**LIBRARY MANAGEMENT SYSTEM**

**Validations:**

**USER:**

1. **Registering the user:**

<http://localhost:8080/users/register>

{

  "username": "nive",

  "password": "nive",

  "role" : "USER"

}

User registered successfully.

1. **Registering the user again:**

{

  "username": "nive",

  "password": "nive",

  "role" : "USER"

}

{

    "error\_message": "Username already exists: nive",

    "error\_type": "UserAlreadyExists",

    "status": "error",

    "timestamp": "2025-08-13T11:12:35.0245744"

}

1. **Registering the user with missed password fields**

{

  "username": "sa",

  "role" : "USER"

}

{

    "error\_message": "rawPassword cannot be null",

    "error\_type": "IllegalArgumentException",

    "status": "error",

    "timestamp": "2025-08-13T11:14:16.5579162"

}

1. **Registering the user with missed role**

{

  "username": "san",

  "password": "san"

}

{

    "error\_message": "Invalid role. Must be ADMIN or USER",

    "error\_type": "RuntimeException",

    "status": "error",

    "timestamp": "2025-08-13T11:15:05.9599858"

}

**Get by username**

<http://localhost:8080/users>

1. **Giving the correct userid**

username: nive in headers

{

    "id": 1,

    "username": "nive",

    "password": "$2a$10$w3YPIZtgGpcocM33oe4l6OZK9TQVn8DMug5M6Zz/ScC0KDxVDaUf.",

    "role": "USER"

}

1. Userid wrong

Username : aaa

{

    "error\_message": "User not found: aaa",

    "error\_type": "UserNotFound",

    "status": "error",

    "timestamp": "2025-08-13T11:18:10.8118355"

}

**Checking whether the user is admin or not:**

<http://localhost:8080/users/isAdmin>

1. Given the unregistered userid:

Username : ASDSADAS

{

    "error\_message": "User not found: ASDSADAS",

    "error\_type": "UserNotFound",

    "status": "error",

    "timestamp": "2025-08-13T11:20:06.1527502"

}

1. Given the non admin userid:

Username : nive

Admin user:false

1. Given the admin userid

Username : san

Admin user:true

**BOOK:**

1. **Add book:**

**1.Adding book with no username(check for role and admin)**

<http://localhost:8080/books>

{

  "title": "sherlock",

  "author": "sherlock",

  "copies" : 4

}

No headers

{

    "error\_message": "Required request header 'username' for method parameter type String is not present",

    "error\_type": "MissingRequestHeaderException",

    "status": "error",

    "timestamp": "2025-08-13T11:25:17.1132131"

}

1. **Adding book with non admin user**

<http://localhost:8080/books>

{

  "title": "sherlock",

  "author": "sherlock",

  "copies" : 4

}

Username : nive

{

    "error\_message": "Only ADMIN can add books",

    "error\_type": "RuntimeException",

    "status": "error",

    "timestamp": "2025-08-13T11:27:03.3753192"

}

1. **Adding book with admin user**

<http://localhost:8080/books>

{

  "title": "sherlock",

  "author": "sherlock",

  "copies" : 4

}

Username : san

Book added successfully.

1. **Adding some missed fields with admin user**

<http://localhost:8080/books>

{

  "title": "sherlock",

  "copies" : 4

}

Username : san

{

    "error\_message": "Validation failed for argument [1] in public org.springframework.http.ResponseEntity<?> com.example.libraryManagement.Controller.BookController.addBook(java.lang.String,com.example.libraryManagement.Entity.Book): [Field error in object 'book' on field 'author': rejected value [null]; codes [NotBlank.book.author,NotBlank.author,NotBlank.java.lang.String,NotBlank]; arguments [org.springframework.context.support.DefaultMessageSourceResolvable: codes [book.author,author]; arguments []; default message [author]]; default message [Author is required]] ",

    "error\_type": "Validation Error",

    "status": "error",

    "timestamp": "2025-08-13T11:29:00.7943231"

