Microservices Spring Boot - Final Project

Overview:

The Book Store REST service using Mysql database show how to use the RESTful service to add and retrieve books from a book store.

The Book Store REST service performs ten operations:

1. printing all books from bookstore,
2. printing book which has zero inventory
3. printing all books from all shelves order by shelf number
4. finding book using shelf number which is passed as a query parameter to the POST operation
5. finding book title using a search string which is passed as a query parameter to the POST operation
6. getting inventory using book id, which is passed as a query parameter to the POST operation
7. finding publisher name using a search string which is passed as a query parameter to the POST operation

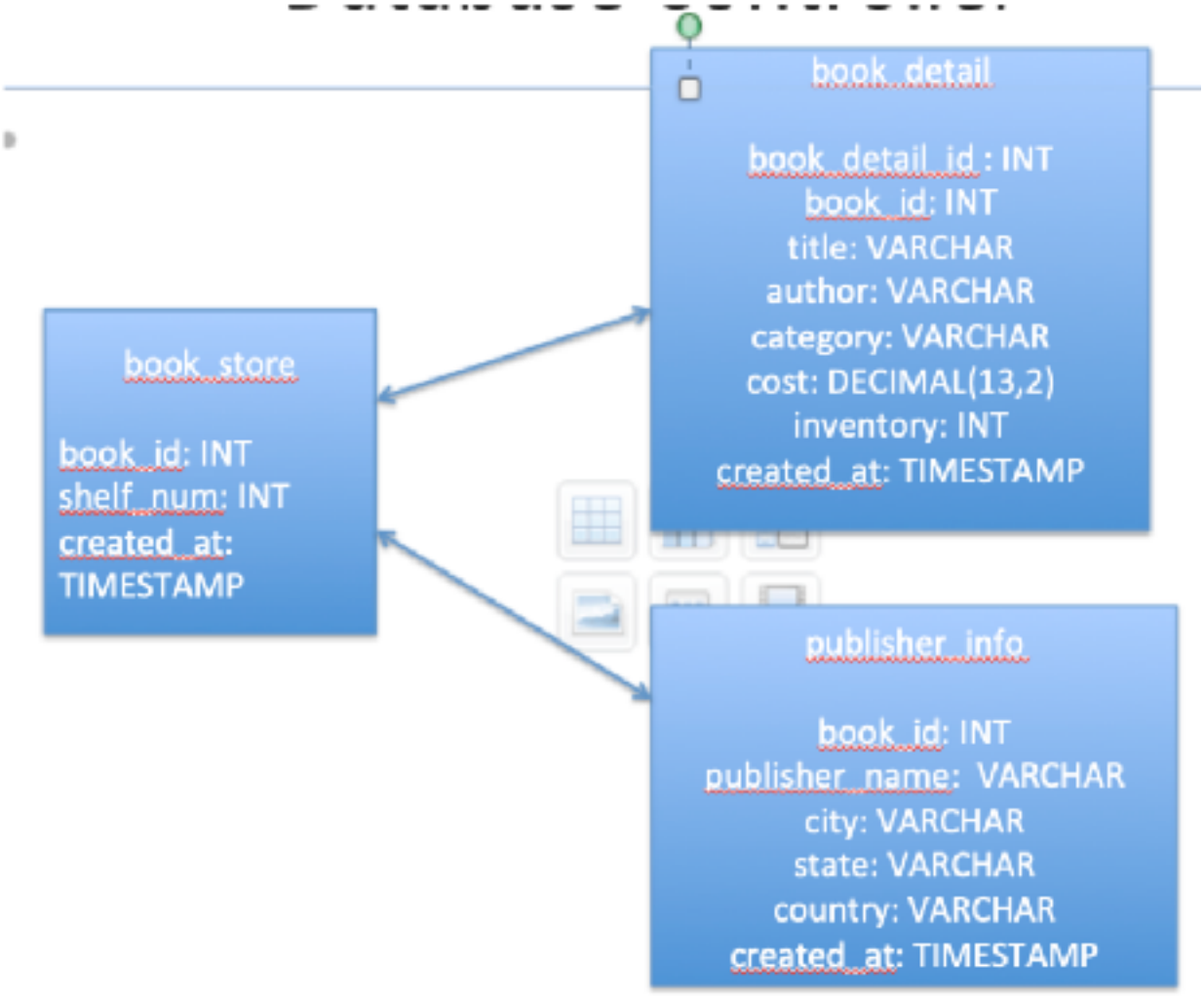
8. adding a book using a java object class which is passed as datas to the POST operation

