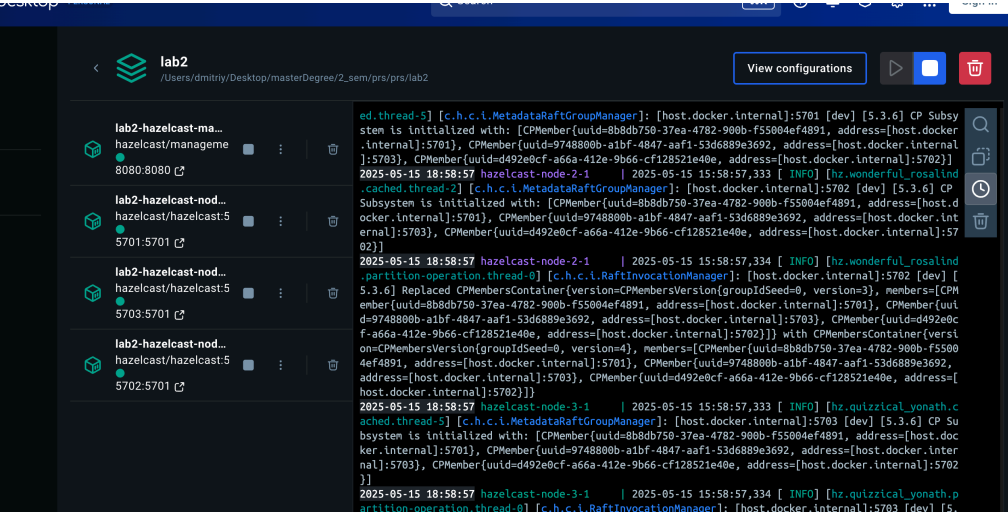


Рябко Дмитро ФБ-42мп



Name	In Memory F...	Entries	Entry Memory	Backup Mem...	Events
messages	BINARY	3	492.00 B	492.00 B	0

Для початку запустимо Config service (порт: 8000) щоб фасад зміг отримувати налаштування:

```

✓ SERVICES = {
✓   "logging-service": [
       "http://127.0.0.1:8011",
       "http://127.0.0.1:8012",
       "http://127.0.0.1:8013",
       ],
✓   "messages-service": [
       "http://127.0.0.1:8004"
       ]
✓ }

✓ class ServiceList(BaseModel):
    instances: list[str]

    @app.get("/services/{service_name}", response_model=ServiceList)

```

```

dmitriy@Dmitrys-MacBook-Air lab3 % python3 config_service.py
INFO: Will watch for changes in these directories: ['/Users/dmitriy/Desktop/masterDegree/2_sem/prs/lab3']
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
INFO: Started reloader process [78427] using StatReload
INFO: Started server process [78429]
INFO: Waiting for application startup.
INFO: Application startup complete.

```

Запуск Facade service та Message service(заглушки):

```

Python x
dmitriy@Dmitrys-MacBook-Air lab3 % python3 facade_service.py
INFO: Will watch for changes in these directories: ['/Users/dmitriy/Desktop/masterDegree/2_sem/prs/lab3']
INFO: Uvicorn running on http://0.0.0.0:8010 (Press CTRL+C to quit)
INFO: Started reloader process [78519] using StatReload
INFO: Started server process [78522]
INFO: Waiting for application startup.
INFO: Application startup complete.

Python x
dmitriy@Dmitrys-MacBook-Air lab3 % python3 messages_service.py
INFO: Will watch for changes in these directories: ['/Users/dmitriy/Desktop/masterDegree/2_sem/prs/lab3']
INFO: Uvicorn running on http://0.0.0.0:8004 (Press CTRL+C to quit)
INFO: Started reloader process [78527] using StatReload
INFO: Started server process [78529]
INFO: Waiting for application startup.
INFO: Application startup complete.

```

Запуск трьох екземплярів Logging Service:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
dmitriy@Dmitrys-MacBook-Air lab3 % uvicorn logging_service:app --host 0.0.0.0 --port 8011 --reload
INFO: Will watch for changes in these directories: ['/Users/dmitriy/Desktop/masterDegree/2_sem/prs/lab3']
INFO: Uvicorn running on http://0.0.0.0:8011 (Press CTRL+C to quit)
INFO: Started reloader process [78639] using StatReload
INFO: Started server process [78641]
INFO: Waiting for application startup.
INFO: Application startup complete.

dmitriy@Dmitrys-MacBook-Air lab3 % uvicorn logging_service:app --host 0.0.0.0 --port 8012 --reload
INFO: Will watch for changes in these directories: ['/Users/dmitriy/Desktop/masterDegree/2_sem/prs/lab3']
INFO: Uvicorn running on http://0.0.0.0:8012 (Press CTRL+C to quit)
INFO: Started reloader process [78645] using StatReload
INFO: Started server process [78647]
INFO: Waiting for application startup.
INFO: Application startup complete.

