

ДОДАТОК А

Перелік джерел посилання за науковими напрямками керівника та науковців
кафедри програмної інженерії

11. Falatiuk H., Shirokopetleva M., Dudar Z. Investigation of Architecture and Technology Stack for e-Archive System / Conference: 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), October 2019, DOI:10.1109/PICST47496.2019.9061407.

12. Afanasieva I., Golian N., Hnatenko O., Daniil Y., Onyshchenko K. Data exchange model in the internet of things concept. Volume 78, Issue 10, 2019, pp. 869-878, 2019, DOI: 10.1615/TelecomRadEng.v78.i10.30.

14. Подорожний М., Ревенчук І. Дослідження методів автоматизації API Governance: керування інтерфейсами у мікросервісній системі. XXV International Scientific and Practical Conference «Current Trends in the Development of Scientific Research in Today's Conditions». Florence, Italy 29-31.05.2024. P.50-52.

ДОДАТОК Б

Слайди презентації

1



Дослідження методів автоматизації API Governance:
керування інтерфейсами у мікросервісній системі

Подорожний Михайло Павлович., ІПЗм-22-6

Науковий керівник: доц. Ревенчук І.А.

14 червня 2024



2

Дослідження

Актуальність та стан розвитку галузі: Галузь API інтерфейсів для моніторингу працездатності, звітів та маршрутизації запитів – були і будуть єдиною та зв'язуючою складовою між сервером та розробником.

Чітке визначення напрямку дослідження: Основна мета – з'ясувати принцип роботи подібних систем, розробити власну систему для роботи із мікросервісними додатками або адаптувати існуючу систему.

Об'єкт дослідження: API Governance системи для роботи із мікросервісними застосунками



Постановка задачі

- Дослідити методи автоматизації API Governance для керування інтерфейсами у мікросервісній системі.
- Вивчити принципи мікросервісної архітектури та роль API Governance.
- Провести аналіз популярних систем автоматизації API Governance та їх архітектури.
- Розробити порівняльний аналіз систем за критеріями ефективності, безпеки та зручності використання.
- Розробити практичні рекомендації для інтеграції API Governance.
- Створити застосунок для автоматизації управління API.
- Проаналізувати результати.



Методологія

Опис використаних методів дослідження: Дослідження були проведені за допомогою відкритих джерел та бібліотек із відкритим вихідним кодом, розширення було розроблено на платформі .Net, на прикладі власного застосунку із Swagger та його інтерфейсу та розробленого розширення.

Інструментарій та технології, використані в роботі:

- Платформи: .Net
- Мова програмування: C#
- Бібліотеки та API: Swagger, Eureka



Дослідження проблематики

- API Governance є критично важливим для забезпечення стабільності, безпеки та зручності використання інтерфейсів, що використовуються для взаємодії між мікросервісами. Без ефективного управління API виникають ризики неконсистентності, проблеми з безпекою, труднощі у версіонуванні та інтеграції, що може призвести до значних втрат часу та ресурсів, а також до виникнення критичних помилок у системах.
- Дослідження методів автоматизації API Governance у мікросервісній системі є надзвичайно важливим і актуальним напрямком, який відповідає потребам сучасної ІТ-індустрії. Впровадження таких методів дозволить підвищити ефективність розробки, забезпечити високу якість та безпеку програмних продуктів, а також створити основу для подальшого розвитку та масштабування програмних систем.



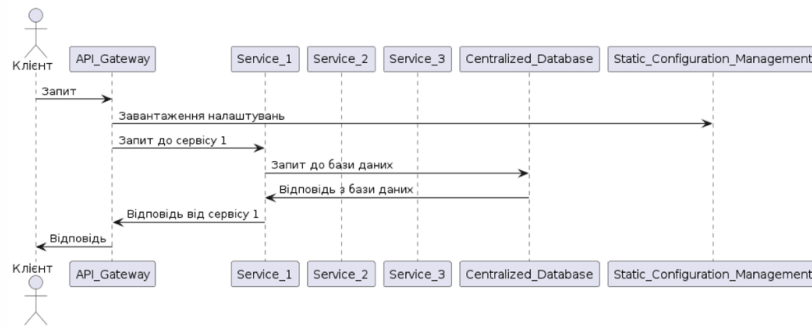
Огляд існуючих рішень

Було розглянуто:

- Swagger
- KONG
- SprigBoot
- Netflix OSS



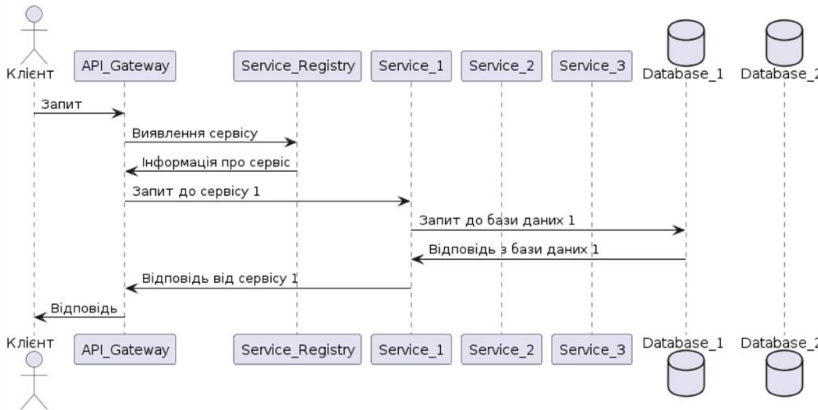
Приклади архітектурних рішень



Система не адаптована під мікросервіси



Приклади архітектурних рішень



Система адаптована під мікросервіси



9. Вирішення проблем за прикладом KONG

| Проблема | Виправлення |
|--|---|
| Централізоване зберігання даних ускладнює масштабування та адаптацію до динамічних змін у мікросервісній архітектурі. Це створює вузьке місце і єдину точку відмови, що може знизити надійність та доступність системи. | Kong підтримує як базу даних, так і режим без бази даних (DB-less). |
| Використання статичних конфігурацій обмежує можливість динамічного виявлення сервісів та адаптації до змін у мікросервісах. Це призводить до складнощів у масштабуванні та оновленні конфігурацій без перезапуску системи. | Kong дозволяє динамічне оновлення конфігурацій через Admin API або використання Kong Ingress Controller для Kubernetes, який автоматично оновлює конфігурації у відповідь на події в кластері, такі як деплоймент нових подів або зміни конфігурацій (Kong Inc.) (Kong Docs). |
| У традиційних системах API Governance відсутність підтримки динамічного виявлення сервісів ускладнює автоматичне додавання нових екземплярів сервісів або видалення неактивних екземплярів, що призводить до додаткових витрат на ручне управління конфігураціями. | Kong використовує Kong Ingress Controller для інтеграції з Kubernetes, що дозволяє автоматично виявляти та реєструвати нові сервіси. |
| Відсутність автоматизованих процесів інтеграції та розгортання (CI/CD) затримує процес оновлення та розгортання нових версій API і сервісів, що збільшує ризик помилок і знижує гнучкість системи. | Kong підтримує автоматизацію CI/CD через інтеграцію з інструментами, такими як Jenkins або GitLab CI, для автоматизованого тестування та розгортання нових конфігурацій. |

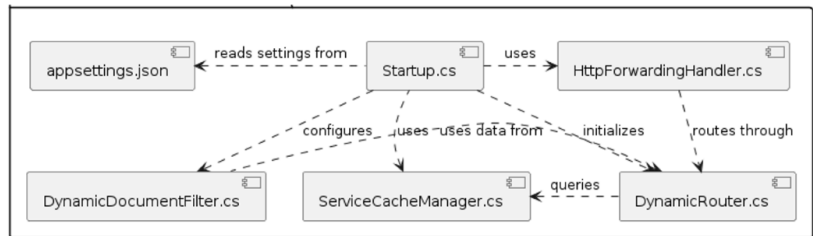


10

Файлова структура розширення

```

Проект DynamicAPIManagement
├── Startup.cs
├── DynamicRouter.cs
├── ServiceCacheManager.cs
├── DynamicDocumentFilter.cs
├── HttpForwardingHandler.cs
└── appsettings.json
    
```



Файлова структура взаємодії

Зображення файлової системи



11

Ключовий функціонал розширення

Клас DynamicRouter виконує декілька ключових функцій в контексті взаємодії між мікросервісами у нашому .NET застосунку, інтегруючи інформацію з сервісу відкриття Eureka для маршрутизації HTTP запитів.

Клас продемонстровано із згорнутими методами

```
public class DynamicRouter
{
    private readonly IDiscoveryClient _discoveryClient;
    private ConcurrentDictionary<string, Uri> _serviceCache;
    private HttpClient _httpClient;

    public DynamicRouter(IDiscoveryClient discoveryClient)
    {
        _discoveryClient = discoveryClient;
        _serviceCache = new ConcurrentDictionary<string, Uri>();
        _httpClient = new HttpClient();
        InitializeServiceCache();
    }

    private void InitializeServiceCache()
    {
    }

    public async Task RouteRequest(HttpContext context)
    {
    }

    private async Task ForwardRequest(HttpContext context, Uri serviceUri, string path)
    {
    }

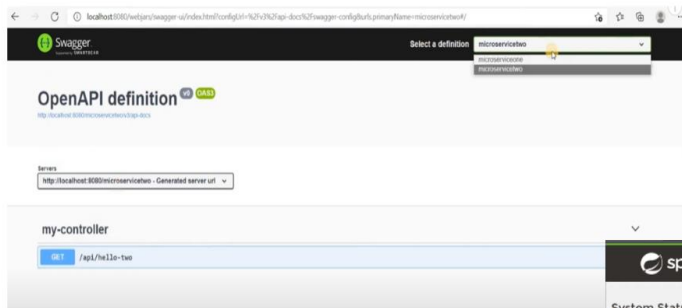
    private HttpRequestMessage CreateHttpRequest(HttpContext context, Uri uri)
    {
    }

    private static async Task CopyResponseAsync(HttpContext context, HttpResponseMessage responseMessage)
    {
    }
}
```



