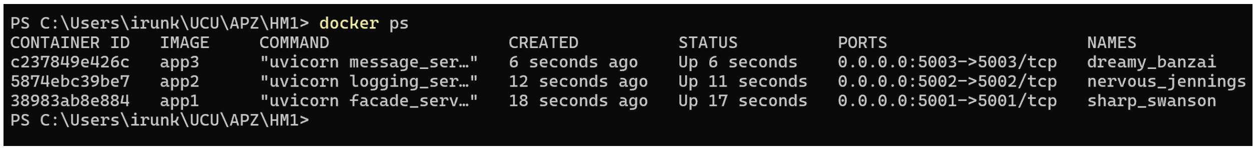
Homework 1

Protocol

After running three containers, we have three separate microservices:



```
PS C:\Users\irunk\UCU\APZ\HM1> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                    NAMES
c237849e426c   app3      "uvicorn message_ser…"   6 seconds ago    Up 6 seconds    0.0.0.0:5003->5003/tcp   dreamy_banzai
5874ebc39be7   app2      "uvicorn logging_ser…"   12 seconds ago   Up 11 seconds   0.0.0.0:5002->5002/tcp   nervous_jennings
38983ab8e884   app1      "uvicorn facade_serv…"   18 seconds ago   Up 17 seconds   0.0.0.0:5001->5001/tcp   sharp_swanson
PS C:\Users\irunk\UCU\APZ\HM1>
```

app1 — *facade service*

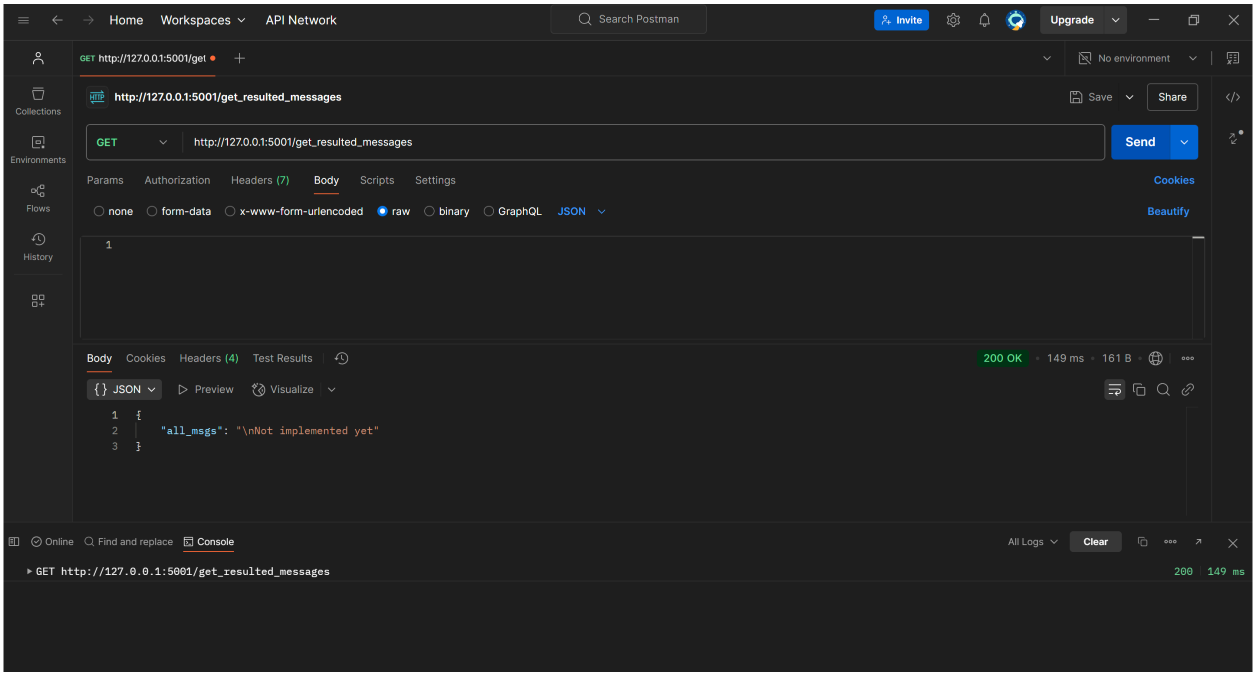
app2 —*logging service*

app3 — *message service*

Due to the fact that these are separate containers, they work as separate services and can exist without each other (generally without taking into account the fact that the post and get methods still require urls).

For testing the **Postman** will be used.

