****

**NT-Encrypt: Документация по проекту**

**Название Проекта:** NT-Encrypt

**Автор:** Красильников Максим Игоревич

**Дата:** 20.03.2024

**Версия:** 1.0

**Аннотация:**

Этот документ предоставляет документацию по проекту NT-Encrypt – Telegram боту для безопасного обмена сообщениями, разработанному для обеспечения конфиденциальности общения пользователей. В документе описаны функциональные возможности проекта, архитектура, меры безопасности, интерфейс пользователя, структура базы данных, интеграция с Telegram, модель подписки и выводы.

Москва, 2024

Содержание

[1 Введение 4](#_Toc161968466)

[1.1 Описание целей и задач проекта NT-Encrypt 4](#_Toc161968467)

[1.2 Обоснование необходимости создания безопасного приложения для обмена сообщениями 4](#_Toc161968468)

[2 Обзор Проекта 5](#_Toc161968469)

[2.1 Общий обзор функциональности и возможностей NT-Encrypt 5](#_Toc161968470)

[2.2 Краткое описание основных компонентов бота 5](#_Toc161968471)

[3 Функциональные Возможности 6](#_Toc161968472)

[3.1 Подробное описание всех функций, которые предоставляет приложение NT-Encrypt. 6](#_Toc161968473)

[3.2 Объяснение процесса шифрования и дешифрования сообщений 6](#_Toc161968474)

[3.3 Обзор возможностей подписки и доступных функций для подписчиков 6](#_Toc161968475)

[4 Архитектура 7](#_Toc161968476)

[4.1 Обзор архитектуры приложения, включая его компоненты и взаимосвязи между ними 7](#_Toc161968477)

[4.2 Описание основных технологий и инструментов 7](#_Toc161968478)

[5 Меры Безопасности 8](#_Toc161968479)

[5.1 Подробное описание мер безопасности, реализованных в NT-Encrypt для защиты сообщений пользователей 8](#_Toc161968480)

[5.2 Объяснение методов шифрования и защиты данных 8](#_Toc161968481)

[6 Интерфейс Пользователя 9](#_Toc161968482)

[6.1 Описание пользовательского интерфейса бота. 9](#_Toc161968483)

[6.2 Обзор основных экранов, кнопок и элементов управления 9](#_Toc161968484)

[7 Структура Базы Данных 10](#_Toc161968485)

[7.1 Описание структуры базы данных 10](#_Toc161968486)

[7.2 Объяснение основных таблиц и их взаимосвязей 10](#_Toc161968487)

[8 Интеграция с Telegram 11](#_Toc161968488)

[8.1 Обзор интеграции приложения с платформой Telegram 11](#_Toc161968489)

[8.2 Описание основных функций и возможностей, доступных через Telegram-интерфейс 11](#_Toc161968490)

[9 Модель Подписки 12](#_Toc161968491)

[9.1 Обзор модели подписки приложения и ее особенностей 12](#_Toc161968492)

[10 Модули бота 13](#_Toc161968493)

[10.1 Структура модуля обработки сообщений (main.py): 13](#_Toc161968494)

[10.1.1 Настройка бота и обработка команд 13](#_Toc161968495)

[10.1.2 Обработка команд старта и описаний 13](#_Toc161968496)

[10.1.3 Обработка шифрования и дешифрования сообщений 13](#_Toc161968497)

[10.1.4 Обработка дополнительных функций 13](#_Toc161968498)

[10.1.5 Обработка подписки 13](#_Toc161968499)

[10.1.6 Обработка неопознанных сообщений 13](#_Toc161968500)

[10.1.7 Заключение 13](#_Toc161968501)

[10.2 Структура модуля для расшифровки сообщений (encrypt.py) 14](#_Toc161968502)

[10.2.1 Описание функции message\_input 14](#_Toc161968503)

[10.3 Структура модуля для расшифровки сообщений (decrypt.py) 15](#_Toc161968504)

[10.3.1 Описание функции message\_output 15](#_Toc161968505)

[10.4 Структура модуля для расшифровки сообщений (SQL\_DB.py) 16](#_Toc161968506)

[10.5 Структура модуля для расшифровки сообщений (text.py) 17](#_Toc161968507)

[10.6 Структура модуля для расшифровки сообщений (state.py) 18](#_Toc161968508)

[10.7 Структура модуля для расшифровки сообщений (buttons.py) 19](#_Toc161968509)

