Тестовое задание

Описание

Система состоит из двух служб: служба пользователей (user-service) и служба задач (task-service). Каждая из служб обеспечивает набор базовых операций для работы с одноименными сущностями: пользователь (user) и задача (task). Основная бизнес логика строится на формировании в системе задач и их исполнении пользователями, которым они назначены на исполнение.

Требования

Общие требования

Обе службы для обращения к ним предоставляют REST API.

Любое обращение по API к одной из служб не должно вынуждать её обращаться к другой службе за информацией, то есть если одна служба оказалась недоступной, то это не должно повлиять на работоспособность другой. Но любая служба может взаимодействовать с другой службой если это не блокирует текущий REST API вызов.

Вся система должна работать в экосистеме Docker под управлением ОС Linux (желательно Ubuntu/Debian).

В исходниках проекта должна быть документация (например README файл) по развертыванию системы на целевой инфраструктуре.

REST API Reference (Swagger) для служб user-service и task-service.

База данных: PostgreSQL

Система управления миграциями БД: Liquibase или Flyway

Брокер сообщений: Apache Kafka

Требования к user-service

Является основным источником информации о пользователях.

Свойства пользователя:

- идентификатор
- логин (username) - уникальное значение
- фамилия
- имя

Поддерживает следующие операции:

- получить всех пользователей,
- получить пользователя по идентификатору,
- получить пользователя по его логину (username)
- редактировать пользователя

Операции записи:

- идентификация пользователя при редактирования осуществляется через его идентификатор
- при редактировании указываются свойства: логин, фамилия, имя
- при редактировании запрос может быть отклонен если новый логин не является старым и при этом он не уникален

Требования к task-service

Является основным источником информации о задачах.

При запросе задач(и) должна возвращаться информация о владельце и исполнителе (не просто их идентификатор, а их свойства)

Свойства задачи:

- идентификатор
- наименование
- владелец - пользователь
- исполнитель - пользователь
- статус - значение из множества (новое, назначено, в работе, выполнено)
- дата и время создания
- дата и время последнего изменения

Поддерживает следующие операции:

- получить все задачи
- получить задачу по идентификатору
- получить список моих задач (те у которых я владелец)
- получить список назначенных мне на исполнение задач
- создать задачу
- редактировать задачу
- изменить статус
- удалить задачу

Операции чтения:

- при получении задач(и) также возвращается информация о владельце и исполнителе в виде отдельной структуры, содержащей в себе все свойства пользователя, но все свойства кроме идентификатора можно агрегировать в одно простой конкатенацией

Операции записи:

- любой пользователь может создавать задачи
- владелецу задачи доступны все действия (просмотр, редактирование, изменение статуса, удаление)
- исполнителю задачи доступны просмотр и изменение статуса с ”новое” на “в работе” и с “в работе” на “выполнено”
- пользователь, не имеющий никакой связи с задачей не может выполнять с ней никаких действий в том числе и просматривать (при получении списка всех задач он должен получить только те задачи в которых является либо владельцем, либо исполнителем)