To show that there is nothing in the hash table at the beginning, let's call `GET` to the facade service:



http://127.0.0.1:5001/get_resulted_messages

GET http://127.0.0.1:5001/get_resulted_messages

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

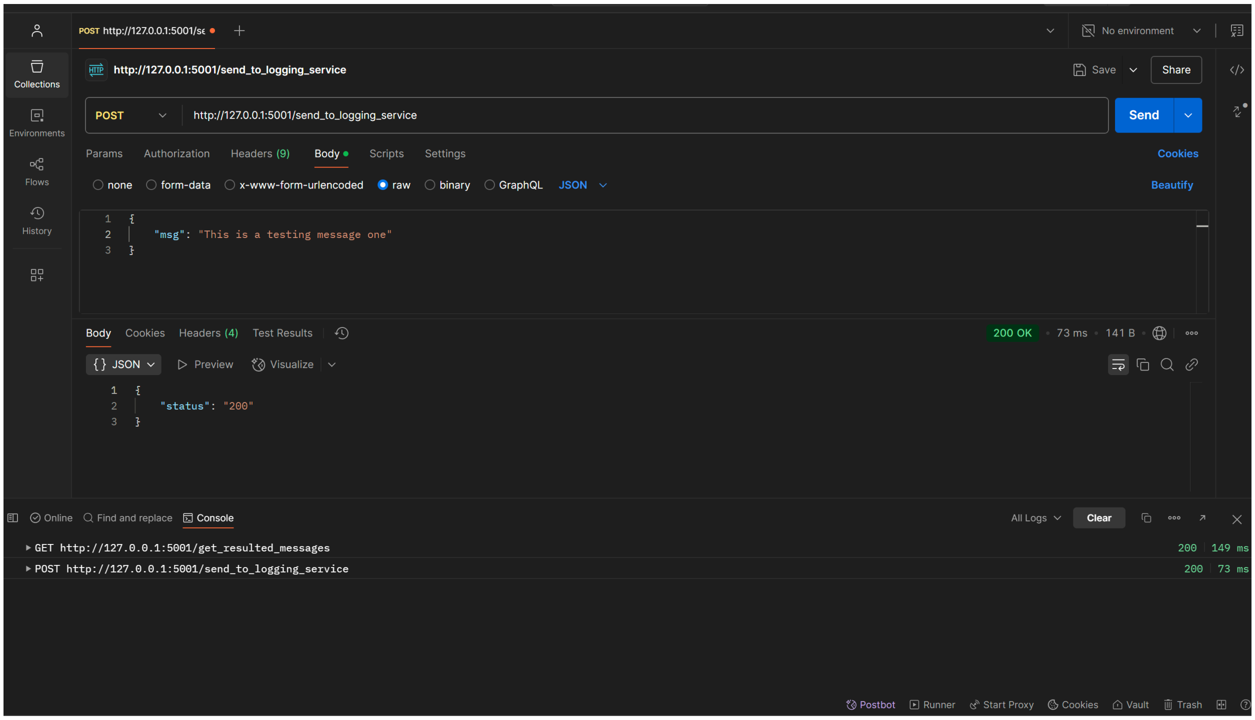
200 OK · 149 ms · 161 B

```
{
    "all_msgs": "\nNot implemented yet"
}
```

```
▸ GET http://127.0.0.1:5001/get_resulted_messages                200  149 ms
```

As we can see, the output contains only “Not implemented” as we are also calling `GET` to the message service, which returns a static text, but there are no messages.

Let’s add some messages:



POST http://127.0.0.1:5001/se

http://127.0.0.1:5001/send_to_logging_service

POST http://127.0.0.1:5001/send_to_logging_service

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

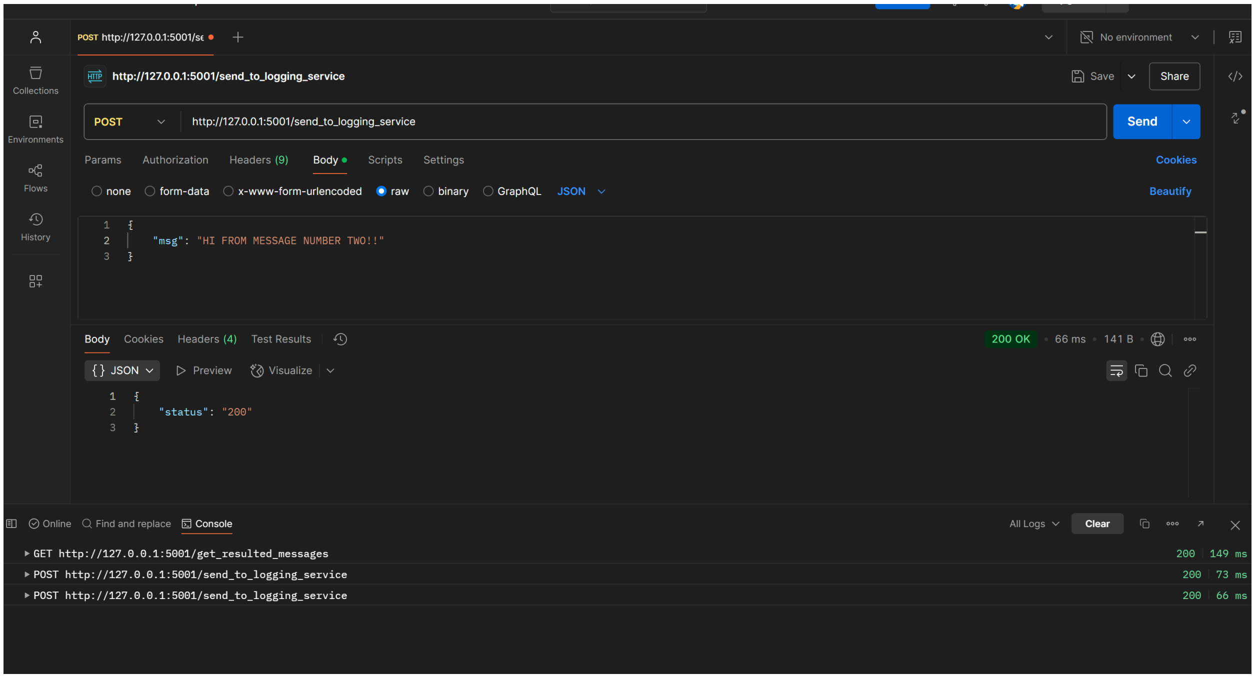
```
{
    "msg": "This is a testing message one"
}
```

