API Specification Doc

(*Library Management System*)

Version	Date	Author	Description
1.0	09 - December - 2024	Udara Liyanage	API that provides endpoints for managing library system.

Table of Contents

Methods

1. Library Book Management/ Register

Register new book to the library.

Request

Method	URL
POST	/api/v1/book/register

Schema	{
	"isbn": "string",
	"title": "string",
	"author": "string"
	}

Response

Status	Response

200	{
	"isbn": "123456789112",
	"title": "Book 1",
	"author": "Auth",
	"borrowed": false,
	"bookId": 1
	}

400	{
	"author": "Author is required",
	"isbn": "ISBN should be between 10 and 13 characters",
	"title": "Title is required"
	}

422	{
	"message": "Error occurred during processing request"
	}

500	Internal Server Error

2. Book Management/ Find Book by ID

Get books by book ID

Request

Method	URL
GET	/api/v1/book/{bookId}

Param	{ bookId } - “int”

Response

Status	Response

200	{
	"isbn": "123456789112",
	"title": "Book 1",
	"author": "Auth",
	"borrowed": false,
	"bookId": 1
	}

404	{
	"message": "Book not found with id: '20'"
	}

500	Internal Server Error

3. Book Management/ Fetch All Books

Fetch All Books data in the system

Request

Method	URL
GET	/api/v1/book/ getAllBooks

Param	{pageNo} - “integer”, {pageSize} - “integer”, {sortBy} - “string”, {sortDir} - “string”

Response

Status	Response

200	{
	"content": [
	{
	"isbn": "123456789112",
	"title": "Book 1",
	"author": "Auth",
	"borrowed": false,
	"bookId": 1
	}
	],
	"pageNo": 0,
	"pageSize": 1,
	"totalElements": 9,
	"numberOfElements": 1,
	"totalPages": 9,
	"last": false
	}

422	{
	"message": " Error occurred during processing request "
	}

500	Internal Server Error

4. Book Management/ Get Available Books to Borrow

Fetch all available books to borrow.

Request

Method	URL

GET	/api/v1/book/ getAvailable

Param	{pageNo} - “integer”, {pageSize} - “integer”, {sortBy} - “string”, {sortDir} - “string”

Response

Status	**Response**

200	{
	"content": [
	{
	"isbn": "123456789112",
	"title": "Book 1",
	"author": "Auth",
	"borrowed": false,
	"bookId": 1
	}
	],
	"pageNo": 0,
	"pageSize": 1,
	"totalElements": 9,
	"numberOfElements": 1,
	"totalPages": 9,
	"last": false
	}

422	{
	"message": " Error occurred during processing request "
	}

500	Internal Server Error

5. Borrower Management/ Register Borrower

Register new Borrower to the Library.

Request

Method	URL
POST	/api/v1/borrower/register

Schema	{
	"name": "string",
	"email": "string"
	}

Response

Status	Response

200	{
	"name": "Udara Liyanage",
	"email": "udara@gmail.com",
	"borrowerId": 2
	}

422	{
	"message": " Error occurred during processing request "
	}

400	{
	"email": "Invalid email id"
	}

500	Internal Server Error

6. Borrower Management/ Find Borrower by ID

Get borrower details by borrower ID

Request

Method	URL
GET	/api/v1/borrower/{borrowerId}

Param	{ borrowerId } - “int”

Response

Status	Response

200	{
	"name": "Udara Liyanage",
	"email": "udara@gmail.com",
	"borrowerId": 2
	}

404	{
	"message": "Borrower not found with id: '20'"
	}

500	Internal Server Error

7. Borrower Management/ Fetch All Borrower

Fetch All Borrower data in the system

Request

Method	URL
GET	/api/v1/borrower

Param	{pageNo} - “integer”, {pageSize} - “integer”, {sortBy} - “string”, {sortDir} - “string”

Response

Status	Response

200	{
	"content": [
	{
	"name": "string",
	"email": "ab@abc.com",
	"borrowerId": 1
	},
	{
	"name": "Udara Liyanage",
	"email": "udara@gmail.com",
	"borrowerId": 2
	}
	],
	"pageNo": 0,
	"pageSize": 10,
	"totalElements": 2,
	"numberOfElements": 2,
	"totalPages": 1,
	"last": true
	}

422	{
	"message": " Error occurred during processing request "
	}

500	Internal Server Error

8. Borrow Management/ Borrow Book

Issues library books to a borrower.

