Report For DOS Project
(Small Microservices application)

Description :

In this homework we built application Based on Microservices approach this photo describes servers and (requests - responses) between them.

Servers :

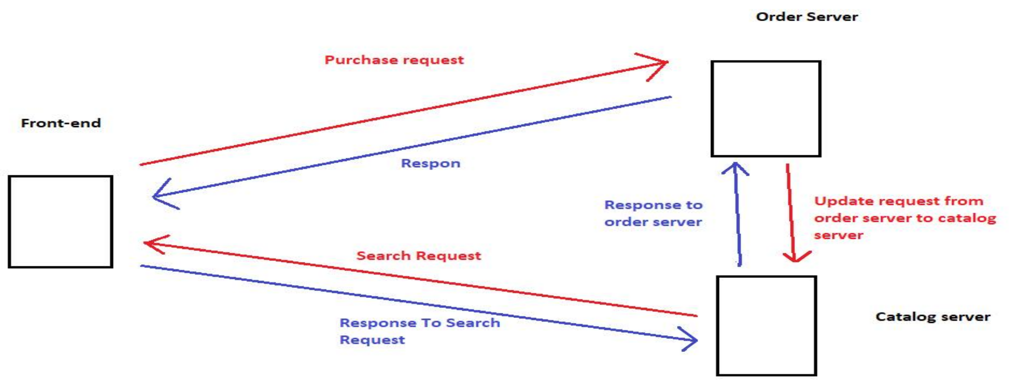
1- Front-end server .
2- Order server .
3- Catalog server .

Operation with front-end :

1- Search (topic) : this request sent to catalog server and the response is all books belongs to specified topic in url .

2- Information (item number): this request sent to catalog server and the response is all information about the book its ID specified in url .

3- Purchase(item number): this request sent to order server to buy a book , then Order server send PUT request to catalog server to update some information in database…after that Catalog server make response to Order server , then Order server send response to front-end server and so on .



Tow types of request in this app.

*** Searching requests :**

To get information about books saved in catalog server , we send requests from Front-end (client) to catalog server .

***Purchase request :**

To Buy a book from this application , Client (front-end) sends request to order server , and order server send update request to catalog server , then catalog server make response to order server . after that order server make response to client (Front-end) .

Response is differ from state to another state . if you want to buy a book ,and this book is exist in catalog . Order server will make update request to catalog server to decrement number of copies of this book .

Catalog server sends response to order server . then Order make response to Client , “The book is available , the operation is done” ,that if the book is exist in Catalog . Otherwise the response will be “The book is not available “ .

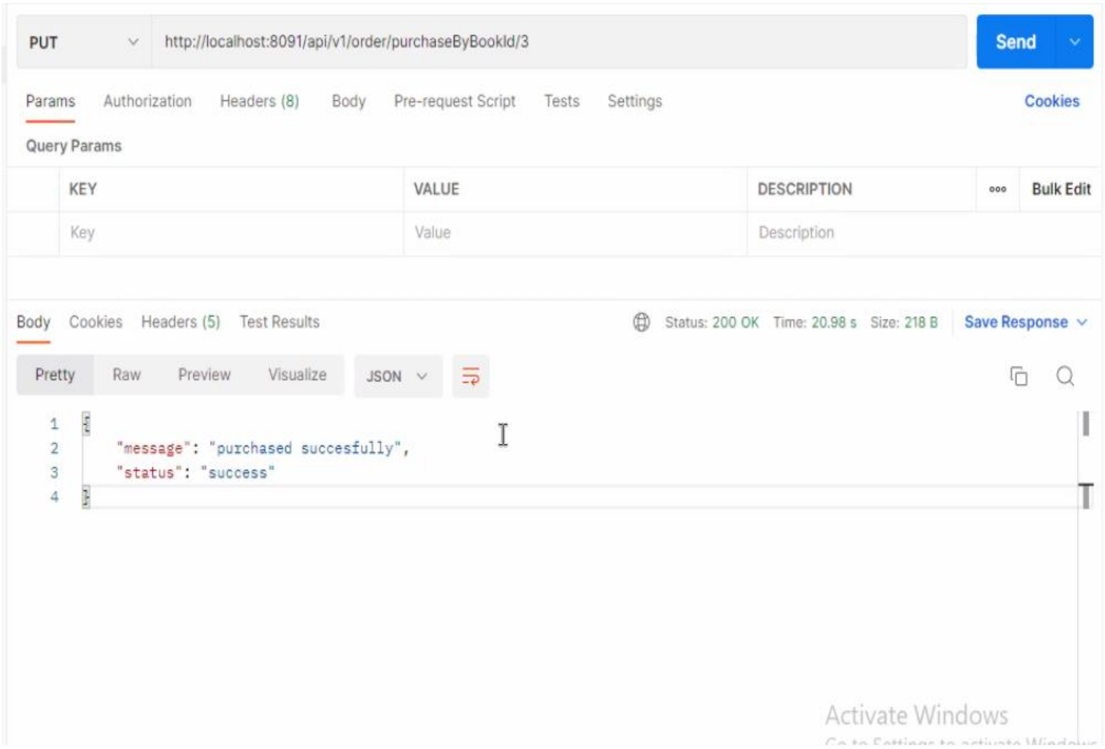
Tools Used In this Project :

