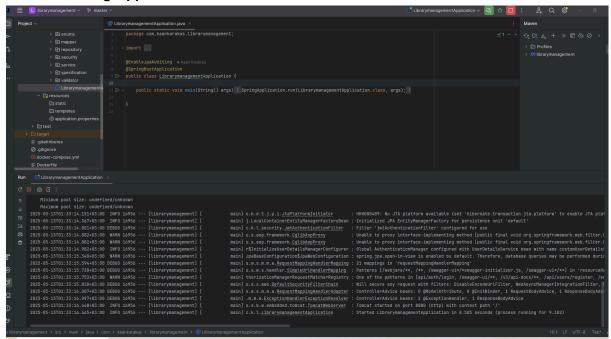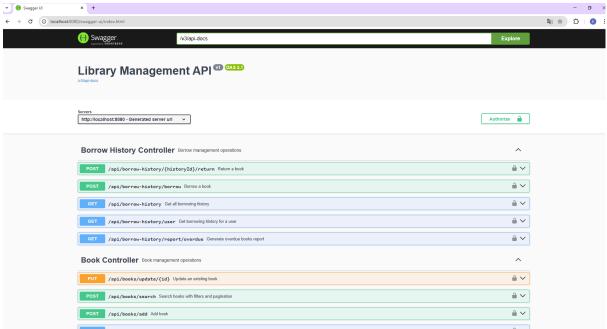# Live Application Screenshots

Kaan Karakaş

- ## Running Application

- ## Swagger

- Register User

POST /api/users/register Register a new user

Registers a new user with provided information like username, email, password, and role.

Parameters

Cancel   Reset

No parameters

Request body *required*

application/json

```
{
  "name": "Kaan Karakas",
  "username": "kaank35",
  "email": "kaan.k@hotmail.com",
  "password": "kaan1234567",
  "role": "LIBRARIAN"
}
```

Execute   Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/api/users/register' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "name": "Kaan Karakas",
  "username": "kaank35",
  "email": "kaan.k@hotmail.com",
  "password": "kaan1234567",
  "role": "LIBRARIAN"
}'
```

Request URL

```
http://localhost:8080/api/users/register
```

Server response

Code	Details
200	Response body

```
{
  "id": 14,
  "name": "Kaan Karakas",
  "username": "kaank35",
  "email": "kaan.k@hotmail.com",
  "role": "LIBRARIAN",
  "userStatus": "ACTIVE"
}
```

Download

- Login - Auth

Auth Controller Auth management operations

POST /api/auth/login

Parameters

Cancel   Reset

No parameters

Request body *required*

application/json

```
{
  "username": "kaank35",
  "password": "kaan1234567"
}
```

Execute   Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/api/auth/login' \
  -H 'accept: application/json;charset=UTF-8' \
  -H 'Content-Type: application/json' \
  -d '{
  "username": "kaank35",
  "password": "kaan1234567"
}'
```

Request URL

```
http://localhost:8080/api/auth/login
```

Server response

Code	Details
200	Response body

```
{
  "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJrYWFuazM1IiwiaWF0IjoxNzQ3MDg5NjExLCJleHAiOjE3NDcwOTMyMTF9.Ri18D7pfKUd7GmgEVqNLeyeSXVgprMY87r57VgLrwgU"
}
```

Download

- Add Book

POST /api/books/add Add book

Add a new book to the library

Parameters

Cancel

No parameters

Request body *required*

application/json

```
{
  "title": "The Great Gatsby",
  "author": "F. Scott Fitzgerald",
  "isbn": "9784507",
  "publicationDate": "1925-04-10",
  "genre": "FICTION"
}
```

| Execute | Clear |

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/api/books/add' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJrYWFuazMlIiwiaWF0IjoxNzQ3MDg5NjExLCJleHAiOjE3NDcwOTMyMTF9.RilBD7pfKUd7GmgEVqNLeyeSXYqprMY87rS7VgLrwgU' \
  -H 'Content-Type: application/json' \
  -d '{
  "title": "The Great Gatsby",
  "author": "F. Scott Fitzgerald",
  "isbn": "9784507",
  "publicationDate": "1925-04-10",
  "genre": "FICTION"
}'
```

Request URL

```
http://localhost:8080/api/books/add
```

Server response

Code	Details
201	Response body

```
{
  "id": 11,
  "title": "The Great Gatsby",
  "author": "F. Scott Fitzgerald",
  "isbn": "9784507",
  "publicationDate": "1925-04-10",
  "genre": "FICTION",
  "bookStatus": "ACTIVE"
}
```
Download

Response headers

- Get Book By Id

GET /api/books/{id} Get book details by ID

Parameters

Cancel

Name	Description
id * *required* integer($int64) (path)	11

| Execute | Clear |

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/api/books/11' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJrYWFuazMlIiwiaWF0IjoxNzQ3MDg5NjExLCJleHAiOjE3NDcwOTMyMTF9.RilBD7pfKUd7GmgEVqNLeyeSXYqprMY87rS7VgLrwgU'
```

