САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа №1

Выполнил:

Пиотуховский Александр

K3341

Проверил:
Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

Нужно написать свой bopilerplate. Должно быть явное разделение на: модели, контроллеры, роуты.

Ход работы

В рамках выполнения лабораторной работы была разработана структура проекта. Основной целью являлось создание boilerplate-приложения с чётким разделением логики по слоям. Также были реализованы CRUD-операции для ранее спроектированных моделей данных, а также API-эндпоинт для получения информации о пользователе по id или email.

1. Структура проекта

Проект располагается в каталоге src. Он состоит из набора технических и бизнес-модулей. В app/ содержатся:

- Инфраструктурные компоненты (core, db, utils),
- Реализация DI-контейнера, настраивающего разрешение зависимостей,
- Роутинг и API-интерфейсы.

На рисунке 1 изображена структура проекта в корне.

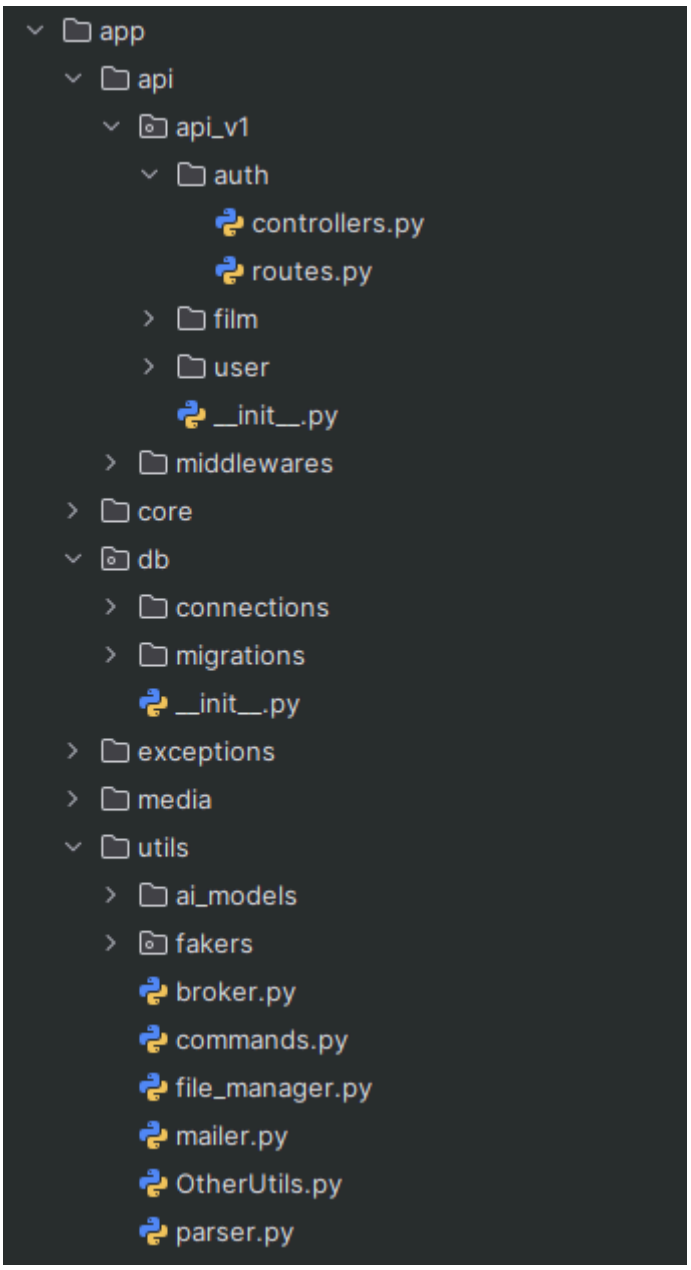

Рисунок 1 – структура проекта в корне репозитория

В модулях film/ и user/ реализованы:

- Репозитории, модели и SQL-запросы,
- Схемы для передачи данных
- Бизнес-логика, изолированная от инфраструктуры

На рисунке 2 изображена структура модуля film.

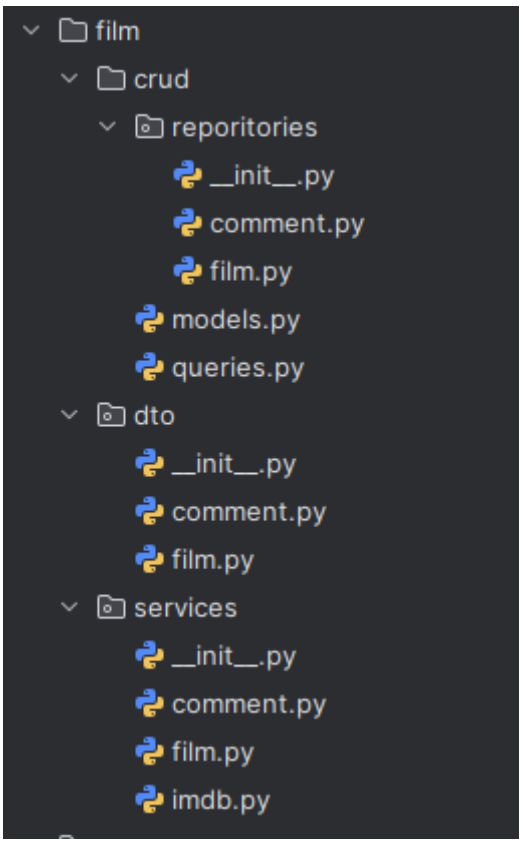

Рисунок 2 – структура модуля film

2. Принципы архитектуры

Проект построен по принципам чистой архитектуры. Слои приложения изолированы: бизнес-логика не зависит от фреймворка, СУБД или способа доставки данных. Интерфейсы определяются в доменных слоях, а инфраструктура реализует эти интерфейсы. Все зависимости передаются через контейнер внедрения зависимостей, что обеспечивает инверсию зависимостей и лёгкую заменяемость компонентов.

Пример: модуль работы с базой данных реализован через абстракции в db/connections/interface.py, а конкретная реализация postgres.py регистрируется в DI-контейнере. Это позволяет легко подменить реализацию, не изменяя бизнес-логику приложения. Реализация представлена на рисунке 3.

Рисунок 3 – Регистрация зависимостей в DI-контейнере

Вывод

В ходе выполнения работы была разработана архитектура приложения с разделением на слои. Была реализована поддержка внедрения зависимостей и принципов чистой архитектуры, что обеспечивает модульность и лёгкую заменяемость компонентов.