

# Тестовое задание – Набор микросервисов

В рамках задания необходимо *разработать три взаимодействующих между собой микросервиса* MS1, MS2, MS3.

## Основа взаимодействия

*Микросервисы* взаимодействуют между собой путем последовательной передачи сообщения `message` следующего формата:

```
{
  "session_id": integer,
  "service1_timestamp": datetime,
  "service2_timestamp": datetime,
  "service3_timestamp": datetime,
  "end_timestamp": datetime
}
```

где `session_id` - идентификатор сеанса взаимодействия, задаваемый MS1, оставшиеся поля – отметки времени прохождения сообщения через оставшиеся сервисы включая MS1 в конце.

## Инициализация – MS1

1. MS1 устанавливает соединение с базой данных
2. MS1 устанавливает соединение с MS2 по протоколу WebSocket
3. MS1 ожидает запрос `HTTP GET /start`
4. MS1 ожидает запрос `HTTP GET /stop` (обработка этого запроса должна привести к прерыванию цикла)

## Цикл взаимодействия

1. После получения запроса `HTTP GET /start` MS1 записывает это сообщение в лог-файл и также выводит его в консоль
2. MS1 создает `message` с заполненными нужными данными полями `session_id` и `service1_timestamp`
3. MS1 отправляет `message` в MS2 через WebSocket
4. MS1 ожидает `HTTP POST /store` с `message`
5. MS2 принимает `message` от MS1, записывает в поле сообщения `service2_timestamp` текущее время и отправляет сообщение в MS3 через топик брокера Kafka
6. MS3 забирает `message` из топика, записывает в поле `service3_timestamp` текущее время и отправляет `message` в MS1 `HTTP POST /store`

*Цикл взаимодействия повторяется* в течение заданного интервала взаимодействия. Длительность интервала взаимодействия задается в секундах *параметром в конфигурационном файле*.

*Завершение взаимодействия индицировать на консоль* с выводом следующих параметров:

- общее время взаимодействия;
- количество сообщений, сгенерированных во время взаимодействия.

## Требования

Для реализации использовать только *стабильные LTS-версии* библиотек и компонентов.

*Для целей трассировки* предусмотреть в микросервисах отправку span-сообщений в систему Jaeger для каждого входящего сообщения.

*В качестве БД использовать СУБД MariaDB*. После остановки контейнеров с микросервисами и окружением база данных должна быть доступна для просмотра средствами СУБД.

*Запуск* микросервисов и окружения необходимо производить *только в контейнерах*.

Код должен быть минимально *покрыт тестами* в части логики обработки сообщений. Измерение покрытия не требуется.

## Расширенные требования – бонус-поинты к оценке

- Опубликовать образы на Docker Hub и предоставить репо с docker-compose и конфигурацией для запуска микросервисов
- Обеспечить 86% покрытие тестами кода микросервисов
- Сгенерировать документацию