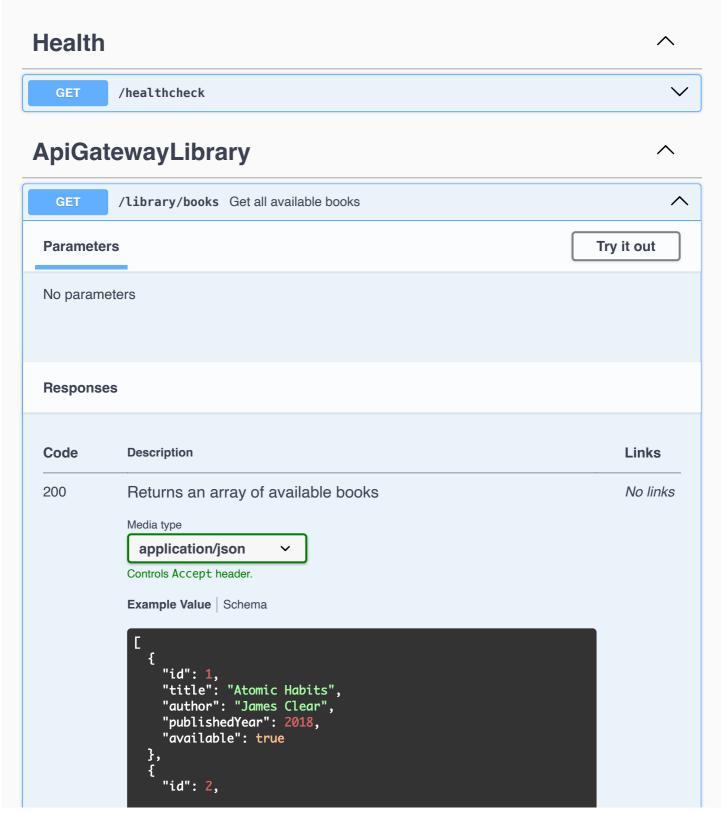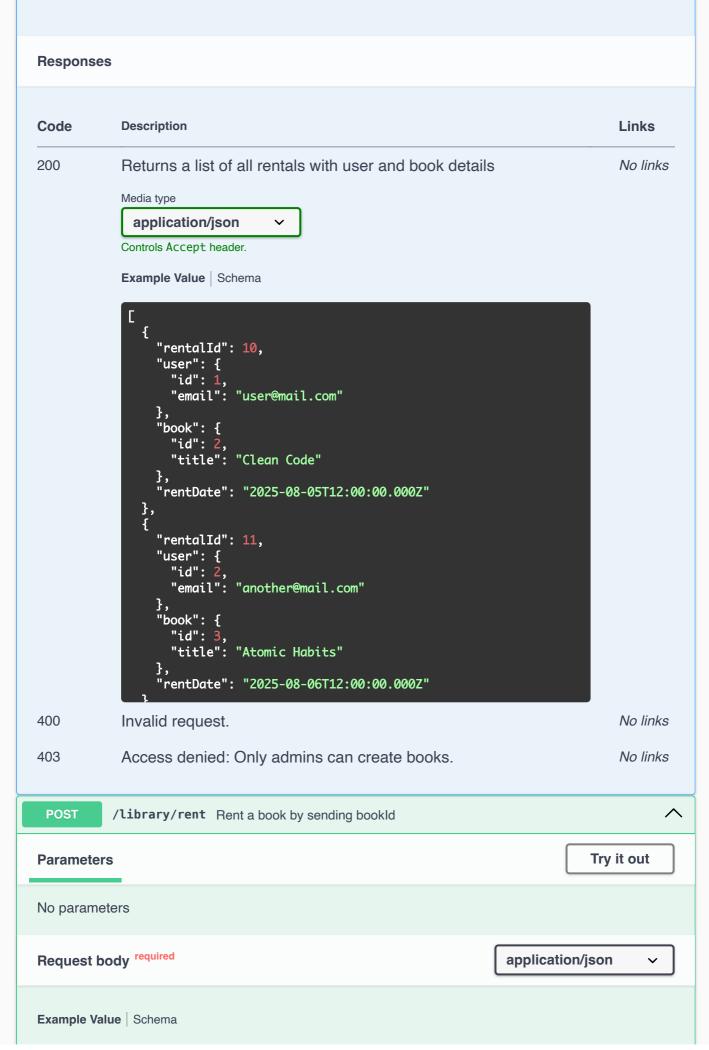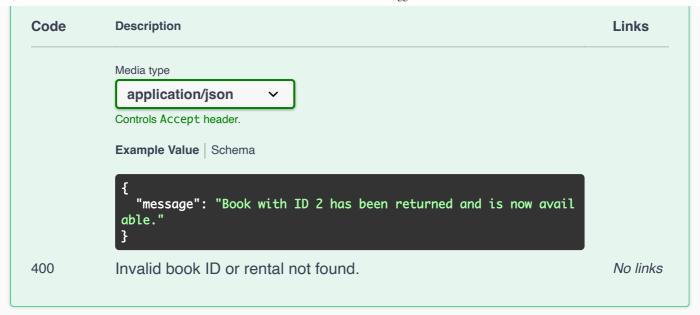# API Gateway 1.0.0 OAS 3.0

