Bazar Bookstore – Program Output

D. Samer Arandi

| Rand Johari | 12027653 |
|---------------|----------|
| Abeer Kharouf | 12028125 |

External via Postman:

- ✓ Search Book by Topic (Frontend → Catalog).
- ✓ Get Book Info by ID (Frontend → Catalog)...
- ✓ Purchase Book (Frontend → Order → Catalog).

Internal via curl:

✓ Internal Request: Update Book Quantity (Order → Catalog).

1. Search Book by Topic.

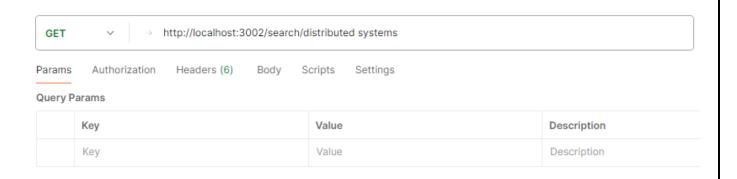
URL: http://localhost:3002/search/distributed systems

Method: GET

Expected Behavior:

Returns a list of books under the topic "distributed systems".

Response output: JSON - 200



2. E Get Book Info by ID.

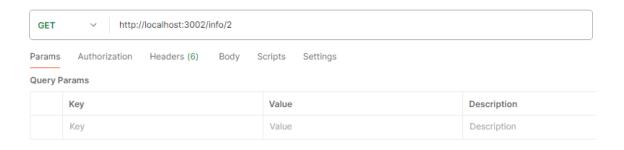
URL: http://localhost:3002/info/2.

Method: GET

Expected Behavior:

Returns detailed information about book with ID 2.

Response output: JSON - 200



Front:

```
2025-04-04 02:41:31 Book details:

2025-04-04 02:41:31 ID: 2

2025-04-04 02:41:31 Fopic: undefined

2025-04-04 02:41:31 Fittle: RPCs for Noobs

2025-04-04 02:41:31 unantity: 5

2025-04-04 02:41:31 Price: 50
```

Catalog:

3. Furchase Book.

URL: http://localhost:3002/purchase/2

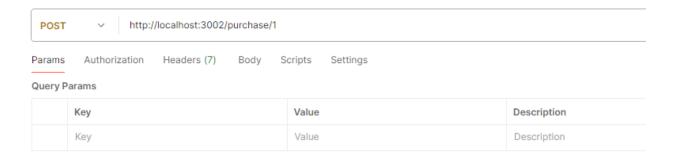
Method: POST.

Expected Behavior:

When a purchase request is made:

- 1. The **frontend** sends a POST request to the **order** service.
- 2. The **order** service:
 - Queries the catalog service using GET /info/:id to retrieve the book's current quantity.
 - If quantity > 0, it proceeds to PUT /update/:id to decrease the quantity by 1.
 - If quantity == 0, the purchase fails.

Response output: JSON - 200





Order:

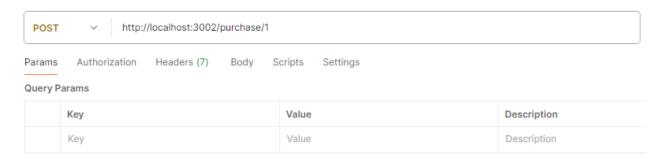
```
2025-04-04 02:59:42 Trying to contact catalog with: http://catalog:3000/info/1 2025-04-04 02:59:42 ✓bought book: How to get a good grade in DOS in 40 minutes a day
```

Fronted:

```
2025-04-04 02:59:42 ought book with ID: 1
```

Catalog:

After 2 purchases:





Order:

Fronted:

```
2025-04-04 03:23:01 Front-end server running on port 3100 2025-04-04 03:23:06 **Mourchase failed - Book with ID 1 is out of stock
```

Catalog:

4. 🏖 Internal Request: Get Book Info.

URL: curl http://catalog:3000/info/2

Method: PUT.

Executed inside: order container

To test updating a book manually (simulate internal communication):

1. Open a shell into the order container:

```
docker exec -it <order_container_id> sh
```

2. Run the following command:

```
curl -X PUT http://catalog:3000/update/2 \
  -H "Content-Type: application/json" \
  -d '{"quantity": 4, "price": 40 }'
```

```
C:\Users\pc\bazar-bookstore\backend\order>docker exec -it e64cfbec0f3b89f2834149feea26d08a7c2a932686d54a80aa8114d82121d1
58 sh
#
# curl -X PUT http://catalog:3000/update/2 \
-H "Content-Type: application/json" \
-d '{"quantity": 4, "price": 40 }'
>> {"status":"success","updated":{"id":2,"price":40,"quantity":4}}#
```

Nothing in Front and Order:

```
2025-04-04 12:28:52 Front-end server running on port 3100

2025-04-04 12:28:52 Order Service running on http://localhost:3001
```

Catalog:

```
2025-04-04 12:28:52 Catalog Service running on http://localhost:3000 2025-04-04 12:34:49 Received update request for ID: 2 2025-04-04 12:34:49 Rew data: { quantity: 4, price: 40 }
```