

Результат выполнения задания должен быть выложен соискателем в публичный репозиторий github и помимо кода проекта содержать подробные инструкции по сборке и запуску. Ссылку на проект необходимо направить на почту: hr@bewise.ai в тексте письма указать ваше ФИО и ссылку на ваше резюме

Задание

Разработайте сервис для обработки заявок пользователей. Сервис должен:

1. Принимать заявки через REST API (FastAPI).
2. Обрабатывать и записывать заявки в PostgreSQL.
3. Публиковать информацию о новых заявках в Kafka.
4. Обеспечивать эндпоинт для получения списка заявок с фильтрацией и пагинацией.
5. Включать Docker-файл для развертывания приложения.

Детали реализации

1. REST API:

- Создайте эндпоинт **POST /applications** для создания новой заявки. Заявка содержит следующие поля:
 - **id** (генерируется автоматически)
 - **user_name** (имя пользователя)
 - **description** (описание заявки)
 - **created_at** (дата и время создания, устанавливается автоматически)
- Создайте эндпоинт **GET /applications** для получения списка заявок:
 - Поддержка фильтрации по имени пользователя (**user_name**).
 - Поддержка пагинации (параметры **page** и **size**).

2. PostgreSQL:

- Спроектируйте таблицу для хранения заявок.
- Используйте SQLAlchemy для работы с базой данных.

3. Kafka:

- Настройте публикацию данных о новых заявках в топик Kafka.

- В сообщении должно содержаться:
 - `id` заявки
 - `user_name`
 - `description`
 - `created_at`
 - 4. **Асинхронность:**
 - Убедитесь, что все взаимодействия с Kafka и PostgreSQL реализованы асинхронно.
 - 5. **Docker:**
 - Подготовьте `Dockerfile` и `docker-compose.yml` для локального запуска:
 - Приложение (FastAPI)
 - PostgreSQL
 - Kafka
 - 6. **Документация:**
 - Опишите инструкцию по запуску проекта.
 - Добавьте пример запроса и ответа для эндпоинтов.
-

Критерии оценки

1. **Работающий код:**
 - Корректная работа API и публикации сообщений в Kafka.
 2. **Качество кода:**
 - Использование стандартов Python (PEP8).
 - Читаемость и структура кода.
 3. **Проектирование БД:**
 - Оптимальное решение для хранения заявок.
 4. **Использование технологий:**
 - Работа с FastAPI, PostgreSQL, Kafka и Docker.
 5. **Документация:**
 - Понятность инструкций по запуску и тестированию.
-

Дополнительно

Реализованные дополнительные функции будут преимуществом:

- Валидация входящих данных с использованием Pydantic.

- Логирование ошибок и событий в приложении.
- Подготовка unit-тестов для ключевых компонентов.

Срок выполнения:

Предоставьте решение в течение **7 дней**.