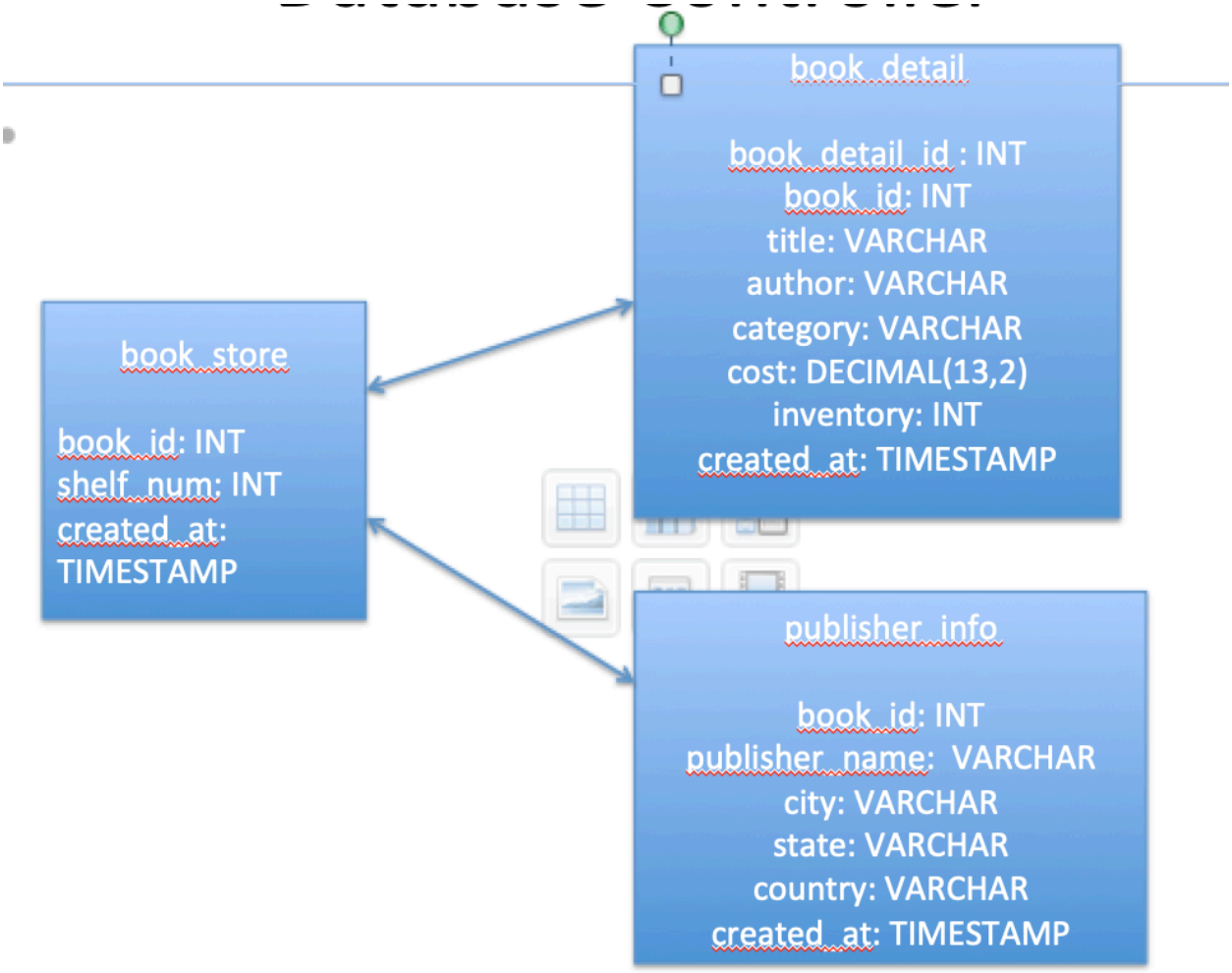
# Microservices Spring Boot - Final Project

Linh Tran

## Overview:

These microservices REST Apis are about to provide the basic functionalities for book store access like search title/publisher, do inventory, add a book/publisher....

