

## 11 лабораторная работа

**Тема: Работа с API на примере мобильного приложения с запущенным Telegram-ботом. Управляйте каналом или группой Telegram с помощью мобильного приложения.**

**Цель работы: Ознакомление с созданием telegram бота и управление каналом и группой.**

Для любого бизнеса важно установить канал общения с клиентами. Но сделать это не так просто. Психология людей такова, что они не хотят захламлять память своего смартфона новым фирменным приложением из того места, которое они посетили. Совсем другое дело — чат-бот. Ненавязчивый и дружелюбный.

А кроме того — со всеми необходимыми фишками: рекламными акциями, скидками и быстрым заказом. И уведомления в одном единственном удобном мессенджере. Почему бы его не реализовать в своем бизнесе? Тем более, что это не так сложно, как вы думаете.

Сегодня мы поговорим о ботах и их API (telegram api) на базе популярного мессенджера Telegram.

### Содержание

Telegram Bot API и Telegram API BotFather: быстрый StartПростой эхо-бот  
aiogram — асинхронная библиотека

Создаем эхо-бота

Оформление сообщений

Учим бот-модерации

Делаем бота админом

### Telegram Bot API и Telegram API

Все началось с того, что Николай Дуров совместно с командой программистов создал криптографический протокол. Его движок задействовал комбинацию симметричного шифрования

AES, протокол Диффи-Хеллмана для обмена ключами шифрования между клиентами и ряд хеш-функций. На основе этого протокола был построен MTProto — механизм, позволяющий пользователям сегодня использовать Telegram-мессенджеры.

На данный момент есть два основных инструмента API, с помощью которых можно задействовать сервисы Telegram — Telegram Bot API и Telegram API.

Первый служит для разработки чат-ботов, второй позволяет делать полностью кастомные Telegram-клиенты. Разработчикам также доступна открытая библиотека TDLib (Telegram Database Library), с помощью которой

можно создавать свою версию мессенджера с уникальными опциями (как например, Telegram X, построенный именно на TDLib).

Telegram Bot API является надстройкой над Telegram API, поэтому пользоваться Bot API можно без знаний о механизме используемого протокола MTProto.

Для его работы задействован промежуточный сервер с HTTPS-интерфейсом, который шифрует трафик и обеспечивает связь с Telegram API. Bot API позволяет легко создавать программы, которые используют интерфейс Telegram для выполнения кода на локальном сервере. Пользователи могут взаимодействовать с ботами, отправляя им сообщения, команды и встроенные запросы.

Принцип работы любого бота заключается в том, что он перманентно направляет запросы на сервер и регулярно получает обновления. Получать их можно двумя способами. Во-первых, можно использовать вебхуки, когда сервер делает обратный вызов на указанный URL. А во-вторых, можно просто «забрасывать» запросами Telegram, получая постоянные ответы.

Обратите внимание — получать уведомления о новых сообщениях в боте и других событиях вы можете всего один раз. Поэтому, если данные чата представляются вам очень важными, то придется самостоятельно сохранять список чатов и историю старых сообщений. Если вы случайно сотрете/потеряете эту информацию, то вы ее больше никак не восстановите.

## Сервер

Bot API имеет возможность настройки порта и локального IP-адреса для вебхука, а также поддерживает до 100000 одновременных подключений — более чем достаточно для большинства задач.