Body Cookies Headers (4) Test Results 200 OK • 73 ms • 141 B

{ } JSON

```
{
    "status": "200"
}
```

```
▸ GET http://127.0.0.1:5001/get_resulted_messages                200  149 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service              200   73 ms
```



POST http://127.0.0.1:5001/se

http://127.0.0.1:5001/send_to_logging_service

POST http://127.0.0.1:5001/send_to_logging_service

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

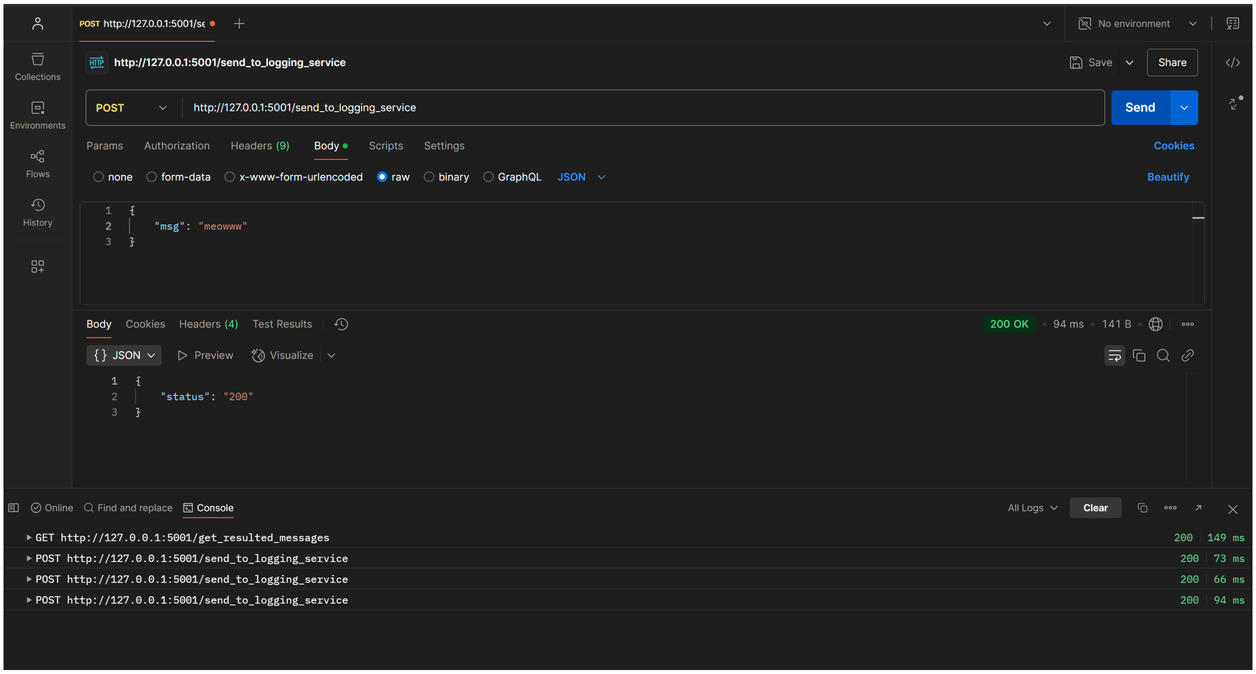
```
{
    "msg": "HI FROM MESSAGE NUMBER TWO!!"
}
```

