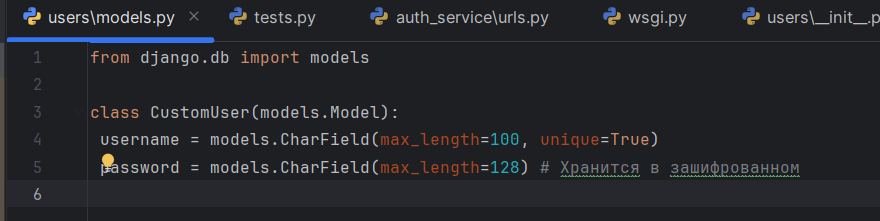


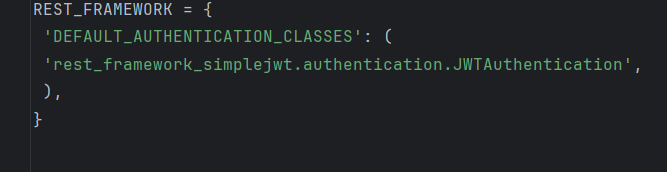
python manage.py runserver\_plus --cert-file cert.pem --key-file key.pem 8000

1. Создали два сервиса AuthService и DataService

2. Настройка Auth Service 2.1 Модель пользователя:



JWT-аутентификация:



. Защита от атак

Rate Limiting:

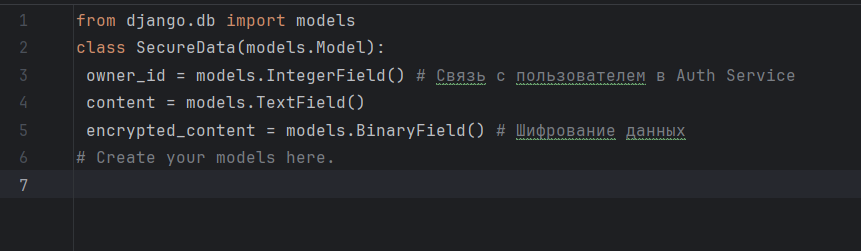
REST\_FRAMEWORK = {

'DEFAULT\_THROTTLE\_RATES': { 'user': '100/hour', }

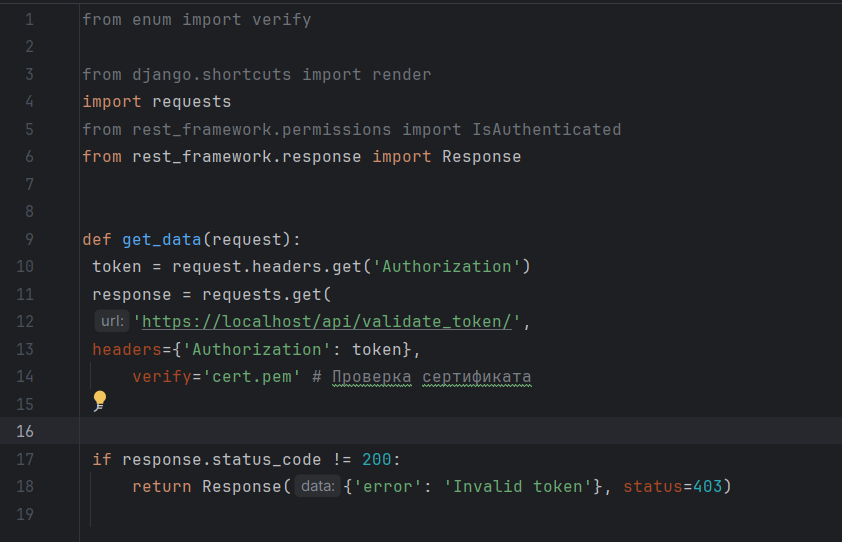
}

3. Настройка Data Service

Модель данных:



4. Проверка JWT-токена:

