

Створення бібліотек на основі Spring Boot для уніфікації розробки інформаційної системи з мікросервісною архітектурою

Виконав Скрипнік А.О., студент ІПЗ МП-2
Науковий керівник – Франчук Олег Васильович



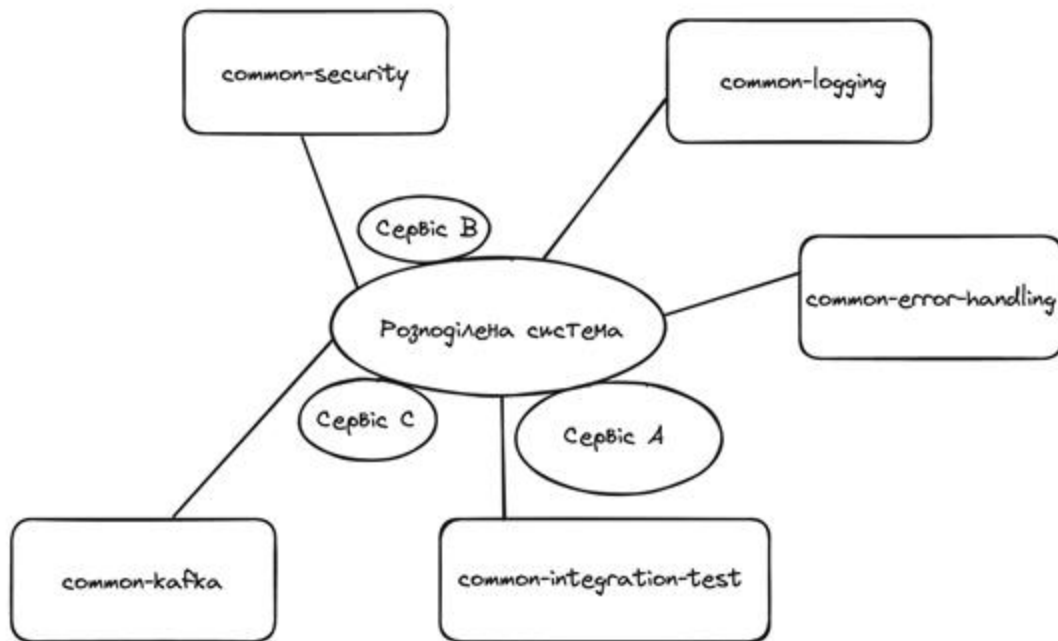
Чому гальмується розробка

- Складність системи
- Крос-командні залежності
- Дублювання коду
- Неоднорідність API
- Неузгоджена архітектура

Постановка задачі

1. Проведення аналізу проблематики, що виникає при розробці системи із мікросервісною архітектурою.
2. Розробити концепцію рішення та реалізувати бібліотеки для уніфікації компонентів системи.
3. Створити демонстраційний проект із використанням розроблених рішень.
4. Оцінити запропоноване рішення та зробити висновки.