[10.8 Структура модуля для расшифровки сообщений (payments.py) 20](#_Toc161968510)

[11 Заключение 21](#_Toc161968511)

[11.1 Подведение итогов и общий вывод о проекте NT-Encrypt 21](#_Toc161968512)

[11.2 Будущие планы развития и улучшения бота 21](#_Toc161968513)

[12 Блок схема проекта 22](#_Toc161968514)

# Введение

## Описание целей и задач проекта NT-Encrypt

Проект NT-Encrypt создан с целью обеспечения безопасного и конфиденциального обмена сообщениями между пользователями. Основной задачей бота является предоставление средства коммуникации, которое гарантирует сохранность личной информации и защиту приватности пользователей в онлайн-среде. Я стремлюсь предложить простой в использовании и надежный инструмент для общения, который обеспечивает шифрование и защиту данных на высоком уровне.

## Обоснование необходимости создания безопасного приложения для обмена сообщениями

С ростом числа онлайн-угроз и рисков нарушения конфиденциальности данных пользователей становится все более актуальной задача обеспечения безопасности при обмене сообщениями. В цифровой эпохе, когда большинство коммуникаций происходит через интернет, важно иметь инструмент, который защищает личную информацию от несанкционированного доступа и перехвата. NT-Encrypt призван удовлетворить эту потребность, предоставляя пользователям защищенное пространство для общения, где они могут быть уверены в безопасности своих персональных данных и сообщений.

# Обзор Проекта

## Общий обзор функциональности и возможностей NT-Encrypt

NT-Encrypt — это инновационный телеграм-бот для обмена сообщениями, который обеспечивает высокий уровень конфиденциальности и безопасности для своих пользователей. С помощью NT-Encrypt пользователи могут обмениваться сообщениями, зная, что их личные данные и личная переписка защищены с использованием надежных методов шифрования. Основанный на шифре Цезаря, этот бот обеспечивает простоту использования и надежную защиту сообщений.

## Краткое описание основных компонентов бота

* **Интерфейс чата** - Интуитивно понятный интерфейс чата позволяет пользователям обмениваться сообщениями с удобством. Пользователи могут отправлять текстовые сообщения, вводя их в специальное поле ввода, и просматривать историю переписки.
* **Шифрование сообщений** - Основной функционал NT-Encrypt заключается в шифровании и расшифровке сообщений с использованием шифра Цезаря. Этот метод обеспечивает конфиденциальность сообщений и защиту от несанкционированного доступа к личным данным пользователей.
* **Подписка на дополнительные возможности** - Пользователи могут подписаться на дополнительные возможности, такие как неограниченное количество сообщений, доступ к групповому чату с разработчиками для обмена сообщениями и предложений по улучшению бота, а также доступ к закрытому информационному каналу «Кодовая семантика». Подписка предоставляет пользователям дополнительные преимущества и функциональность в боте.
* **Профиль пользователя** - Каждый пользователь имеет свой профиль, в котором отображается информация о текущей подписке, статусе аккаунта и другие настройки.

NT-Encrypt объединяет простоту использования с высоким уровнем безопасности, предлагая пользователям надежное решение для обмена сообщениями в мессенджере Telegram.

# Функциональные Возможности

## Подробное описание всех функций, которые предоставляет приложение NT-Encrypt.

* **Шифрование и дешифрование сообщений -** NT-Encrypt позволяет пользователям шифровать отправляемые сообщения, чтобы обеспечить их конфиденциальность. Для расшифровки сообщений получатель также может использовать этот бот. Это обеспечивает безопасный обмен информацией между пользователями.
* **Подписка на дополнительные функции -** Подписка на NT-Encrypt предоставляет пользователям доступ к дополнительным функциям, таким как неограниченное количество сообщений, участие в групповом чате с разработчиками для предложений по улучшению приложения и доступ к закрытому информационному каналу «Кодовая семантика».
* **Профиль пользователя -** Каждый пользователь имеет свой профиль, где отображается информация о текущей подписке, статусе аккаунта и другие настройки.

## Объяснение процесса шифрования и дешифрования сообщений

Процесс шифрования в NT-Encrypt осуществляется путем замены каждой буквы в сообщении на другую букву алфавита с определенным сдвигом. Этот сдвиг определяется ключом шифрования, который выбирает отправитель сообщения. Получатель, зная этот ключ, может произвести обратную операцию - дешифровку сообщения и прочитать исходный текст.