1- Spring framework .
2- postresql database .

Some Results :

Request to order (PUT Request) :

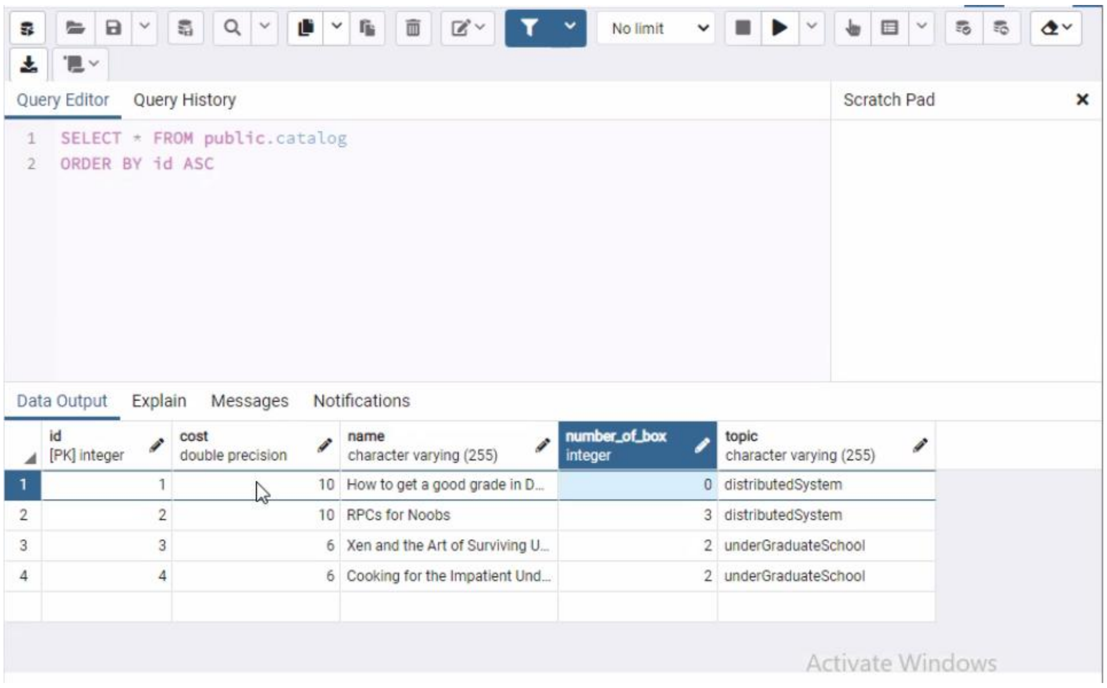
* This is purchase request -> to buy a book from catalog server.
* This operation is done successfully .


