**Урок 9**

**Отправка и получение разных файлов**

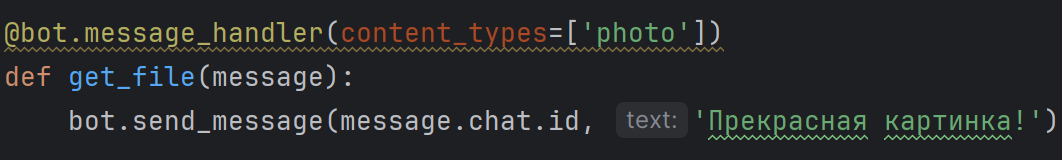
В прошлый раз мы научились менять расположение кнопок в **reply-клавиатуре** и **inline-клавиатуре**. Познакомились с встроенными методами для создания более сложных клавиатур — **row**, **insert** и **add.** Использовали новые функции кнопок, например, узнали, что кнопка может отправлять контакт или геолокацию.

На этом уроке мы научимся отвечать и отправлять различные файлы пользователю.

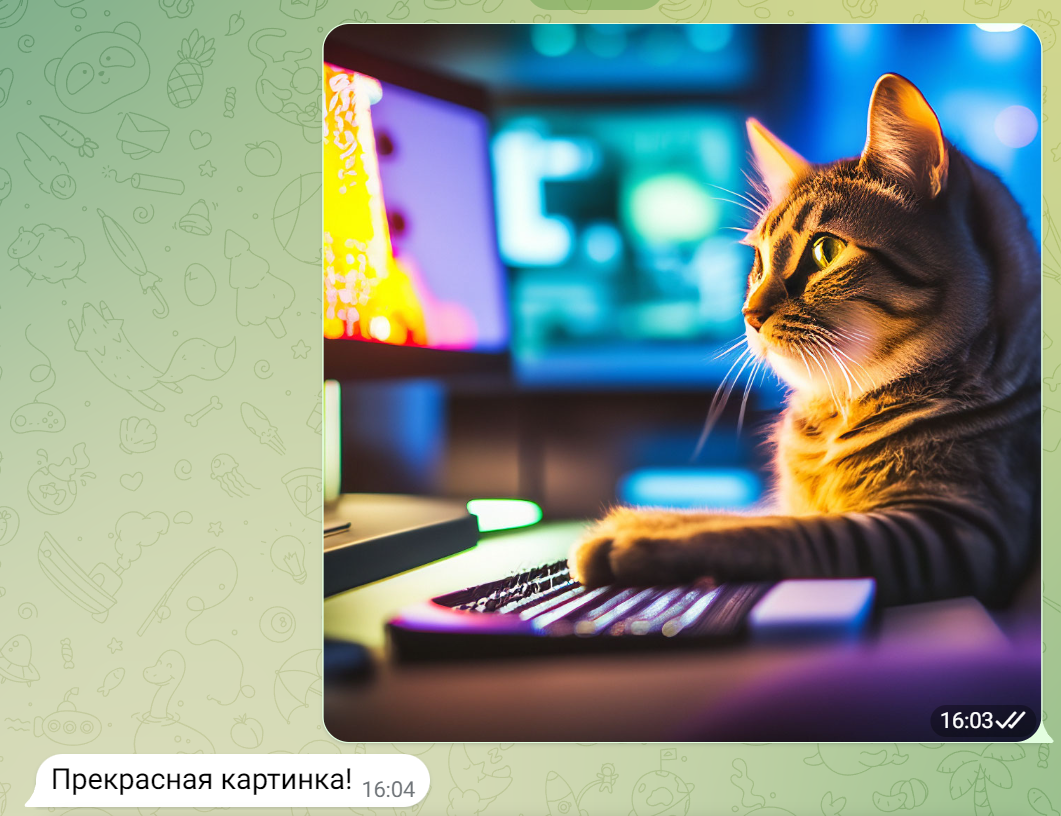
**Получение и ответ на файлы пользователя**

В прошлых уроках мы научились отслеживать команды, а также текст. А как сделать так, чтобы бот реагировал на некий файл (видео, фото, аудио и т.д.)?

Для этого используется тот же принцип, как и для отслеживания команд и текста. Мы создаем декоратор **@bot.message\_handler()**, но теперь вместо **commands** мы будем обращаться к **content\_types** и внутри мы указываем тип файла — 'audio', 'video', 'photo', 'sticker' . Данный метод может срабатывать на несколько типов файлов, если указать их через запятую (например: content\_types = 'photo', 'video').



Теперь давайте запустим программу и посмотри, что получилось.



Таким образом можно обрабатывать получение различных типов объектов и в зависимости от того, что пользователь вам отправляет, вы можете выполнять абсолютно любой код.

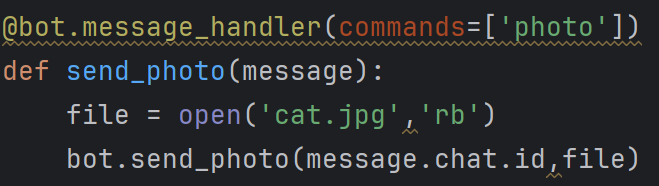
**Самостоятельная работа**

Создайте программу, чтобы бот реагировал и что-то отвечал на видео и аудио файлы.

