# Library Management API Documentation

## Overview

The Library Management API provides functionalities for managing a library system. It allows users to perform CRUD operations for books, manage users, and handle borrowing and returning of books. The API is built using Express.js and follows RESTful design principles.

## Base URL

https://<your-api-base-url>

## Authentication

All endpoints require an authentication token to access. Add the token in the `Authorization` header as:  
Authorization: Bearer <your-token>

## Endpoints

### Books

#### 1. Get All Books

URL: /api/books  
Method: GET  
Description: Retrieves a list of all available books.

Response:

[  
 {  
 "id": 1,  
 "title": "1984",  
 "author": "George Orwell",  
 "isbn": "123-456-789",  
 "status": "available"  
 }  
 ]

#### 2. Add a New Book

URL: /api/books  
Method: POST  
Headers:

Content-Type: application/json  
Authorization: Bearer <your-token>

Body:

{  
 "title": "1984",  
 "author": "George Orwell",  
 "isbn": "123-456-789"  
 }

Response:

{  
 "message": "Book added successfully",  
 "bookId": 1  
 }

#### 3. Update Book Details

URL: /api/books/:id  
Method: PUT  
Headers:

Content-Type: application/json  
Authorization: Bearer <your-token>

Body:

{  
 "title": "Animal Farm",  
 "author": "George Orwell"  
 }

Response:

{  
 "message": "Book updated successfully"  
 }

#### 4. Delete a Book

URL: /api/books/:id  
Method: DELETE  
Headers:

Authorization: Bearer <your-token>

Response:

{  
 "message": "Book deleted successfully"  
 }

### Users

### Borrowing & Returning

## Error Handling

All endpoints return errors in the following format:  
  
 {  
 "error": {  
 "message": "Error message here",  
 "status": 400  
 }  
 }

## Technologies Used

Framework: Express.js  
Database: MySQL  
Authentication: JWT-based authentication