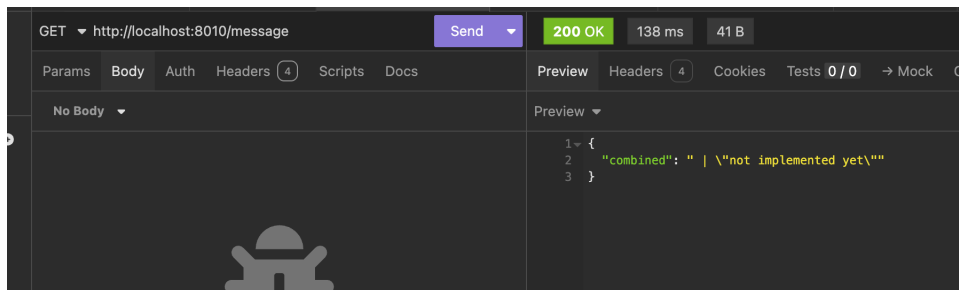
dmitriy@Dmitrys-MacBook-Air lab3 % uvicorn logging_service:app --host 0.0.0.0 --port 8013 --reload
INFO: Will watch for changes in these directories: ['/Users/dmitriy/Desktop/masterDegree/2_sem/prs/lab3']
INFO: Uvicorn running on http://0.0.0.0:8013 (Press CTRL+C to quit)
INFO: Started reloader process [78651] using StatReload
INFO: Started server process [78653]
INFO: Waiting for application startup.
INFO: Application startup complete.

```

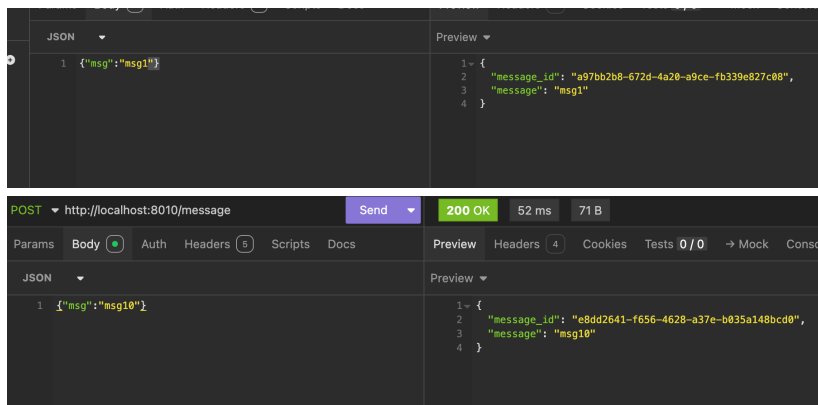
```
config_service.py
3 import uvicorn
4
5 app = FastAPI()
6
7 SERVICES = {
8     "logging-service":
9         "http://127.0.0.1:8011",
10     "http://127.0.0.1:8011",
11     "http://127.0.0.1:8011",
12 },
13 "messages-service":
14     "http://127.0.0.1:8011",
15 }
16
17
18 class ServiceList(BaseModel):
19     instances: list[str] = Field()
20
21
