CST438 Assignment 6

Design and code a REST service that could be used by a library to process book checkouts and returns. This api could be used by library equipment to scan and checkout and return books and by a library app for a library patron to get a list of books that are currently checked out under their patron id.

http verb	URL	Return code and data
PUT	/book/{bookId}/checkout/{patronId}	Success: response code 200 OK response code 404 : bookId or patronId invalid
PUT	/book/{bookId}/return	Success: response code 200 OK response code 404 : bookId invalid
GET	/patron/{patronId}	Success: http response code 200 OK Return data see below response code 404 : patronId invalid

Example of JSON response from get patron

```
{
    "patronId": 425198003,
    "name": "David",
    "books": [
        {
            "bookId": 981003724,
            "title": "Biography of Tom",
            "author": "Tom",
            "checkoutDate": "2023-04-23"
        },
        {
            "bookId": 981456123,
            "title": "All about Mary's Lamb",
            "author": "Mary",
            "checkoutDate": "2023-04-23"
        }
    ]
}
```

Fields in the JSON response:

patronId	patron identifier
name	patron name
books	list of currently checked out books for patron
bookId	book number
title	book title
author	book author
checkoutDate	when the book was checked out

Create Entity classes and Spring repositories for the following SQL tables

```
create table patrons (
      patron_id bigint primary key,
      name varchar(250) not null );


create table books (
       book_id bigint primary key,
       title varchar(250) not null,
       author varchar(250) not null,
       checkout_patron_id bigint,
       checkout_date date,
       foreign key (checkout_patron_id) references patrons(patron_id));
```

When a book is checked out, checkout_patron_id and checkout_date fields are updated. Checkout date is set to the current date. When a book is returned, these fields are set to null.

You will need to

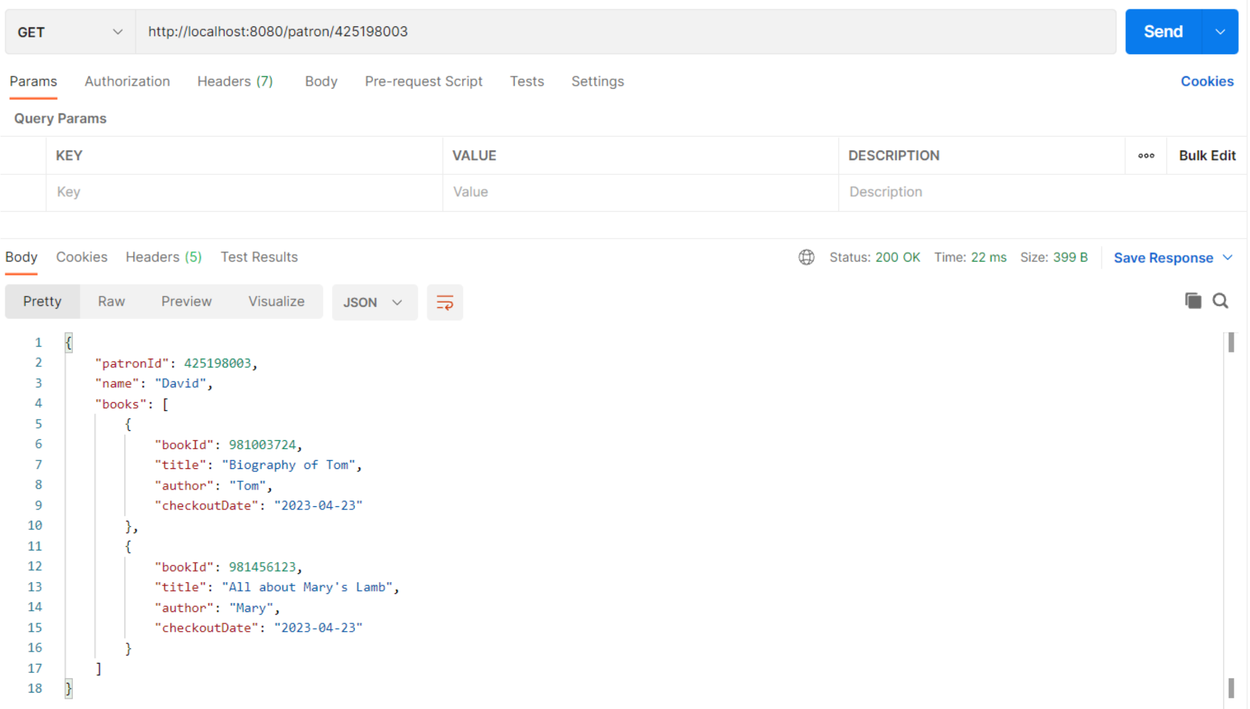
- code a rest controller class (all URL endpoints should be in one controller class).
- code entity classes and respositories
- create a MySQL database and tables and insert test data
- test the rest controller with PostMan
- unit tests and ReactJS front end are not required.

What to submit for this assignment

- create test data
  - rows for two books with id=981003724 and id=981456123
  - patron id=425198003

- create a document (docx or pdf) with the following screenshots. Label each screenshot and include your name on the document.
  - POSTMAN call to checkout book 981003724 and patron 425198003. Show both the call data and the return status in the screenshot. See example below.
  - call to checkout book 981456123
  - list patron information for 425198003 showing the 2 books checked out
  - show a listing of the books table from MySQL Workbench
  - checkin the book 981003724
  - list patron information for 425198003 showing 1 book checked out
  - checkin with an invalid book number
  - list patron information with an invalid patron number
  - The document should include your name and labels for each screenshot.

Example screenshot of get patron showing return data.



GET http://localhost:8080/patron/425198003

Status: 200 OK Time: 22 ms Size: 399 B

```
{
    "patronId": 425198003,
    "name": "David",
    "books": [
        {
            "bookId": 981003724,
            "title": "Biography of Tom",
            "author": "Tom",
            "checkoutDate": "2023-04-23"
        },
        {
            "bookId": 981456123,
            "title": "All about Mary's Lamb",
            "author": "Mary",
            "checkoutDate": "2023-04-23"
        }
    ]
}
```