## Обзор возможностей подписки и доступных функций для подписчиков

Подписка на NT-Encrypt открывает доступ к ряду дополнительных функций, среди которых:

* **Неограниченное количество сообщений** - подписчики могут отправлять сколько угодно сообщений без ограничений.
* **Участие в групповом чате** - подписчики могут присоединиться к групповому чату с разработчиками, чтобы обсудить идеи, предложения и получить помощь.
* **Доступ к информационному каналу «Кодовая семантика»** - подписчики получают доступ к закрытому информационному каналу с актуальными новостями и статьями о кодировании, разработке программного обеспечения и другим темам.

Эти дополнительные возможности делают подписку на NT-Encrypt привлекательным выбором для пользователей, желающих получить больше от своего опыта обмена сообщениями в Telegram.

# Архитектура

## Обзор архитектуры приложения, включая его компоненты и взаимосвязи между ними

Архитектура приложения NT-Encrypt построена на основе модульной структуры, которая включает в себя следующие ключевые компоненты:

* **Telegram Bot API** - Основной интерфейс для взаимодействия с пользователем. Бот использует API Telegram для получения и отправки сообщений, управления состояниями чата и обработки команд.
* **Бизнес-логика** - Этот компонент отвечает за обработку команд пользователя, шифрование и дешифрование сообщений, управление подпиской и профилем пользователя.
* **База данных** - Хранит информацию о пользователях, их сообщениях, состояниях чата и подписках.
* **States и StatesGroup** - Компоненты, предоставляемые библиотекой aiogram для управления состояниями чата и обработки пользовательских действий.
* **Клавиатура** - Отображает пользовательские интерфейсы в виде кнопок для удобства использования и навигации по функциям бота.

## Описание основных технологий и инструментов

* **Python** - Язык программирования Python используется для разработки бота благодаря своей простоте, гибкости и богатой экосистеме библиотек, таких как aiogram.
* **Aiogram** — Это мощная библиотека Python для разработки Telegram ботов. Она предоставляет удобные инструменты для взаимодействия с Telegram Bot API, управления состояниями чата и обработки команд.
* **SQLite** - Компактная реляционная база данных SQLite используется для хранения информации о пользователях, сообщениях и других данных, необходимых для работы бота.
* **Git** - Система контроля версий Git используется для управления исходным кодом проекта, обеспечения совместной разработки и отслеживания изменений.
* **Telegram Bot API** - Официальный API Telegram используется для взаимодействия с мессенджером Telegram, отправки и получения сообщений, управления состояниями чата и других операций.

Эти технологии и инструменты обеспечивают надежность, производительность и удобство использования приложения NT-Encrypt.

# Меры Безопасности

## Подробное описание мер безопасности, реализованных в NT-Encrypt для защиты сообщений пользователей

NT-Encrypt предпринимает ряд мер для обеспечения конфиденциальности и безопасности обмена сообщениями пользователей:

* **Шифрование сообщений** - Все сообщения, отправляемые и принимаемые через NT-Encrypt, подвергаются шифрованию с использованием надежных алгоритмов шифрования для предотвращения доступа к содержимому третьих лиц.
* **Уникальные ключи шифрования** - Каждый пользователь сам устанавливает уникальный ключ шифрования, который используется для защиты их сообщений. Эти ключи хранятся в безопасном хранилище и недоступны для сторонних лиц.
* **Защита данных пользователя** - NT-Encrypt обеспечивает защиту личных данных пользователей, включая их идентификационные данные и историю сообщений, от несанкционированного доступа.
* **Регулярные обновления безопасности** - Команда разработчиков NT-Encrypt выпускает регулярные обновления для исправления обнаруженных уязвимостей и повышения общего уровня безопасности приложения.

## Объяснение методов шифрования и защиты данных

NT-Encrypt применяет передовые методы шифрования для защиты данных пользователей. Каждое сообщение шифруется с использованием уникального ключа шифрования, доступ к которому имеют только отправитель и получатель.