PUT http://localhost:8091/api/v1/order/purchaseByBookId/3

Status: 200 OK Time: 20.98 s Size: 218 B

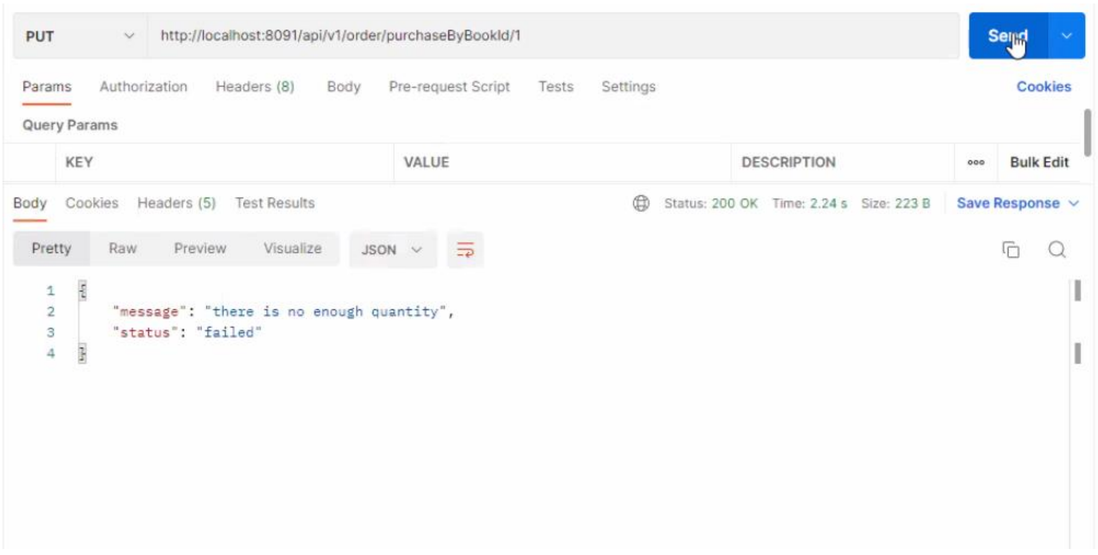
```
{
    "message": "purchased succesfully",
    "status": "success"
}
```

* After this purchase request , there is no book available in database.


```
SELECT * FROM public.catalog
ORDER BY id ASC
```

	id [PK] integer	cost double precision	name character varying (255)	number_of_box integer	topic character varying (255)
1	1	10	How to get a good grade in D...	0	distributedSystem
2	2	10	RPCs for Noobs	3	distributedSystem
3	3	6	Xen and the Art of Surviving U...	2	underGraduateSchool
4	4	6	Cooking for the Impatient Und...	2	underGraduateSchool

* if we want to resend purchase request to buy the same book again
This operation is failed because there is no enough copies from this book in database .



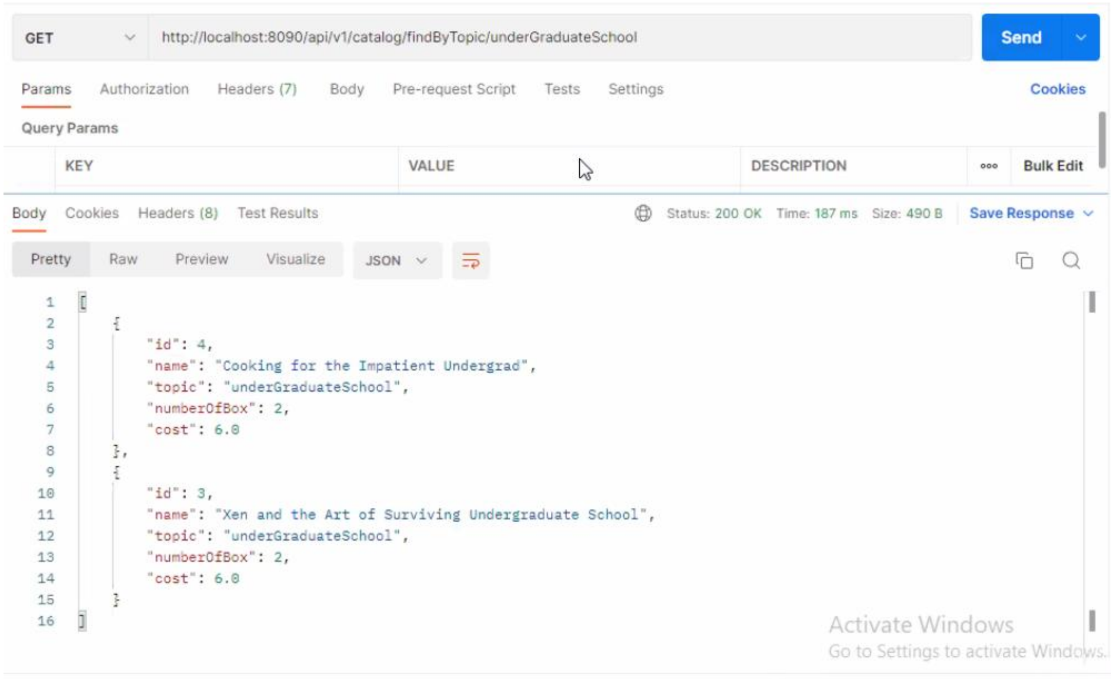
http://localhost:8091/api/v1/order/purchaseByBookId/1

