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	Токен для авторизации (Token for authorization)

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

Parameter	Type	In	Description
page	integer	query	Page number (default: 0)
size	integer	query	Page size (default: 10)
sort	string	query	Sort by title or year

author	string	query	Filter by author
genre	string	query	Filter by genre
available	boolean	query	Filter by availability
Authorization	Bearer-Token	header	Token for authorization

Response Examples:

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"}

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	Token for authorization

Request Example:

```
json
{
  "title": "New Book",
  "author": "Author Name",
  "year": 2024,
  "genre": "Sci-Fi",
  "available": true
}
```

Response:

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

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

Parameter	Type	In	Description
id	integer	path	Book ID

Response:

Status Code	Example
200 OK	{ "id": 1, "title": "Example Book" }
404 Not Found	{ "detail": "Book not found" }

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

PUT /api/v1/books/{id} — Update entire book

Parameter	Type	In	Description
id	integer	path	Book ID
body	JSON	body	All book fields
Authorization	Bearer-Token	header	Token for authorization

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

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	Example
204 No Content	— (book successfully deleted)
404 Not Found	{ "detail": "Book not found" }