}

**Get All Books:**

<http://localhost:8080/books>

[

    {

        "id": 1,

        "title": "sherlock",

        "author": "sherlock",

        "copies": 4

    },

    {

        "id": 2,

        "title": "charles",

        "author": "char",

        "copies": 6

    }

]

**Getbook by id:**

1. Getting book with non matching id:

<http://localhost:8080/books/12>

{

    "error\_message": "Book not found with id: 12",

    "error\_type": "ResourceNotFound",

    "status": "error",

    "timestamp": "2025-08-13T11:31:31.989263"

}

1. Getting book with matching id

<http://localhost:8080/books/2>

{

    "id": 2,

    "title": "charles",

    "author": "char",

    "copies": 6

}

**UpdateBook Details:**

1. **Updating book with no username in header**

<http://localhost:8080/books/1>

{

  "title": "life ",

  "author": "jacob",

  "copies" : 4

}

{

    "error\_message": "Required request header 'username' for method parameter type String is not present",

    "error\_type": "MissingRequestHeaderException",

    "status": "error",

    "timestamp": "2025-08-13T11:33:10.6875682"

}

1. **Updating book with non admin user**

<http://localhost:8080/books/1>

{

  "title": "life ",

  "author": "jacob",

  "copies" : 4

}

Username: nive

{

    "error\_message": "Only ADMIN can update books",

    "error\_type": "RuntimeException",

    "status": "error",

    "timestamp": "2025-08-13T11:34:23.8282921"

}

1. **Updating book with admin user:**

<http://localhost:8080/books/1>

username : san

{

  "title": "life ",

  "author": "jacob",

  "copies" : 4

}

Book with ID 1 updated successfully.

1. **Updating book with missing fields**

<http://localhost:8080/books/1>

{

  "title": "life ",

  "copies" : 4

}

Username : san

{

    "error\_message": "Validation failed for argument [2] in public org.springframework.http.ResponseEntity<?> com.example.libraryManagement.Controller.BookController.updateBook(java.lang.Long,java.lang.String,com.example.libraryManagement.Entity.Book): [Field error in object 'book' on field 'author': rejected value [null]; codes [NotBlank.book.author,NotBlank.author,NotBlank.java.lang.String,NotBlank]; arguments [org.springframework.context.support.DefaultMessageSourceResolvable: codes [book.author,author]; arguments []; default message [author]]; default message [Author is required]] ",

    "error\_type": "Validation Error",

    "status": "error",

    "timestamp": "2025-08-13T11:36:17.9535583"

}

**5.if id is not present**

<http://localhost:8080/books/5>

username : san

{

  "title": "life ",

  "author": "jacob",

  "copies" : 4

}

{

    "error\_message": "Book not found with id: 5",

    "error\_type": "ResourceNotFound",

    "status": "error",

    "timestamp": "2025-08-13T11:37:16.4705903"

}

**Deleting the book:**

1. **Deleting the book with no username in header**

<http://localhost:8080/books/5>

{

    "error\_message": "Required request header 'username' for method parameter type String is not present",

    "error\_type": "MissingRequestHeaderException",

    "status": "error",

    "timestamp": "2025-08-13T11:38:41.8601935"

}

1. **Deleting the book with non admin user**

<http://localhost:8080/books/1>

username : nive

{

    "error\_message": "Only ADMIN can delete books",

    "error\_type": "RuntimeException",

    "status": "error",

    "timestamp": "2025-08-13T11:39:29.3240357"

}

1. **Deleting the book with admin access**

<http://localhost:8080/books/1>

username : san

Book with ID 1 deleted successfully.

1. **Deleting the book with irrelevant id:**

<http://localhost:8080/books/12>

username : san

{

    "error\_message": "Book not found with id: 12",

    "error\_type": "ResourceNotFound",

    "status": "error",

    "timestamp": "2025-08-13T11:41:07.5035403"

}