

ТЗ №1 — Backend-разработчик

Задание 1

Цель задания

Создание API-сервиса, предназначенного для простого управления задачами пользователя (Task Management Service).

Функциональные требования:

1. **Создание, удаление и получение задач пользователя**
 - a. Каждая задача должна содержать:
 - i. Название задачи (Title)
 - ii. Описание задачи (Description)
 - iii. Статус задачи (новая, в работе, выполненная) (*Status*)
 - iv. Дату создания и дату изменения (*CreatedAt*, *UpdatedAt*)
2. **Синхронное взаимодействие между сервисами**
 - a. Реализовать отдельный сервис-слушатель, который предоставляет методы получения данных о событиях изменения задачи и логирует их.
3. **(Опционально) Асинхронное взаимодействие между сервисами через очередь сообщений Kafka или RabbitMQ**
 - a. При создании, изменении или удалении задачи отправлять события (producer).
 - b. Создать отдельный сервис-слушатель очереди (consumer), который принимает события об изменении задачи и логирует их.

Технические требования и стек:

- .NET 8 / ASP.NET Core
- Использование ORM системы для работы с БД. Предпочтительно - EF Core
- БД: желательно использовать PostgreSQL или Microsoft SQL Server
- Kafka или RabbitMQ
- Observability: логирование (вывод на консоль или в текстовый файл)

Требования к реализации кода:

- Чистая и понятная структура кода, следование принципам SOLID.
- Наличие файла README с инструкцией по запуску проекта.

Дополнительные требования (опционально, плюсом будет считаться):

- Dockerfile и возможность запустить проект через Docker compose.
- Покрытие кода unit-тестами.
- Наличие XML-документация к классам, методам и свойствам.
- Добавление трассировки (OpenTelemetry)

Задание 2

Цель задания:

Необходимо написать табличную функцию SQL, которая будет возвращать по ClientId и интервалу дат (тип Date) поденные суммы платежей. Если за указанный день не было платежей, то функция должна возвращать 0. Интервалы дат могут охватывать несколько лет.

Структура данных:

Есть таблица платежей клиентов ClientPayments следующего вида:

ClientPayments

```
(  
    Id bigint, -- первичный ключ таблицы  
    ClientId bigint, -- Id клиента  
    Dt datetime2(0), -- дата платежа  
    Amount money --сумма платежа  
)
```

Пример входных данных:

Таблица client.Payments

Id	ClientId	Dt	Amount
1	1	2022-01-03 17:24:00	100
2	1	2022-01-05 17:24:14	200
3	1	2022-01-05 18:23:34	250
4	1	2022-01-07 10:12:38	50
5	2	2022-01-05 17:24:14	278
6	2	2022-01-10 12:39:29	300

Примеры работы:

Результат работы функции №1:

Входные данные:

- ClientId = 1
- Sd = 2022-01-02
- Ed = 2022-01-07

Результат функции:

Dt	Сумма
2022-01-02	0
2022-01-03	100
2022-01-04	0
2022-01-05	450
2022-01-06	0
2022-01-07	50

Результат работы функции №2:

Входные данные:

- ClientId = 2
- Sd = 2022-01-04
- Ed = 2022-01-11

Результат функции:

Dt	Сумма
2022-01-04	0
2022-01-05	278
2022-01-06	0
2022-01-07	0
2022-01-08	0
2022-01-09	0
2022-01-10	300
2022-01-11	0