# How to use the APIs:

A screenshot of a computer

AI-generated content may be incorrect.

1. **Access to Swagger API Docs**:

<http://localhost:8080/library-system/swagger-ui/index.html>

1. **Register borrowers:**

Expand POST /borrowers

Click "Try it out" button

Replace Request body with the following and click Execute button to register a borrower:

{

"name": "Adrian Tee",

"email": "adrian.tee@aeonbank.com"

}

Add more borrowers:

{

"name": "Alex Pee",

"email": "alex.pee@merz.com"

}

1. **Register books:**

Expand POST /books

Click "Try it out" button

\*\*NOTE: ISBN must be a valid one. Invalid ISBN is not allowed to be registered. Validation implemented.

\*\*ISBN will be normalized by replacing - (hyphen) and \\s (any whitespace character: space, tab, etc.)

Replace Request body with the following and click Execute button to register a book:

{

"isbn": "9780132350884",

"title": "Clean Code",

"author": "Robert Cecil Martin"

}

Click multiple times with the same request body allow adding the same book with different IDs.

Add more books:

{

"isbn": "978-0007179732",

"title": "The Monk Who Sold His Ferrari",

"author": "Robin Sharma"

}

1. **Get a list of all books in the library:**

EXPAND GET /books/all

Click "Try it out" button

Click Execute button to list all books.

1. **Get a paginated list of books in the library:**

EXPAND GET /books

Click "Try it out" button

Replace pageable with the following to sort the list by title with descending order:

\*\*you can filter by isbn, title, author and loanStatus.

\*\*you can sort by isbn, title and author.

{

"page": 0,

"size": 10,

"sort": "title,desc"

}

Click Execute button to list all books.

1. **Borrow a book:**

EXPAND POST /loans

Click "Try it out" button

Make sure to set valid bookId and borrowerId. Invalid bookId or borrowerId will throw exception.

Book already on loan cannot be borrowed by other at the same time.

A borrower can borrow multiple books.

Replace Request body with the following and click Execute button to borrow a book:

{

"bookId": 1,

"borrowerId": 1

}

Borrow more books:

{

"bookId": 2,

"borrowerId": 2

}

{

"bookId": 3,

"borrowerId": 1

}

Click Execute button to to borrow book.

1. **Return a borrowed book on behalf of a borrower:**

EXPAND PATCH /loans/{loanId}/return  
Click "Try it out" button

Insert a valid loanId.

Click Execute button to return a loaned book.

1. **Get a list of loans records**

EXPAND GET /loans

Click "Try it out" button

Replace pageable with the following:

{

"page": 0,

"size": 10

}

Click Execute button to list all loan records.

\*\*Can be filtered by:

loanStatus (unselected will list all loan records)

borrowerName (partial value)

borrowerEmail (exact value)