{···}
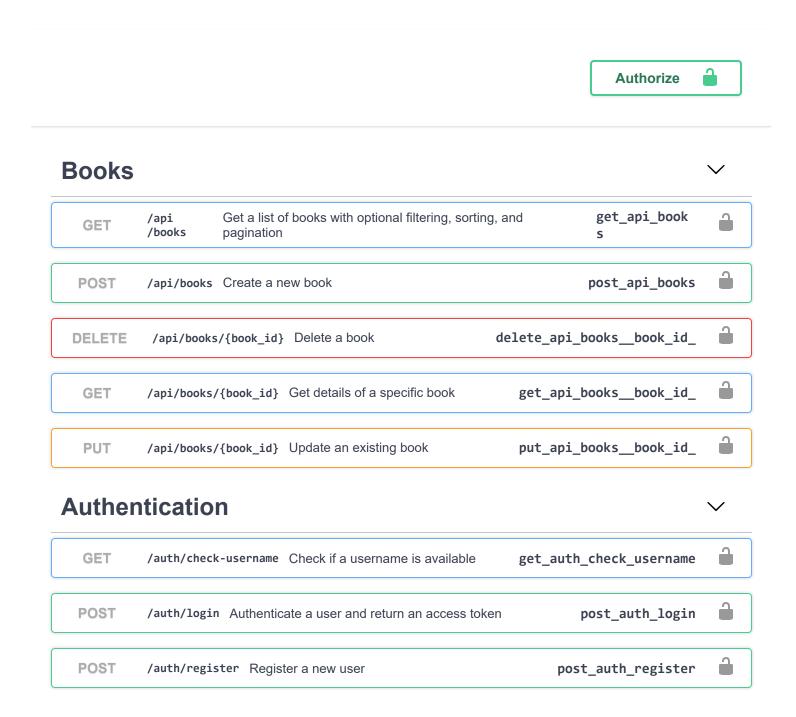
/apispec_1.json   Explore

Book Progress Tracker API 1.0.0

/apispec_1.json

API for tracking book progress

Authorize 🔓

Books   ⌄

GET	**/api /books**	Get a list of books with optional filtering, sorting, and pagination	**get_api_book s**	🔓
POST	**/api/books**	Create a new book	**post_api_books**	🔓
DELETE	**/api/books/{book_id}**	Delete a book	**delete_api_books__book_id_**	🔓
GET	**/api/books/{book_id}**	Get details of a specific book	**get_api_books__book_id_**	🔓
PUT	**/api/books/{book_id}**	Update an existing book	**put_api_books__book_id_**	🔓

Authentication   ⌄

GET	**/auth/check-username**	Check if a username is available	**get_auth_check_username**	🔓
POST	**/auth/login**	Authenticate a user and return an access token	**post_auth_login**	🔓
POST	**/auth/register**	Register a new user	**post_auth_register**	🔓

Models   ⌄

AccessToken   {

    access_token   string

                        JWT access token

}

Book   {

    genre   string
    id   integer
    is_completed   boolean
    pages_read   integer
    pages_total   integer
    progress_percent   number
    title   string

}

BookCreate   {

    genre*   string
                        example: Fiction

                        Genre of the book

                        Enum:

                          Array [3]
    is_completed   boolean
                        example: false

                        Whether the book is completed

    pages_read   integer
                        example: 50

                        Number of pages read so far

    pages_total*   integer
                        example: 193

                        Total number of pages in the book

    title*   string
                        example: The Hitchhiker's Guide to the Galaxy

                        Title of the book

}

BookList   {

books
                                    [...]
current_page   integer

                         Current page number

pages   integer

                         Total number of pages

total   integer

                         Total number of books

}

BookUpdate   {

genre   string
                         example: Science Fiction

                         New genre of the book

                         Enum:

                             Array [3]
is_completed   boolean
                         example: true

                         Whether the book is completed

pages_read   integer
                         example: 100

                         New number of pages read

pages_total   integer
                         example: 250

                         New total number of pages

title   string
                         example: The Restaurant at the End of the Universe

                         New title of the book

}

Error   {

error   string

                         Error message

}

UserLogin   {

password*   **string**
                          example: SecureP@ss1

                          User's password

username*   **string**
                          example: testuser

                          User's username

}

UserRegistered   {

id   **integer**

                          User ID

username   **string**

                          Registered username

}

UserRegistration   {

password*   **string**
                          example: SecureP@ss1

                          User's chosen password (min 8 chars, 1 uppercase, 1 lowercase, 1
                          number, 1 special char)

username*   **string**
                          example: testuser

                          User's chosen username

}

UsernameAvailability   {

available   **boolean**

                          True if username is available, False otherwise

message   **string**

                          Optional message (e.g., "Username is already taken")

}

[Powered by Flasgger 0.9.7.1]