## Database Architecture



# Rest Apis:

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

    SELECT * from book_detail:
    book_id,title, author, category, cost, inventory
    1, Absalom,Absalom, Willam Kaulkner, action, 34.80, 0,
    2, A time to kill, John Grisham, action, 22.50, 1,
    3, East of Edden, John Steinbeck, romance, 22.50, 10,
    4, Vile Bodies, Evelyn Waugh, romance, 32.50, 7,
    5, Behold, here is poison, Georette Heyer, romance, 12.50, 20,
    6, Band of brothers, Stephen E. Ambrose, fiction, 12.50, 20,
    7, Mortal Engines, Philip Reeve, history, 18.50, 0,

---

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

- 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//printBookHasZeroInventory
  - Expected Output:

    SELECT * from book_detail where inventory = 0:
    book_id,title, author, category, cost, inventory
    1, Absalom,Absalom, Willam Kaulkner, action, 34.80, 0,
    7, Mortal Engines, Philip Reeve, history, 18.50, 0,

---

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:

    select t1.shelf_number, t2.book_id, t2.title, t2.author, t2.category, t2.inventory, t2.cost from book_store t1 INNER JOIN book_detail t2 ON t1.book_id = t2.book_id order by t1.shelf_number;

    shelf_number, book_id, title , author, category, inventory , cost
    111, 6, Band of brothers, Stephen E. Ambrose, fiction, 12.50, 20,
    111, 1, Absalom,Absalom, Willam Kaulkner, action, 34.80, 0,
    112, 2, A time to kill, John Grisham, action, 22.50, 1,
    113, 5, Behold, here is poison, Georette Heyer, romance, 12.50, 20,
    113, 7, Mortal Engines, Philip Reeve, history, 18.50, 0,
    113, 3, East of Edden, John Steinbeck, romance, 22.50, 10,
    114, 4, Vile Bodies, Evelyn Waugh, romance, 32.50, 7,

---

4)/findBookOnShelf : list all books from a specific shelf

- Method: POST

- URL Param:
  - shelfNum=[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//findBookOnShelf?shelfNum=111"
  - Expected Output:

    select t1.shelf_number, t2.book_id, t2.title, t2.author, t2.category, t2.inventory, t2.cost from book_store t1 INNER JOIN book_detail t2 ON t1.book_id = t2.book_id where t1.shelf_number = 111;

    shelf_number, book_id, title , author, category, inventory , cost
    111, 1, Absalom,Absalom, Willam Kaulkner, action, 34.80, 0,
    111, 6, Band of brothers, Stephen E. Ambrose, fiction, 12.50, 20,
    111, 20, Mortal Enginges, Philip Reeve, action, 21.90, 20,
    111, 21, Mortal Enginges, Philip Reeve, action, 21.90, 20,
    111, 22, Mortal Enginges, Philip Reeve, action, 21.90, 20,
    111, 24, Mortal Enginges, Philip Reeve, action, 21.90, 20,

---

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:

    select * from book_detail where title like '%Ba%';
    book_id, title , author, category, cost, inventory
    6, Band of brothers, Stephen E. Ambrose, fiction, 12.50, 20,

---

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:

    select book_id, title, author, inventory, category from book_detail where book_id = 1;

    book_id, title, author, inventory, category
    1, Absalom,Absalom, Willam Kaulkner, 0, action,

---

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:

    select * from publisher_info where publisher_name like '%Ha%';

    publisher_id, book_id , publisher_name, city, state, country
    2, 2, Hachette Livre, Cupertino, CA, USA,
    3, 3, HarperCollins, San Jose, TX, USA,

---

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:

    Number of row added into 'book_detail' and 'book_store' = 1  with book_id  = 22

---

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:

    ** Added 1 row in publisher_info

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:

select * from book_detail where category like '%action%';

book_id, title , author, category, cost, inventory
1, Absalom,Absalom, Willam Kaulkner, action, 34.80, 0,
2, A time to kill, John Grisham, action, 22.50, 1,
20, Mortal Enginges, Philip Reeve, action, 21.90, 20,
21, Mortal Enginges, Philip Reeve, action, 21.90, 20,
22, Mortal Enginges, Philip Reeve, action, 21.90, 20,
24, Mortal Enginges, Philip Reeve, action, 21.90, 20,
25, Mortal Enginges, Philip Reeve, action, 21.90, 20,

## 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