This is the API Gateway for the NestJS Microservices Bookstore project. It serves as the single entry point, forwarding requests to Books, Library, Users, and Roles services.

Health   ∧

GET	/healthcheck	∨

ApiGatewayLibrary   ∧

GET	/library/books Get all available books	∧

Parameters   **Try it out**

No parameters

Responses

Code	Description	Links
200	Returns an array of available books	*No links*

Media type

application/json   ∨

Controls `Accept` header.

Example Value | Schema

```
[
  {
    "id": 1,
    "title": "Atomic Habits",
    "author": "James Clear",
    "publishedYear": 2018,
    "available": true
  },
  {
    "id": 2,
```

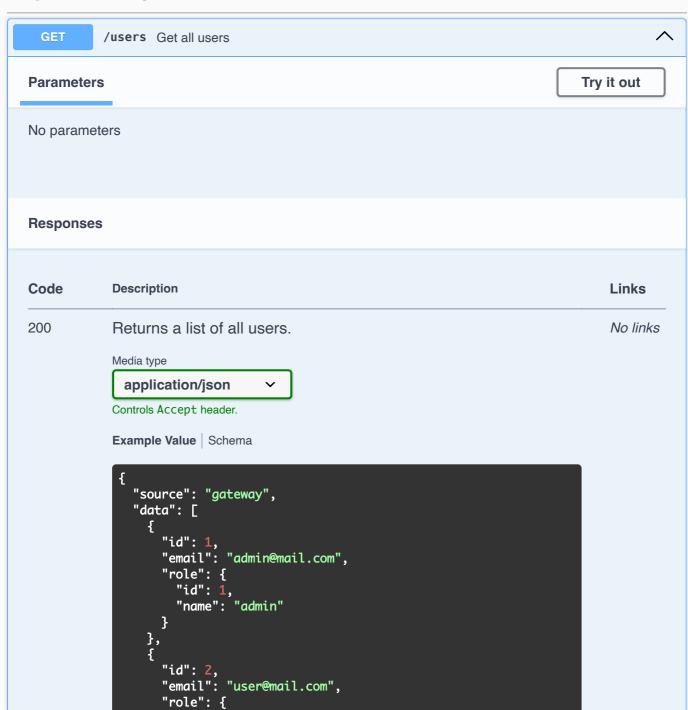| Code	Description	Links

```
      "title": "Clean Code",
      "author": "Robert C. Martin",
      "publishedYear": 2008,
      "available": true
    }
  ]
```

| 400 | Invalid data provided. | *No links* |

GET **/library/myBooks** Get all books rented by the current user ∧

Parameters   **Try it out**

No parameters

Responses

Code	Description	Links
200	Returns an array of rentals belonging to the current user	*No links*

Media type

application/json   ∨

Controls `Accept` header.

Example Value | Schema

```
[
  {
    "rentalId": 10,
    "book": {
      "id": 1,
      "title": "Clean Code",
      "author": "Robert C. Martin"
    },
    "rentDate": "2025-08-05T12:00:00.000Z"
  }
]
```

| 400 | Invalid data provided. | *No links* |

GET **/library/repository** List all rentals and their status (Admin Only) ∧

Parameters   **Try it out**

No parameters

Responses

Code	Description	Links
200	Returns a list of all rentals with user and book details	*No links*

Media type

application/json ⌄

Controls `Accept` header.

Example Value | Schema

```
[
  {
    "rentalId": 10,
    "user": {
      "id": 1,
      "email": "user@mail.com"
    },
    "book": {
      "id": 2,
      "title": "Clean Code"
    },
    "rentDate": "2025-08-05T12:00:00.000Z"
  },
  {
    "rentalId": 11,
    "user": {
      "id": 2,
      "email": "another@mail.com"
    },
    "book": {
      "id": 3,
      "title": "Atomic Habits"
    },
    "rentDate": "2025-08-06T12:00:00.000Z"
  }
```

Code	Description	Links
400	Invalid request.	*No links*
403	Access denied: Only admins can create books.	*No links*

POST	**/library/rent** Rent a book by sending bookId	⌃

Parameters

Try it out

No parameters

Request body **required**

application/json ⌄

Example Value | Schema

```
{
  "bookId": 120
}
```

Responses

Code	Description	Links
201	Successfully rented the book	No links

Media type

```
application/json                    ⌄
```

Controls `Accept` header.

