

Thanks for the detailed description—your system already has a solid foundation. Here's a complete set of API endpoints that will round out your library management system and give you a solid backend project for practice:

Book Management

Method	Endpoint	Description
POST	/books	Add a new book to the library
PATCH	/books/{bookId}	Update book info (e.g., title, author, quantity)
PATCH	/books/quantity/{bookId}	Update quantity (trigger will auto-update status)
GET	/books	Get all books
GET	/books/available	Get all books with status AVAILABLE
GET	/books/search?query=	Search books by title, author, genre, or ISBN
GET	/books/{bookId}	Get specific book by ID
DELETE	/books/{bookId}	Delete a book

User Management

Method	Endpoint	Description
POST	/users	Register a new user
PATCH	/users/{userId}	Update user info
GET	/users	List all users
GET	/users/{userId}	Get specific user
PATCH	/users/ban/{userId}	Manually ban a user
PATCH	/users/unban/{userId}	Manually unban a user

Method	Endpoint	Description
GET	/users/status/{status}	Get users by status (ACTIVE or BANNED)
GET	/users/credits/low	Get users with low credit (e.g., <= -5)

Borrow/Return Management

Method	Endpoint	Description
POST	/borrow	Borrow a book (insert into BORROW_RECORD, reduce book quantity)
PATCH	/return/{borrowId}	Return a book (update return date, update user credit, update book quantity)
GET	/borrow	Get all borrow records
GET	/borrow/active	Get all books currently borrowed
GET	/borrow/user/{userId}	Get all books borrowed by a user
GET	/borrow/overdue	Get all overdue borrow records

Background Tasks

Method	Endpoint	Description
POST	/cron/decrement-credits	Cron-like endpoint that you can trigger daily to reduce credit for overdue books

You can simulate this with a `@Scheduled` method in Spring later or use a cron job if hosted.

Bonus Ideas

- Authentication system (later): login, roles (admin vs user).
- Borrow limits: max number of books per user.
- Genre filtering: `/books/genre/{genre}`.
- Book recommendations: based on user's previous borrow history.