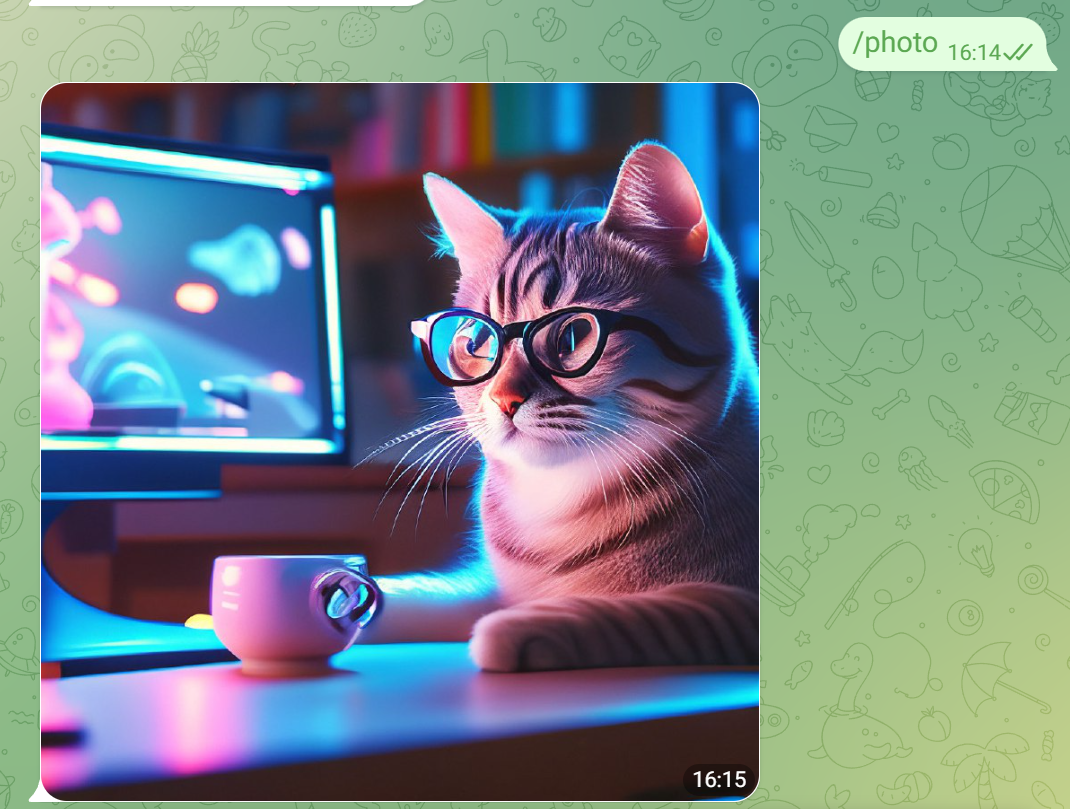
**Отправка различных файлов пользователю**

Мы уже умеем отправлять нашему пользователю различный текст, но как же нам отправлять ему разный типы файлов? Давайте для начала научимся отправлять некое изображение. Будем при получении команды **/photo**, отправлять ему картинку с котиком.

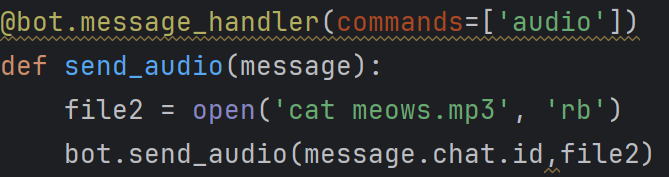
Важно чтобы ваш проект и файл находились **в одной папке**, иначе вам придётся в программе указывать путь до файла! Далее мы с вами открываем этот файл, для этого обращаемся к методу **open**. Потом мы говорим, что конкретно мы хотим открыть. Также нам надо указать формат открытия файла. Тут мы указываем что открываем его для чтения — **'rb'**. После того как мы все это сделали, обращаемся к боту и к методу под названием **send\_photo()**. Этот метод работает также, как работал **send\_message**. Разница только в том, что вместо теста мы отправляем изображение. В качестве первого параметр мы указываем также **message.chat.id**, а вторым параметром указываем какой файл мы будем отправлять.

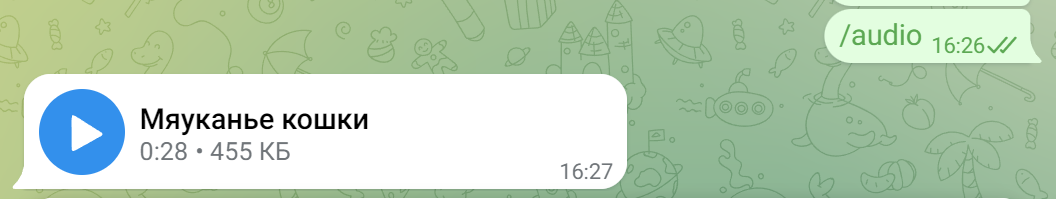


Давайте проверим, что у нас получилось.