Status: 200 OK Time: 2.24 s Size: 223 B

```
{
    "message": "there is no enough quantity",
    "status": "failed"
}
```

Search requests :

Search (topics) : we send this get request and the result are all books under specified topic .

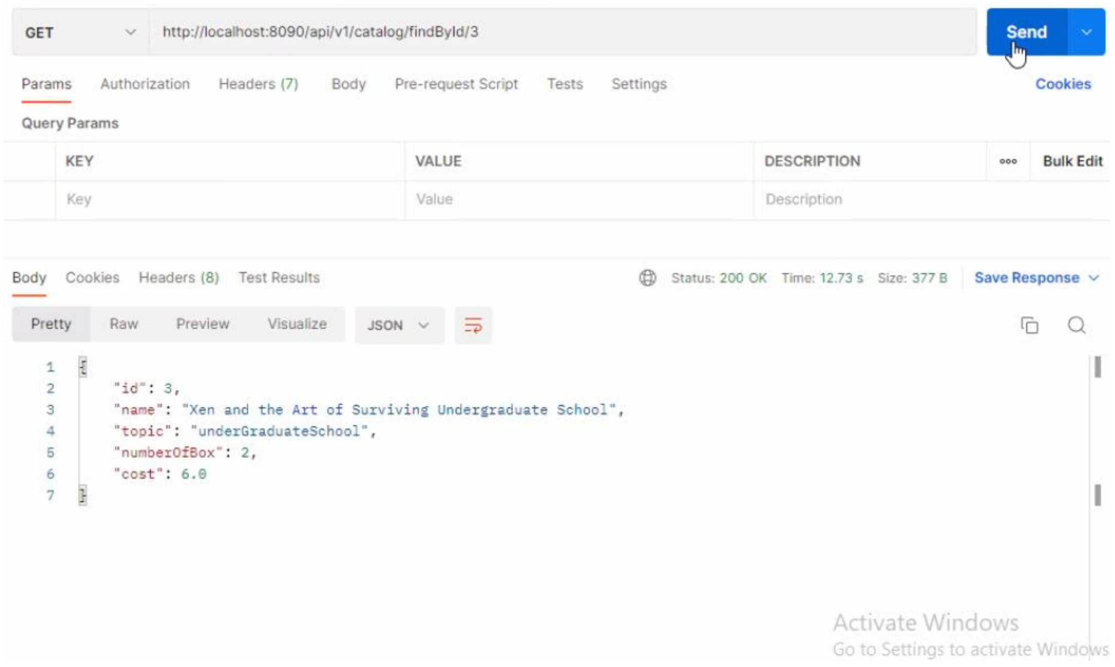


http://localhost:8090/api/v1/catalog/findByTopic/underGraduateSchool

Status: 200 OK Time: 187 ms Size: 490 B

```
[
    {
        "id": 4,
        "name": "Cooking for the Impatient Undergrad",
        "topic": "underGraduateSchool",
        "numberOfBox": 2,
        "cost": 6.0
    },
    {
        "id": 3,
        "name": "Xen and the Art of Surviving Undergraduate School",
        "topic": "underGraduateSchool",
        "numberOfBox": 2,
        "cost": 6.0
    }
]
```

