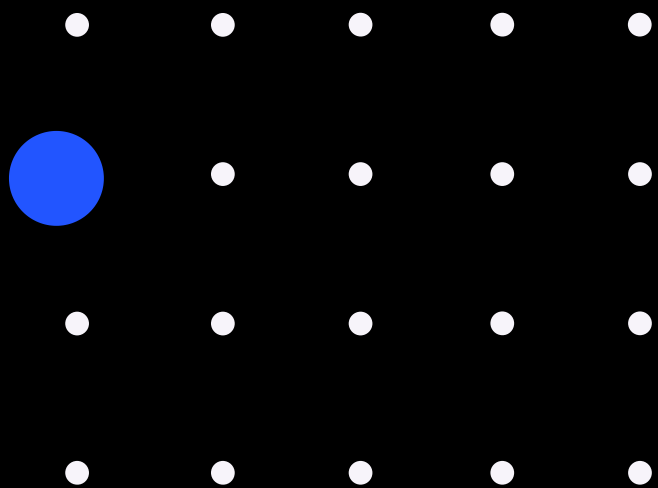


Как создать свой
чат-бот Telegram?



Что вам понадобится?

- Python 3.0
- TeleBot

Позволяет взаимодействовать с ботом, настраивать сообщения, ответы, реакции бота на определенные действия, кнопки.

С чего начнем?

1. Находим в Telegram бота BotFather. В нем мы регистрируемся и задаем имя нашему боту. На выходе BotFather дает нам токен (подчеркнуто красным в тексте) нашего бота (в дальнейшем мы будем его использовать в коде для контроля)

The screenshot shows a Telegram chat interface with BotFather. The chat header shows '796 Чаты' and 'BotFather бот'. The chat history includes:

- A message from BotFather: "Что умеет этот бот? BotFather is the one bot to rule them all. Use it to create new bot accounts and manage your existing bots. About Telegram bots: <https://core.telegram.org/bots> Bot API manual: <https://core.telegram.org/bots/api> Contact @BotSupport if you have questions about the Bot API."
- A user message: `/deletegame` - delete an existing game
- A user message: `/newbot`
- A BotFather response: "Alright, a new bot. How are we going to call it? Please choose a name for your bot."
- A user message: TestBot
- A BotFather response: "Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot."
- A user message: VichkaMmf2021bot
- A BotFather response: "Done! Congratulations on your new bot. You will find it at t.me/VichkaMmf2021bot. You can now add a description, about section and profile picture for your bot, see `/help` for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this. Use this token to access the HTTP API: 1827009943:AAGZRVLSu9WQ29tZx4XDMmZJZtgGKb_2Lx4 Keep your token **secure** and **store it safely**, it can be used by anyone to control your bot. For a description of the Bot API, see this page: <https://core.telegram.org/bots/api>"
- A user message: `/start`
- A BotFather response: "I can help you create and manage Telegram bots. If you're new to the Bot API, please see the manual. You can control me by sending these commands: `/newbot` - create a new bot `/mybots` - edit your bots [beta] Edit Bots `/setname` - change a bot's name `/setdescription` - change bot description `/setabouttext` - change bot about info `/setuserpic` - change bot profile photo `/setcommands` - change the list of"

2. В нашем проекте создаем файл `config.py`, в котором будет храниться наш токен:

```
1 # -*- coding: utf-8 -*-
2 |
3
4 TOKEN = "1693797099:AAHZ_SV0BJkR7SoIryN8MPwwb7-cpFR2h4Q"
```

3. Создаем главный файл нашего проекта, который называем, например `bot.py`. В него мы импортируем библиотеку `telebot` и файл `config` (созданный ранее). Далее создаем переменную `bot`, через которую будем управлять нашим ботом:

```
9 import config
10 import telebot
11 FEED_EXPORT_ENCODING = 'utf-8'
12
13
14 bot = telebot.TeleBot(config.TOKEN)
```

Эту переменную мы создаем для того, чтобы каждый раз не прописывать `telebot.TeleBot()` с нашим токеном.

Строка `FEED_EXPORT_ENCODING = 'utf-8'` нужна нам для того, чтобы могли работать с кодировкой `utf-8`

4. Далее добавляем обработчик команды запуска (`/start`), которое будет выводить приветственное сообщение и выводит меню бота с кнопками.

```
16 @bot.message_handler(commands=['start'])
17 def start_handler(message):
18     msg = 'Hallo, nice to meet you'
19     msg_out = bot.send_message(message.chat.id, msg)
20     menu.main_menu(bot, message.chat.id)
```

- строка 16 - обработчик команды `/start`
- строка 17 - создаем функцию, которая запускается при команде `start`
- строка 19 - выводит наше сообщение (задаем его в строке 18)
- строка 20 - `main_menu` (хранится в файле `menu`, о нем будет позже) выводит меню с кнопками