Body Cookies Headers (4) Test Results 200 OK • 66 ms • 141 B

{ } JSON

```
{
    "status": "200"
}
```

```
▸ GET http://127.0.0.1:5001/get_resulted_messages                200  149 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service              200   73 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service              200   66 ms
```



http://127.0.0.1:5001/send_to_logging_service

POST http://127.0.0.1:5001/send_to_logging_service

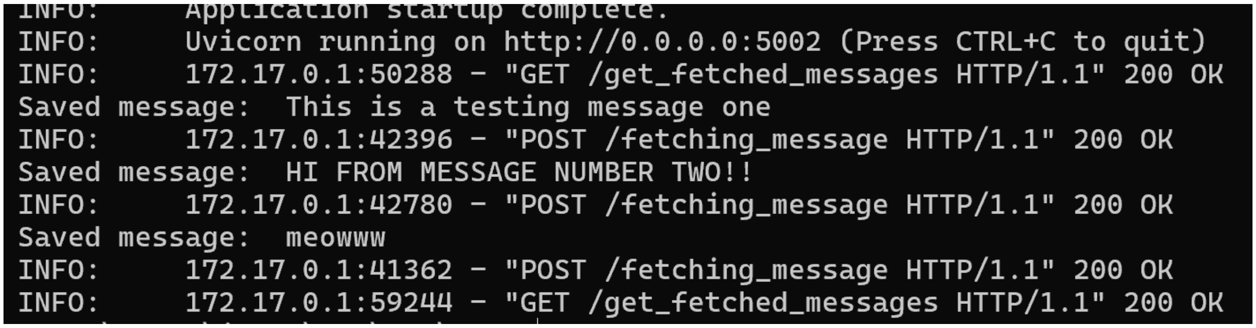
```
{
    "msg": "meowww"
}
```

200 OK · 94 ms · 141 B

```
{
    "status": "200"
}
```

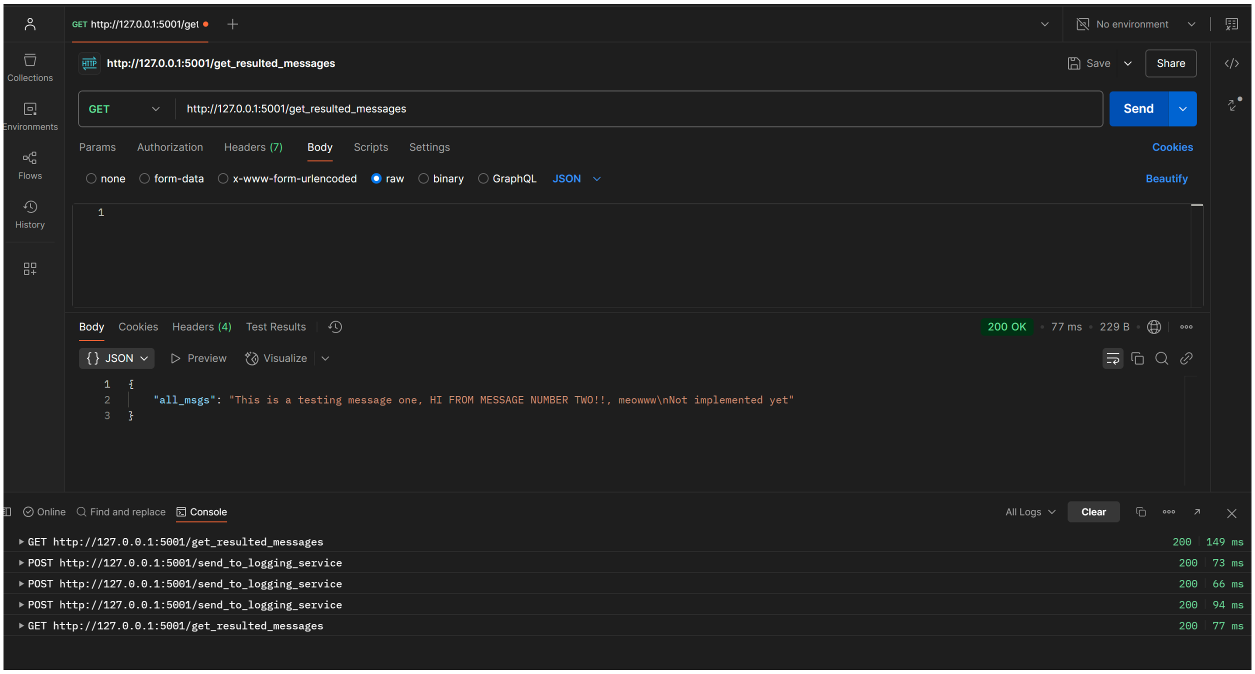
```
▸ GET http://127.0.0.1:5001/get_resulted_messages          200  149 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service       200   73 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service       200   66 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service       200   94 ms
```

We can take a look into the console to see if it printed messages that it recorded in the logging microservice:



```
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:5002 (Press CTRL+C to quit)
INFO:     172.17.0.1:50288 - "GET /get_fetched_messages HTTP/1.1" 200 OK
Saved message:  This is a testing message one
INFO:     172.17.0.1:42396 - "POST /fetching_message HTTP/1.1" 200 OK
Saved message:  HI FROM MESSAGE NUMBER TWO!!
INFO:     172.17.0.1:42780 - "POST /fetching_message HTTP/1.1" 200 OK
Saved message:  meowww
INFO:     172.17.0.1:41362 - "POST /fetching_message HTTP/1.1" 200 OK
INFO:     172.17.0.1:59244 - "GET /get_fetched_messages HTTP/1.1" 200 OK
```

If we do a GET method to the facade service, we can see all messages that we provided:



