

Movie API Template

Description

The purpose of this CRUD API is to be able to fetch, add, edit and delete movies.

Prefix

The prefix (or base URL) for all API requests is:

`http://localhost:8888/movie-api.php`

Content Type

The default content type for this API is JSON (application/json).

Status Codes

This API uses the following status codes:

- **200 OK** : The request was successful. Standard status code.
- **400 Bad Request** : The request was malformed or missing required parameters.
- **401 Unauthorized** : You're unauthorized to make the attempted request.
- **404 Not Found** : The requested movie was not found.
- **405 Method Not Allowed** : The requested method was not allowed.

Endpoints

GET /movie-api.php

Returns a list of all movies in the database.

Response

Returns a JSON array consisting of objects with the following properties:

- `id`: The unique identifier of the movie.
- `title`: The title of the movie.
- `releaseYear`: The release year of the movie.
- `director`: The director of the movie.
- `language`: An array of the language(s) spoken in the movie.
- `runtime`: The length of the movie.
- `avgRating`: The average rating of the movie.

Example

Request:

```
GET /movie-api.php
```

Response:

```
[
  {
    "id": 1,
    "title": "Perfect Blue",
    "releaseYear": 1997,
    "director": "Satoshi Kon",
    "language": [
      "Japanese"
    ],
    "runtime": 81,
    "avgRating": 4.4
  },
  {
    "id": 2,
    "title": "Her",
    "releaseYear": 2013,
```

```
        "director": "Spike Jonze",
        "language": [
            "English"
        ],
        "runtime": 126,
        "avgRating": 4
    },
    ...
]
```

GET /movie-api.php?parameter=value

Returns a list of all movies in the database that fulfill the parameter requirements.

Parameters

Only the following parameters are allowed to use for filtering.

PARAMETER	TYPE	DESCRIPTION
<code>id</code>	integer	The id of a specific movie.
<code>releaseYear</code>	integer	Movies released up and until that year.
<code>director</code>	string	The director of the movie(s).
<code>language</code>	array of strings	An array of the language(s) spoken in the movie.
<code>runtime</code>	integer	Movies no longer than this runtime.
<code>avgRating</code>	integer	Movies rated with a minimum of this average.

Example

Request:

```
GET /movie-api.php?id=15
```

Response:

```
{
  "id": 15,
  "title": "Nope",
  "releaseYear": 2022,
  "director": "Jordan Peele",
  "language": [
    "English"
  ],
  "runtime": 130,
  "avgRating": 3.7
}
```

Errors

- **404 Not Found**: If the URL parameters don't follow the right syntax or don't exist in the database, you'll receive the following response:

```
{"ERROR": "Movie not found."}
```

POST /movie-api.php

Adds a movie to the JSON database and returns the added movie.

Structure

To add a movie, the object should have the following structure:

PARAMETER	TYPE	DESCRIPTION
title	string	The title of the movie.
releaseYear	integer	The release year of the movie.
director	string	The director of the movie.

PARAMETER	TYPE	DESCRIPTION
<code>language</code>	array of strings	An array of the language(s) spoken in the movie.
<code>runtime</code>	integer	The length of the movie.
<code>avgRating</code>	integer	The average rating of the movie.

Example

Request:

```
POST /

{
  "title": "Citizen Kane",
  "realeaseYear": 1941,
  "director": "Orson Welles",
  "language": [
    "English"
  ],
  "runtime": 119,
  "avgRating": 4.2
}
```

Response:

```
{
  "title": "Citizen Kane",
  "realeaseYear": 1941,
  "director": "Orson Welles",
  "language": [
    "English"
  ],
  "runtime": 119,
```

```
"avgRating": 4.2
}
```

Errors

- **400 Bad Request**: If one or several keys are missing and/or incorrect, you'll receive the following response:

```
{"ERROR": "Bad Request: One or more keys are missing."}
```

PATCH /movie-api.php

Changes the average rating of the movie.

Structure

To add a movie, the object should have the following structure:

PARAMETER	TYPE	DESCRIPTION
<code>id</code>	string	The id of the movie you wanna edit.
<code>avgRating</code>	integer	The new average rating of the movie.

Example

Request:

```
PATCH /

{
  "id": 19,
  "avgRating": 4.2
}
```

Response:

```
{
  "id": 19,
  "title": "La Noire de...",
  "releaseYear": 1966,
  "director": "Ousmane Semb\u00e8ne",
  "language": [
    "French"
  ],
  "runtime": 60,
  "avgRating": 4.2
}
```

Errors

- **400 Bad Request**: If the ID is missing, you'll receive the following response:

```
{"ERROR": "Bad Request: Movie ID is missing."}
```

- **401 Unauthorized**: If attempted to edit any other key than "avgRating", you'll receive the following response:

```
{"ERROR": "Unathorised access: You are not allowed to edit one or more of these keys!"}
```

- **404 Not Found**: If the ID number doesn't exist, you'll receive the following response:

```
{"ERROR": "Not Found: Movie not found."}
```

DELETE /movie-api.php

Deletes a movie with a specific ID from the JSON database.

Structure

To add a movie, the object should have the following structure:

PARAMETER	TYPE	DESCRIPTION
<code>id</code>	string	The id of the movie you wanna delete.

Example

Request:

```
DELETE /  
  
{  
  "id": 3,  
}
```

Response:

```
{  
  "id": 3,  
  "title": "Guillermo Del Toro's Pinocchio",  
  "releaseYear": 2022,  
  "director": "Guillermo Del Toro",  
  "language": [  
    "English"  
  ],  
  "runtime": 117,  
  "avgRating": 4  
}
```

Errors

- **400 Bad Request**: If the ID is missing, you'll receive the following response:


```
{"ERROR": "Bad Request: Movie ID required."}
```

- **404 Not Found**: If the ID number doesn't exist, you'll receive the following response:

```
{"ERROR": "Not Found: Movie not found."}
```