Хорошо — у тебя уже готов основной запускной модуль FastAPI-приложения. Вот что важно проверить и уточнить по структуре и инициализации:

**✅ Что уже реализовано:**

1. **FastAPI и Telegram интеграция** через AdvancedApplication.
2. **Маршруты** (файлы, email, суммирование, документы).
3. **CORS** политика открыта для всех источников.
4. **Healthcheck и root-роут**.
5. **Асинхронный запуск** через app\_instance.start().

**📌 Что важно убедиться:**

1. **app/core/config.py содержит settings.VERSION** и использует pydantic.BaseSettings, подгружая .env:

python

КопироватьРедактировать

from pydantic import BaseSettings

from dotenv import load\_dotenv

import os

load\_dotenv()

class Settings(BaseSettings):

VERSION: str = "0.1.0"

DATABASE\_URL: str = os.getenv("DATABASE\_URL")

AI\_PROVIDER: str = os.getenv("AI\_PROVIDER")

...

settings = Settings()

1. **AdvancedApplication в advanced\_architecture.py должен загружать БД и Telegram**:

python

КопироватьРедактировать

class AdvancedApplication:

def \_\_init\_\_(self):

self.rest\_adapter = FastAPIAdapter() # FastAPI instance + routers

self.bot\_adapter = TelegramAdapter() # Telegram bot instance

async def start(self):

await self.rest\_adapter.startup()

await self.bot\_adapter.start\_polling()

1. **Инициализация БД через Alembic или SQLAlchemy должна быть вызвана в startup()** — или из FastAPI событий:

python

КопироватьРедактировать

@app.on\_event("startup")

async def startup\_event():

from app.db.session import init\_db

await init\_db()

Если хочешь, я помогу сформировать config.py и init\_db() с привязкой к SQLAlchemy и проверкой наличия .env. Нужно?