

## # Тестовое задание: Разработка микросервиса для подготовки дайджестов контента

### ## Цель

Разработайте микросервис, который будет формировать дайджесты контента для пользователей на основе их подписок. Дайджест представляет собой выборку постов из различных источников, на которые подписан пользователь. Вы можете выбрать любую технологию и форматы обмена данными.

### ## Требования к микросервису

1. Получение запроса на формирование дайджеста: Микросервис должен уметь принимать запросы от основного приложения на формирование дайджеста для пользователя, идентифицируемого по уникальному ID.
2. Определение подписок пользователя: После получения запроса, микросервис должен определить источники, на которые подписан пользователь, используя информацию о подписках пользователя.
3. Сбор постов из подписок: Зная подписки пользователя, микросервис должен собирать посты из этих источников. Подумайте о нём как о "сканере" подписок пользователя в поисках нового контента.
4. Фильтрация постов: Из собранных постов отфильтруйте те, которые не соответствуют интересам пользователя или недостаточно популярны. Микросервис должен использовать определенные критерии для фильтрации.
5. Создание дайджеста: После фильтрации, оставшиеся посты упаковываются в дайджест. Дайджест - это совокупность постов, отобранных для пользователя.
6. Отправка дайджеста: Сформированный дайджест возвращается в главное приложение, которое предоставит его пользователю.

### ## Структура данных

- Модель User: Хранит данные о пользователе, включая ID и имя.
- Модель Subscription: Содержит информацию о подписках пользователя, включая ID подписки, название источника и ID пользователя.
- Модель Post: Включает информацию о постах из подписок пользователя, включая ID поста, его содержание и популярность.
- Модель Digest: Содержит информацию о сформированном дайджесте, включая ID дайджеста, список постов и ID пользователя.

### ## Дополнительные требования (по желанию)

- Использование Docker для создания контейнеризованного микросервиса.
- Использование SQLAlchemy для взаимодействия с базой данных.

- Использование RabbitMQ для асинхронной обработки запросов на формирование дайджеста.
- Написание автоматических тестов для проверки функциональности микросервиса.

### ## Критерии оценки

- Работоспособность микросервиса.
- Качество кода (организованность, чистота, следование стандартам).
- Соответствие функциональным требованиям.
- Реализация дополнительных требований (если выбраны).

Удачи в выполнении задания!

P.S. Все непонятные моменты интерпретируйте по своему усмотрению.

P.P.S. Спросили про сроки, это задание реально сделать за 2 дня. По факту у нас вышло около 80 строчек без моделей в бд, но, опять таки. Задание на джуниор разработчика, поэтому вы можете выкручиваться или сделать упрощенную версию от этой. Все обсуждаемо, можно даже обсудить сроки, если они будут аргументированы, но давайте в любом случае уважать время друг друга!

P.P.P.S. Вопросы задавать можно.

Пример готового дайджеста в приложении

- Суд в Флориде приказал «Макдональдсу» и компании Urchurh Foods выплатить 800 тысяч долларов семье девочки, получившей ожог от горячего куриного наггетса в 2019 году. Семья требовала 15 миллионов долларов компенсации. #1
- Банк "Тинькофф" подтвердил попадание под санкции США и Канады, однако заверил, что продукты и услуги банка будут функционировать без изменений для клиентов. Банк также проводит анализ влияния санкций на свою деятельность. #2
- Евросоюз ввел санкции «за нарушения прав человека» — «против тех, кто использовал технологию распознавания лиц для массовых произвольных арестов в России, а также против тех, кто выносил политически мотивированные судебные решения в отношении Алексея Навального и Владимира Кара-Мурзы». #3