

# Backend assignment

## Description

John Doe works in the IT department of a library that uses paper records as a means to keep records of books, authors, and customers. One day, John's manager assigned John to develop a backend application that functions as a record system as well as a rental software.

## Instructions

**Task 1** - Implement the JPA's ORM model from the schema provided in the sql (please use **postgresql** as your database)

### Acceptance criteria

- Entity should follow the sql definition

**Task 2** - Write query to pull from database (given a statement below)

Give me the top 3 most rented books and for each book give me top 3 people who rented it in the specific country. Sort the customers in descending order of their total number of books rented." This sql should use the country short code example: SG, MY, US to do the filter.

Use the most optimized way you can to retrieve the dataset.

### Acceptance criteria

- Should return the correct result
- Response time of the query will be compared with a predefined baseline for grading (will be tested against our test server database, which will included millions rows of book\_rent records)

**Task 3** - Implement a GET api endpoint (/getTop3ReadBook?country\_code=XX) with the data retrieved from Task 2 to return to the caller with the given JSON structure below.

```
[
  {
    "author": "author_value",
    "name": "book_name_value",
    "borrower": [
      "top_1_borrower_value",
      "top_2_borrower_value",
      "top_3_borrower_value"
    ]
  }
]
```

**Acceptance criteria**

- Should convert the sql query result to the json structure above correctly
- Should validate the country\_code parameter, should return 400 with error message { message: "invalid param" } when the param is invalid
- Should implement error message { message: "no result" } when there is no result return from sql query