

РОЗРОБКА МІКРОСЕРВІСНОЇ СИСТЕМИ СКЛАДАННЯ РОЗКЛАДУ ВНТУ

Вінницький національний технічний університет

Анотація

Запропоновано метод створення WEB-додатку для створення, перегляду, редагування та видалення розкладу для університету на основі мікросервісної архітектури. Така архітектура забезпечить незалежність роботи WEB-додатків один від одного.

Ключові слова: WEB-додаток, мікросервіси, C#, JS, REST API.

Abstract

The method of creating a WEB application for creating, viewing, editing and deleting a schedule for a university based on microservice architecture is proposed. Such an architecture will ensure the independence of WEB applications from each other.

Keywords: WEB application, microservices, C#, JS, REST API.

Вступ

Довгий час найпопулярнішою архітектурою для будь-якого додатку був моноліт. Моноліт складається з кількох шарів – інтерфейс користувача, обробка та збереження інформації. Недоліками такої архітектури є розмір, велика кількість зав'язків між модулями що ускладнює модифікацію окремого модуля і у випадку якщо один з модулів дає збій, то припиняє роботу повністю весь додаток.

На даний момент набирає популярність мікросервісна архітектура. Найбільший плюс такої архітектури – це незалежність модулів. Тому, для декількох модулів може використовуватись декілька мов програмування. Сервіси за своїми розмірами невеликі, що дозволяє підтримувати один або декілька сервісів одному розробнику.

Метою роботи є розробка додатку на основі мікросервісної архітектури, за допомогою якого буде можливість скласти розклад для ВНТУ.

Результати дослідження

Мікросервісна архітектура дозволяє створювати великі, за своїм функціоналом, додатки. Але у такої архітектури є свої переваги та недоліки. Серед переваг виділяють: незалежність сервісів один від одного, один сервіс вирішує одну чітку задачу, висока відмовостійкість наприклад якщо сервіс авторизації відмовить це не зупинить роботу усього додатку, вільний вибір технологій для кожного із сервісів що дозволяє вибрати найбільш зручний інструмент для вирішення конкретної задачі, відносно легкий процес розгортання оскільки кожний сервіс працює як окремий додаток то й розгортання невеликого додатку займає менше часу, завдяки невеликому розміру сервіса є можливість одному розробнику підтримувати декілька сервісів. Серед недоліків можна виділити невелику швидкість розробки додатку оскільки потрібно спроектувати систему та складність підтримки цілої системи. Приклад такої архітектури зображено на рис. 1. Для зв'язку між сервісами часто використовують черги повідомлень, наприклад Azure Service Bus, RabbitMQ, Kafka. Це потрібно на випадок коли один із сервісів дає збій і щоб не втратити ці повідомлення вони зберігаються в цих чергах.

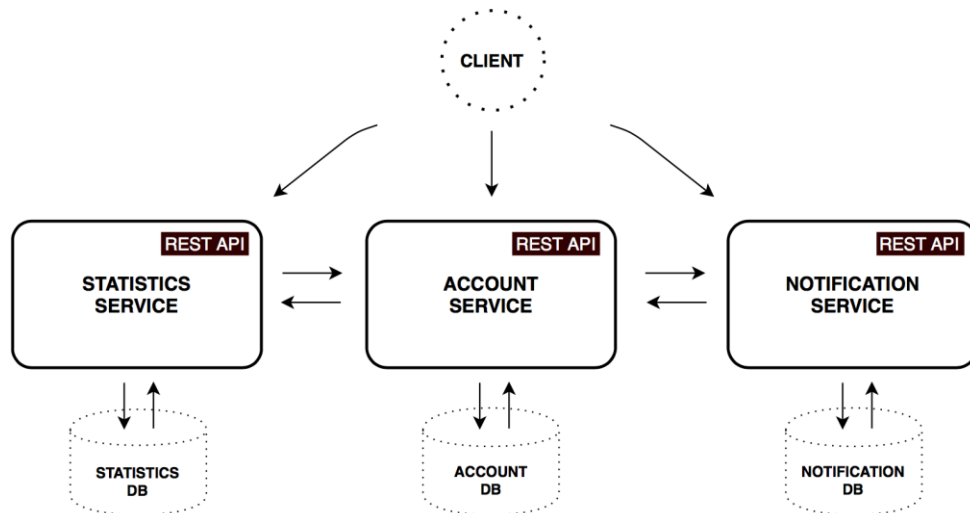


Рис. 1 – приклад використання мікросервісної архітектури

Висновки

За допомогою висвітленої архітектури є можливість розробити відмовостійкий додаток для створення, перегляду та редагування навчального розкладу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. 4. Фаулер М. Шаблини корпоративних додатків. Брюс Е. – СПб.: Издательский дом "Питер", 2016, 544 с. ISBN: 978-5-8459-1611-2.
2. [Електронний ресурс]: Режим доступу: <https://microservices.io/>. Дата звернення: Січень 24, 2020.

Мамашвілі Леонід Олегович — студент групи ІСІ-16Б, факультет комп'ютерних систем та автоматики, Вінницький національний технічний університет, Вінниця, e-mail: mamashvili.leonid@gmail.com.

Довгалець Сергій Михайлович — к.т.н, доцент, професор кафедри автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, м. Вінниця

Leonid Mamashvili — student of Automation and Intelligent Information Technology department, Vinnytsia National Technical University, Vinnytsia, email: mamashvili.leonid@gmail.com.

Dovgalets Sergey M. Professor of Automation and Intelligent Information Technology department, Vinnytsia National Technical University, Vinnytsia