

```
Too many requests, please try again later.
```

POST `/user/logout`

Log the user out, invalidating the refresh token

Parameters Try it out

No parameters

Request body required application/json

The refresh token

Example Value Schema

```
{
  "refreshToken": "ajsonwebtoken"
}
```

Responses

Code	Description	Links
200	Token successfully invalidated Media type application/json Controls `Accept` header. **Example Value** Schema `{ "error": false, "message": "Token successfully invalidated" }`	*No links*

Code	Description	Links

400	Invalid refresh request
	
	Media type
	application/json
	
	Example Value Schema
	
	{
	"error": true,
	"message": "Request body incomplete, refresh token required"
	}
		No links

401	Unauthorized. Click on 'Schema' below to see the possible error responses.
	
	Media type
	application/json
	
	Example Value Schema
	
	{
	"error": true,
	"message": "JWT token has expired"
	}
		No links

429	Rate limit exceeded. Media type **text/html** **Example Value** Schema Too many requests, please try again later.	*No links*