Designing a REST API: a library – books

Functional Requirements:

- Add, update, delete, and retrieve books.
- Search books by title, author, genre, etc.
- Paginate large book collections.
- Link related resources (HATEOAS).
- Provide full error details.
- Secure access with JWT.
- Cache read operations for performance.

Non-Functional Requirements:

- **Security**: Token-based auth, input validation.
- **Caching**: GET operations cached; POST/PUT/DELETE not cached.

Entities

1. **Book**

   Attributes:

   - id
   - title
   - author

REST API Design

Authentication

Parameter	Type	In	Description
Authorization	Bearer-Token	Headers	Token for authorization

GET /api/v1/books — Get all books (with filters, pagination)

Parameter	Type	In	Description

author	string	query	Filter by author
title	string	query	Filter by title
Authorization	Bearer-Token	headers	Token for authorization

Response:

Status Code	Content-Type	Example
200 OK	application/json	[{"id": 1, "title": "Example Book"}, {"id": 2, "title": "Example Book"}]
403 Forbidden	application/json	{"detail": "unauthorized"}
404 Not Found	application/json	{ "detail": "Book not found" }

Caching: Cache-Control: max-age=10min

GET /api/v1/books/{id} — Get a book by ID

Parameter	Type	In	Description
id	integer	path	Book ID
Authorization	Bearer-Token	headers	Token for authorization

Response:

Status Code	Content-Type	Example
200 OK	application/json	[{"id": 1, "title": "Example Book"}, {"id": 2, "title": "Example Book"}]
403 Forbidden	application/json	{"detail": "unauthorized"}
404 Not Found	application/json	{ "detail": "Book not found" }

Caching: Cache-Control: max-age=10min

POST /api/v1/books — Add a new book

Parameter	Type	In	Description
body	JSON	body	Book data (title, author, etc.)

Authorization	Bearer-Token	header s	Token for authorization
Content-Type		header s	application/json

Response:

Status Code	Content-Type	Example
201 Created	application/json	{ "id": 101, "title": "New Book" }
400 Bad Request	application/json	{ "detail": "Missing field" }
403 Forbidden	application/json	{"detail": "Unauthorized "}

Caching: this type of operation is not cached.

PATCH /api/v1/books/{id} — Update part of a book

Parameter	Type	In	Description
id	integer	path	Book ID
body	JSON	body	Partial fields to update
Authorization	Bearer-Token	header	Token for authorization

Response:

Status Code	Content-Type	Example
200 OK	application/json	[{"id": 1, "title": "Example Book"}, {"id": 2, "title": "Example Book"}]
400 Bad Request	application/json	{ "detail": "Missing field" }
403 Forbidden	application/json	{"detail": "Unauthorized"}
404 Not Found	application/json	{ "detail": "Book not found" }

Caching: this type of operation is not cached.

DELETE /api/v1/books/{id} — Delete a book

Parameter	Type	In	Description

id	integer	path	Book ID
Authorization	Bearer-Token	header	Token for authorization

Response:

Status Code	Content-Type	Example
204 No Content	application/json	does not return anything
403 Forbidden	application/json	{"detail": "unauthorized"}

Caching: this type of operation is not cached.