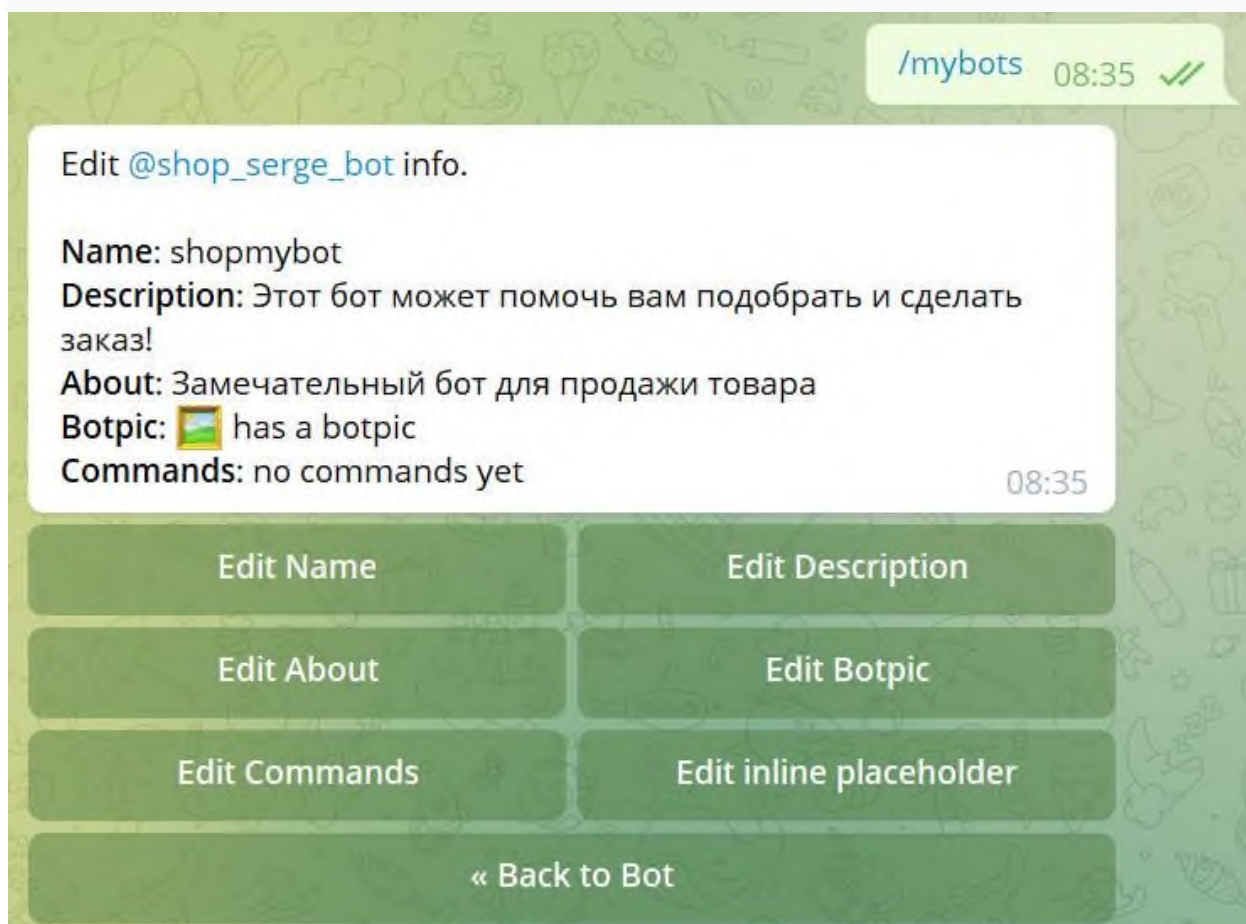
BotFather: быстрый Start Чтобы приступить к созданию собственного бота, необходимо получить токен для авторизации и подключения через API. Делается это при помощи служебного бота. Введите в поиске Telegram его имя — BotFather. Далее следует выбрать команду newbot и дать имя боту.

Затем BotFather спросит вас имя, которое обязательно должно заканчиваться на bot, например, shop\_serge\_bot. Далее для бота будет сгенерирован уникальный токен, который будет выглядеть примерно так — 2093336709:AAGiH64Ec1R8r222sM9IywvlIGFkb7wFqyo.

Всего можно генерировать не более 20 ботов на одного пользователя. Управление ботами также происходит через меню команд служебного бота BotFather. Например, если вам потребуется настроить какой-то из ваших ботов, вы должны перейти по командам **mybots** и затем нажать на кнопку Edit Bot. Здесь вы сможете настроить имя — Edit Name

и указать описание — Edit Description.

Настраивая description для бота вы задаете текст, который будет показываться пользователям под сообщением «Что может делать этот бот?» когда они его будут подключать. В профиле бота также будут показаны сведения, которые вы задали в поле «About» (команда Edit About). В меню команд имеется возможность установить аватар — BotPic.



В меню настроек BotSettings включается режим встроенных запросов (по умолчанию эта опция отключена). Когда встроенные запросы активированы, пользователи могут вызвать вашего бота, просто введя имя пользователя в поле для ввода текста в любом чате, группе или канале.

Если Telegram использовать в коммерческих целях, чат-бот можно вооружить средствами для приема платежей. Стоит обратить внимание, что сам Telegram не занимается проведением транзакций, он лишь дает возможность подключить услуги длинного списка провайдеров.

В их числе такие платежные системы, как Stripe, YooMoney, Сбербанк, PayMaster, PSB, Tranzzo, Payme, CLICK, LiqPay, Portmone, Paymega, ECOMMPAY и др. Разумеется, чтобы использовать эти платежные системы, нужно быть юридическим лицом.

aiogram — асинхронная библиотека

Реализовывать свой проект удобнее при помощи библиотек, таких как aiogram— это, пожалуй, лучшая стабильная асинхронная библиотека для Python (<https://pypi.org/project/aiogram/>).

