GEEK TEXT API: BOOK BROWSING & SORTING

Niccholas Reiz (nreiz001@fiu.edu)

ABOUT

The Book Browsing and Sorting feature of the Geek Text bookstore RestAPI consists of retrieving and updating books in a database. With this feature, users are allowed to retrieve books by genre, rating, and most copies sold. Users are also given the functionality to discount books by a given publisher. The Springboot RestAPI is written in Java using the IntelliJ IDE. MySQL was the database our team used to store our entities.

HTTP REQUESTS

➔ BASE ENDPOINT: https://localhost:8085/book-browsing

**Sample responses are shortened for simplicity*

1. Retrieve a list of books by genre (GET request)

GET ENDPOINT: /genre:{genre}

Sample request:

GET: https://localhost:8085/book-browsing/genre:**Poetry**

Response:

```
[{
      "isbn": "23421",
```

```
        "bookName": "The Raven",
        "description": "This is a gothic book written by Poe",
        "price": 25.93,
        "author": "Edgar Allan Poe",
        "genre": "Poetry",
        "publisher": "penguin",
        "yearPublished": 1980,
        "copiesSold": 10000,
        "rating": 2.5
    }, ...]
```

2. Retrieve a book list of top sellers (GET REQUEST)

GET ENDPOINT: /genre:{genre}

- Returns the ten most sold books in descending order.

  Sample request:

  GET: https://localhost:8085/book-browsing/top-sellers

  Response:

```
[{
        "isbn": "23421",
        "bookName": "The Raven",
        "description": "This is a gothic book written by Poe",
        "price": 25.93,
        "author": "Edgar Allan Poe",
        "genre": "Poetry",
        "publisher": "penguin",
        "yearPublished": 1980,
        "copiesSold": 10000,
        "rating": 2.5
    },
    {
        "isbn": "45667",
        "bookName": "The Call of Cthulhu",
```

```
    "description": "Cthulhu is a cosmic entity created by writer H.
  P. Lovecraft. It was introduced in his short story \"The Call of
  Cthulhu\"",
    "price": 100.0,
    "author": "H.P.Lovecraft",
    "genre": "Poetry",
    "publisher": "atlas",
    "yearPublished": 1850,
    "copiesSold": 700,
    "rating": 3.5
},...]
```


3. Retrieve books by rating or higher (GET request)

`GET` ENDPOINT: /rating:{rating}

- Returns a list of books with the given rating and higher.
- Book rating is on a scale of 0.0 to 5.0.

  Sample request:

  GET: https://localhost:8085/book-browsing/rating:3.5

  Response:


```
[{
    "isbn": "1000",
    "bookName": "Name",
    "description": "john-Gonzalez",
    "price": 68.0,
    "author": "john doee",
    "genre": "non-fiction",
    "publisher": "test",
    "yearPublished": 2020,
    "copiesSold": 100,
    "rating": 5.0
},
{
```

```
    "isbn": "45667",
    "bookName": "The Call of Cthulhu",
     "description": "Cthulhu is a cosmic entity created by writer H.
  P. Lovecraft. It was introduced in his short story \"The Call of
  Cthulhu\"",
    "price": 100.0,
    "author": "H.P.Lovecraft",
    "genre": "Poetry",
    "publisher": "atlas",
    "yearPublished": 1850,
    "copiesSold": 700,
    "rating": 3.5
}]
```


4. Discount books by publisher (PATCH request)

PATCH ENDPOINT: /discount:{discount}/publisher:{publisher}

- Updates the prices of books by a given discount under a given publisher.
- The API assumes that the discount is given in percent form, not decimal form.
- Discount must be a number in between 0 and 100.

  Sample request:

  PATCH: https://localhost:8085/book-browsing/discount:5/publisher:penguin

  Response:

  NONE

➔ *books under “penguin” publisher will have their price discounted by 5%*

ERROR HANDLING

The RestAPI implements error handling to catch any issues. Error handling messages are received by the backend side of the program.

For example, if I request a genre that does not appear in the database, the IDE console will output:

“No book found with genre: Non-fiction”