9. adding a publisher information using a java object class which is passed as datas to the POST operation

10. Finding book category using a search string which is passed as a query parameter to the POST operation

This is a REST Client process which uses Invoke REST API activity to retrieve data from the REST Server. You can POST and GET books by specifying the HTTP Client, HTTP Method and Request Type in the Invoke REST API activity.

DataBase Architecture



REST Servicies:

GitHub:

https://github.com/learn-OC/dev-ent-java-microserv-spring-final-proj-template

URL :

http://ec2-18-216-171-213.us-east-2.compute.amazonaws.com:8080/ spring-proj-template/

Methods:

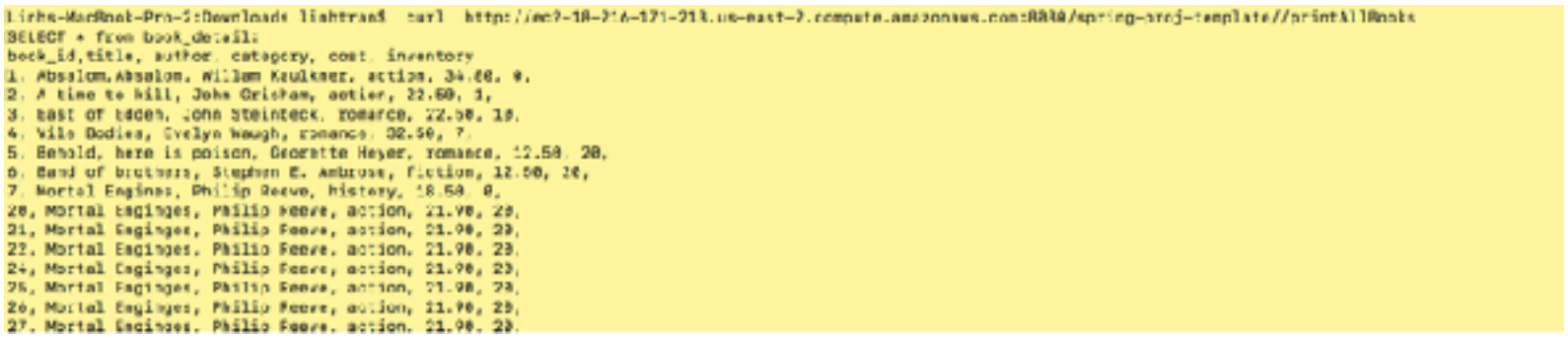
---

1)/printAllBooks: print all books from book detail

- Method: GET
- URL Param: None
- Success Response : 200
- Sample Curl call:

  http://ec2-18-216-171-213.us-east-2.compute.amazonaws.com:8080/ spring-proj-template// printAllBooks

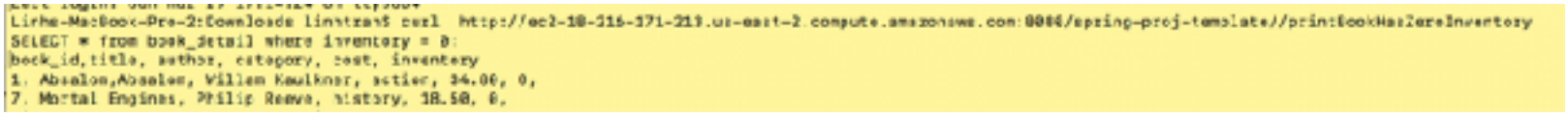
- Expected Output:



---

2)/printBookHasZeroInventoty : print all books which have zero inventory

- Method: GET
- URL Param: None
- Success Response : 200
- Sample Curl:
  - curl http://ec2-18-216-171-213.us-east-2.compute.amazonaws.com:8080/spring-proj-template// printBookHasZeroInventory
  - Expected Output:

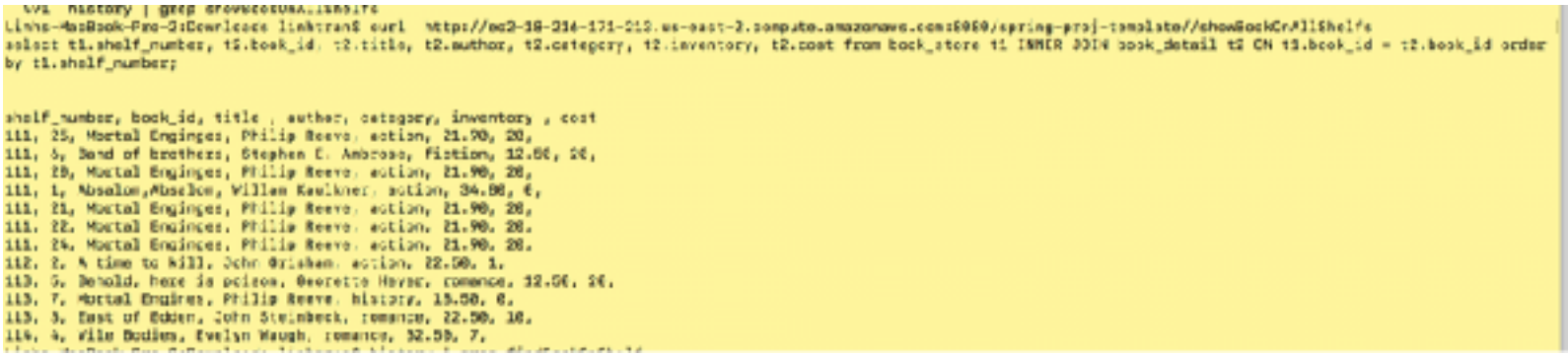


---

3)/showBookOnAllShelfs: list all books on shelves and order them by shelf number

- Method: GET
  URL Param: None
- Success Response : 200

- Sample Curl call:
  - curl http://ec2-18-216-171-213.us-east-2.compute.amazonaws.com:8080/spring-proj-template// showBookOnAllShelfs
  - Expected Output:



---

4)/findBookOnShelf : list all books on shelves and order them by shelf number

- Method: GET
- URL Param: shelfNum=[integer]
- Success Response : 200
- Sample Curl call:
  - curl -i -H "Accept: application/json" -H "Content- Type:application/json" -X POST “http://ec2-18-216-171-213.us-east-2.compute.amazonaws.com:

8080/spring-proj-template//
findBookOnShelf?shelfNum=111”

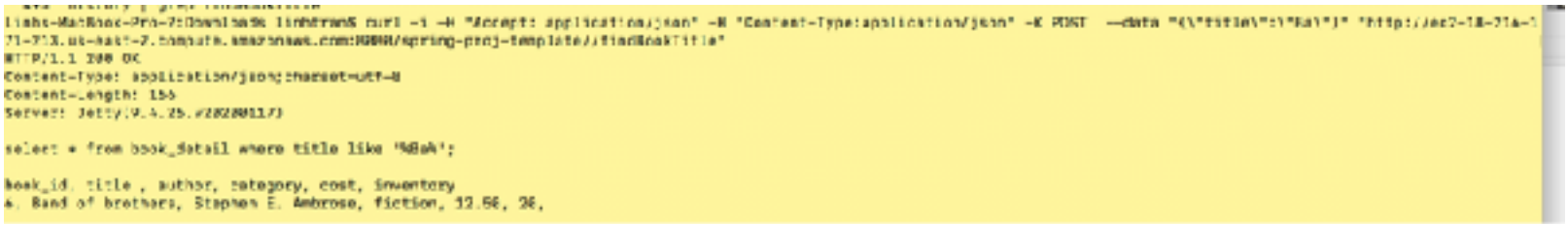
- Expected Output:

---

5)/findBookTitle : search all books which have a substring which provided from URL param

- Method: POST
- URL Param:
- title=[String] Required
- Success Response : 200
- Sample Curl call:
  - curl -i -H "Accept: application/json" -H "Content- Type:application/json" -X POST --data "{\"title\":\"Ba\"}" “http://ec2-18-216-171-213.us-east-2.compute.amazonaws.com:8080/spring-proj-template//findBookTitle”