# Интерфейс Пользователя

## Описание пользовательского интерфейса бота.

Пользовательский интерфейс бота NT-Encrypt обладает интуитивно понятным и простым дизайном, который обеспечивает удобство использования и быстрый доступ к функциям приложения. Основные элементы интерфейса включают в себя:

* **Клавиатура с кнопками** - Бот предоставляет пользователю клавиатуру с кнопками для быстрого доступа к основным функциям. Эти кнопки отображаются внизу экрана чата и включают в себя команды для шифрования, дешифрования, подписки, получения информации и другие функции.
* **Сообщения бота** - Бот активно взаимодействует с пользователем, отправляя информационные сообщения, подсказки и ответы на запросы. Эти сообщения содержат полезную информацию о функциях бота, инструкции по использованию и другие сведения.

## Обзор основных экранов, кнопок и элементов управления

* **Кнопки шифрования и дешифрования** - Позволяют пользователю зашифровать и дешифровать сообщения соответственно.
* **Кнопка подписки** - Позволяет пользователю оформить подписку для получения дополнительных функций и возможностей.
* **Кнопка описания** - Предоставляет пользователю информацию о функциональности и возможностях бота.
* **Кнопка общего чата** - Позволяет пользователю общаться с другими участниками чата и разработчиками, делиться идеями и предложениями.
* **Кнопка информационного канала** - Предоставляет доступ к информационному каналу «Кодовая семантика» с актуальными новостями и сведениями.

Эти элементы управления обеспечивают удобное и эффективное взаимодействие пользователя с ботом NT-Encrypt, позволяя ему максимально использовать его функциональность.

# Структура Базы Данных

База данных, используемая в NT-Encrypt, разработана для эффективного хранения пользовательских данных и сообщений с соблюдением принципов безопасности и надежности. Она состоит из нескольких таблиц, каждая из которых отвечает за определенный тип данных.

## Описание структуры базы данных

* **Таблица Users\_Messages\_Shirf** - В этой таблице хранятся зашифрованные сообщения пользователей. Она содержит следующие поля:
* **Users\_id** - уникальный идентификатор пользователя;
* **Message** - зашифрованное сообщение;
* **Step** - значение шага для шифрования;
* **Result** - результат зашифрования.
* **Таблица Users\_Messages\_Deshirf** - Здесь хранятся расшифрованные сообщения пользователей. Она имеет аналогичную структуру, как и Users\_Messages\_Shirf, но содержит дешифрованные данные.
* **Таблица Payment\_subscription** - В этой таблице хранится информация о подписках пользователей. Она включает следующие поля:
* **Users\_id** - уникальный идентификатор пользователя;
* **User\_NicName** - никнейм пользователя;
* **Time\_sub** - время окончания подписки.

## Объяснение основных таблиц и их взаимосвязей

Таблицы Users\_Messages\_Shirf и Users\_Messages\_Deshirf содержат зашифрованные и дешифрованные сообщения пользователей соответственно. Они связаны общим полем Users\_id, которое позволяет сопоставить сообщения с конкретными пользователями.

Таблица Payment\_subscription содержит информацию о подписках пользователей. Ее поля Users\_id и User\_NicName связаны с таблицами Users\_Messages\_Shirf и Users\_Messages\_Deshirf, что позволяет идентифицировать пользователей и управлять их подписками.

Эта структура базы данных обеспечивает эффективное хранение и управление данными пользователей в NT-Encrypt, обеспечивая безопасность и надежность функционирования приложения.

# Интеграция с Telegram

NT-Encrypt тесно интегрирован с платформой Telegram, обеспечивая пользователям простой и удобный способ безопасного обмена сообщениями.

## Обзор интеграции приложения с платформой Telegram

Приложение NT-Encrypt представляет собой чат-бот, доступный через мессенджер Telegram. Пользователи могут получить доступ к функциям бота, общаясь с ним через интерфейс Telegram.

## Описание основных функций и возможностей, доступных через Telegram-интерфейс

* **Зашифрование сообщений** - Пользователи могут отправлять сообщения боту в Telegram для их зашифровки перед отправкой. Для этого они должны ввести текст сообщения, после чего добавить символ "#" и указать ключ шифрования.
* **Расшифровка сообщений** - Пользователи также могут отправлять зашифрованные сообщения боту для их расшифровки. Для этого они должны ввести зашифрованный текст сообщения, после чего добавить символ "#" и указать ключ дешифрования.
* **Подписка на услуги** - Бот предоставляет возможность подписки на дополнительные услуги, такие как неограниченное количество сообщений в течение месяца и доступ к группе общего чата с разработчиками.
* **Доступ к информационному каналу** - Подписчики могут получать доступ к закрытому информационному каналу "Кодовая семантика", который предоставляет свежую информацию о мире IT, новости и советы по безопасности.