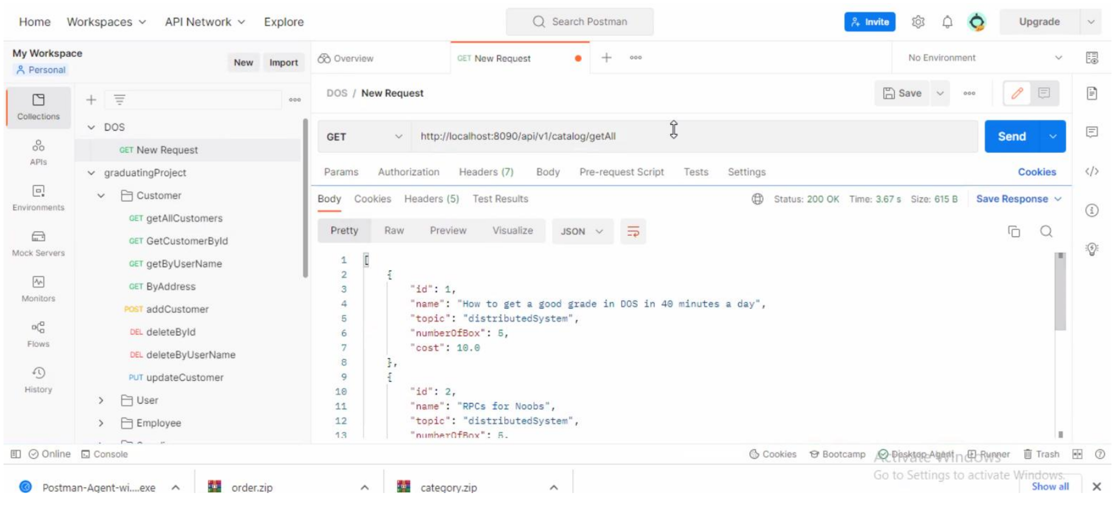
Search (ID) : we send this get request and the result is the book with specified ID .


GET http://localhost:8090/api/v1/catalog/findById/3

Status: 200 OK Time: 12.73 s Size: 377 B

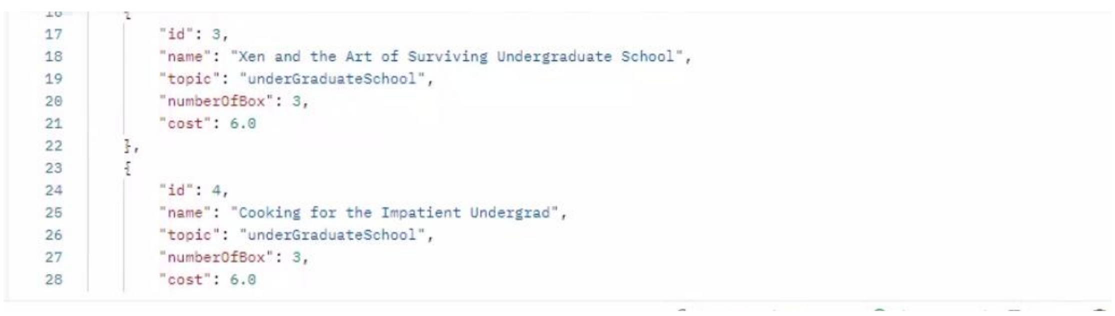
```
{
    "id": 3,
    "name": "Xen and the Art of Surviving Undergraduate School",
    "topic": "underGraduateSchool",
    "numberOfBox": 2,
    "cost": 6.0
}
```

Get request to view all books from catalog .


GET http://localhost:8090/api/v1/catalog/getAll

Status: 200 OK Time: 3.67 s Size: 615 B

```
[
    {
        "id": 1,
        "name": "How to get a good grade in DOS in 40 minutes a day",
        "topic": "distributedSystem",
        "numberOfBox": 5,
        "cost": 10.0
    },
    {
        "id": 2,
        "name": "RPCs for Noobs",
        "topic": "distributedSystem",
        "numberOfBox": 6,
```


```
        "id": 3,
        "name": "Xen and the Art of Surviving Undergraduate School",
        "topic": "underGraduateSchool",
        "numberOfBox": 3,
        "cost": 6.0
    },
    {
        "id": 4,
        "name": "Cooking for the Impatient Undergrad",
        "topic": "underGraduateSchool",
        "numberOfBox": 3,
        "cost": 6.0
```

ابراهيم عارضة
أحمد نبهان