API Endpoints

For postman collection refer to BMS.postman\_collection.json. It can be imported into Postman tool.

**Note: For trying out the endpoints, Sign up using /User/signup endpoint and authenticate using /User/login endpoint. Then always attach bearer token in the headers for every other request. More details of the signup and login endpoints below.**

User endpoints:

**GET**

/User

Returns a list of all users with their details in the system. This is allowed for users with admin role only.

/User/{id}

Returns a single user by Id with all its details.

/User/filter?

Returns a list of all users filtered by the allowed query parameters – First Name, Last Name, Role, Status, Min Date of Birth, Max Date of Birth. This is only allowed for admins.

**POST**

/User/signup

Registers a user in the system. Here is a sample payload where Date of Birth can be empty/null:

{

"UserId": "1234567894",

"UserName": "johndoe12351",

"FirstName": "John",

"LastName": "Doe",

"Email": "johndoe@example.com",

"Password": "hashedPass221@game",

"DateOfBirth": "1990-01-01",

"Role": "Admin",

"Status": "Active",

"CreatedAt": "2024-12-11 14:30:00",

"LastModifiedAt": "2024-12-11 14:30:00",

"LastLogin": "2024-12-11 14:30:00"

}

/User/login

Authenticates a user by validating email and hashed password and assigns a JWT token for use. Here is a sample payload:

{

"UserName": "johndoe12350",

"Password": "hashedPass221@game"

}

Here is the response:

{

"message": "User authenticated successfully",

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9uYW1lIjoiam9obmRvZTEyMzUwIiwic3ViIjoiam9obmRvZTEyMzUwIiwiaHR0cDovL3NjaGVtYXMubWljcm9zb2Z0LmNvbS93cy8yMDA4LzA2L2lkZW50aXR5L2NsYWltcy9yb2xlIjoiQWRtaW4iLCJqdGkiOiI4MWZjZDExZi1hZDIzLTQ2YTktODI5Mi0xOTBiYjA4NmEwYjAiLCJleHAiOjE3MzU3ODM3MTgsImlzcyI6IkJvb2stTWFuYWdlbWVudC1TeXN0ZW0iLCJhdWQiOiJ1c2VycyJ9.Ct7cwKc\_ZQhOtyLcNKD\_SsrEZ68F8ekjAOEX05kq-Fw"

}

**PATCH**

/User

Modify details of a user. Allowed for both customers and admins. Here is a sample payload where only User Id is required, rest everything can be null/empty:

{

"UserId": "1234567892",

"UserName": "johndoe12349",

"FirstName": "is it working",

"LastName": "Doe",

"Email": "johndoe@example.com",

"Password": "pass@221",

"DateOfBirth": "1990-01-01",

"Role": "Admin",

"Status": "Active",

"CreatedAt": "2024-12-11 14:30:00",

"LastModifiedAt": "2024-12-11 14:30:00",

"LastLogin": "2024-12-11 14:30:00"

}

**DELETE**

/User/{id}

Deletes a user from the system specified by the user Id. Admin only operation.

Book endpoints:

**GET**

/Book

returns a list of all books with all details in the system.

/Book/{id}

returns details of a single book specified by book Id.

/Book/filter?

returns filtered books with details. Allowed query parameters – Genre, Author, MinRating, MaxRating, MinPrice, MaxPrice

**POST**

/Book

create a book in the system. Here is a sample payload where Description and Rating can be null:

{

"BookId": "B002",

"Title": "The Great Gatsby",

"Author": "F. Scott Fitzgerald",

"Publisher": "Scribner",

"ISBN": "9780743273565",

"Genre": "Fiction",

"PublishedDate": "1925-04-10",

"Edition": 1,

"Description": "A classic novel set in the 1920s about wealth, love, and the American Dream.",

"Price": 15.99,

"Rating": 4.7

}

**PATCH**

/Book

Modify details of a book in the system. Here is a sample payload where everything can be null except for the book Id:

{

"BookId": "B001",

"Title": "The Great Gatsby",

"Author": "F. Scott Fitzgerald",

"Publisher": "Scribner",

"ISBN": "9780743273565",

"Genre": "Fiction",

"PublishedDate": "1925-04-10",

"Edition": 1,

"Description": "A classic novel set in the 1920s about wealth, love, and the American Dream.",

"Price": 15.99

}

**DELETE**

/Book/{id}

Delete a book from the system using book Id.

Borrowed Book endpoints:

**GET**

**/**availableBooks

Returns a list of all books that can be borrowed with their book details.

/borrowedBooks

Returns a list of all books that are currently borrowed with their book details.

/borrowDetails

Returns a list of all borrowed books with their borrow details only.

/borrowDetails/{id}

Returns borrow details with books details for a single borrowed book specified by borrow Id.

/borrowDetails/filter?

Returns a list of all borrowed books with their borrow details with filter. Allowed query parameters - Min Borrow Date, Max Borrow Date, Min Return Date, Max Return Date, Status.

**POST**

/BorrowedBook/borrow

allows a user to borrow a book. Here is sample payload:

{

"BorrowId": "BR003",

"BookId": "B001",

"UserId": "1234567892"

}

/BorrowedBook/return/{id}

Allows a user to return borrowed book specified by borrow Id.

Reviews Endpoints:

**GET**

/Review/{id}

Returns all reviews of a book specified by book Id.

**POST**

/Review

Allows a user to add a review for a book. Here is sample payload where Description can be empty/null:

{

"ReviewId": "R1",

"BookId": "B001",

"UserId": "1234567892",

"Description": "Testing if update review works",

"Rating": 5

}