Request

Method	URL
POST	/api/v1/borrowing/{bookId}/borrow

Param	Path Param {bookId} – “integer”, Query Param {borrowerId} – “integer”

Response

Status	Response

200	{
	"borrower": {
	"name": "string",
	"email": "ab@abc.com",
	"borrowerId": 1
	},
	"bookInfo": {
	"isbn": "123456789112",
	"title": "Book 1",
	"author": "Auth",
	"borrowed": true,
	"bookId": 1
	},
	"borrowDate": "2024-12-08T22:33:33.4277313",
	"returnDate": null,
	"borrowed": true,
	"borrowingId": 1
	}

422	{
	"message": "Book is already borrowed by someone"
	}

404	{
	"message": "Book not found with id: '50'"
	}

500	Internal Server Error

9. Borrow Management/ Return Book

Returns the book to the library by the borrower.

Request

Method	URL
POST	/api/v1/borrowing/{borrowingId}/return

Param	Path Param {borrowingId} – “integer”,

Response

Status	Response

200	{
	"borrower": {
	"name": "string",
	"email": "ab@abc.com",
	"borrowerId": 1
	},
	"bookInfo": {
	"isbn": "123456789112",
	"title": "Book 1",
	"author": "Auth",
	"borrowed": false,
	"bookId": 1

	}, "borrowDate": "2024-12-08T22:33:33.427731", "returnDate": "2024-12-08T22:40:17.8324196", "borrowed": false, "borrowingId": 1 }

404	{
	"message": "Borrowing not found with id: '10'"
	}

500	Internal Server Error

10. Borrow Management/ Filter Books by Borrower ID

Get all books borrowed by borrower

Request

Method	URL
GET	/api/v1/borrowing/{borrowerId}

Param	Query Param {pageNo} - “integer”, {pageSize} - “integer”, {sortBy} - “string”, {sortDir} - “string” Path Param {borrowerId} – “integer”

Response

Status	Response

200	{
	"content": [
	{
	"borrower": {
	"name": "string",
	"email": "ab@abc.com",
	"borrowerId": 1
	},
	"bookInfo": {
	"isbn": "123456789112",
	"title": "Book 1",
	"author": "Auth",
	"borrowed": false,
	"bookId": 1
	},
	"borrowDate": "2024-12-08T22:33:33.427731",
	"returnDate": "2024-12-08T22:44:15.517888",
	"borrowed": false,
	"borrowingId": 1
	}
	],
	"pageNo": 0,
	"pageSize": 10,
	"totalElements": 1,
	"numberOfElements": 1,
	"totalPages": 1,
	"last": true
	}

500	Internal Server Error

11. Borrow Management/ Filter borrow by borrower and Book

Get borrowing info by borrower ID and book ID

Request

Method	URL
GET	/api/v1/borrowing/getBorrowingInfo/{borrowerId}/{bookId}

Param	{borrowerId} – “integer”, {bookId} – “integer”

Response

Status	Response

200	{
	"borrower": {
	"name": "string",
	"email": "ab@abc.com",
	"borrowerId": 1
	},
	"bookInfo": {
	"isbn": "123456789112",
	"title": "Book 1",
	"author": "Auth",
	"borrowed": false,
	"bookId": 1
	},
	"borrowDate": "2024-12-08T22:33:33.427731",
	"returnDate": "2024-12-08T22:44:15.517888",
	"borrowed": false,
	"borrowingId": 1
	}

500	Internal Server Error

Glossary

Status Codes

All status codes are standard HTTP status codes. The below ones are used in this API.

2XX - Success of some kind
4XX - Error occurred in client's part
5XX - Error occurred in server's part

Status Code	Description
200	OK
201	Created
400	Bad request
403	Forbidden
404	Resource not found
405	Method Not Allowed
409	Conflict
412	Precondition Failed
413	Request Entity Too Large
500	Internal Server Error
501	Not Implemented
503	Service Unavailable