BookWise Online Bookstore

API Documentation

February 1, 2026

Contents

1 Introduction

This document provides comprehensive documentation for the BookWise Online Bookstore RESTful API. The API supports JWT-based authentication, role-based access control, and dual storage modes (JSON and SQLite).

2 Base Information

- **Base URL**: `http://localhost:1010`
- **Authentication**: JWT Bearer Token
- **Content-Type**: `application/json`
- **Rate Limiting**: Implemented for security

3 Authentication

3.1 POST /auth/register

Register a new user account.

Request Body:

```
{
  "name": "string",
  "username": "string",
  "email": "string",
  "password": "string"
}
```

Success Response (201):

```
{
  "user_id": 1,
  "username": "string",
  "role": "customer",
  "name": "string",
  "email": "string",
  "token": "jwt_token"
}
```

Error Responses:

- `400 Bad Request` - Missing fields, invalid email, weak password, or duplicate username
- `429 Too Many Requests` - Rate limit exceeded (5 requests per 10 seconds)

3.2 POST /auth/login

Authenticate existing user.

Request Body:

```
{
  "username": "string",
  "password": "string"
}
```

Success Response (200):

```
{
  "user_id": 1,
  "username": "string",
  "role": "customer",
  "name": "string",
  "email": "string",
  "token": "jwt_token"
}
```

Error Responses:

- `401 Unauthorized` - Invalid credentials
- `429 Too Many Requests` - Rate limit exceeded (5 requests per 10 seconds)

4 Books Endpoints

4.1 GET /books

Get all books (public endpoint).

Success Response (200):

```
[
  {
    "id": 1,
    "title": "string",
    "author": {
      "id": 1,
      "first_name": "string",
      "last_name": "string",
      "bio": "string"
    },
    "genres": ["string"],
    "published_at": "2023-01-01T00:00:00Z",
    "price": 19.99,
    "stock": 10
  }
]
```

4.2 GET /books/{id}

Get specific book by ID (public endpoint).

Success Response (200): Same structure as above, single object.

Error Responses:

- `404 Not Found` - Book not found

4.3 POST /books (Admin Only)

Create new book.

Headers:

```
Authorization: Bearer <admin_token>
```

Request Body:

```
{
  "title": "string",
  "author": {
    "id": 1
  },
  "genres": ["Fiction"],
  "price": 19.99,
  "stock": 10
}
```

Success Response (201): Complete book object with author details.

Error Responses:

- `401 Unauthorized` - Missing/invalid token
- `403 Forbidden` - User not admin
- `400 Bad Request` - Invalid input
- `429 Too Many Requests` - Rate limit exceeded

4.4 PUT /books/{id} (Admin Only)

Update existing book.

Headers:

```
Authorization: Bearer <admin_token>
```

Request Body: Same as POST /books.

Success Response (200): Updated book object.

Error Responses: Same as POST /books plus `404 Not Found`.

4.5 DELETE /books/{id} (Admin Only)

Delete book.

Headers:

```
Authorization: Bearer <admin_token>
```

Success Response (204): No content.

Error Responses: Same as PUT /books/{id}.

5 Authors Endpoints

5.1 GET /authors

Get all authors (public endpoint).

Success Response (200):

```
[
  {
    "id": 1,
    "first_name": "string",
    "last_name": "string",
    "bio": "string"
  }
]
```

5.2 GET /authors/{id}

Get specific author by ID (public endpoint).

Success Response (200): Single author object.

Error Responses:

- `404 Not Found` - Author not found

5.3 POST /authors (Admin Only)

Create new author.

Headers:

```
Authorization: Bearer <admin_token>
```

Request Body:

```
{
  "first_name": "string",
  "last_name": "string",
  "bio": "string"
}
```

Success Response (201): Complete author object.