GET http://127.0.0.1:5001/get_resulted_messages

```
{
    "all_msgs": "This is a testing message one, HI FROM MESSAGE NUMBER TWO!!, meowww\nNot implemented yet"
}
```

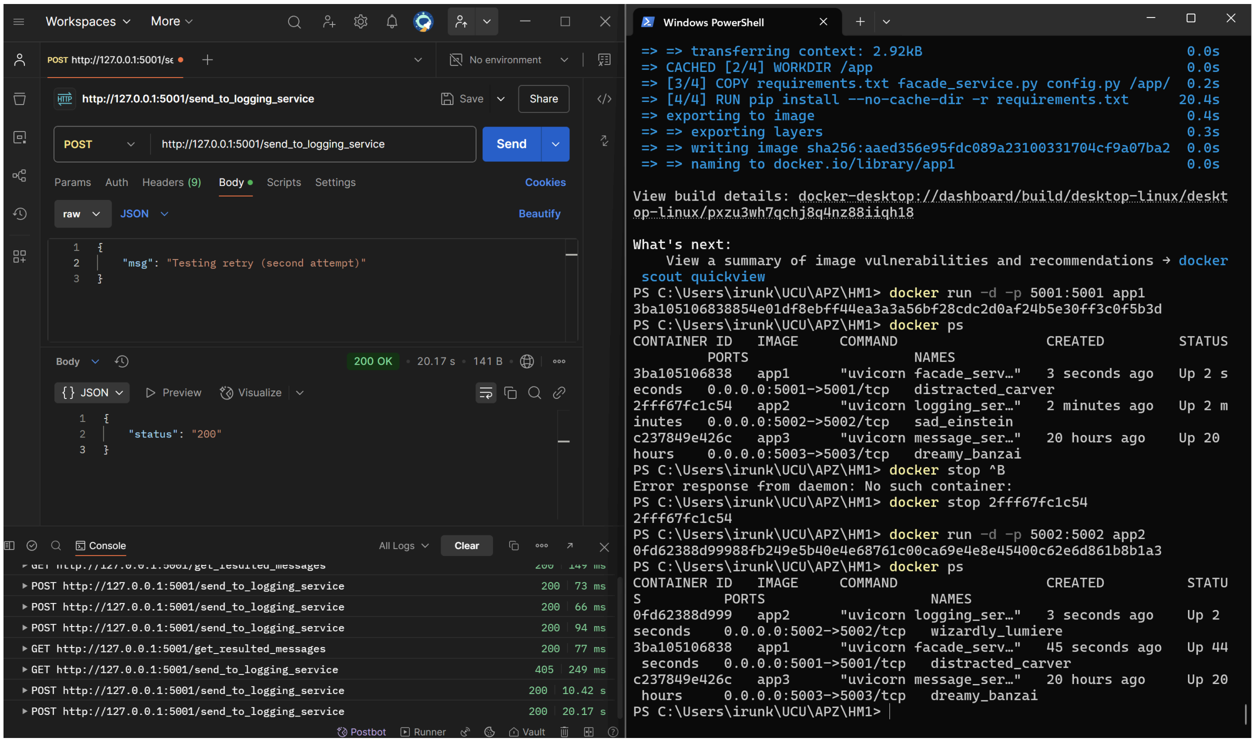
```
▸ GET http://127.0.0.1:5001/get_resulted_messages          200  149 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service       200  73 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service       200  66 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service       200  94 ms
▸ GET http://127.0.0.1:5001/get_resulted_messages          200  77 ms
```

Retrying and deduplication

Basically my implementation does 5 retries with a difference of 10 seconds.

To test, you can stop the docker container with the second server and start it after some time (within the time when the retries are still happening). This will help us see that when it finds a connection, it can successfully send a message (i.e. test the retries functionality itself) and in addition, we can see that even if multiple `POST` requests occur, it will only save one such message (i.e. deduplication work).

Example:


Workspaces
More
POST http://127.0.0.1:5001/se
No environment
http://127.0.0.1:5001/send_to_logging_service
Save
Share
POST
http://127.0.0.1:5001/send_to_logging_service
Send
Params
Auth
Headers (9)
Body
Scripts
Settings
Cookies
raw
JSON
Beautify

```
{
    "msg": "Testing retry (second attempt)"
}
```

```
{
    "status": "200"
}
```

```
▸ GET http://127.0.0.1:5001/get_resulted_messages          200   149 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service       200    73 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service       200    66 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service       200    94 ms
▸ GET http://127.0.0.1:5001/get_resulted_messages          200    77 ms
▸ GET http://127.0.0.1:5001/send_to_logging_service        405   249 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service       200   10.42 s
▸ POST http://127.0.0.1:5001/send_to_logging_service       200   20.17 s
```