Относительно асинхронности нужно сказать несколько слов. Python — язык однопоточный, и любые команды в нем выполняются только после того, как завершилось выполнение предыдущей команды. Однако в случае с ботом программе необходимо отвечать на множество сообщений и делать это очень быстро. Можно, конечно, плодить множество потоков выполнений, но такой вариант чаще всего неприемлем, и альтернативой выступает асинхронная реализация задачи.

Когда в потоке выполнения команды возникает какая-то пауза, например, работает функция и у нее при этом есть какой-то таймаут ожидания, мы получаем участок времени, который не занят решением алгоритма. Асинхронность в aiogram позволяет заполнить эти промежутки, ускоряя работу нашего бота и быстро отвечая на сообщения. Устанавливается эта библиотека следующей командой:

```
pip install -U aiogram.
```

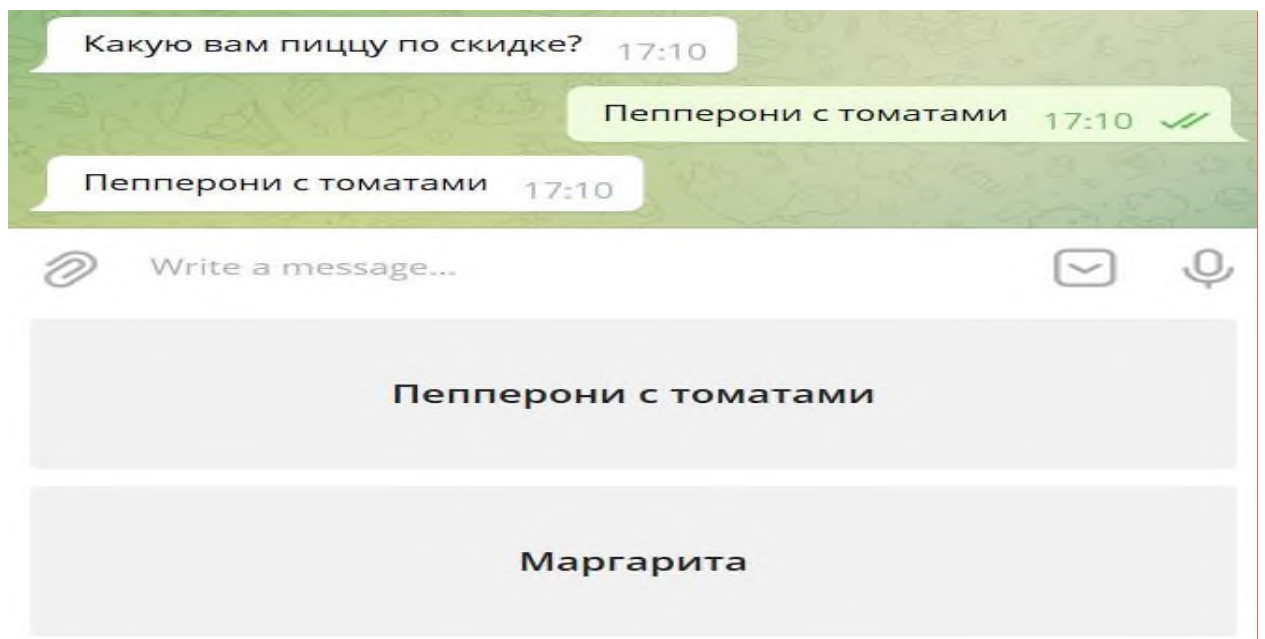
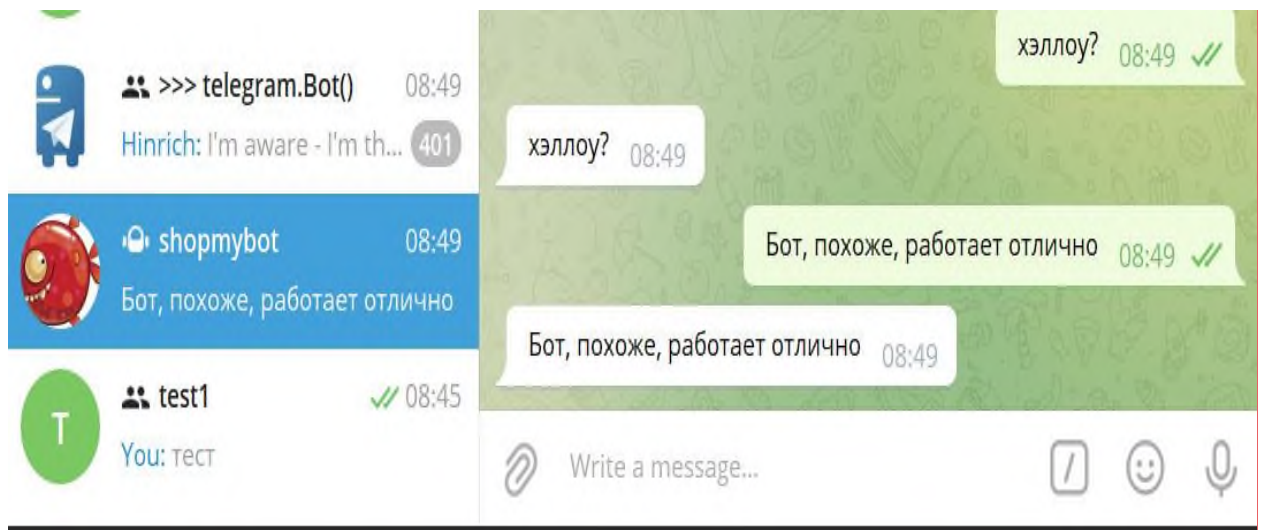
Создаем эхо-бот