22 @app.get("/services/{service}")
23 async def get_service(service: str):
24     instances = SERVICE_LIST[service]
25     return ServiceList(instances=instances)
```

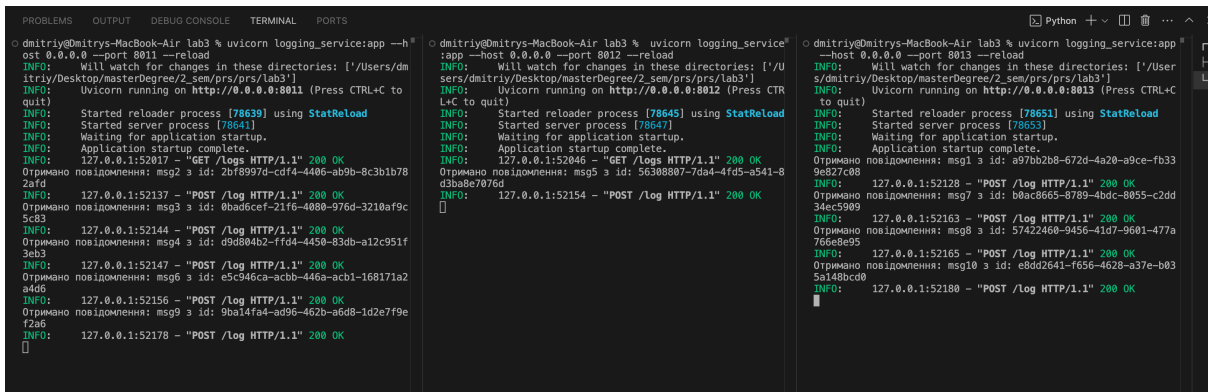
```
terminal
dmitriy@Dmitrys-MacBook-Air lab3 % python3 config_service.py
INFO: Will watch for changes in these directories: ['/Users/dmitriy/Desktop/masterDegree/2_sem/prs/prs/lab3']
INFO: Uvicorn running on http://0.0.0.0:8011 (Press CTRL+C to quit)
INFO: Started reloader process [78427] using StatReload
INFO: Started server process [78429]
INFO: Waiting for application startup.
INFO: Application startup complete.
```

Спробуємо отримати статичне повідомлення від Message service:



Відправимо 10 повідомлень через Facade service:

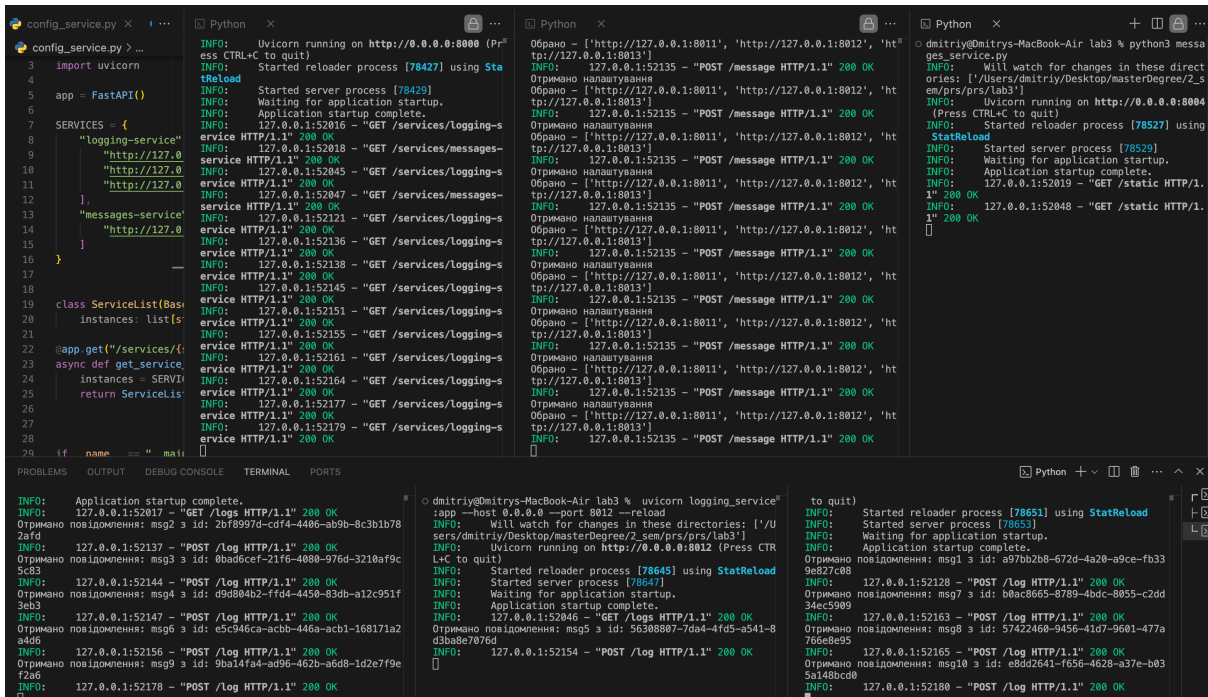




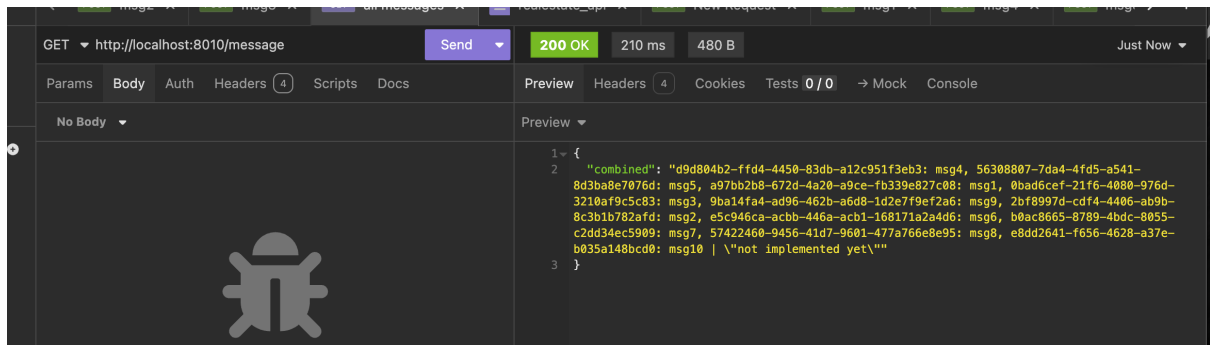
На скріншоті видно, що вибір екземпляра сервісу відбувається рандомно