Example Value | Schema

```
{
  "rentalId": 15,
  "user": {
    "id": 1,
    "email": "user@mail.com"
  },
  "book": {
    "id": 2,
    "title": "Clean Code"
  },
  "rentDate": "2025-08-05T12:00:00.000Z"
}
```

POST   **/library/return/{bookId}**   Return a rented book by its bookId   ∧

Parameters

Try it out

Name	Description
bookId * required **string** *(path)*	ID of the book to return `2`

Responses

Code	Description	Links
200	Book returned successfully	No links

Code	Description	Links

Media type

<div style="border:1px solid green; display:inline-block; padding:4px">application/json ▾</div>

Controls Accept header.

Example Value | Schema

```
{
  "message": "Book with ID 2 has been returned and is now available."
}
```

400	Invalid book ID or rental not found.	*No links*

ApiGatewayUsers ⌃

GET **/users** Get all users ⌃

Parameters   Try it out

No parameters

Responses

Code	Description	Links
200	Returns a list of all users.	*No links*

Media type

<div style="border:1px solid green; display:inline-block; padding:4px">application/json ▾</div>

Controls Accept header.

Example Value | Schema

```
{
  "source": "gateway",
  "data": [
    {
      "id": 1,
      "email": "admin@mail.com",
      "role": {
        "id": 1,
        "name": "admin"
      }
    },
    {
      "id": 2,
      "email": "user@mail.com",
      "role": {
```

Code	Description	Links

```
        "id": 2,
        "name": "user"
      }
    }
  ]
}
```

ApiGatewayRoles ⌃

GET /roles Fetch all roles ⌃

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Returns a list of all roles.	*No links*

Media type

application/json ⌄

Controls `Accept` header.

Example Value | Schema

```
[
  {
    "id": 1,
    "name": "admin"
  },
  {
    "id": 2,
    "name": "user"
  }
]
```

POST /roles/create Create a new role ⌃

Parameters

Try it out

No parameters

Request body required

<div style="text-align: right">application/json ⌄</div>

Example Value | Schema

```
{
  "name": "manager"
}
```

Responses

Code	Description	Links
201	Role created successfully.	No links

Media type

application/json ⌄

Controls `Accept` header.

Example Value | Schema

```
{
  "id": 3,
  "name": "manager"
}
```

| 400 | Creation failed: Invalid data or role already exists. | No links |

ApiGatewayBooks   ⌃

POST   **/books/create**   Create a new book (Admin only)   ⌃
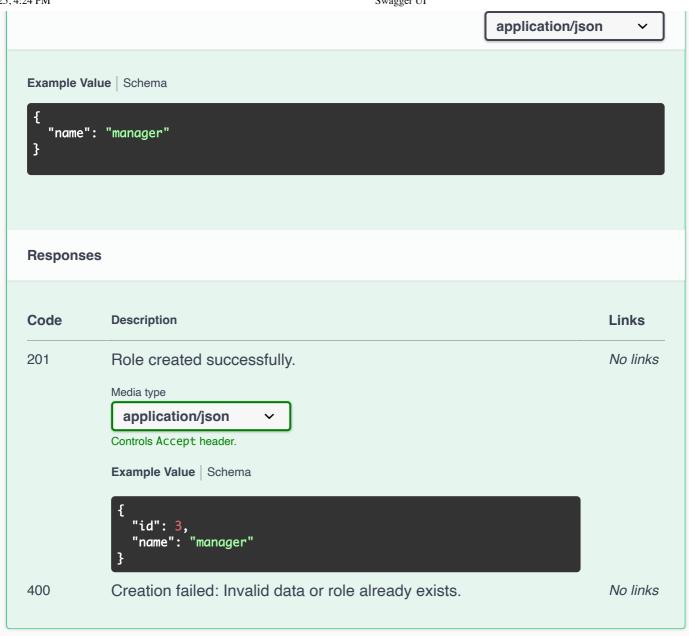
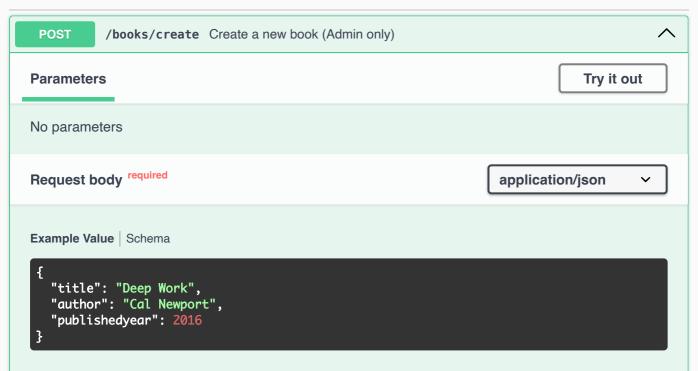**Parameters**   Try it out

