

Домашнее задание 06: Проектирование Event-Driven архитектуры

****Цель работы:**** Получить навыки проектирования событийно-ориентированной архитектуры, работы с брокерами сообщений и применения паттерна CQRS.

****Время выполнения:**** 3–4 часа

Задание

Для своего варианта задания выполните следующие задачи:

1. Анализ событий в системе

- Изучите выбранный вариант задания
- Определите события (events), которые происходят в вашей системе
- Определите команды (commands), которые инициируют события
- Определите, какие сервисы должны быть уведомлены о каждом событии

2. Проектирование Event-Driven архитектуры

- Определите компоненты системы, которые будут производителями событий (event producers)
 - Определите компоненты, которые будут потребителями событий (event consumers)
 - Определите типы событий и их структуру (payload)
 - Опишите поток событий в системе

3. Проектирование взаимодействия через брокер сообщений

- Выберите RabbitMQ или Apache Kafka
- Определите формат сообщений для событий
- Опишите гарантии доставки сообщений (at-least-once, exactly-once)

4. Применение паттерна CQRS

- Определите, можно ли применить CQRS в вашей системе
- Если да, разделите операции на команды (write) и запросы (read)
- Опишите, как события синхронизируют read и write модели

5. Реализация простого Event-Driven сервиса**

- Настройте RabbitMQ/Kafka (использовать Docker)
- Реализуйте простой producer, который публикует события
- Реализуйте простой consumer, который обрабатывает события
- Протестируйте взаимодействие

6. Документация событий

- Создайте каталог событий (event catalog) с описанием всех событий
- Для каждого события укажите:
 - Название события
 - Структуру payload
 - Производителя события
 - Потребителей события
 - Гарантии доставки

Результат

Домашнее задание по курсу «Архитектура программных систем»

Результат должен быть оформлен в виде следующих файлов, размещенных в вашем GitHub репозитории:

- `event_driven_design.md` – описание Event-Driven архитектуры
- `event_catalog.md` – каталог событий с описанием
- Исходный код с реализацией producer/consumer
- `docker-compose.yml` – для запуска RabbitMQ/Kafka
- `README.md` – описание проекта и инструкции по запуску

Критерии оценки:

- Корректность определения событий и команд
- Качество проектирования Event-Driven архитектуры
- Правильность выбора типов exchange и routing
- Применение паттерна CQRS (если применимо)
- Качество каталога событий
- Работоспособность реализации (если реализовано)