Через Telegram-интерфейс NT-Encrypt обеспечивает удобный и защищенный способ общения, позволяя пользователям обмениваться сообщениями с максимальной конфиденциальностью и безопасностью.

# Модель Подписки

Модель подписки в приложении NT-Encrypt предоставляет пользователям доступ к дополнительным функциям и возможностям за определенную ежемесячную плату.

## Обзор модели подписки приложения и ее особенностей

Подписка в NT-Encrypt предоставляет пользователям следующие возможности:

* **Неограниченное количество сообщений** - Подписчики могут отправлять и получать любое количество сообщений в течение месяца без каких-либо ограничений.
* **Доступ к группе общего чата -** Подписчики получают доступ к группе общего чата, где они могут общаться с разработчиками, задавать вопросы, предлагать идеи и участвовать в опросах и конкурсах.
* **Доступ к информационному каналу «Кодовая семантика»** - Подписчики получают доступ к закрытому информационному каналу, где предоставляется свежая информация о мире IT, новости, советы и обновления по безопасности.

**Описание доступных функций для подписчиков и бонусов подписки**

* **Неограниченное количество сообщений -** Подписчики могут отправлять и получать неограниченное количество сообщений в течение месяца, что обеспечивает им свободу общения без ограничений.
* **Доступ к группе общего чата** - Подписчики могут общаться с разработчиками и другими пользователями в группе общего чата, обсуждая вопросы, задавая вопросы и предлагая свои идеи для улучшения приложения.
* **Доступ к информационному каналу «Кодовая семантика»** - Подписчики получают доступ к закрытому информационному каналу, где они могут узнавать последние новости и обновления о мире IT, получать советы по безопасности и быть в курсе всех изменений и обновлений в приложении.

Подписка в NT-Encrypt предоставляет пользователям дополнительные преимущества и возможности для комфортного и безопасного общения, делая использование приложения еще более удобным и эффективным.

# Модули бота

## Структура модуля обработки сообщений (main.py):

### Настройка бота и обработка команд

* Определение токена бота и создание экземпляра бота с использованием библиотеки aiogram.
* Настройка объекта диспетчера для обработки сообщений и команд от пользователей.
* Определение токена платежной системы для возможности оформления подписки на дополнительные функции.

### Обработка команд старта и описаний

* Функция cmd\_start отвечает за обработку команды /start и приветствие пользователя при первом входе.
* Функция description отвечает за вывод описания функционала бота при запросе пользователем.

### Обработка шифрования и дешифрования сообщений

* Функции shifr\_message и deshifr\_message осуществляют шифрование и дешифрование сообщений соответственно, а также сохраняют данные в базе данных.
* Проверка корректности формата сообщения перед шифрованием/дешифрованием.

### Обработка дополнительных функций

* Обработчики команд для доступа к общему чату и информационным каналам бота.
* Обработчик команды для просмотра профиля пользователя с информацией о подписке.

### Обработка подписки

* Обработчик команды для оформления подписки на дополнительные функции бота.
* Обработка успешного платежа и активация подписки пользователя.

### Обработка неопознанных сообщений

Обработчик для сообщений, не соответствующих ожидаемым командам, выводит сообщение о непонимании и предлагает использовать кнопки для взаимодействия.

### Заключение

Модуль NT-Encrypt представляет собой комплексный инструмент для безопасного обмена сообщениями в Telegram. Его функционал включает шифрование и дешифрование сообщений, доступ к общим чатам и информационным каналам, а также возможность оформления подписки на дополнительные функции. Этот модуль обеспечивает удобное и безопасное общение для всех пользователей Telegram.

## Структура модуля для расшифровки сообщений (encrypt.py)

Этот модуль содержит функцию для шифрования текстовых сообщений с использованием шифра Цезаря. Функция принимает сообщение и ключ шифрования (шаг), а затем возвращает зашифрованное сообщение.

### Описание функции message\_input

Функция message\_input принимает два аргумента:

* message - исходное сообщение, которое необходимо зашифровать.
* step - ключ шифрования (шаг), используемый для шифрования сообщения.

Для каждого символа в исходном сообщении производится следующее:

* Если символ не является буквой, то он остается без изменений.
* Для буквенных символов производится шифрование с учетом регистра и языка (кириллица или латиница) с использованием шифра Цезаря.
* Зашифрованный символ добавляется к строке с зашифрованным сообщением.

