kluchikbot

Как менять текст?

Структура бота организована таким образом, чтобы легко было обновить любой текст в меню. Для этого каждое меню хранит свои тексты в отдельных файлах. Все тексты разделены по категориям, чтобы вам было удобно находить и менять нужную строчку.

Сама структура:

```
├──bot
│   handlers
│   ├── main_menu
│   │   ├── messages_for_main_menu.py
│   ├── make_orders
│   │   ├── messages_for_orders.py
│   ├── registration
│   │   ├── messages_for_registration.py
│   ├── service_workshop
│   │   └── text_for_message_workshop.py
│   └── smart_keys
│       └── text_for_smartkeys.py
```

Чтобы изменить текст:

1. Перейдите в папку **bot**, где находится вся структура бота.
2. Найдите папку, которая отвечает за нужное вам меню:
   - **main_menu** — это текст для главного меню.
   - **make_orders** — текст, связанный с оформлением заказов.
   - **registration** — текст для регистрации.
   - **service_workshop** — текст для мастерской.
   - **smart_keys** — текст для смарт ключей.
3. Внутри выбранной папки найдите файл с текстами, например, `messages_for_main_menu.py` или `messages_for_orders.py`.
4. Откройте файл и найдите строчку, которую нужно изменить.
5. После изменения текста просто сохраните файл и перезапустите бота.

Теперь ваш бот будет использовать новые тексты!

Запуск бота

Шаг 1: Установка Python

1. **Для Windows**:
   - Перейдите на сайт python.org и скачайте последнюю стабильную версию Python.
   - Во время установки убедитесь, что выбрали опцию "Add Python to PATH".
2. **Для macOS**:
   - Откройте Terminal и выполните команду:
     `brew install python`
   - Если у вас нет Homebrew, установите его, следуя инструкциям на официальном сайте Homebrew.
3. **Для Linux**:
   - На большинстве систем Python уже установлен. Для его установки выполните команду:
     `sudo apt update sudo apt install python3 python3-pip`

Шаг 2: Создание виртуального окружения

После установки Python необходимо создать виртуальное окружение для вашего проекта.

1. Откройте командную строку (или Terminal на macOS и Linux).
2. Перейдите в папку с проектом:
   `cd /путь/к/проекту`
3. Создайте виртуальное окружение:
   `python -m venv venv`
4. Активируйте виртуальное окружение:

- **Для Windows**:
  `.\venv\Scripts\activate`
- **Для macOS и Linux**:
  `source venv/bin/activate`

Шаг 3: Установка зависимостей

После активации виртуального окружения установите все зависимости из файла `requirements.txt`:

`pip install -r requirements.txt`

Шаг 4: Создание `.env` файла

Перед запуском бота необходимо создать файл `.env`, который будет содержать конфиденциальные данные, такие как API-ключ бота и пароли администраторов.

1. Скопируйте файл `example_env` и переименуйте его в `.env`.

2. Откройте `.env` и введите следующие данные:

```
API_TOKEN=YOUR_API_TOKEN
ADMIN_PASSWORD=ADMIN_PASSWORD
BIG_ADMIN_PASSWORD=BIG_ADMIN_PASSWORD
```

Замените `YOUR_API_TOKEN`, `ADMIN_PASSWORD` и `BIG_ADMIN_PASSWORD` на соответствующие значения.

Шаг 5: Настройка PYTHONPATH

Для того чтобы ваш бот работал корректно, необходимо изменить переменную окружения `PYTHONPATH` на папку `kluchikbot`.

1. **Для Windows**:
   `set PYTHONPATH=путь\к\папке\kluchikbot`
2. **Для macOS и Linux**:
   `export PYTHONPATH=/путь/к/проекту/kluchikbot`

Шаг 6: Запуск бота

Теперь, когда всё настроено, вы можете запустить бота.
`python bot/main.py`

Структура проекта.

```
├── bot
│   ├── handlers                # Обработчики команд и сообщений
│   │   ├── admin_panel         # Админ-панель
│   │   ├── main_menu           # Главныое меню бота
│   │   ├── make_orders         # Обработка заказов
│   │   ├── registration        # Обработка регистрации пользователей
│   │   ├── service_workshop    # Обработка мастерской
│   │   └── smart_keys          # Обработка смарт ключей
│   ├── main.py                 # Основной файл для запуска бота
│   └── utils                   # Утилиты, декораторы, клавиатуры, работа
с базой данных
├── config                      # Конфигурации
│   ├── load_env.py             # Загрузка переменных окружения
│   ├── logger_config.py        # Настройки логгера
├── database                    # База данных и миграции
│   ├── alembic.ini             # Конфигурация Alembic для миграций
│   ├── confdb.py               # Конфигурации SQLAlchemy.
│   ├── database.db             # Файл базы данных SQLite
│   ├── migrations              # Миграции базы данных
│   ├── models.py               # Модели базы данных
├── docs                        # Документация
```

```
├── example_env                 # Пример файла с переменными окружения
├── media                       # Медиа-файлы
└── requirements.txt            # Список зависимостей
```

Команды администратора.

- `/admin` - Команда для получения административной панели.

Авторизация

- `/bigadminlogin` — Вход старшего администратора. Требуется пароль.
- `/adminlogin` — Вход администратора. Требуется пароль.

Работа с бонусами клиентов

- `/sale` — Начисление или списание бонусов у клиентов по номеру телефона.

Команды старшего администратора

- `/make_sending` — Массовая рассылка текстового сообщения всем пользователям.
- `/make_sending_with_photo` — Массовая рассылка сообщения с фото.
- `/deleteadmin` — Удаление статуса администратора у указанного пользователя.
- `/getadminlog` — Функция для получения файла с логами, действия младшего администратора.