No parameters

Request body required   application/json ⌄

Example Value | Schema

```
{
  "title": "Deep Work",
  "author": "Cal Newport",
  "publishedyear": 2016
}
```

Responses

Code	Description	Links
200	Book created successfully.	*No links*

Media type

┌─────────────────────────────┐
│ **application/json**   ⌄ │
└─────────────────────────────┘

Controls Accept header.

Example Value | Schema

```
{
  "id": 3,
  "title": "Deep Work",
  "author": "Cal Newport",
  "publishedYear": 2016
}
```

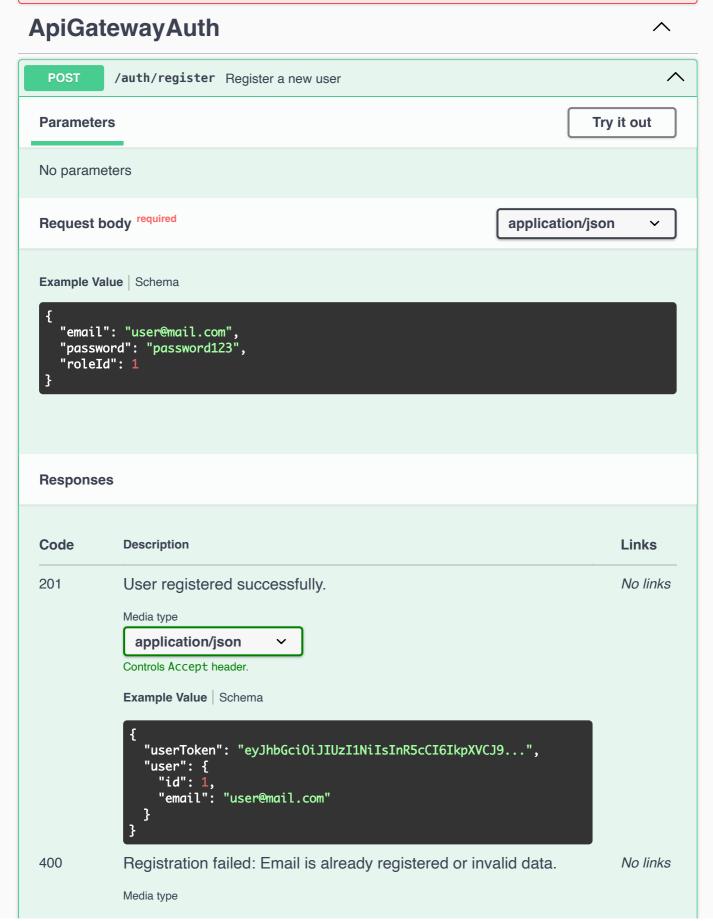| Code	Description	Links
400	Invalid data provided.	*No links*
403	Access denied: Only admins can create books.	*No links*

PUT   **/books/update/{id}**   Update a book by ID (Admin only)   ⌃

Parameters

┌──────────────┐
│ Try it out   │
└──────────────┘

Name	Description
id * required **string** *(path)*	┌──────────────────────────────┐ │ id │ └──────────────────────────────┘

Request body required   ┌──────────────────────────┐
                                                        │ application/json   ⌄ │
                                                        └──────────────────────────┘

Example Value | Schema

```
{
  "title": "Atomic Habits (Revised Edition)",
  "author": "James Clear",
  "publishedyear": 2025
}
```

Responses

Code	Description	Links
200	Book updated successfully.	*No links*

Media type

application/json   ⌄

Controls `Accept` header.

Example Value | Schema

```
{
  "id": 3,
  "title": "Deep Work (Updated Edition)",
  "author": "Cal Newport",
  "publishedYear": 2025
}
```

400	Invalid data or book not found.	*No links*
403	Access denied: Only admins can update books.	*No links*

DELETE   **/books/delete/{id}**   Delete a book by ID (Admin only)   ∧

Parameters   **Try it out**

Name	Description
id * required **string** *(path)*	id

Responses

Code	Description	Links
200	Book deleted successfully.	*No links*

Media type

application/json   ⌄

Controls `Accept` header.

