JBDL-53

Digital Library

29th July, 2023

OVERVIEW

GOALS

1. You have to create an Entity Relationship Diagram based on a database schema.
2. Create API/endpoints as per Controllers.
3. Implement Security [Hold For Now].

SPECIFICATIONS

Entity Involved:

1. Book
2. Author
3. User
4. Student
5. Card
6. Transaction

Database Schema

Author

1. Id
2. Name
3. Email
4. Age
5. Country

Book

1. Id

2. Name
3. author_id(mapping)
4. Number of Pages
5. Language
6. Available
7. Genre
8. ISBN Number
9. Published Date

For Example:

Aditya, Rohan has Java

Aditya, John on spring Boot

Aditya on Hibernate

User (Hold It for Now)

1. Id
2. Name
3. Username
4. Email
5. Password
6. Authority

Student

1. Id
2. Age
3. Name
4. Country
5. Email
6. Phone Number
7. CreatedOn
8. UpdatedOn
9. CardId

Card

1. Id
2. Status

3. Email
4. ValidUpto
5. CreatedOn
6. UpdatedOn

Transaction

1. Id
2. CardID
3. BookID
4. TransactionDate
5. BookDueDate
6. IsIssued
7. IsReturned
8. FineAmount
9. Status
10. CreatedOn
11. UpdatedOn

Relations

Source	**Destination**	**Relation**
Student	Card	1-1
Book	Author	1-1
Book	Transaction	1-N
Card	Transaction	1-N

Controller/Endpoints/API

Student Controller

Request Mapping: /student/<Endpoint>

1. CRUD API for Student
2. Allow Student to update Password (HOLD For now)

Book Controller

Request Mapping: /book/<Endpoint>

1. CRUD API for Books

Authors Controller

Request Mapping: /author/<Endpoint>

1. CRUD API for Author

Transactions Controller

Request Mapping: /transact/<Endpoint>

1. Issue Book (Parameters: BookId, Card Id)
    a. If the Card/Student is active
    b. If the book is available
    c. Each card can have utmost 3 books issued, check if this book does not go beyond the limit.
    d. If all of the above are ok, then issue the book,
        i. Create a new transaction
        ii. Mark the book status as Unavailable
        iii. Kafka: Drop a kafka message for an email to be sent to the user, email Id has to be passed with the details of book issued and the last date of returning the book.
    e. If any of the above return an error, insert a transaction as failure.
2. Return Book (Parameters: BookId, Card Id)
    a. If the BookId and Card Id is a valid combination
    b. If the Card/Student is active
    c. If all of the above are ok, then issue the book,
        i. Make the book is available
        ii. Each card can have utmost 3 books issued, check if this book does not go beyond the limit.
        iii. In the transaction table, you have to set card_id as null and
        iv. Calculate if the book is delayed, calculate the fine and send a kafka to send email.
3. Get Details of a Transaction

Reports Controller [Optional]

Request Mapping: /report/<Endpoint>

1. All Books Issued on a specific date (Parameters: StartDate, EndDate)
2. Total Fine Collected by Date, Date Range (Parameters: StartDate, EndDate)
3. Total Students Signed Up on a day (Parameters: StartDate, EndDate)
4. Number of books returned (Parameters: StartDate, EndDate)
5. List of all active Students
6. List of all Issued Books