

README: Архитектура проекта "Signal Stats Tracker"

Мы переходим к профессиональной, модульной структуре. Это позволит нам легко тестировать, отлаживать и расширять проект.

Структура папок

```
/signal-tracker-project
  └── src/          # <== Весь наш код живет здесь
      ├── handlers/    # <== Логика Telegram-команд (commands.py)
      │   └── __init__.py
      ├── models/       # <== Модели данных Pydantic (signal.py)
      │   └── __init__.py
      └── signal.py
      ├── exchanges/    # <== Адаптеры для бирж (binance.py, bybit.py)
      │   └── __init__.py
      ├── base.py
      ├── factory.py
      └── ...
      ├── services/     # <== "Мозг" (price_checker.py, signal_tracker.py)
      │   └── __init__.py
      ├── storage/       # <== Логика "базы данных" (json_storage.py)
      │   └── __init__.py
      └── base.py
      ├── utils/         # <== Вспомогательные модули
      │   └── __init__.py
      ├── config.py      # <== Ваша конфигурация
      └── logger.py     # <== Ваш логгер
      └── main.py        # <== Главная точка входа, запускающая все

  └── tests/         # <== ВСЕ ТЕСТЫ ЖИВУТ ЗДЕСЬ
      ├── __init__.py
      └── test_signal_model.py # <== Пример теста

  └── data/          # <== Папка для JSON базы данных
      └── signals.json  # <== Наша "база" (появится при первом запуске)

  └── .env           # <== (Секрет!) Файл с вашими API-ключами
  └── .env.example   # <== Пример файла .env
  └── requirements.txt # <== Все зависимости проекта
```

```
└── leapcell.yaml      # <== Файл для деплоя (он уже настроен на src/main.py)
```

Как запускать тесты

1. Убедитесь, что вы в **корневой** папке проекта (той, где src и tests).
2. Установите pytest (он есть в requirements.txt).
3. Просто запустите в терминале:
pytest

Pytest автоматически найдет и выполнит все тесты из папки tests/. Это и есть "модульное" и "регрессионное" тестирование.