

## Тестовое задание python developer

**Задание:** Необходимо реализовать Rest-like API, через которое можно создавать и получать дневники с записями.

### Описание моделей:

Модель Diary:

- 1.1 Поля
  1. title: string
    - 2.1 Название дневника
    - 2.2 Обязательное поле
  2. expiration
    - 3.1 Дата, после которой можно удалить дневник
    - 3.2 Может быть назначена только у private дневников
  3. kind: enum
    - 4.1 Тип дневника, значения: public, private
    - 4.2 Обязательное поле
4. Связан с user one-to-many. У юзера много дневников, у дневника один юзер Модель

Note:

1. Поля
2. text: text
3. связан с Diary one-to-many. У дневника много страниц, у страницы один дневник

Модель User

1. Поля
2. login: string
  - 2.1 Логин пользователя в виде email Обязательное поле
3. password
  - 3.1 Обязательное поле

Описанные выше поля не являются обязательными, если нужны дополнительные поля либо нужно поменять эти, то это можно сделать.

### Описание функционала

1. Реализовать CRUD для моделей Diary, Note
  - 1.1 все пользователи могут смотреть дневники и страницы всех пользователей
  - 1.2 пользователь может управлять только своими дневниками и страницами
2. Для работы с api Diary и Note необходимо авторизоваться, используя модель User

3. Формат передачи данных - json
4. Фоново удалять старые дневники каждые 10 минут
5. Интерфейс реализовывать не нужно, только API
6. В роутах с выдачей всех объектов в формате list настроить пагинацию, использовать любое готовое решение
7. В роутах с выдачей всех объектов в формате list добавить возможность делать выборки по полям
8. Реализовать автогенерацию API в swagger
9. Настроить запуск всего приложения в docker-compose
10. Покрыть юнит тестами реализованный функционал

### **Технологии и библиотеки**

django  
celery  
redis  
postgres  
docker  
pytest

**Итоговое решение выложить на github с описанным readme.**