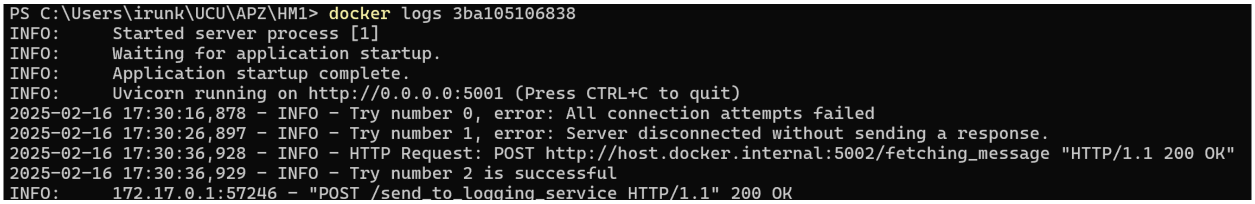
```
 => => transferring context: 2.92kB                                   0.0s
 => CACHED [2/4] WORKDIR /app                                         0.0s
 => [3/4] COPY requirements.txt facade_service.py config.py /app/     0.2s
 => [4/4] RUN pip install --no-cache-dir -r requirements.txt         20.4s
 => exporting to image                                                0.4s
 => => exporting layers                                               0.3s
 => => writing image sha256:aaed356e95fdc089a23100331704cf9a07ba2     0.0s
 => => naming to docker.io/library/app1                               0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/deskt
op-linux/pxzu3wh7qchj8q4nz88iiqh18

What's next:
    View a summary of image vulnerabilities and recommendations → docker
 scout quickview
PS C:\Users\irunk\UCU\APZ\HM1> docker run -d -p 5001:5001 app1
3ba105106838854e01df8ebff44ea3a3a56bf28cdc2d0af24b5e30ff3c0f5b3d
PS C:\Users\irunk\UCU\APZ\HM1> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS
         PORTS                    NAMES
3ba105106838   app1      "uvicorn facade_serv…"   3 seconds ago    Up 2 s
econds   0.0.0.0:5001->5001/tcp   distracted_carver
2fff67fc1c54   app2      "uvicorn logging_ser…"   2 minutes ago    Up 2 m
inutes   0.0.0.0:5002->5002/tcp   sad_einstein
c237849e426c   app3      "uvicorn message_ser…"   20 hours ago     Up 20
hours    0.0.0.0:5003->5003/tcp   dreamy_banzai
PS C:\Users\irunk\UCU\APZ\HM1> docker stop ^B
Error response from daemon: No such container:
PS C:\Users\irunk\UCU\APZ\HM1> docker stop 2fff67fc1c54
2fff67fc1c54
PS C:\Users\irunk\UCU\APZ\HM1> docker run -d -p 5002:5002 app2
0fd62388d99988fb249e5b40e4e68761c00ca69e4e8e45400c62e6d861b8b1a3
PS C:\Users\irunk\UCU\APZ\HM1> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATU
S          PORTS                    NAMES
0fd62388d999   app2      "uvicorn logging_ser…"   3 seconds ago     Up 2
seconds    0.0.0.0:5002->5002/tcp   wizardly_lumiere
3ba105106838   app1      "uvicorn facade_serv…"   45 seconds ago    Up 44
 seconds   0.0.0.0:5001->5001/tcp   distracted_carver
c237849e426c   app3      "uvicorn message_ser…"   20 hours ago      Up 20
 hours     0.0.0.0:5003->5003/tcp   dreamy_banzai
PS C:\Users\irunk\UCU\APZ\HM1>
```

At the right you can see a screenshot of the terminal, where you can see that the container stops and then starts. On the left you can see the response: `{"status": "200"}` that means that it sent a message.