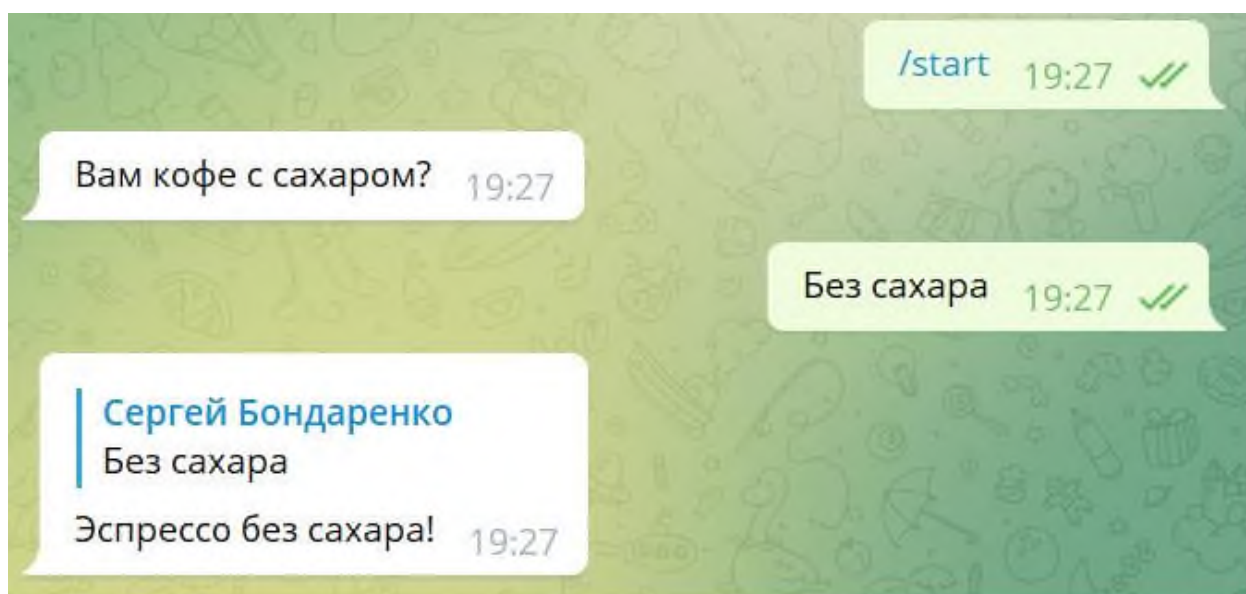
Перед тем как начать разбираться с Telegram-ботами, попробуйте создать простенький эхо-бот, который будет возвращать все поступающие сообщения. Это поможет понять механизм работы и разобраться с его функциями.

Создаем конфигурационный файл config.py и указываем в нем значение, сгенерированное ботом BotFather. Приступаем к написанию кода самого чат-бота.

Для начала импортируем конфигурацию и систему логирования. Затем подключаем все необходимые модули из aiogram. Указываем уровень логирования, а затем инициализируем бота, создав две переменные —

Bot и Dispatcher





 | Write a message...  

С сахаром

Без сахара