Компоненти, що можна уніфікувати



Використані технології



API-first підхід та генерація коду

Account Service API 1.0.0 OAS3

API documentation for the Account Service

Servers

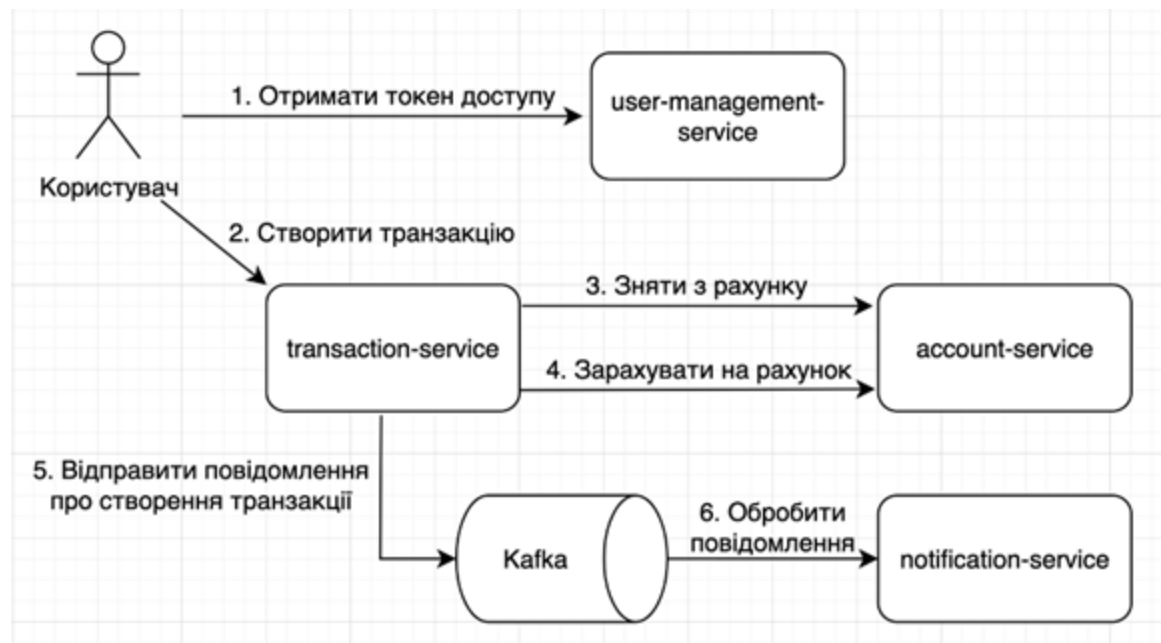
/api/v1

Account Management API for managing accounts

- POST** /accounts Create a new account
- GET** /accounts/{accountId} Get account by ID
- POST** /accounts/{accountId}/deduct Deduct money from account balance
- POST** /accounts/{accountId}/add Add money to account balance

```
<plugin>
  <groupId>org.openapitools</groupId>
  <artifactId>openapi-generator-maven-plugin</artifactId>
  <version>6.3.0</version>
  <executions>
    <execution>
      <phase>generate-sources</phase>
      <goals>
        <goal>generate</goal>
      </goals>
      <configuration>
        <inputSpec>${project.basedir}/openapi-spec/AccountServiceV1.yaml</inputSpec>
        <generatorName>spring</generatorName>
        <configOptions>
          <useJakartaEe>true</useJakartaEe>
          <useSpringBoot3>true</useSpringBoot3>
          <documentationProvider>none</documentationProvider>
          <skipDefaultInterface>true</skipDefaultInterface>
          <dateLibrary>java8</dateLibrary>
          <modelPackage>com.skrypnik.accountservice.v1.model</modelPackage>
          <apiPackage>com.skrypnik.accountservice.api.controller.v1</apiPackage>
          <invokerPackage>com.skrypnik.accountservice</invokerPackage>
          <interfaceOnly>true</interfaceOnly>
        </configOptions>
      </configuration>
    </execution>
  </executions>
</plugin>
```

Демонстраційна розподілена система



Результат

1. Проаналізовано й досліджено проблематику, що виникає при розробці системи з мікросервісною архітектурою.
2. Були визначені компоненти, реалізація яких має бути єдиною та уніфікованою в усіх частинах розподіленої системи задля ефективної роботи над кодом, його підтримкою і розширенням.
3. Розроблено 5 бібліотек (Spring Boot starters): `common-logging`, `common-error-handling`, `common-kafka`, `common-integration-test`, `common-security`.
4. Побудовано демонстраційну розподілену систему із використанням API-first підходу та вищезазначених бібліотек.

Висновок

Підсумовуючи, використання розроблених бібліотек в демонстраційному проекті показало, що для прискорення, паралелізації й уніфікації розробки розподіленої системи слід розглянути запропоновані вище підходи

Розроблені у форматі Spring Boot старетрів бібліотеки дозволяють зробити конфігураційний файл сервісу `application.yaml` єдиним місцем налаштування асинхронного спілкування, обробки помилок та безпеки, що також робить розробку простішою, швидшою та зменшує дублювання коду в системі.