

Лабораторна робота №1

Золотов Іван ФБ-31мп

Посилання на GitHub

<https://github.com/SideOctopus/Lab1>

Facade-service

Цей сервіс приймає повідомлення від клієнта, генерує унікальний ідентифікатор для кожного повідомлення і передає його іншим мікросервісам для подальшої обробки.

```
1  from flask import Flask, request, jsonify
2  import uuid
3  import requests
4
5  app = Flask(__name__)
6
7  @app.route('/data', methods=['GET', 'POST'])
8  def handle_data():
9      if request.method == 'POST':
10         # Get the message from the POST request
11         message = request.get_data(as_text=True)
12         # Generate a unique ID for the message
13         unique_id = str(uuid.uuid4())
14         # Create the payload to send to external services
15         payload = {'id': unique_id, 'message': message}
16
17         try:
18             # Send the payload to the logging service
19             log_response = requests.post('http://localhost:5001/log', json=payload)
20             log_response.raise_for_status()
21
22             # Send the payload to the messages service
23             message_response = requests.post('http://localhost:5002/message', json=payload)
24             message_response.raise_for_status()
25
26         except requests.exceptions.RequestException as e:
27             # Return an error response if any request fails
28             return jsonify({"status": "error", "message": str(e)}), 500
29
30         # Return a success response with the unique ID
31         return jsonify({"status": "message received", "id": unique_id}), 201
32
33     elif request.method == 'GET':
34         try:
35             # Get logs from the logging service
36             log_response = requests.get('http://localhost:5001/logs')
37             log_response.raise_for_status()
38
39             # Get messages from the messages service
40             message_response = requests.get('http://localhost:5002/messages')
41             message_response.raise_for_status()
42
43         except requests.exceptions.RequestException as e:
44             # Return an error response if any request fails
45             return jsonify({"status": "error", "message": str(e)}), 500
46
47         # Return the logs and messages in the response
48         return jsonify({
49             "logs": log_response.json(),
50             "messages": message_response.json()
51         })
52
53 if __name__ == '__main__':
54     # Start the Flask application
55     with app.app_context():
56         app.run(port=5000, debug=True)
```

Logging-service

Цей сервіс зберігає отримані повідомлення разом з їхніми унікальними ідентифікаторами. У разі отримання GET-запиту він повертає всі збережені повідомлення.

```

1  from flask import Flask, request, jsonify
2
3  app = Flask(__name__)
4
5  # Dictionary to store logs
6  log_storage = {}
7
8  @app.route('/log', methods=['POST'])
9  def log_message():
10     # Get the JSON data from the POST request
11     data = request.get_json()
12     # Store the message in log_storage with the unique ID as the key
13     log_storage[data['id']] = data['message']
14     # Return a success response
15     return jsonify({"status": "logged"}), 201
16
17 @app.route('/logs', methods=['GET'])
18 def get_logs():
19     # Return all stored logs
20     return jsonify(log_storage)
21
22 if __name__ == '__main__':
23     # Start the Flask application for logging service
24     app.run(port=5001, debug=True)
25

```

Messages-service

Цей сервіс відповідає за зберігання повідомлень. При отриманні GET-запиту він повертає всі збережені повідомлення.

```

1  from flask import Flask, request, jsonify
2
3  app = Flask(__name__)
4
5  # Dictionary to store messages
6  messages_storage = {}
7
8  @app.route('/message', methods=['POST'])
9  def save_message():
10     # Get the JSON data from the POST request
11     data = request.get_json()
12     # Store the message in messages_storage with the unique ID as the key
13     messages_storage[data['id']] = data['message']
14     # Return a success response
15     return jsonify({"status": "message saved"}), 201
16
17 @app.route('/messages', methods=['GET'])
18 def get_messages():
19     # Return all stored messages
20     return jsonify(messages_storage)
21
22 if __name__ == '__main__':
23     # Start the Flask application for messages service
24     app.run(port=5002, debug=True)
25

```

Застосунок клієнта

```

1  import requests
2
3  def main():
4      while True:
5          print("1) Send POST request")
6          print("2) Send GET request")
7          print("3) Exit")
8          choice = int(input("Your choice: "))
9
10         if choice == 1:
11             message = input("Enter your message: ")
12             response = requests.post('http://localhost:5000/data', data=message)
13             print('Response:', response.json())
14         elif choice == 2:
15             response = requests.get('http://localhost:5000/data')
16             print('Response:', response.json())
17         elif choice == 3:
18             break
19         else:
20             print("Invalid choice. Try again.")
21
22 if __name__ == "__main__":
23     main()
24

```

Запустимо застосунок клієнта

```

1) Send POST request
2) Send GET request
3) Exit
Your choice: 1
Enter your message: Hello, World!
Response: {'id': '597899bb-0431-4fc9-aca6-b218f094b388', 'status': 'message received'}
1) Send POST request
2) Send GET request
3) Exit
Your choice: 2
Response: {'logs': {'597899bb-0431-4fc9-aca6-b218f094b388': 'Hello, World!'}, 'messages': {'597899bb-0431-4fc9-aca6-b218f094b388': 'Hello, World!'}}
1) Send POST request
2) Send GET request
3) Exit
Your choice: 

```