

# Тестовое задание №1 Проект “Task Tracker”

## 1. Структуру данных (основные сущности и поля).

Основными сущностями будет модель пользователя и задачи,

Модель пользователя будет иметь такие поля как:

- email - тип string - уникальный email пользователя
- password - тип string - хешированный пароль (bcrypt)
- created\_at - тип datetime - дата регистрации
- tasks - связь с задачами, таким образом что при удалении пользователя удаляются все его задачи

Модель задач будет иметь такие поля:

- title- тип string - Заголовок задачи
  - description - тип string - Описание задачи
  - user\_id - тип integer - для связи с пользователем
  - due\_date - тип datetime - дедлайн задачи
- Дальше, что я бы добавил в дальнейшем
- completed - тип bool - выполнена задача или нет
  - priority - тип enum - приоритет задачи
  - поле для уведомления через Celery

## 2. Общая структура проекта

В качестве бекенда использую:

- FastAPI быстрый фреймворк, за счет асинхронности
- В качестве базы PostgreSQL
- Для кеширования, хранения токенов, будущем для работы с Celery использовать Redis
- Для хеширования паролей bcrypt
- Для аутентификации JWT(pyjwt)
- Валидация данных pydantic
- Alembic для работы с PostgreSQL

В качестве фронтенда будь я фронтендером использовал бы:

- React + Axios + React Query/TanStack Query - самый популярный фреймворк с библиотекой для http запросов

## 3. Короткое описание логики авторизации и работы с задачами.

1. Регистрация пользователя по email и. password, пароль хранится в виде хэша

2. Вход в систему по той почте и паролю, проверка почты и пароля, создание токенов, access token возвращается пользователю, refresh в cookie и в redis с TTL, для дальнейшего их сравнения.
3. Создание задачи отправляется запрос заголовком в котором указан access token:
  - Authorization: Bearer access\_tokenдалее будет проверка истек ли он, либо находится ли в черном списке который находится в redis и добавляется туда при выходе из системы. Если токен истек можно обновить с помощью refresh токена. Тоже самое происходит при просмотре всех задач пользователя, своего профиля
4. Выход из системы оба токена добавляются в черный список в redis с TTL