Request URL

```
http://localhost:8080/api/books/11
```

Server response

Code	Details
200	Response body

```
{
  "id": 11,
  "title": "The Great Gatsby",
  "author": "F. Scott Fitzgerald",
  "isbn": "9784507",
  "publicationDate": "1925-04-10",
  "genre": "FICTION",
  "bookStatus": "ACTIVE"
}
```
Download

Response headers

```
cache-control: no-cache,no-store,max-age=0,must-revalidate
connection: keep-alive
content-type: application/json;charset=UTF-8
date: Mon,12 May 2025 22:41:51 GMT
expires: 0
keep-alive: timeout=60
pragma: no-cache
transfer-encoding: chunked
x-content-type-options: nosniff
x-frame-options: DENY
```

- Update Book

PUT   /api/books/update/{id}   Update an existing book

Parameters   Cancel   Reset

Name   Description

id * required
integer($int64)   [11]
(path)

Request body required   application/json ▼

{
  "title": "Harry Potter"
}

[Execute   ]   [Clear   ]

Responses

Curl

curl -X 'PUT' \
  'http://localhost:8080/api/books/update/11' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJrYWFuazM1IiwiaWF0IjoxNzQ3MDg5NjExLCJleHAiOjE3NDcwOTMyMTF9.Ril8D7pfKUd7GmgEVqNLeyeSXYqprHVB7r57VgLrwgU' \
  -H 'Content-Type: application/json' \
  -d '{
  "title": "Harry Potter"
}'

Request URL

http://localhost:8080/api/books/update/11

Server response

Code   Details

200
        Response body

        {
          "id": 11,
          "title": "Harry Potter",
          "author": "F. Scott Fitzgerald",
          "isbn": "9784507",
          "publicationDate": "1925-04-10",
          "genre": "FICTION",
          "bookStatus": "ACTIVE"
        }

                                                        Download

        Response headers

- Search Book

POST   /api/books/search   Search books with filters and pagination

Parameters   Cancel   Reset

No parameters

Request body required   application/json ▼

{
  "title": "Harry Potter",
  "page": 0,
  "size": 10
}

[Execute   ]   [Clear   ]

Responses

Curl

curl -X 'POST' \
  'http://localhost:8080/api/books/search' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJrYWFuazM1IiwiaWF0IjoxNzQ3MDg5NjExLCJleHAiOjE3NDcwOTMyMTF9.Ril8D7pfKUd7GmgEVqNLeyeSXYqprHVB7r57VgLrwgU' \
  -H 'Content-Type: application/json' \
  -d '{
  "title": "Harry Potter",
  "page": 0,
  "size": 10
}'

Request URL

http://localhost:8080/api/books/search

Server response

Code   Details

200
        Response body

        {
          "books": [
            {
              "id": 11,
              "title": "Harry Potter",
              "author": "F. Scott Fitzgerald",
              "isbn": "9784507",
              "publicationDate": "1925-04-10",
              "genre": "FICTION",
              "bookStatus": "ACTIVE"
            }
          ],
          "totalElements": 1,
          "totalPages": 1,
          "page": 0,
          "size": 10
                                                        Download

- Delete Book

DELETE `/api/books/delete/{id}` Soft delete a book 🔒 ∧

Parameters   Cancel

Name   Description

id * required
integer($int64)   `11`
(path)

            Execute   Clear

Responses

Curl
```
curl -X 'DELETE' \
  'http://localhost:8080/api/books/delete/11' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJrYWFuazM1IiwiaWF0IjoxNzQ3MDg5NjExLCJleHAiOjE3NDcwOTMyMTF9.Ril8D7pfKUd7GmgEVqNLeye5XYqprMY87r57VgLrwgU'
```

Request URL
```
http://localhost:8080/api/books/delete/11
```

Server response

Code   Details

200
         Response body
```
{
  "id": 11,
  "title": "Harry Potter",
  "author": "F. Scott Fitzgerald",
  "isbn": "9784507",
  "publicationDate": "1925-04-10",
  "genre": "FICTION",
  "bookStatus": "DELETED"
}
```
                                                                     Download

         Response headers

- Get User By Id

GET `/api/users/{id}` Get user by ID 🔒 ∧

Retrieves detailed information of a user by their ID.