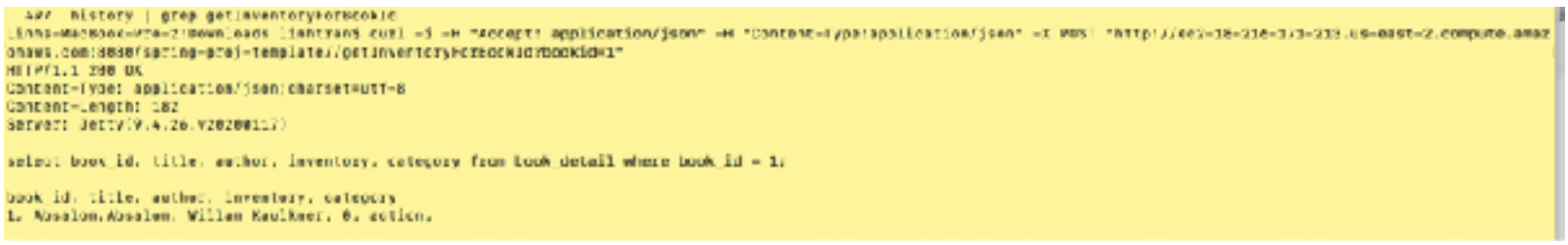
- Expected Output:



---

6)/getInventoryForBookId : get an inventory for specific bookid

- Method: POST URL Param:
- bookid =[integer] Required
- Success Response : 200
- Sample Curl call:
  - curl -i -H "Accept: application/json" -H "Content- Type:application/json" -X POST “http://ec2-18-216-171-213.us-east-2.compute.amazonaws.com:8080/spring-proj-template//getInventoryForBookId?bookid=1”
  - Expected Output:



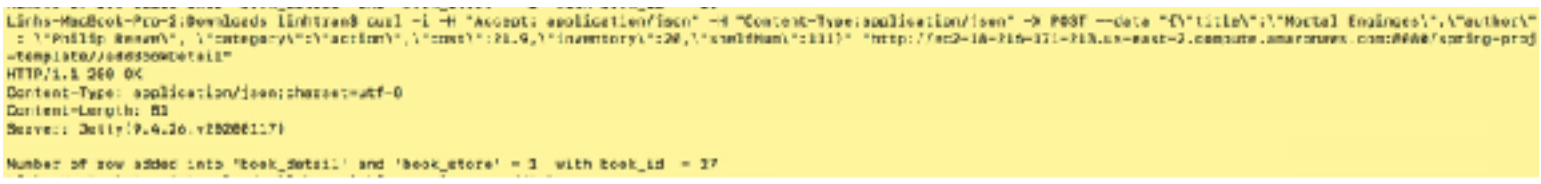
7)/findPublisherName: find publisher name based on user input

- Method: POST URL Param:
- name =[String] Required
- Success Response : 200
- Sample Curl call:
  - curl -i -H "Accept: application/json" -H "Content- Type:application/json" -X POST --data "{\"name\":\"Ha\"}" “http://ec2-18-216-171-213.us-east-2.compute.amazonaws.com:8080/spring-proj-template// findPublisherName"
  - Expected Output:

8)/addBookDetail: add a book into book_detail database table

- Method: POST

- URL Param: None
- Data Param:
  - {“title”:[String],"author" : [String], “category”: [String],”cost”:[Decimal], “inventory”: [Int],”shelfNum”:[Int]}"
    Required
- Success Response : 200
- Sample Curl call:
  - curl -i -H "Accept: application/json" -H "Content- Type:application/json" -X POST --data "{\"title\":\"Mortal Enginges\",\"author\" : \"Philip Reeve\", \"category\":\"action\", \"cost\":21.9,\"inventory\": 20,\"shelfNum\":111}" “http://ec2-18-216-171-213.us-east-2.compute.amazonaws.com:8080/ spring-proj-template// addBookDetail”
  - Expected Output:

