APIs

1) Create a Book

Endpoint
POST / books /

Creates a new book entry in the system.

Request body
{
	name: “book_name”,
	author: “book_author”,
	publisher: “publisher”,
	publicationDate: “date of publication”,
	“ISBN”: “ISBN number of the book”,
	“coverPhoto: “URL of the cover photo”
}

Responses:
201 – Created
Successfully created

Response body
{
	id: “id of the book entry”,
	name: “book_name”,
	author: “book_author”,
	publisher: “publisher”,
	publicationDate: “date of publication”,
	“ISBN”: “ISBN number of the book”,
	“coverPhoto: “URL of the cover photo”
{

400 – Bad Request
Invalid format or missing fields

401 – Unauthorized
Does not have proper permissions to create a book entry

500 – Server problem
Something went wrong

2) Update a book's cover photo

Endpoint:
PATCH / books / {book_id} / coverPhoto /

Updates the cover photo of a given book

Request body:
{
    coverPhoto: "new url with the cover photo"
}

Responses:

200 – OK
Cover photo successfully updated

Response body:
{
    id: "id"
    coverPhoto: "new URL"
}

404 – Not found
No matching book id record

401 – Unauthorized
Does not have proper permissions to create a book entry

500 – Server problem
Something went wrong

3) List books

GET / books / ?author=Rowling&sort_by=title&index=1&page_size=3

List all the available books

Query params:

index (int, the current results set) = 0 (default)
page_size (int, results per page) = 20 (default)
author (string, author filter)
publisher(string, publisher filter)
title(string, book title filter)
sort_by (string, - for DESC)

Responses

200 OK
Successfully retrieved matching records

{
index: 1,
page_size = 3,
total_results = 8,
books: [
{
id: “1234”,
name: “Harry Potter and the Chamber of Secrets”,
author: “JK Rowling”,
publisher: “publisher”,
publicationDate: “date of publication”,
“ISBN”: “ISBN number of the book”,
“coverPhoto: “URL of the cover photo”
},
{……..
}
]
}

400 – Bad Request
Invalid query parameters

4. Upload a book's content as a file

Endpoint
POST / books / {book_id} / upload /

Uploads a book file as a binary

Request body:

file <file in binary format>

Responses:

200 OK
File successfully uploaded

Response body:
{
    id: "id",
    file_url: "downloadable url of the file as a PDF"
}

400 Bad Request
Invalid file

404 Not Found
Book id not found

5. Processing a Book

Send for processing

Endpoint
POST / books / {book_id} / process /

Start the processing

Responses:

202 – Accepted
Transferred successfully for processing

Response body:
{
taskId : “task id returned by the process”
status : “In progress”
bookId : book_id
}

404 Not Found
Book id not found

Check the status

Endpoint
GET / processes / task_id

Returns the current status of the book processing

Responses:

200 OK
Processing has completed
{
taskId : task_id
status : “Completed”
bookId : book_id
start: “Start time”
end: “Finish time”
}

202 Accepted
{
taskId : task_id
status : “In progress”
}

404 Not Found
Task id not found

Retrieve processed data

Endpoint
GET / books / {book_id} / insights /

Retrieve the insights of the book

Responses:
200 OK

Response body:
{
    bookID : book_id
    summary : “book summary”
    …..
}

202 Accepted
Book is still processing

404 Not found
Book id not found