### Руководство программиста

«Telegram-бот для просмотра информации о рыночных активах»

### 1. Установка и запуск программы

### 1.1. Подготовка окружения

### Скачайте файлы программы из репозитория: [GitHub Repository](https://github.com/bigidulka/4k-educational-practice" \t "_new).

### Убедитесь, что на вашем компьютере установлен Python 3.x.

### Убедитесь, что у вас есть следующие библиотеки:

### yfinance

### investpy

### pytest

### pytest-asyncio

### pytest-cov

### investiny

### tinkoff-investments

### aiogram

### Для установки зависимостей выполните команду:

### pip install -r requirements.txt

### 1.2. Настройка конфигурации

### Настройте config файл. Создайте файл config в корне проекта и добавьте в него:

### BOT\_TOKEN=<Ваш\_Telegram\_токен>

### 1.3. Подключение базы данных

### Для локальной базы данных проверьте файл database.py в директории bots/data. Убедитесь, что путь и параметры подключения соответствуют вашей конфигурации.

### 1.4. Запуск бота и API приложения

### Запустите основное приложение:

### python app/main.py

### Запустите бота:

### python bots/main.py

### 2. Проверка работоспособности

### 2.1. Проверка запуска

### Убедитесь, что приложение на базе app/main.py запускается корректно и отвечает на запросы.

### Проверьте, что бот (bots/main.py) доступен в Telegram через команду /start.

### 2.2. Тестирование функциональности

### Проверьте работу всех основных функций:

### Просмотр списка активов (акции, валюты, криптовалюты).

### Добавление активов в избранное.

### Построение графиков.

### Настройка уведомлений о ценах.

### Для проверки взаимодействия с базой данных используйте тесты:

### pytest --cov=app

### 3. Поддержка и обновление программы

### 3.1. Обновления

### Регулярно обновляйте зависимости:

### pip install --upgrade -r requirements.txt

### Следите за изменениями в библиотеках aiogram, investpy, yfinance.

### 3.2. Мониторинг ошибок

### Используйте встроенные логеры для отслеживания ошибок:

### Лог-файлы находятся в директории logs/ (если настроены).

### При необходимости настройте уровень логирования в config.py.

### 3.3. Документация и уведомления

### Обновляйте README.md и документацию о новых функциях в репозитории.

### Уведомляйте пользователей через GitHub Issues или личные сообщения в Telegram.

### 3.4. Резервное копирование

### Регулярно сохраняйте резервные копии базы данных PostgreSQL:

### pg\_dump -U <user> -d <database> > backup.sql

### 4. Модернизация программы

### 4.1. Разработка новых функций

### Создайте новую ветку для разработки:

### git checkout -b feature/<название\_функции>

### Реализуйте функции в соответствующих модулях:

### API-запросы: app/src/services/.

### Обработчики команд: bots/handlers/.

### 4.2. Тестирование

### Напишите тесты для новой функции в директории tests/:

### pytest tests/

### Убедитесь, что покрытие кода тестами выше 80%:

### pytest --cov=app --cov=bots

### 4.3. Внесение изменений

### Сливайте изменения в основную ветку только после успешного тестирования:

### git push origin feature/<название\_функции>

### 5. Основная структура проекта

|  |  |
| --- | --- |
| Директория | Описание |
| app/src/ | API-запросы и обработка данных из внешних сервисов. |
| bots/handlers/ | Обработчики команд и событий Telegram. |
| bots/keyboards/ | Inline и reply клавиатуры. |
| bots/utils/ | Утилиты, промежуточное ПО, хелперы. |
| bots/data/ | Файлы базы данных и настройки подключения. |
| tests/ | Тесты для проверки функциональности. |

### 6. Часто задаваемые вопросы (FAQ)

### Ошибка: Invalid BOT\_TOKEN

### Убедитесь, что в .env указан правильный токен бота.

### Ошибка базы данных

### Проверьте подключение к базе данных и правильность настроек в database.py.

### Бот не отвечает на команды

### Убедитесь, что bots/main.py запущен и токен в .env указан правильно.