Documentation Settings

JSON storage for albums, books and films as well as image uploading for the covers/posting.

Introduction

This API uses POST methods to retrieve the information of an album, a book or a film from form data and adds the information to a JSON file (containing an array) in a specific folder for the item as well as uploading the image of the cover/poster.

Overview

Can easily alter the API code to compensate for items other than albums, books and films.

Authentication

For listing and sorting items.

Error Codes

What errors and status codes can a user expect? 200 for successful requests and 404 in cases where information is missing.

POST http://localhost:3000/albums/add

```
http://localhost:3000/albums/add
```

Uses form data (from an HTML form, with user inputs) as the body to upload an image of the album cover to the folder ./api/albums/albumImages (using multer) and insert the information of this album (including the title, artist, release year and the source of the image) into the array in the JSON file .api/albums/albums.json. A unique ID is generated and given to the album, and this is used as the image filename (plus the file format). If successful, the status is 200 and the new album constant is sent.

Example Request

Default

```
var raw = "";

var requestOptions = {
  method: 'POST',
  body: raw,
  redirect: 'follow'
};

fetch("http://localhost:3000/albums/", requestOptions)
  .then(response => response.text())
```

View More

Documentation Settings

GET http://localhost:3000/albums/

http://localhost:3000/albums/

Sends the contents of the JSON file albums.json (from ./api/albums/albums.json).

Example Request

Default

```
var raw = "";

var requestOptions = {
  method: 'GET',
  body: raw,
  redirect: 'follow'
};

fetch("http://localhost:3000/albums/", requestOptions)
  .then(response => response.text())
```

View More

DEL http://localhost:3000/albums/:Id

http://localhost:3000/albums/:Id

Uses the data in the body to retrieve the Id of the album (from data.Id) and uses this Id to find the album associated with the Id in the albums.json file and delete this. Also finds the file in the albumImages folder containing this Id in its name, and therefore representing the album cover of this specific album, and deletes this. If successful, the status is 200 and a string stating the album being deleted is sent.

PATH VARIABLES

Documentation Settings

Id

Example Request

Default

```
var raw = "";

var requestOptions = {
  method: 'DELETE',
  body: raw,
  redirect: 'follow'
};

fetch("http://localhost:3000/albums/:Id", requestOptions)
  .then(response => response.text())
```

View More

POST http://localhost:3000/books/add

http://localhost:3000/books/add

Uses form data (from an HTML form, with user inputs) as the body to upload an image of the book cover to the folder ./api/books/bookImages (using multer) and insert the information of this book (including the title, author, release year and the source of the image) into the array in the JSON file books.json. A unique ID is generated and given to the book, and this is used as the image filename (plus the file format). If successful, the status is 200 and the new book constant is sent.

Example Request

Default

```
var raw = "";

var requestOptions = {
  method: 'POST',
  body: raw,
  redirect: 'follow'
};

fetch("http://localhost:3000/books/", requestOptions)
  .then(response => response.text())
```

View More

GET http://localhost:3000/books/

Documentation Settings

http://localhost:3000/books/

Sends the contents of the JSON file albums.json (from ./api/albums/albums.json).

Example Request

Default

```
var raw = "";

var requestOptions = {
  method: 'GET',
  body: raw,
  redirect: 'follow'
};

fetch("http://localhost:3000/books/", requestOptions)
  .then(response => response.text())
```

View More

DEL http://localhost:3000/books/:Id

http://localhost:3000/books/:Id

Uses the data in the body to retrieve the Id of the book (from data.Id) and uses this Id to find the book associated with the Id in the books.json file and delete this. Also finds the file in the bookImages folder containing this Id in its name, and therefore representing the book cover of this specific book, and deletes this. If successful, the status is 200 and a string stating the book being deleted is sent.

PATH VARIABLES

Id

Example Request

Default

Documentation Settings

```
var raw = "";

var requestOptions = {
  method: 'DELETE',
  body: raw,
  redirect: 'follow'
};

fetch("http://localhost:3000/books/:Id", requestOptions)
  .then(response => response.text())
```

View More

POST http://localhost:3000/films/add

http://localhost:3000/films/add

Uses form data (from an HTML form, with user inputs) as the body to upload an image of the film poster to the folder ./api/films/filmImages (using multer) and insert the information of this film (including the title, director, release year and the source of the image) into the array in the JSON file films.json. A unique ID is generated and given to the film, and this is used as the image filename (plus the file format). If successful, the status is 200 and the new film constant is sent.

Example Request

Default

```
var raw = "";

var requestOptions = {
  method: 'POST',
  body: raw,
  redirect: 'follow'
};

fetch("http://localhost:3000/films/", requestOptions)
  .then(response => response.text())
```

View More

GET http://localhost:3000/films/

http://localhost:3000/films/

Sends the contents of the JSON file albums.json (from ./api/albums/albums.json).

Documentation Settings

Example Request

Default

```
var raw = "";

var requestOptions = {
  method: 'GET',
  body: raw,
  redirect: 'follow'
};

fetch("http://localhost:3000/films/", requestOptions)
  .then(response => response.text())
```

View More

DEL http://localhost:3000/films/:Id

http://localhost:3000/films/:Id

Uses the data in the body to retrieve the Id of the film (from data.Id) and uses this Id to find the film associated with the Id in the films.json file and delete this. Also finds the file in the filmImages folder containing this Id in its name, and therefore representing the poster of this specific film, and deletes this. If successful, the status is 200 and a string stating the film being deleted is sent.

PATH VARIABLES

Id

Example Request

Default

```
var raw = "";

var requestOptions = {
  method: 'DELETE',
  body: raw,
  redirect: 'follow'
};

fetch("http://localhost:3000/films/:Id", requestOptions)
  .then(response => response.text())
```

View More