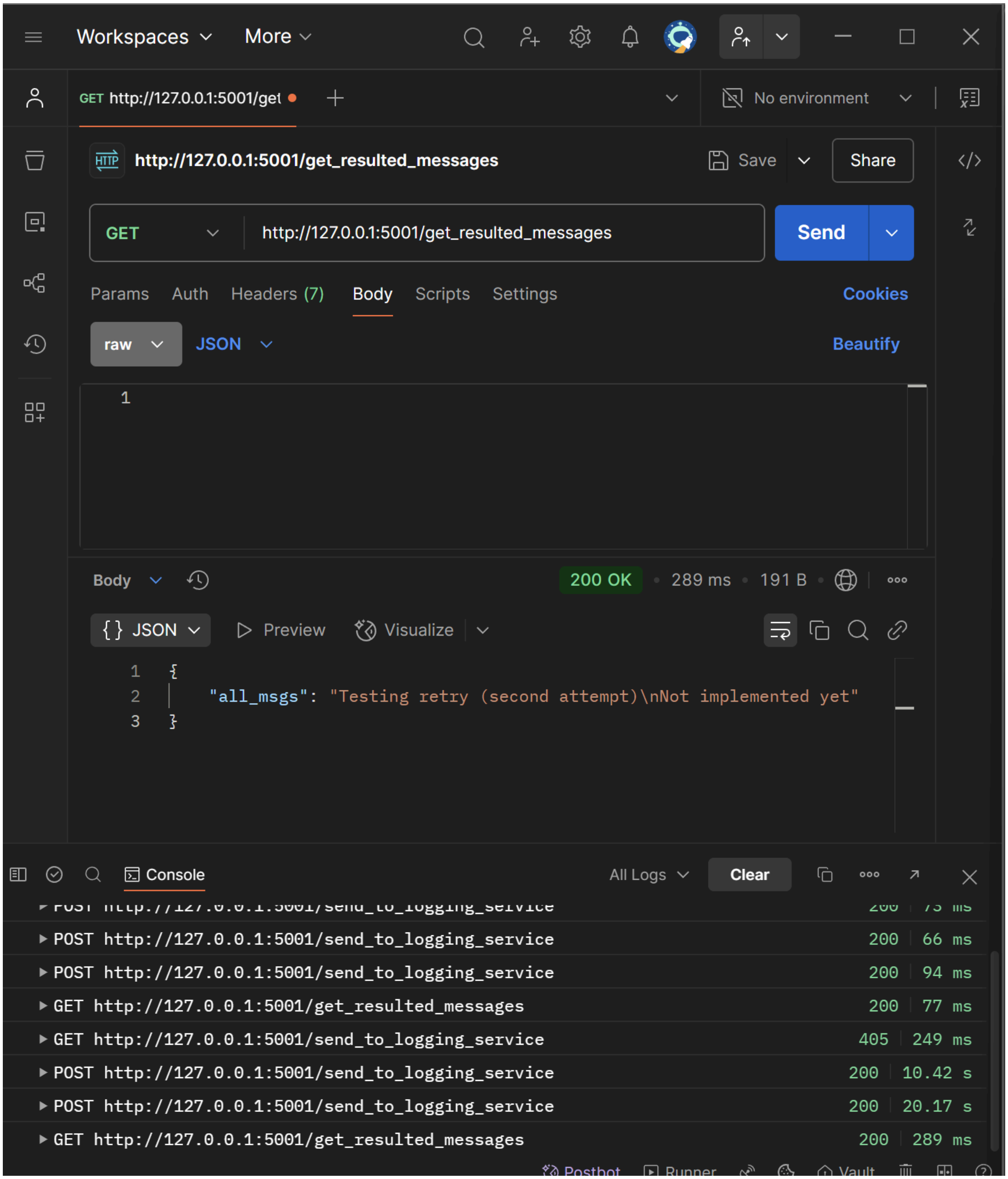
Here is the logging during this action:



```
PS C:\Users\irunk\UCU\APZ\HM1> docker logs 3ba105106838
INFO:     Started server process [1]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:5001 (Press CTRL+C to quit)
2025-02-16 17:30:16,878 - INFO - Try number 0, error: All connection attempts failed
2025-02-16 17:30:26,897 - INFO - Try number 1, error: Server disconnected without sending a response.
2025-02-16 17:30:36,928 - INFO - HTTP Request: POST http://host.docker.internal:5002/fetching_message "HTTP/1.1 200 OK"
2025-02-16 17:30:36,929 - INFO - Try number 2 is successful
INFO:     172.17.0.1:57246 - "POST /send_to_logging_service HTTP/1.1" 200 OK
```

It can be seen that there were three retry operations, the third of which was successful.

If we try to get the message, we can see that it received and saved one such message (i.e. after retry we have no repeated identical messages):



http://127.0.0.1:5001/get_resulted_messages

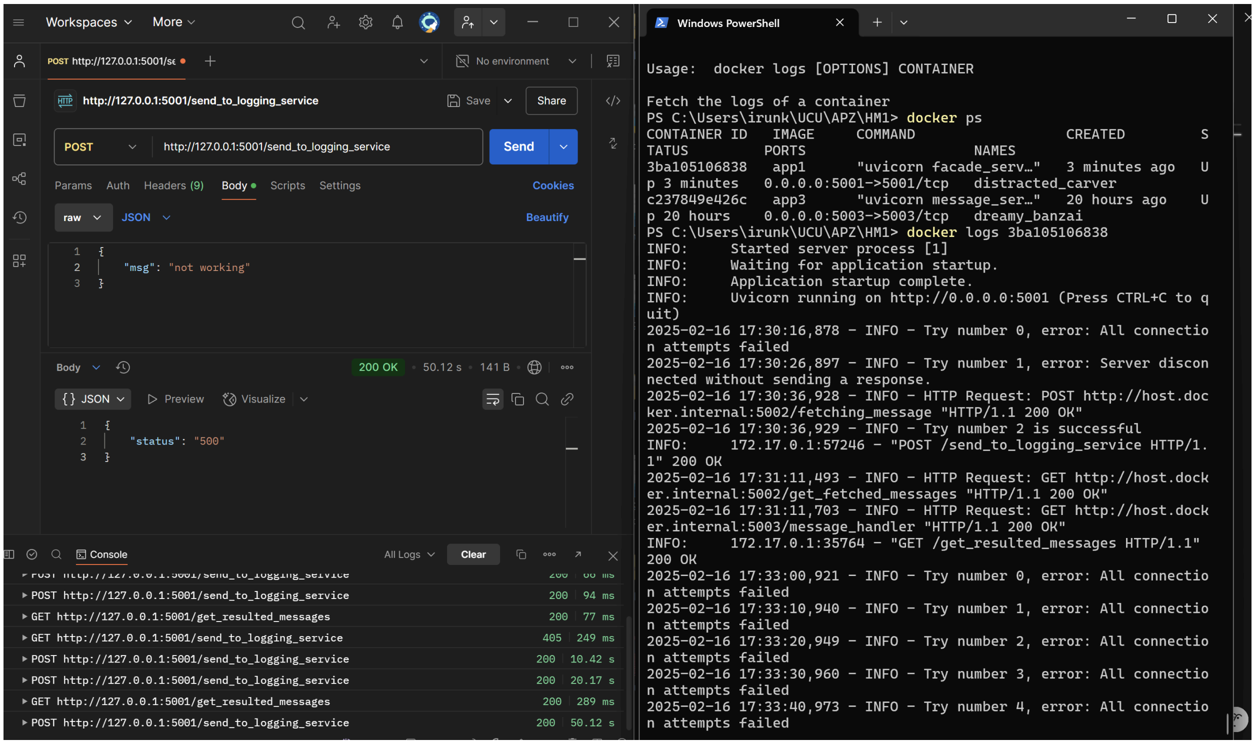
GET http://127.0.0.1:5001/get_resulted_messages

