

Лабораторна робота №3

Золотов Іван ФБ-31мп

Посилання на GitHub

https://github.com/SideOctopus/Lab_3

Реалізація мікросервісів:

```
1 from flask import Flask, request, jsonify
2 import requests
3 import random
4
5 app = Flask(__name__)
6
7 # Список URL-адрес, які вказують на сервіси журналювання
8 logging_services = [
9     "http://localhost:5001",
10    "http://localhost:5002",
11    "http://localhost:5003"
12 ]
13
14 @app.route('/data', methods=['POST', 'GET'])
15 def handle_data():
16     if request.method == 'POST':
17         # Обробка POST-запиту
18         message = request.data.decode('utf-8') # Отримання повідомлення з тіла запиту
19         target_service = random.choice(logging_services) # Вибір випадкового сервісу журналювання
20         response = requests.post(f'{target_service}/data', json={'message': message}) # Відправка POST-запиту до вибраного сервісу
21         return response.text # Повернення текстової відповіді від сервісу журналювання
22
23     elif request.method == 'GET':
24         # Обробка GET-запиту
25         target_service = random.choice(logging_services) # Вибір випадкового сервісу журналювання
26         response = requests.get(f'{target_service}/data') # Відправка GET-запиту до вибраного сервісу
27         return jsonify(response.json()) # Повернення JSON-відповіді від сервісу журналювання
28
29 if __name__ == '__main__':
30     app.run(port=5000, debug=True)
```

```
1 from flask import Flask, request, jsonify
2 import hazelcast
3 import sys
4
5 app = Flask(__name__)
6
7 # Підключення до кластера Hazelcast з вказаними учасниками
8 hz = hazelcast.HazelcastClient(cluster_members=[
9     "127.0.0.1:5701",
10    "127.0.0.1:5702",
11    "127.0.0.1:5703"
12 ])
13
14 # Отримання розподіленої мапи з назвою "messages"
15 distributed_map = hz.get_map("messages").blocking()
16
17 @app.route('/data', methods=['POST', 'GET'])
18 def data():
19     if request.method == 'POST':
20         # Обробка POST-запиту
21         message = request.json['message'] # Витягнення повідомлення з JSON-тіла запиту
22         key = str(len(distributed_map.key_set()) + 1) # Генерація унікального ключа для повідомлення
23         distributed_map.put(key, message) # Збереження повідомлення в розподілену мапу
24         return 'Message stored successfully' # Повернення підтвердження успішного збереження
25
26     elif request.method == 'GET':
27         # Обробка GET-запиту
28         # Отримання всіх повідомлень з розподіленої мапи
29         messages = {key: distributed_map.get(key) for key in distributed_map.key_set()}
30         return jsonify(messages) # Повернення повідомлень у вигляді JSON-відповіді
31
32 if __name__ == '__main__':
33     port = int(sys.argv[1]) # Отримання номера порту з аргументу командного рядка
34     app.run(port=port, debug=True) # Запуск додатка Flask на вказаному порту в режимі налагодження
```

```
1 from flask import Flask, jsonify, request
2
3 app = Flask(__name__)
4
5 messages = {}
6
7 @app.route('/data', methods=['POST', 'GET'])
8 def data():
9     if request.method == 'POST':
10         # Обробка POST-запиту
11         message = request.json['message'] # Отримання повідомлення з JSON-тіла запиту
12         key = len(messages) + 1 # Генерація унікального ключа для повідомлення
13         messages[key] = message # Збереження повідомлення в словник
14         return 'Message stored successfully' # Повернення підтвердження успішного збереження
15
16     elif request.method == 'GET':
17         # Обробка GET-запиту
18         return jsonify(messages) # Повернення всіх збережених повідомлень у вигляді JSON
19
20 if __name__ == '__main__':
21     app.run(port=5005, debug=True)
```

Клієнт

```

1  import requests
2
3  def main():
4      base_url = 'http://127.0.0.1:5000/data'
5
6      while True:
7          print("1) Send POST-request")
8          print("2) Send GET-request")
9          print("3) Exit")
10         choice = int(input("Your choice: "))
11
12         if choice == 1:
13             message = input("Enter your message: ")
14             response = requests.post(base_url, data=message)
15             print('POST response:', response.text)
16         elif choice == 2:
17             response = requests.get(base_url)
18             print('GET response:', response.json())
19         elif choice == 3:
20             break
21         else:
22             print("Invalid choice. Please try again.")
23
24     if __name__ == "__main__":
25         main()

```

Запускаємо loggingService

```

* Serving Flask app 'logging_service'
* Debug mode: on
WARNING: This is a development server. Do not use it in production.
Use a production WSGI server instead.
* Running on http://127.0.0.1:5001
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!

```

facadeService

```

* Serving Flask app 'facade_service'
* Debug mode: on
WARNING: This is a development server. Do not use it in production.
Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!

```

messagesService

```

* Serving Flask app 'messages_service'
* Debug mode: on
WARNING: This is a development server. Do not use it in production.
Use a production WSGI server instead.
* Running on http://127.0.0.1:5005
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!

```

Далі я ввів 10 повідомлень і перевіряв, що вони збереглись.

Переглянув ноди, після чого я відключив loggingService – це призвело до перемішання даних які я вводив до цього. Самі дані не були ушкодженні).