Example Value | Schema

```
{
    "message": "Book with ID 2 has been deleted."
}
```

Code	Description	Links
400	Book not found.	*No links*
403	Access denied: Only admins can delete books.	*No links*

ApiGatewayAuth ⌃

| **POST** | **/auth/register** Register a new user | ⌃ |

Parameters   [**Try it out**]

No parameters

Request body required   [application/json ⌄]

Example Value | Schema

```
{
  "email": "user@mail.com",
  "password": "password123",
  "roleId": 1
}
```

Responses

Code	Description	Links
201	User registered successfully.	*No links*

Media type

[**application/json** ⌄]

Controls `Accept` header.

Example Value | Schema

```
{
  "userToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "user": {
    "id": 1,
    "email": "user@mail.com"
  }
}
```

| 400 | Registration failed: Email is already registered or invalid data. | *No links* |

Media type

Code	Description	Links

application/json ⌄

Example Value | Schema

```
{
  "statusCode": 400,
  "message": "Registration failed: This email is already regist
ered!",
  "error": "Bad Request"
}
```

POST **/auth/login** User login   ⌃

Parameters   Try it out

No parameters

Request body **required**   application/json ⌄

Example Value | Schema

```
{
  "email": "user@mail.com",
  "password": "password123"
}
```

Responses

Code	Description	Links
200	User logged in successfully.	*No links*

Media type

application/json ⌄

Controls `Accept` header.

Example Value | Schema

```
{
  "message": "SUCCESS: Logged in!",
  "user": {
    "id": 1,
    "email": "user@mail.com",
    "role": "admin"
  }
}
```

Code	Description	Links
401	Login failed: Invalid email or password.	*No links*

Media type

application/json   ∨

Example Value | Schema

```
{
  "statusCode": 401,
  "message": "Login failed: Password is incorrect!",
  "error": "Unauthorized"
}
```

POST **/auth/logout** User logout   ∧

Parameters   **Try it out**

No parameters

Responses

Code	Description	Links
200	User logged out successfully.	*No links*

Media type

application/json   ∨

Controls `Accept` header.

Example Value | Schema

```
{
  "message": "SUCCESS: Logged out!"
}
```

Code	Description	Links
401	Unauthorized: No active session.	*No links*

Media type

application/json ⌄

Example Value | Schema

```
{
  "statusCode": 401,
  "message": "Unauthorized: No active session.",
  "error": "Unauthorized"
}
```

GET	/auth/whoami Get current authenticated user	⌃

Parameters   Try it out

No parameters

Responses

Code	Description	Links
200	Returns the details of the currently logged-in user.	*No links*

Media type

application/json ⌄

Controls `Accept` header.

Example Value | Schema

```
{
  "id": 1,
  "email": "user@example.com",
  "role": {
    "name": "admin"
  }
}
```

| 401 | Unauthorized: Token is missing or invalid. | *No links* |

Media type

application/json ⌄

Example Value | Schema

```
{
  "statusCode": 401,
```

Code	Description	Links

```
    "message": "Unauthorized",
    "error": "Unauthorized"
}
```

Schemas   ∧

CreateRentalDto ⌄ {
  bookId*
              ⌄ number
              example: 120
              Id of the book

}

CreateRoleDto ⌄ {
  name*
              ⌄ string
              example: manager
              Name of the role to create (ex: admin, user)

}

CreateBookDto ⌄ {
  title*
              ⌄ string
              example: Deep Work
              Title of the book

  author*
              ⌄ string
              example: Cal Newport
              Author of the book

  publishedyear
              ⌄ number
              example: 2016
              Year of the book was published

}

UpdateBookDto ˅ {
   title
                        ˅ **string**
                      example: Atomic Habits (Revised Edition)
                      Updated title of the book

   author
                        ˅ **string**
                      example: James Clear
                      Updated author name

   publishedyear
                        ˅ **number**
                      example: 2025
                      Updated year of publication

}

CreateUserDto ˅ {
   email*
                        ˅ **string**
                      example: user@mail.com
                      Email address of the user

   password*
                        ˅ **string**
                      example: password123
                      Password for the user account

   roleId
                        ˅ **number**
                      example: 1
                      Role ID of the user (optional)

}

LoginUserDto ˅ {
   email*
                        ˅ **string**
                      example: user@mail.com
                      Email address of the user

   password*
                        ˅ **string**
                      example: password123
                      Password for the user account

}