Parameters   Cancel

Name   Description

id * required
integer($int64)   `2`
(path)

            Execute   Clear

Responses

Curl
```
curl -X 'GET' \
  'http://localhost:8080/api/users/2' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJrYWFuazM1IiwiaWF0IjoxNzQ3MDg5NjExLCJleHAiOjE3NDcwOTMyMTF9.Ril8D7pfKUd7GmgEVqNLeye5XYqprMY87r57VgLrwgU'
```

Request URL
```
http://localhost:8080/api/users/2
```

Server response

Code   Details

200
         Response body
```
{
  "id": 2,
  "name": "Alp Krakas",
  "username": "akppas",
  "email": "alp.kara@gmail.com",
  "role": "PATRON",
  "userStatus": "ACTIVE"
}
```
                                                                     Download

         Response headers

DELETE `/api/books/delete/{id}` Soft delete a book

id * required
integer($int64)
(path)

- Update User

PUT /api/users/update/{id} Update user by ID 🔒 ⌃

Updates user details for a given ID. Checks for uniqueness of username and email before updating.

Parameters   Cancel   Reset

Name   Description

id * required
integer($int64)   `2`
(path)

Request body required   application/json ⌄

```
{
  "name": "Mehmet"
}
```

| Execute | Clear |

Responses

Curl

```
curl -X 'PUT' \
  'http://localhost:8080/api/users/update/2' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJrYWFuazM1IiwiaWF0IjoxNzQ3MDg5NjExLCJleHAiOjE3NDcwOTMyMTF9.Ril8D7pfKUd7GmgEVqNLeyeSXYqprMY87rS7VgLrwgU' \
  -H 'Content-Type: application/json' \
  -d '{
  "name": "Mehmet"
}'
```

Request URL

```
http://localhost:8080/api/users/update/2
```

Server response

Code   Details

200
        Response body
        ```
        {
          "id": 2,
          "name": "Mehmet",
          "username": "akppas",
          "email": "alp.kara@gmail.com",
          "role": "PATRON",
          "userStatus": "ACTIVE"
        }
        ```
        Download

        Response headers

- Delete User

DELETE /api/users/delete/{id} Soft delete user by ID 🔒 ⌃

Performs a soft delete (marks user as deleted) instead of physically removing them from the database.

Parameters   Cancel

Name   Description

id * required
integer($int64)   `2`
(path)

| Execute | Clear |

Responses

Curl

```
curl -X 'DELETE' \
  'http://localhost:8080/api/users/delete/2' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJrYWFuazM1IiwiaWF0IjoxNzQ3MDg5NjExLCJleHAiOjE3NDcwOTMyMTF9.Ril8D7pfKUd7GmgEVqNLeyeSXYqprMY87rS7VgLrwgU'
```

Request URL

```
http://localhost:8080/api/users/delete/2
```

Server response

Code   Details

200
        Response body
        ```
        {
          "id": 2,
          "name": "Mehmet",
          "username": "akppas",
          "email": "alp.kara@gmail.com",
          "role": "PATRON",
          "userStatus": "DELETED"
        }
        ```
        Download

        Response headers

- **Borrow Book**

POST /api/borrow-history/borrow Borrow a book

Allows a user to borrow a book if available and eligible

Parameters   Cancel

No parameters

Request body required   application/json

{
  "bookId": 5,
  "borrowDate": "2025-05-05",
  "dueDate": "2025-05-15"
}

Execute   Clear

Responses

Curl

curl -X 'POST' \
  'http://localhost:8080/api/borrow-history/borrow' \
  -H 'accept: application/json;charset=UTF-8' \
  -H 'Authorization: Bearer eyJzdWIiOiJrYWFuazM1MyIsImlhdCI6MTc0NzA5MDY1OSwiZXhwIjoxNzQ3MDk0MjU5fQ.YJOdxmDSrOHKPU9prkbTCahAmx_aKqFzceDvBA17cgk' \
  -H 'Content-Type: application/json' \
  -d '{
  "bookId": 5,
  "borrowDate": "2025-05-05",
  "dueDate": "2025-05-15"
}'

Request URL

http://localhost:8080/api/borrow-history/borrow

Server response

Code   Details

201
            Response body

            {
              "id": 3,
              "userId": 15,
              "bookId": 5,
              "userName": "Kaan Karakas",
              "bookTitle": "Book Title",
              "borrowDate": "2025-05-05",
              "dueDate": "2025-05-15",
              "returnDate": null,
              "isReturned": false
            }
                                                          Download

- **Return Book**

Borrow History Controller Borrow management operations

POST /api/borrow-history/{historyId}/return Return a book

Marks a borrowed book as returned and updates availability

Parameters   Cancel

Name   Description

historyId * required
integer($int64)   3
(path)