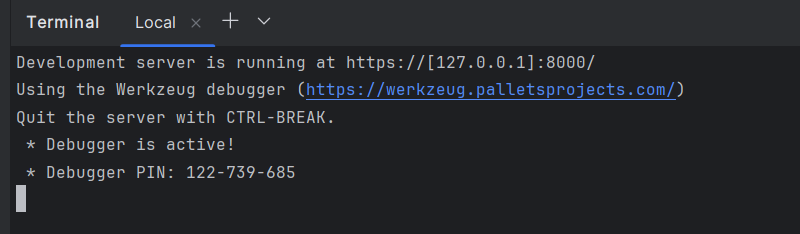


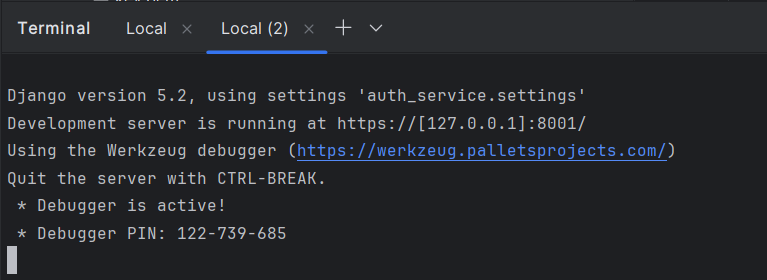
5. сгенерировали сертификаты

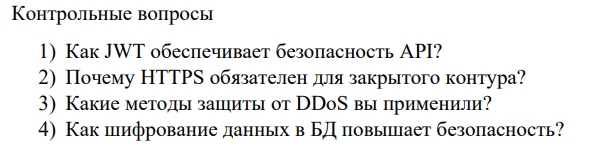
python manage.py runserver\_plus --cert-file cert.pem --key-file key.pem 8000

Запустите серверы с HTTPS:

python manage.py runserver\_plus --cert-file cert.pem --key-file key.pem 8000







**1) Как JWT обеспечивает безопасность API?**

JWT (JSON Web Token) обеспечивает безопасность API за счет:

* **Цифровой подписи** – токен подписывается с использованием секретного ключа (HMAC) или пары ключей (RSA), что гарантирует его целостность и защиту от подделки.
* **Валидации данных** – сервер проверяет подпись токена, его срок действия (exp), издателя (iss) и аудиторию (aud).
* **Stateless-аутентификации** – токен содержит всю необходимую информацию (например, roles), что избавляет от необходимости хранить сессии на сервере.
* **Ограниченного времени жизни** – короткоживущие токены (access) и механизм обновления (refresh) снижают риски при компрометации.

**Угрозы:**

* Без HTTPS токен может быть перехвачен.
* Хранение в localStorage уязвимо к XSS, в куках — к CSRF (рекомендуется HttpOnly, Secure флаги).

**2) Почему HTTPS обязателен для закрытого контура?**

HTTPS обеспечивает:

* **Шифрование трафика** – защищает данные от перехвата (MitM-атак).
* **Аутентификацию сервера** – SSL/TLS-сертификаты подтверждают легитимность узла.
* **Целостность данных** – исключает подмену пакетов злоумышленником.
* **Защиту токенов** – JWT, передаваемые по HTTP, могут быть украдены через сниффинг.

**Без HTTPS:**

* Учетные данные и токены передаются в открытом виде.
* Возможны атаки типа "человек посередине".

**3) Какие методы защиты от DDoS вы применили?**

Для защиты от DDoS используются:

* **Rate Limiting** – ограничение запросов с одного IP

**4) Как шифрование данных в БД повышает безопасность?**

Шифрование решает следующие проблемы:

* **Утечка данных** – даже при взломе БД злоумышленник не прочитает зашифрованные поля (например, пароли, PII).
* **Соответствие стандартам** – GDPR, PCI DSS требуют шифрования конфиденциальных данных.