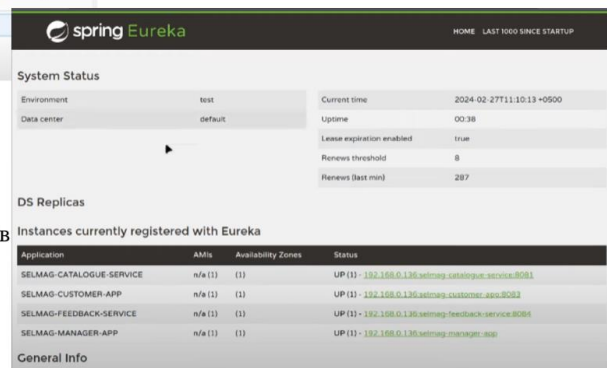
12

Результат тестового запуску



Інтеграція у Swagger

Інтерфейс налаштувань документації до роутингів



Висновки

- Проведено детальний аналіз методів автоматизації API Governance у мікросервісних системах. Розглянуто технології Consul, Eureka та Apache ZooKeeper, їх переваги та недоліки.
- Розроблено практичні рекомендації для інтеграції автоматизованих систем управління API у мікросервісні архітектури.
- Створено власну систему автоматизації управління API, інтегровану з існуючими мікросервісами. Розроблений застосунок успішно протестовано на сумісність з різними технологічними стеками.
- Висновки свідчать про важливість автоматизації управління API для гнучкості, масштабованості та безпеки сучасних програмних рішень. Подальші дослідження спрямовані на розширення функціональності та адаптацію застосунку до специфічних вимог різних галузей.

ДОДАТОК В

Результат проходження на академічний плагіат



Ім'я користувача:
Кардаш Євген Вікторович каф.ПІ

ID перевірки:
1016363297

Дата перевірки:
15.06.2024 15:11:56 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
15.06.2024 15:12:20 EEST

ID користувача:
100013622

Назва документа: 2024_М_ПІ_ІПЗм_22_6_Подорожний_М_П_Скорочений

Кількість сторінок: 35 Кількість слів: 6140 Кількість символів: 51707 Розмір файлу: 834.18 KB ID файлу: 1016168685

1.37% Схожість

Найбільша схожість: 0.62% з Інтернет-джерелом (<http://pygments.org/demo/2691682>)

1.24% Джерела з Інтернету

84

Сторінка 37

0.44% Джерела з Бібліотеки

28

Сторінка 37

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

ДОДАТОК Г

Апробація результатів роботи

Подорожний М., Ревенчук І. Дослідження методів автоматизації API Governance: керування інтерфейсами у мікросервісній системі. XXV International Scientific and Practical Conference «Current Trends in the Development of Scientific Research in Today's Conditions». Florance, Italy 29-31.05.2024. P.50-52. (<https://isu-conference.com/arkhiv/current-trends-in-the-development-of-scientific-research-in-todays-conditions/>)

ДОДАТОК Д

Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008:2015

Експертний висновок результатів перевірки кваліфікаційної роботи

студент
(посада)

програмної інженерії
(кафедра)

ІПЗМ-22-6
(група)

Подорожний Михайло Павлович

(прізвище, ім'я, по батькові)

Зауваження

| Пункт ДСТУ 3008-2015 | Зміст пункту | Сторінка кваліфікаційної роботи |
|----------------------|---|---------------------------------|
| 1 | 2 | 3 |
| | 7.1 Загальні положення | |
| | 7.3 Нумерація сторінок звіту | |
| | 7.4 Нумерація розділів, підрозділів, пунктів, підпунктів | |
| | 7.5 Рисунки | |
| | 7.6 Таблиці | |
| | 7.7 Переліки | |
| | 7.8 Примітки | |
| | 7.9 Виноски | |
| | 7.10 Формули та рівняння | |
| | 7.11 Посилання | |
| | 7.13 Список авторів | |
| | 7.14 Скорочення та умовні позначки | |
| | 7.15 Додатки | |

зауважень немає

Експерт

_____ (підпис)

Олена ОЛІЙНИК

(прізвище, ініціали)

18.06.2024