```
for inst in random.sample(instances, len(instances)):  
    for attempt in range(3):  
        try:  
            r = requests.post(f"{inst}/log", json=payload, timeout=5)  
            r.raise_for_status()  
            return {"message_id": message_id, "message": message.msg}  
        except Exception as e:  
            last_error = e  
            time.sleep(2)  
    raise HTTPException(status_code=500, detail=f"Не вдалося надіслати повідомлення: {last_error}")
```

Ось логи всіх сервісів:



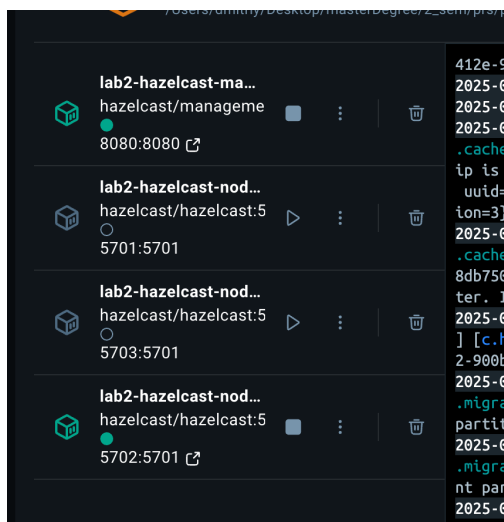
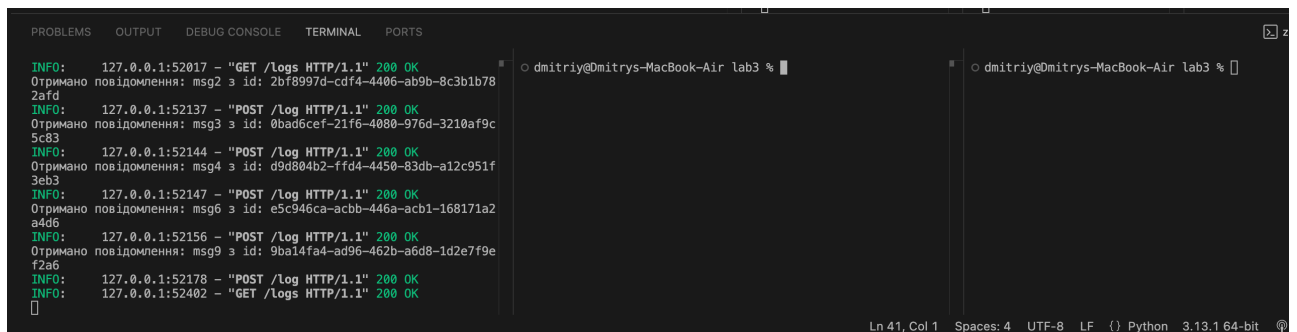
Прочитаємо всі повідомлення:

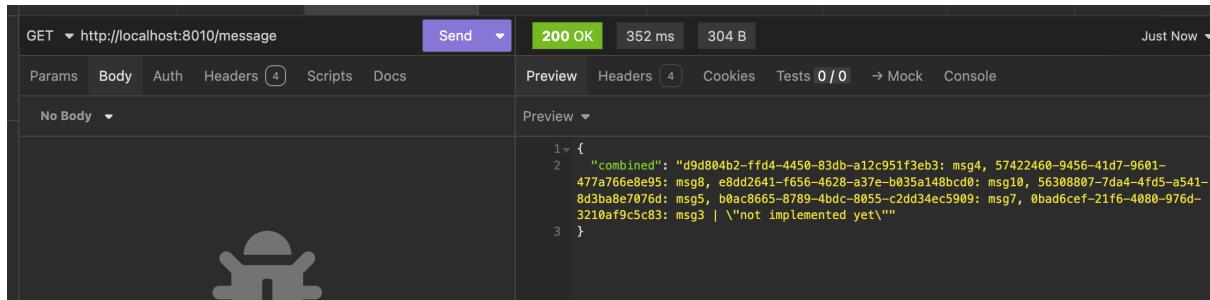


Повідомлення збереглися без дуплікатів:

```
def log_message(log_msg: LogMessage):
    existing = messages_map.get(log_msg.id)
    if existing is not None:
        return {"status": "ok", "id": log_msg.id, "note": "Ігнорування дуплікату"}
    messages_map.put(log_msg.id, log_msg.msg)
    print(f"Отримано повідомлення: {log_msg.msg} з id: {log_msg.id}")
    return {"detail": "Повідомлення успішно збережено"}
```

Вимкнемо два екземпляри Logging service:





На скріншоті можна побачити, що частина повідомлень була втрачена, так як у нас не налаштована реплікація Distrubuted Map