200 OK • 289 ms • 191 B

```
{
    "all_msgs": "Testing retry (second attempt)\nNot implemented yet"
}
```

```
▸ POST http://127.0.0.1:5001/send_to_logging_service          200 | 73 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service          200 | 66 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service          200 | 94 ms
▸ GET http://127.0.0.1:5001/get_resulted_messages             200 | 77 ms
▸ GET http://127.0.0.1:5001/send_to_logging_service          405 | 249 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service        200 | 10.42 s
▸ POST http://127.0.0.1:5001/send_to_logging_service        200 | 20.17 s
▸ GET http://127.0.0.1:5001/get_resulted_messages            200 | 289 ms
```

If we test with the second server completely disconnected, we can see that it returns a status of 500 and all retries are simply not successful:



http://127.0.0.1:5001/send_to_logging_service

POST http://127.0.0.1:5001/send_to_logging_service

```
{
    "msg": "not working"
}
```

200 OK • 50.12 s • 141 B

```
{
    "status": "500"
}
```

```
▸ POST http://127.0.0.1:5001/send_to_logging_service       200  66 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service       200  94 ms
▸ GET http://127.0.0.1:5001/get_resulted_messages          200  77 ms
▸ GET http://127.0.0.1:5001/send_to_logging_service        405  249 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service       200  10.42 s
▸ POST http://127.0.0.1:5001/send_to_logging_service       200  20.17 s
▸ GET http://127.0.0.1:5001/get_resulted_messages          200  289 ms
▸ POST http://127.0.0.1:5001/send_to_logging_service       200  50.12 s
```

```
Usage:  docker logs [OPTIONS] CONTAINER

Fetch the logs of a container
PS C:\Users\irunk\UCU\APZ\HM1> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         S
TATUS          PORTS                    NAMES
3ba105106838   app1      "uvicorn facade_serv…"   3 minutes ago   U
p 3 minutes   0.0.0.0:5001->5001/tcp   distracted_carver
c237849e426c   app3      "uvicorn message_ser…"   20 hours ago    U
p 20 hours    0.0.0.0:5003->5003/tcp   dreamy_banzai
PS C:\Users\irunk\UCU\APZ\HM1> docker logs 3ba105106838
INFO:     Started server process [1]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:5001 (Press CTRL+C to q
uit)
2025-02-16 17:30:16,878 - INFO - Try number 0, error: All connectio
n attempts failed
2025-02-16 17:30:26,897 - INFO - Try number 1, error: Server discon
nected without sending a response.
2025-02-16 17:30:36,928 - INFO - HTTP Request: POST http://host.doc
ker.internal:5002/fetching_message "HTTP/1.1 200 OK"
2025-02-16 17:30:36,929 - INFO - Try number 2 is successful
INFO:     172.17.0.1:57246 - "POST /send_to_logging_service HTTP/1.
1" 200 OK
2025-02-16 17:31:11,493 - INFO - HTTP Request: GET http://host.dock
er.internal:5002/get_fetched_messages "HTTP/1.1 200 OK"
2025-02-16 17:31:11,703 - INFO - HTTP Request: GET http://host.dock
er.internal:5003/message_handler "HTTP/1.1 200 OK"
INFO:     172.17.0.1:35764 - "GET /get_resulted_messages HTTP/1.1"
200 OK
2025-02-16 17:33:00,921 - INFO - Try number 0, error: All connectio
n attempts failed
2025-02-16 17:33:10,940 - INFO - Try number 1, error: All connectio
n attempts failed
2025-02-16 17:33:20,949 - INFO - Try number 2, error: All connectio
n attempts failed
2025-02-16 17:33:30,960 - INFO - Try number 3, error: All connectio
n attempts failed
2025-02-16 17:33:40,973 - INFO - Try number 4, error: All connectio
n attempts failed
```