В конце функция возвращает строку с зашифрованным сообщением.

Этот модуль представляет собой важную часть функционала бота, отвечая за обработку текстовых сообщений от пользователей и их шифрование для последующей отправки.

## Структура модуля для расшифровки сообщений (decrypt.py)

Этот модуль предоставляет функцию для расшифровки текстовых сообщений, зашифрованных с использованием шифра Цезаря. Он содержит одну функцию message\_output, которая принимает зашифрованное сообщение и ключ шифрования (шаг), а затем возвращает расшифрованное сообщение.

### Описание функции message\_output

Функция message\_output принимает два аргумента:

* **shifr\_message** - зашифрованное сообщение, которое необходимо расшифровать.
* **step** - ключ шифрования (шаг), используемый для расшифровки сообщения.

Для каждого символа в зашифрованном сообщении производится следующее:

* Если символ не является буквой, то он остается без изменений.
* Для буквенных символов производится расшифровка с учетом регистра и языка (кириллица или латиница) с использованием шифра Цезаря.
* Расшифрованный символ добавляется к строке с расшифрованным сообщением.

В конце функция возвращает строку с расшифрованным сообщением.

Этот модуль представляет собой важную часть функционала бота, отвечая за обработку зашифрованных сообщений и предоставляя возможность пользователям получить их расшифровку.

## Структура модуля для расшифровки сообщений (SQL\_DB.py)

Этот модуль обеспечивает взаимодействие с базой данных SQLite, в которой хранится информация о пользователях, их сообщениях и времени подписки.

**Класс Data\_Base**

**Метод \_\_init\_\_**

Инициализирует подключение к базе данных и создает необходимые таблицы, если они еще не существуют:

* Users\_Messages\_Shirf для хранения зашифрованных сообщений.
* Users\_Messages\_Deshirf для хранения расшифрованных сообщений.
* Payment\_subscription для хранения информации о пользователях, их именах и времени подписки.

**Метод \_\_del\_\_**

Закрывает соединение с базой данных при уничтожении объекта.

**Методы add\_data\_shifr, add\_data\_deshifr, add\_users**

Добавляют информацию о зашифрованных и расшифрованных сообщениях, а также о пользователях в соответствующие таблицы базы данных.

**Метод get\_nik**

Получает ник пользователя по его идентификатору.

**Методы set\_sub\_time, get\_time\_sub**

Устанавливают и получают время подписки пользователя.

**Метод get\_sub\_status**

Проверяет статус подписки пользователя на основе текущего времени и времени подписки из базы данных.

**Метод get\_id\_status**

Проверяет наличие пользователя в базе данных по его идентификатору.

Этот модуль играет ключевую роль в сохранении и управлении данными, необходимыми для функционирования телеграм-бота, такими как сообщения пользователей, информация о подписках и другие данные о пользователях.

## Структура модуля для расшифровки сообщений (text.py)

Модуль Text содержит текстовые сообщения, которые используются ботом для общения с пользователями. Они представляют собой информацию о функциональности бота, инструкции по использованию и различные приветственные сообщения.

**greeting\_message**

Приветственное сообщение для новых пользователей. Оно предоставляет краткое описание бота, подчеркивает его основные возможности и безопасность использования.

**description**

Подробное описание функциональности бота. Здесь описываются шаги по шифрованию и дешифрованию сообщений, а также дается информация о доступе к подписке, обратной связи и других особенностях бота.

**shifr\_text**

Сообщение, предлагающее пользователю ввести текст для шифрования в определенном формате с указанием ключа.

**deshifr\_text**

Сообщение, предлагающее пользователю ввести зашифрованный текст для дешифрования в определенном формате с указанием ключа.

**Использование:**

* Этот модуль предоставляет текстовые сообщения, которые бот отправляет пользователям в различных ситуациях, таких как начало диалога, инструкции по использованию и запросы на ввод текста.
* Текст сообщений сформулирован таким образом, чтобы быть понятным и информативным для пользователей, помогая им понять, как взаимодействовать с ботом и использовать его функционал.

Модуль обеспечивает эффективное коммуникацию с пользователями и помогает им в использовании функций бота.

