

Task340. Брокеры сообщений (на П)

Дано:

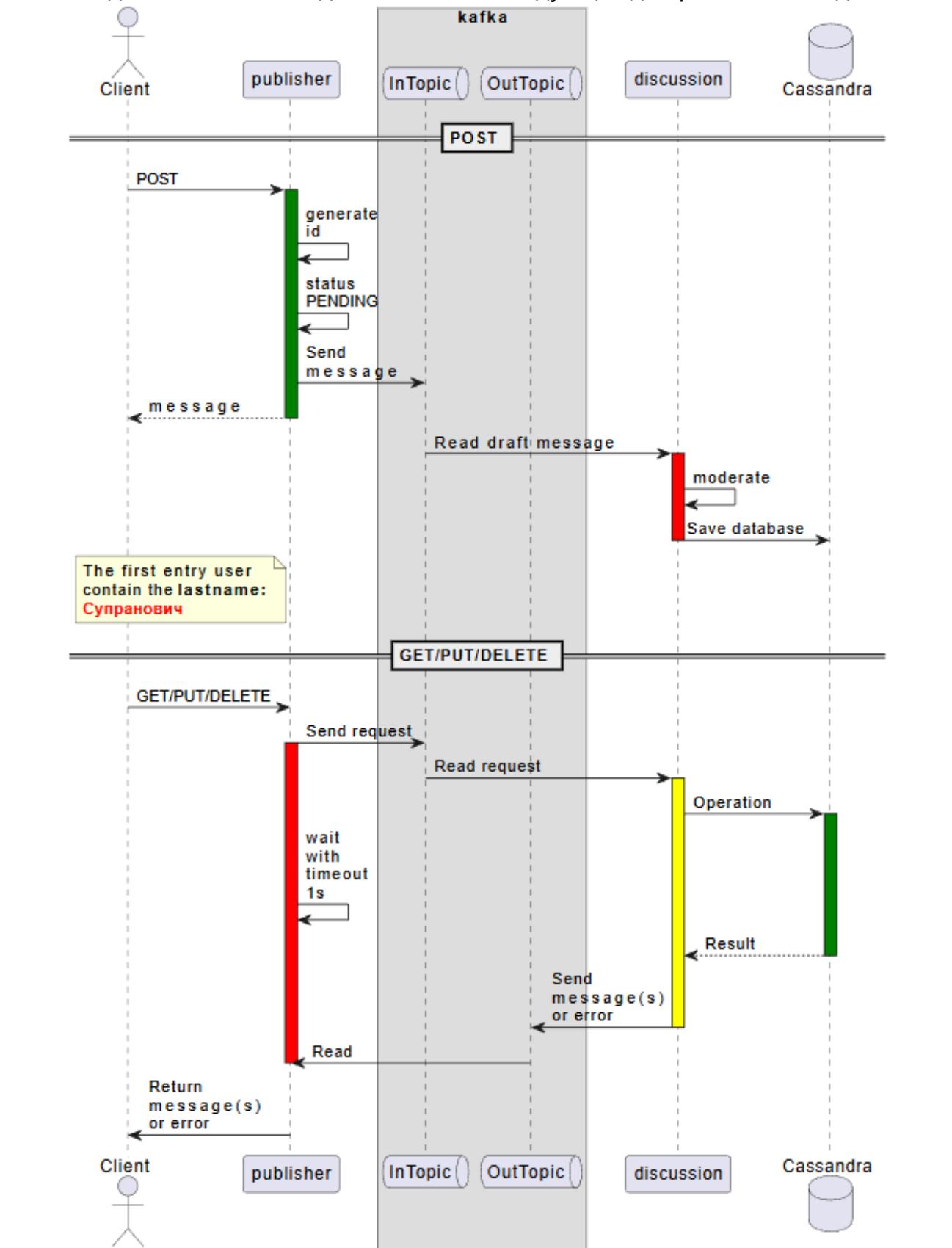
- Разрабатываемая система обрабатывает сущности **User**, **Tweet**, **Label** и **Message**, которые логически связаны отношениями (см. предыдущие задачи)
 - один-ко-многим (**User** и **Tweet**, **Tweet** и **Message**)
 - многие-ко-многим (**Tweet**, **Label**).
- Существует и работает REST-передача между модулями **publisher** и **discussion** сущности **Message**

Задание

Необходимо в ранее разработанном распределенном приложении реализовать транспорт между модулями **publisher** и **discussion** для сущности **Message** через топики Kafka.

р
а
с
х
е
К
а
ф
к
а

Основой для выполнения задания является следующая диаграмма взаимодействия



Технические требования

- Используйте префикс **/api/v1.0/** для контроллеров REST и их методов,
- Используйте адрес и порт **localhost:24110** для самого приложения (это модуль **publisher**).
- Не отключайте REST **localhost:24130** для модуля **discussion**.
- Используйте обязательный префикс **tbl_** для таблицы в базе(ах) данных

Рекомендации

Подготовительные шаги:

- Изучить документацию по Apache Kafka.
- Установить и настроить Apache Kafka локально, используя docker.
- Создать поле state в сущности **Message** с тремя состояниями PENDING APPROVE DELCINE
- Создать топики **InTopic** и **OutTopic** для обмена сообщениями или автоматизировать создание топиков при их отсутствии (Spring это делает автоматически)

Интеграция Kafka с publisher:

- Создать Kafka Producer в модуле **publisher** для отправки сообщений в топик **InTopic**.
- Настроить отправку **Message** так, чтобы в топике **InTopic** сообщения принадлежащие одной **Tweet** всегда оказывались в одной **partition** (независимо от числа **partition**).
- Создать Kafka Consumer **publisher** для чтения сообщений из **OutTopic**.

Интеграция Kafka с discussion:

- Создать Kafka Consumer в модуле **discussion** для чтения запросов из топика **InTopic**:
 - модуль должен обновлять статус после автоматической модерации на APPROVE или DELCINE
 - алгоритм модерации может быть любым, например на основе стоп-слов
- Создать Kafka Producer в модуле **discussion** для отправки ответов в топик **OutTopic**.

Тестирование и отладка:

- Протестировать передачу сообщений через Kafka.
- Проверить регрессионно работоспособность системы после интеграции Kafka.

Масштабирование и оптимизация:

- Оптимизировать настройки Kafka для повышения производительности.
- Рассмотреть возможность масштабирования Kafka для обеспечения отказоустойчивости.
- Определить минимальное число брокеров, партиций и RF для обеспечения отказоустойчивости кластера Kafka.