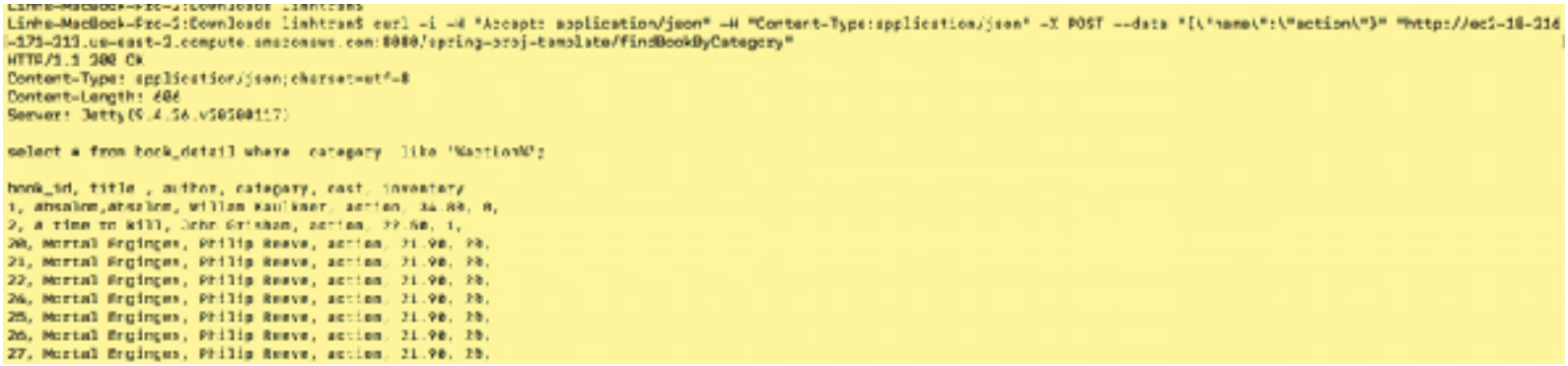


9)/addpublisher : add publisher info into publisher_info

- Method: POST
- URL Param: None
- Data Param:
  - {“bookid”:[Int], “publisher_name": [String],"city": [String], "state": [String], "country": [String]}
- Required: “bookid” must existing in book_store
- Success Response : 200
- Sample Curl call:
  - curl -i -H "Accept: application/json" -H "Content-Type:application/json" -X POST --data "{\"bookid\":1, \"publisher_name\": \"Stephen King\",\"city\": \"San Jose\", \"state\": \"CA\", \"country\": \"USA\"}" “http:// ec2-18-216-171-213.us-east-2.compute.amazonaws.com:8080/ spring- proj-template/addPublisher”
  - Expected Output:

10)/findBookWithCategory : find all books has the search category

- Method: POST
- URL Param: None
- Data Param:
  - {”name”:[String]} Required
- Success Response : 200
- Sample Curl call:
  - curl -i -H "Accept: application/json" -H "Content-Type:application/json" - X POST --data "{\"name\":\"action\"}" “http://ec2-18-216-171-213.us-east-2.compute.amazonaws.com:8080/spring-proj-template/ findBookByCategory”
  - Expected Output:



How to run it locally:

Assume you have IntelliJ installed in your laptop and your AWS server is ready
1)Goto https://github.com/learn-OC/dev-ent-java-microserv-spring-final-proj-template and ‘Fork’
2)Open IntelliJ
3)File->New->Project From Version Control
4)enter your github
5)right click on ‘src/main/java/JDBCApplication.java and ‘Run’
6)Goto the terminal and run Curl command like

- curl http://localhost:8080/printAllBooks
- curl -i -H "Accept: application/json" -H "Content-Type:application/json" -X POST “http://localhost:8080/findBookOnShelf?shelfNum=111”

7)If you want to use Postman, use below steps:

- for ‘GET’ call, put http://localhost:8080/printAllBooks into url
- for ‘POST’ call, put http://localhost:8080/findBookTitle into url and put {“title”: “Ba”} into ‘Body’ tab

How to debug it locally:

Assume you have IntelliJ installed in your laptop and your AWS server is ready
1)Goto https://github.com/learn-OC/dev-ent-java-microserv-spring-final-proj-template and ‘Fork’
2)Open IntelliJ
3)File->New->Project From Version Control
4)enter your github

5)right click on ‘src/main/java/JDBCApplication.java and ‘Debug JDBCApplication’

6)In Postman,

- for ‘GET’ call, put http://localhost:8080/printAllBooks into url

7)Set a breakpoint where you want to trace
8)From Postman, hit ‘Send’ and debug it from there
