

Students : Ru'ya Atatreh & Suha Al_Taher

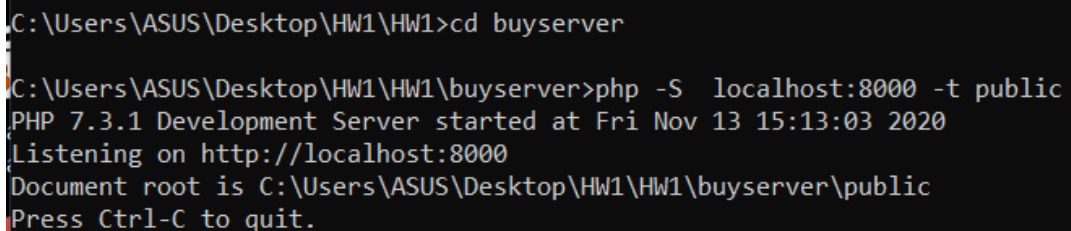
In order to solve this problem we used Lumen as our lightweight micro web framework for PHP by using it we were able to create two microservices, the first is the buyserver that allows the client to send an HTTP REST request to the server in order to buy a book based on its number and returns (bought book <BOOK_NAME>) in order to achieve that the server has to send a query to the catalog server to see if the book is in stock or not, if not, if there is it completes the buying and decrements the number of items in stock, if not it returns (no book in store), the other server is the catalog server which contains information about the book like name, price, item_no etc. it enables the user to find info about the book based on its number or topic. We used Postman or the browser as our client.

One of the trade-offs of this design is that it contains two servers for simple and small task which will result in lots of queries to the catalog server causing traffic between the servers, if the catalog server went down the buying operation from the buyserver will not be completed because it can't check the number of items in stock as there is a dependency between the servers.

Our code works properly but there was a problem with the IP address when sending it to the other device, the server on the other device as the request arrives to the server but with no reply received because of an error in the IP address.

To run the program you have to have lumen installed in your device, follow the following steps:

- Open the command prompt
- Go to the directory where the files are installed
- Write this command `php -S localhost:8000 -t public` as in figure below



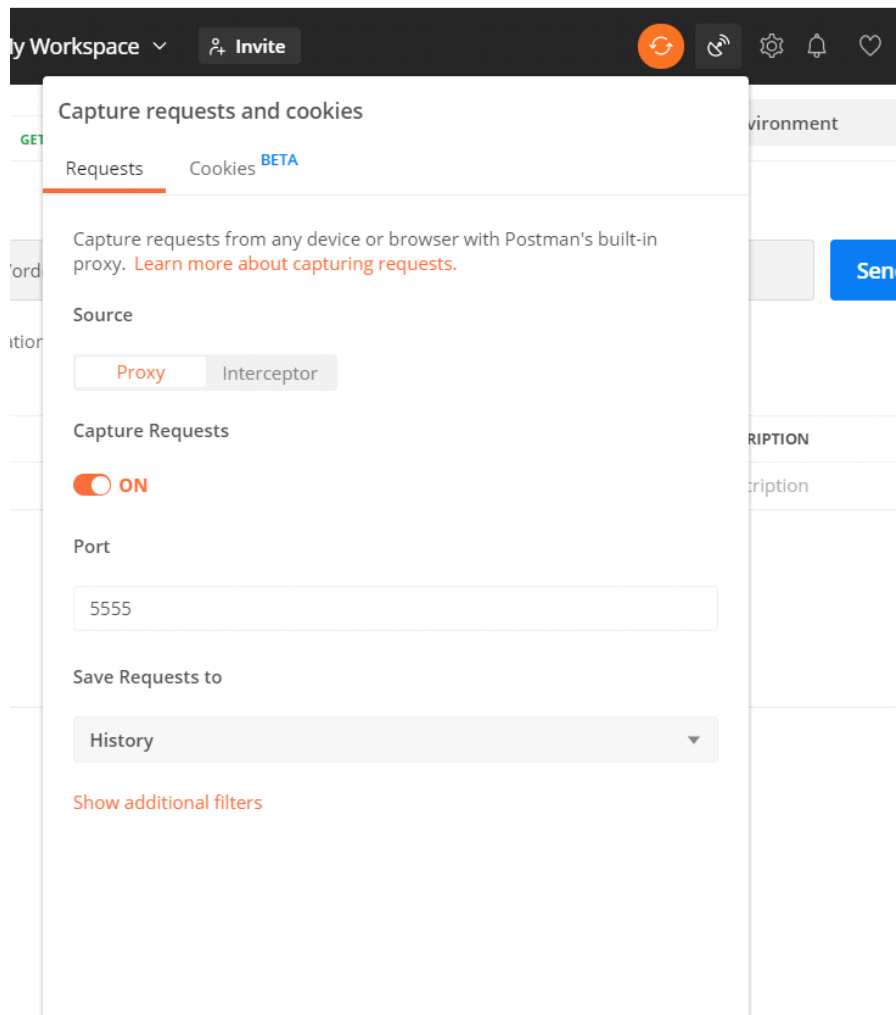
```
C:\Users\ASUS\Desktop\HW1\HW1>cd buyserver
C:\Users\ASUS\Desktop\HW1\HW1\buyserver>php -S localhost:8000 -t public
PHP 7.3.1 Development Server started at Fri Nov 13 15:13:03 2020
Listening on http://localhost:8000
Document root is C:\Users\ASUS\Desktop\HW1\HW1\buyserver\public
Press Ctrl-C to quit.
```

- Repeat the steps to the other server

For the client side we either use the browser or POSTMAN in this case we used both.

postman:

- Open the request capture



then user enter the request URI

search for book :`https:// CATALOG_WEBSERVICE_IP/search/<book name>`



lookup book enter:https:// CATALOG_WEBSERVICE_IP/lookup/<book_id>

← → ↻ ⓘ localhost:8000/CATALOG_WEBSERVICE_IP/lookup/2

```
{ "id": 2, "name": "RPCs for Dummies", "cost": 300, "quantity": 5 }
```

GET http://localhost:8000/CATALOG_WEBSERVICE_IP/lookup/4

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

● none ● form-data ● **x-www-form-urlencoded** ● raw ● binary ● GraphQL

	KEY	VALUE
<input checked="" type="checkbox"/>	cost	170
	Key	Value

Body Cookies Headers (7) Test Results ⚙ Status

Pretty Raw Preview Visualize JSON ↕

```
1 {
2   "id": 4,
3   "name": "Cooking for the Impatient Graduate Student",
4   "cost": 170,
5   "quantity": 18,
```

update book information:https:// CATALOG_WEBSERVICE_IP/book/<book_id>

the type of method:PUT

PUT ▼ http://localhost:8000/CATALOG_WEBSERVICE_IP/book/4

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	cost	170	
	Key	Value	Description

Body Cookies Headers (7) Test Results 🌐 Status: 200 OK Time: 921

Pretty Raw Preview Visualize JSON 🔧

```
1 {  
2   "id": 4,  
3   "name": "Cooking for the Impatient Graduate Student",  
4   "cost": "170",  
}
```

PUT ▼ http://localhost:8000/CATALOG_WEBSERVICE_IP/book/4

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	KEY	VALUE
<input checked="" type="checkbox"/>	quntity	20
	Key	Value

Body Cookies Headers (7) Test Results 🌐 Sta

Pretty Raw Preview Visualize JSON 🔧

```
1 {  
2   "id": 4,  
3   "name": "Cooking for the Impatient Graduate Student",  
4   "cost": 170,  
5   "quntity": "20",  
}
```

if user want to buy a book enter: https://order_webservice_ip/buy/<book_id>

if book is exist the program print a message "bought book book_name " :

POST http://localhost:8000/order_webservice_ip/buy/2

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings

Query Params

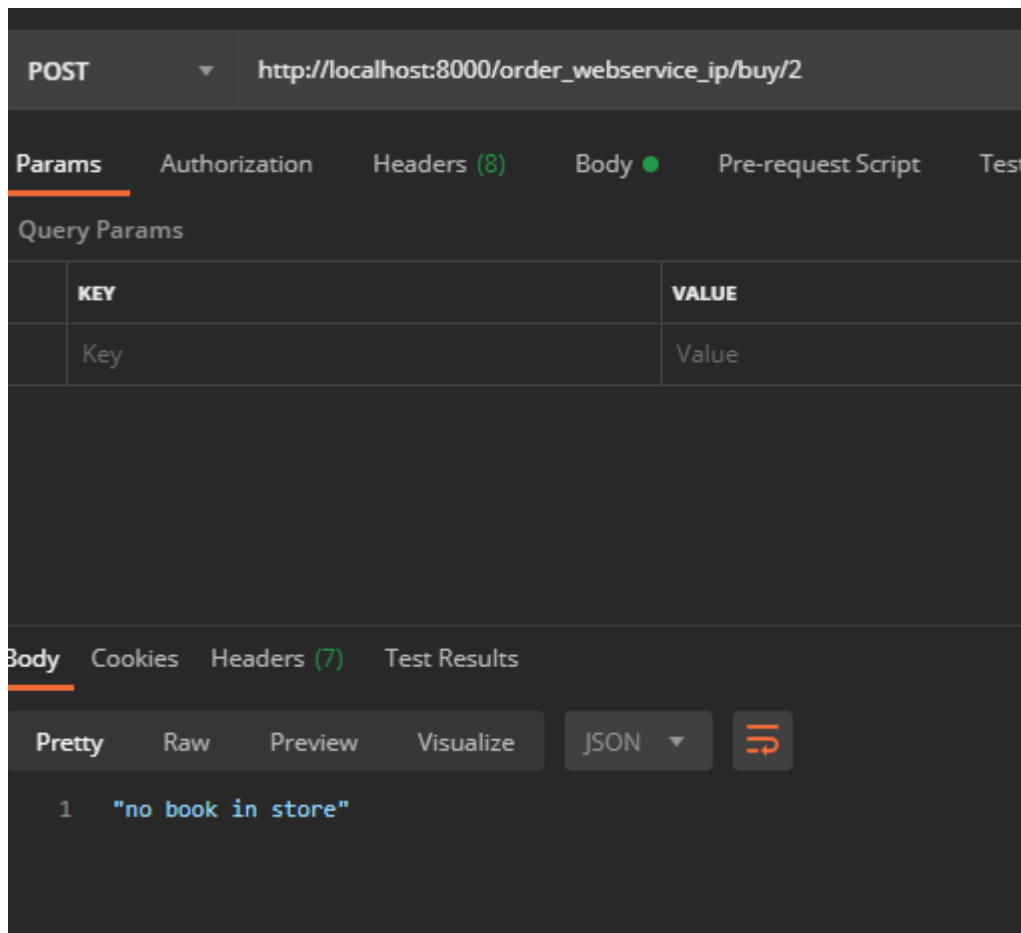
KEY	VALUE
Key	Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 {"bought book RPCs for Dummies"}
```

if not:



Appendix :

<https://github.com/suhaTaher/HW1>

github code:<https://github.com/ruyaatatreh99/DosPro.git>