Problem statement

You are requested to create REST API for book store system. You should expose CRUD operations for Book, and User models (or tables). Also you should be able to lend or return the book for the specific user.

```
class Book{
        int book_id;
        String book_name;
        String author;
}

class User{
        int user_id; //automatic unique sequence number (primary key)
        String name;
}

class BookLog{
        int book_id;
        int user_id;
}
```

The list of api operations required are,

Api Path	Type	Description
/bookstore/user/{id}	GET	Get user by id
/bookstore/user/all	GET	Get all users from the table
/bookstore/user/{id}	PUT	Update user by id
/bookstore/user/{id}	DELETE	Delete user by id
/bookstore/user/	POST	Create new user
/bookstore/book/{id}	GET	Get book by id
/bookstore/book/all	GET	Get all books from the table
/bookstore/book/{id}	PUT	Update book by id
/bookstore/book/{id}	DELETE	Delete book by id
/bookstore/book/	POST	Create new book
/bookstore/lend/{book_id}/{user_id}	PUT	This operation should, 1. select user from user table and book from book table 2. throw error message if user/book record does not exists 3. throw error if an entry already exists with given Book_id, user_id in booklog table 4. Else, Add an entry in booklog table with the user_id and book_id

/bookstore/return/{book_id}/{user_id}	PUT	This operation should, 1. select user from user table and book from book table 2. throw error message if user/book record does not exists 3. throw error if an entry does not exists with given Book_id, user_id in booklog table 4. Else, remove the entry in booklog table with the user_id and book_id

NOTE:

1. You are required to use,
* Springboot
* Maven
* SpringREST
* Spring JPA
For this above application.

2. You can choose the same database which you have gone through during your tutorial. I believe it is mysql.

3. You are free to choose your own coding style and approach as long as it works

4. Below are the things which are good to have but you can ignore this if don’t know what they are,
* Swagger for documentation
* Junit for unit test cases