Если же мы хотим отправить пользователю аудио файл, то в таком случае мы делаем то же самое, только указываем вместо изображения аудио файл, а также вместо **send\_photo** указываете **send\_audio**. Остальное не будет изменяться.

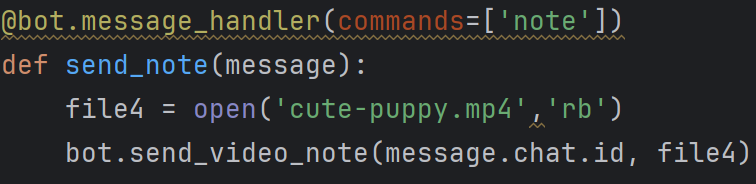




**Самостоятельная работа**

Сделайте так, чтобы бот отправлял видео файлы и текстовые документы. Для этого вы открываете ваш файл (как мы делали это на примере аудио и фото) и указываете команду **send\_video** или **send\_document**.

В приложении Telegram можно отправлять видео в виде кружка. Как это реализовать для нашего бота? Для этого мы сохраняем открытый файл в переменную, как мы это делали и с другими файлами. Далее используем команду **send\_video\_note.**

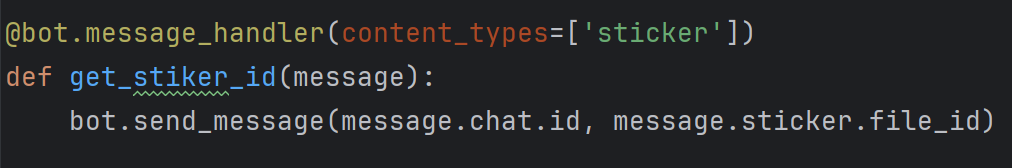




**Работа со стикерами**

Мы переходим к наиболее интересным файлам — стикерам. Отправление стикеров почти ничем не отличается от отправления картинок, видео и аудио. Но вместо файла мы должны указать id (ссылку на стикер, который хотим отправить). А где можно найти эту ссылку? Конечно, в Telegram существуют специальные боты, которые могут возвращать нам id стикера, который мы отправили. Мы научим своего бота отправлять пользователю ссылку на любой стикер.

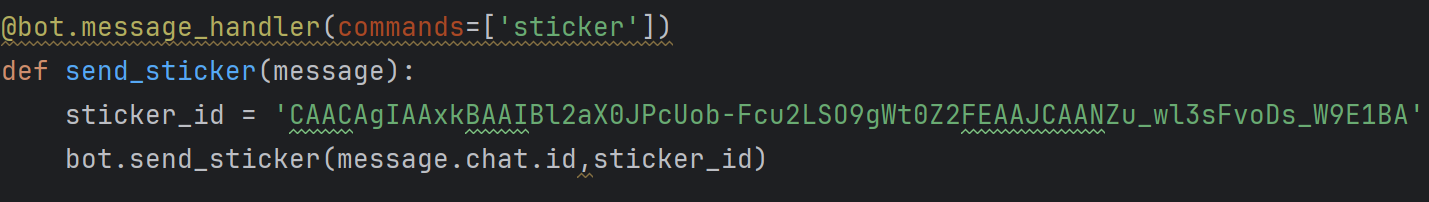
Для начала сделаем так, чтобы бот реагировал на любые сообщения со стикерами. Как это можно сделать? Обращаемся к **content\_types** и внутри указываем тип файла —**'sticker'**. Теперь сделаем так, чтобы бот отправлял пользователю информацию о стикере. Где она может храниться? В параметре **message,** чтобы получить id стикера пишем — **message.sticker.file\_id**



Протестируем программу, для этого необходимо отправить нашему боту любой стикер:



Готово! Теперь мы сможем узнать id стикера. Скопируйте ссылку на стикер, мы сохраним её в переменную. Далее отправляем стикер с помощью команды **send\_sticker**, в параметрах указываем переменную id стикера:





**Самостоятельная работа**

Сделайте reply-кнопки, нажимая на которые пользователь будет получать различные файлы (например, смешные видео и картинки)

**Рефлексия**

* Сегодня мы научили нашего бота отвечать на различные файлы, отправленные пользователем
* Сделали так, чтобы бот отправлял изображения, видео, аудио, текстовые документы, стикеры
* Научились узнавать id стикера с помощью бота

***Тайминг.***

|  |  |  |
| --- | --- | --- |
|  | Раздел урока | Время с начала урока, мин. |
|  | Повторение материала предыдущего урока. Постановка цели урока | 0 |
|  | Получение и ответ на файлы пользователя | 10 |
|  | Самостоятельная работа | 25 |
|  | Отправка различных файлов пользователю | 35 |
|  | Самостоятельная работа | 40 |
|  | Бескомпьютерная деятельность | 55 |
|  | Работа со стикерами | 60 |
|  | Самостоятельная работа | 70 |
|  | Рефлексия | 85 |