Error Responses:

- `401 Unauthorized` - Missing/invalid token
- `403 Forbidden` - User not admin
- `400 Bad Request` - Missing first/last name
- `429 Too Many Requests` - Rate limit exceeded

5.4 PUT /authors/{id} (Admin Only)

Update existing author.

Headers:

```
Authorization: Bearer <admin_token>
```

Request Body: Same as POST /authors.

Success Response (200): Updated author object.

Error Responses: Same as POST /authors plus `404 Not Found`.

5.5 DELETE /authors/{id} (Admin Only)

Delete author.

Headers:

```
Authorization: Bearer <admin_token>
```

Success Response (204): No content.

Error Responses: Same as PUT /authors/{id}.

6 Orders Endpoints

6.1 POST /orders

Create new order.

Headers:

```
Authorization: Bearer <customer_token>
```

Request Body:

```
{
  "items": [
    {
      "book_id": 1,
      "quantity": 2
    }
  ],
  "total_price": 39.98
}
```

Success Response (201):

```
{
  "id": 1,
  "user_id": 1,
  "user": {
    "id": 1,
    "username": "string",
    "role": "customer",
    "name": "string",
    "email": "string"
  },
  "items": [
    {
      "book": {
        "id": 1,
        "title": "string",
        "author": {
          "first_name": "string",
          "last_name": "string"
        },
        "price": 19.99
      },
      "quantity": 2
    }
  ],
  "total_price": 39.98,
  "created_at": "2023-01-01T00:00:00Z",
  "status": "confirmed"
}
```

Error Responses:

- `401 Unauthorized` - Missing/invalid token
- `400 Bad Request` - Invalid input, insufficient stock, or book not found
- `429 Too Many Requests` - Rate limit exceeded

6.2 GET /orders

Get current user's orders.

Headers:

```
Authorization: Bearer <customer_token>
```

Success Response (200): Array of order objects.

Error Responses:

- `401 Unauthorized` - Missing/invalid token
- `429 Too Many Requests` - Rate limit exceeded

7 Users Endpoints

7.1 GET /users/me

Get current user profile.

Headers:

```
Authorization: Bearer <token>
```

Success Response (200):

```
{
  "id": 1,
  "username": "string",
  "role": "customer",
  "name": "string",
  "email": "string"
}
```

Error Responses:

- `401 Unauthorized` - Missing/invalid token
- `429 Too Many Requests` - Rate limit exceeded

7.2 GET /users?role=customer (Admin Only)

Get all customers.

Headers:

```
Authorization: Bearer <admin_token>
```

Success Response (200): Array of customer objects.

Error Responses:

- `401 Unauthorized` - Missing/invalid token
- `403 Forbidden` - User not admin
- `429 Too Many Requests` - Rate limit exceeded

8 Reports Endpoint

8.1 GET /reports (Admin Only)

Get sales reports.

Headers:

```
Authorization: Bearer <admin_token>
```

Success Response (200):

```
{
  "total_orders": 5,
  "total_revenue": 199.90,
  "average_order_value": 39.98
}
```

Error Responses:

- `401 Unauthorized` - Missing/invalid token
- `403 Forbidden` - User not admin
- `429 Too Many Requests` - Rate limit exceeded

9 Rate Limiting

Endpoint Type	**Limit**	**Purpose**
Authentication (/auth/*)	5 requests / 10 seconds	Prevent brute force attacks
Admin endpoints	10 requests / 30 seconds	Prevent admin spam
API endpoints	20 requests / second	Allow normal usage
Public endpoints	No rate limiting	Allow public browsing

When rate limited:

- **Status**: `429 Too Many Requests`
- **Header**: `Retry-After: 60`
- **Body**: `{"error":"rate limit exceeded - too many requests"}`

10 Data Types

- **Content-Type**: Always `application/json`
- **Date Format**: ISO 8601 (`2023-01-01T00:00:00Z`)
- **IDs**: Always positive integers
- **Stock**: Non-negative integers
- **Passwords**: Minimum 8 characters