## Структура модуля для расшифровки сообщений (state.py)

Модуль States представляет собой класс, определяющий состояния для управления различными действиями и процессами в боте Telegram. Каждое состояние соответствует определенному функционалу или процессу, такому как шифрование сообщений, оформление подписки и т.д. Этот модуль используется вместе с модулем управления ботом для управления его поведением в зависимости от текущего состояния.

**Класс Состояний**

* Класс States является наследником класса StatesGroup из модуля aiogram.fsm.state и определяет различные состояния, доступные для управления ботом.
* В классе определены следующие состояния:
* **shifr** - состояние шифрования сообщений.
* **deshifr** - состояние дешифрования сообщений.
* **buy\_subscription** - состояние оформления подписки.
* **feedback** - состояние для обратной связи.
* **description** - состояние для просмотра описания функционала бота.
* **info\_channel** - состояние для получения информации о информационном канале.

Модуль States важный инструмент для организации процессов в боте. Он определяет различные состояния, которые могут быть использованы для управления поведением бота в зависимости от текущего контекста или запроса пользователя.

## Структура модуля для расшифровки сообщений (buttons.py)

Модуль Buttons содержит определение клавиатуры для бота, которая используется для взаимодействия с пользователем. Клавиатура предоставляет удобный способ выбора опций и выполнения действий.

**kb**

Список списков, представляющих собой разметку клавиатуры. Каждый внутренний список содержит кнопки для одной строки клавиатуры.

**keyboard**

Объект типа ReplyKeyboardMarkup, который представляет собой готовую к использованию клавиатуру. Он инициализируется с помощью разметки kb и дополнительных параметров для настройки внешнего вида и поведения клавиатуры.

**Использование:**

* Этот модуль определяет структуру клавиатуры, которая включает в себя кнопки для различных действий, таких как шифрование сообщения, подписка, доступ к чату и профилю и т. д.
* Клавиатура создается с помощью объекта ReplyKeyboardMarkup, который управляет внешним видом и поведением клавиатуры. Он может быть настроен на изменение размера и отображение подсказки в поле ввода.
* После создания клавиатура может быть отправлена пользователю вместе с сообщением, чтобы предоставить ему удобный способ взаимодействия с ботом.

Модуль обеспечивает удобный и интуитивно понятный интерфейс для пользователей, позволяя им легко выбирать нужные действия и использовать функционал бота.

## Структура модуля для расшифровки сообщений (payments.py)

Модуль Payments содержит информацию о тестовой карте, которая может использоваться для оплаты услуг или товаров. Он предоставляет данные тестовой карты, которые могут быть использованы для тестирования платежной системы в боте или приложении.

**Тестовая карта**

* **Номер карты -** 1111 1111 1111 1026.
* **Срок действия** - 12/22.
* **CVC** – 000.

**Использование**

* Этот модуль предоставляет информацию о тестовой карте, которая может быть использована для тестирования функционала оплаты в боте.
* Важно помнить, что эти данные являются тестовыми и предназначены исключительно для тестирования. Не рекомендуется использовать реальные кредитные или дебетовые карты для тестовых целей.

Модуль обеспечивает удобный способ тестирования функционала оплаты без необходимости использования реальных финансовых средств.

# Заключение

NT-Encrypt — это инновационное приложение, разработанное для обеспечения безопасного и конфиденциального обмена сообщениями между пользователями. Сочетая в себе простоту использования с высоким уровнем безопасности, NT-Encrypt становится надежным инструментом для всех, кто ценит конфиденциальность своей переписки.

## Подведение итогов и общий вывод о проекте NT-Encrypt

В ходе разработки NT-Encrypt мы стремились создать мощный и удобный инструмент для обмена сообщениями, который был бы прост в использовании, но при этом обеспечивал бы высокий уровень безопасности и конфиденциальности данных. Мы гордимся тем, что достигли этой цели, предоставив пользователям надежное и эффективное решение для безопасного общения.

## Будущие планы развития и улучшения бота

Я продолжаю работать над улучшением NT-Encrypt, внедряя новые функции и усовершенствования, которые сделают его еще более удобным и функциональным для пользователей. Мои планы включают в себя расширение списка поддерживаемых языков, добавление новых функций безопасности, а также улучшение интерфейса пользователя для более комфортного взаимодействия с приложением. Я также стремлюсь к активной обратной связи с нашими пользователями, чтобы учитывать их потребности и предложения при дальнейшем развитии приложения.

# Блок схема проекта

