Endpoints

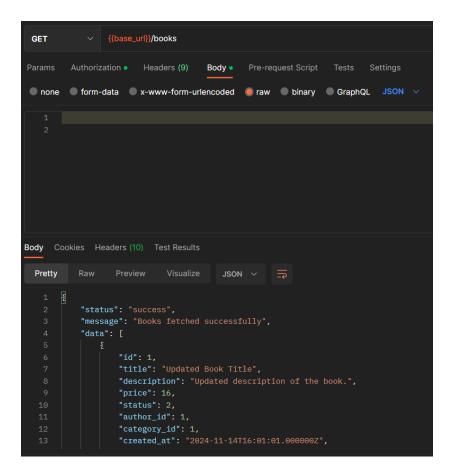
1. Get All Books

Endpoint: GET /books

Description: Fetch a list of all books in the library system.

Response:

• 200 OK: Returns an array of books.



2. Create a New Book

Endpoint: POST / books/create

Description: Add a new book to the library.

Request Body:

Response:

- 201 Created: Returns the created book.
- 400 Bad Request: If validation fails.

3. Update an Existing Book

Endpoint: PUT /books/edit/{id}

Description: Update the details of an existing book.

Path Parameter:

• id (integer): The ID of the book to update.

Request Body:

```
PUT 

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON 

"price": 195.99,

"price": 195.99,

"category": 1,

"category": 1,

"message": "Book updated description of the book."

Pretty Raw Preview Visualize JSON 

"book": f
"idd": 1,
"title": "Updated Book Title",
"book": f
"idd": 1,
"title": "Updated description of the book.",
"price": 135.99,
"status": 2,
"author_id": 1,
"categoryid": 1,
"categoryid": 1,
"categoryid": 1,
"categoryid": 1,
"categoryid": 1,
"created_at": "2024-11-14T16:01:01.0000002",
"updated_at": "2024-11-17T13:33:43.0000002"
```

Response:

- 200 OK: Returns the updated book.
- 404 Not Found: If the book ID does not exist.

4. Delete a Book

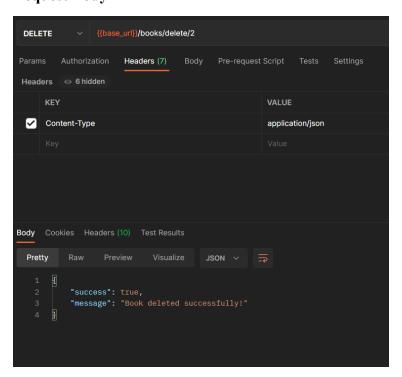
Endpoint: DELETE /books/{id}

Description: Delete a book from the library system.

Path Parameter:

• id (integer): The ID of the book to delete.

Request Body:



Response:

• **204 No Content**: If the deletion is successful.

• 404 Not Found: If the book ID does not exist.