Request body required   application/json

{
  "returnDate": "2025-05-05"
}

Execute   Clear

Responses

Curl

curl -X 'POST' \
  'http://localhost:8080/api/borrow-history/3/return' \
  -H 'accept: application/json;charset=UTF-8' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJrYWFuazM1MyIsImlhdCI6MTc0NzA5MDY1OSwiZXhwIjoxNzQ3MDk0MjU5fQ.YJOdxmDSrOHKPU9prkbTCahAmx_aKqFzceDvBA17cgk' \
  -H 'Content-Type: application/json' \
  -d '{
  "returnDate": "2025-05-05"
}'

Request URL

http://localhost:8080/api/borrow-history/3/return

Server response

Code   Details

200
            Response body

            {
              "id": 3,
              "userId": 15,
              "bookId": 5,
              "userName": "Kaan Karakas",
              "bookTitle": "Book Title",
              "borrowDate": "2025-05-05",
              "dueDate": "2025-05-15",
              "returnDate": "2025-05-05",
              "isReturned": true
            }
                                                          Download

            Response headers

- All Borrow History

GET `/api/borrow-history` Get all borrowing history

Returns borrowing history for all users (for librarians)

Parameters   Cancel

No parameters

| Execute | Clear |

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/api/borrow-history' \
  -H 'accept: application/json;charset=UTF-8' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJrYWFuazM1IiwiaWF0IjoxNzQ3MDkwODIxLCJleHAiOjE3NDcwOTQ0MjF9.tiPYaGGs6J52DmdMbVG3HRoFzCDxSOWTWGh72dIN1Sw'
```

Request URL

```
http://localhost:8080/api/borrow-history
```

Server response

Code	Details
200	Response body

```
[
  {
    "id": 1,
    "userId": 5,
    "bookId": 2,
    "userName": "Kaan Karakas",
    "bookTitle": "The Great Gatsby",
    "borrowDate": "2025-05-05",
    "dueDate": "2025-05-15",
    "returnDate": "2025-05-05",
    "isReturned": true
  },
  {
    "id": 2,
    "userId": 12,
    "bookId": 3,
    "userName": "newUser",
    "bookTitle": "The Great Gatsby",
    "borrowDate": "2023-12-12",
    "dueDate": "2024-02-02",
    "returnDate": "2025-05-12",
    "isReturned": true
  },
  {
    "id": 3,
    "userId": 15,
    "bookId": 5,
    "userName": "Kaan Karakas",
    "bookTitle": "Book Title",
```

Response headers

- Get Borrow History For User

GET `/api/borrow-history/user` Get borrowing history for a user

Returns the complete borrowing history for the given user ID

Parameters   Cancel

No parameters

| Execute | Clear |

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/api/borrow-history/user' \
  -H 'accept: application/json;charset=UTF-8' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJrYWFuazM1IiwiaWF0IjoxNzQ3MDkwODIxLCJleHAiOjE3NDcwOTQ0MjF9.tiPYaGGs6J52DmdMbVG3HRoFzCDxSOWTWGh72dIN1Sw'
```

Request URL

```
http://localhost:8080/api/borrow-history/user
```

Server response

Code	Details
200	Response body

```
[]
```

Response headers

```
cache-control: no-cache,no-store,max-age=0,must-revalidate
connection: keep-alive
content-type: application/json;charset=UTF-8
date: Mon,12 May 2025 23:01:13 GMT
expires: 0
keep-alive: timeout=60
pragma: no-cache
transfer-encoding: chunked
x-content-type-options: nosniff
x-frame-options: DENY
x-xss-protection: 0
```

Responses

- Generate Report

GET /api/borrow-history/report/overdue Generate overdue books report 🔒 ∧

Generates a report of all books that are overdue with user and book details

| Parameters | Cancel |

No parameters

| Execute | Clear |

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/api/borrow-history/report/overdue' \
  -H 'accept: text/plain' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJrYWFuazM1IiwiaWF0IjoxNzQ3MDkwODIxLCJleHAiOjE3NDcwOTQ0MjF9.tiPYaGGs6J52DmdMbVG3HRoFzCDxSOWTWGh72dIN1Sw'
```

Request URL

```
http://localhost:8080/api/borrow-history/report/overdue
```

Server response

Code	Details
200	**Response body**

```
OVERDUE BOOKS REPORT--------------------Total Overdue Records: 0
Users with Overdues: 0

Details:

Generated At: 2025-05-13T02:01:49.420939Z
```

Download

Response headers

```
cache-control: no-cache,no-store,max-age=0,must-revalidate
connection: keep-alive
content-length: 140
content-type: text/plain;charset=UTF-8
date: Mon,12 May 2025 23:01:49 GMT
expires: 0
keep-alive: timeout=60
pragma: no-cache
x-content-type-options: nosniff
x-frame-options: DENY
x-xss-protection: 0
```

Responses