• • • • •

• • • • •

• • • • •

• • • • •

Создаем кнопки для нашего бота

5. Создаем файл `menu.py`, в котором будут находиться команды вывода различных кнопок для различных действий.

```
1 import telebot
2 import json
3
4 def main_menu(bot, chat_id):
5     main_menu_buttons = telebot.types.ReplyKeyboardMarkup(row_width=1,
6                                                             resize_keyboard=True)
7
8     hello_button = telebot.types.KeyboardButton('Hello')
9     how_button = telebot.types.KeyboardButton('How are you?')
10    cute_button = telebot.types.KeyboardButton('You are cute')
11    lol_button = telebot.types.KeyboardButton('Register')
12    new_button = telebot.types.KeyboardButton('Send message')
13
14    main_menu_buttons.add(hello_button, how_button, cute_button,
15                          lol_button, new_button)
16    bot.send_message(chat_id, 'XD',
17                    reply_markup=main_menu_buttons)
```

- строка 2 - импортируем библиотеку `json` для того, чтобы хранить записи о пользователях
- строка 4 - создаем функцию, которая отвечает за меню и его кнопки
- строка 5 - задаем размеры для наших кнопок (`row_width=1`). `resize_keyboard` - убирает клавиатуру пользователя, если он открывает меню.
- строки 8-12 - задаем кнопки и называем их
- строка 14 - загружаем все кнопки в меню
- строка 16 - выводим наше меню

Кнопки не вызывают никаких функций, они только отправляют текст за пользователя.

6. Далее добавляем в исходный файл (bot.py) `import menu` (файл, в котором хранятся наши кнопки).

```
25 @bot.message_handler(content_types=['text'])
26 def text_handler(message):
27     try:
28         if hasattr(message.reply_to_message, 'text'):
29             msg = f'сообщение: ' + message.reply_to_message.text + 'ответ: ' + message.text + f'Send me this number: ' \
30                 f'{random.randint(0, 10)} '
31             msg=message.reply_to_message.text.translate(coding("dec",int(message.text)))
32             msg_out = bot.send_message(message.chat.id, msg)
33         else:
34             if message.text == 'Hello':
35                 hello_out = greetings.hello_out(bot, message.chat.id)
36                 bot.send_sticker(message.chat.id,
37                                 'CAACAgIAAxkBAAECGmdgXD9IAQdmC9xsEuaei6XWWSfbbgAC9QADe04qEEI1AAFA$1pyZh4E')
38             elif message.text == 'Register':
39                 global che
40                 che = random.randint(0, 10)
41                 msg=f'Send me this number: {che}'
42                 msg_out=bot.send_message(message.chat.id, msg)
43                 bot.register_next_step_handler(msg_out, ask_nick)
44             elif message.text == 'How are you?':
45                 hello_out = greetings.how_out(bot, message.chat.id)
46                 bot.send_sticker(message.chat.id,
47                                 'CAACAgIAAxkBAAECGmlgXEARqf8ZbQEByBYAAcq8u81INxMAAhYCAALsxmEPqfUh5GhpX1EeBA')
48             elif message.text == 'You are cute':
49                 hello_out = greetings.cute_out(bot, message.chat.id)
50                 bot.send_sticker(message.chat.id,
51                                 'CAACAgIAAxkBAAECGmtgXEDM7Sg3Gu6eN8S0o3SbQ-n1wwACkwADE2HxBzpVc-VYZqd6HgQ')
52             elif message.text == 'LoL':
53                 hello_out = greetings.loL_out(bot, message.chat.id)
54                 bot.send_sticker(message.chat.id,
55                                 'CAACAgIAAxkBAAECGm9gXEEuEqJMA1HFkynenVnYssS3rgACsGADYmSaJFnWvDC0YiAHHgQ')
56             elif message.text == 'Send message':
57                 msg = 'Enter message'
58                 msg_out = bot.send_message(message.chat.id, msg)
59                 bot.register_next_step_handler(msg_out, ask_who)
60             else:
61                 bot.send_sticker(message.chat.id,
62                                 'CAACAgIAAxkBAAECGpFgXFtHscdGzzVeKp7UIh-kVr6aMgACugADwPsIAAGSbw-9i6NJSx4E')
63     except:
64         bot.send_sticker(message.chat.id,
65                         'CAACAgIAAxkBAAECGpFgXFtHscdGzzVeKp7UIh-kVr6aMgACugADwPsIAAGSbw-9i6NJSx4E')
```

- строка 25 - отвечает за реакцию на сообщения пользователя
- строка 26 - создаем функцию, в которой описываем реакции бота на сообщения пользователя

Во всех переменных `msg` будет задаваться текст сообщения, который мы хотим вывести.

Во всех переменных `msg_out` будет выводиться сообщение пользователя.

Как отправлять сообщения

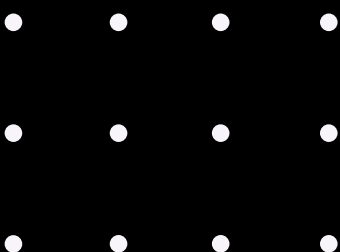
7. Создаем переменную с сообщением, которое хотим отправить.

Через команду `bot.send_message` отправляем сообщение. Первый аргумент - чат, в который надо отправить; второй - само сообщение).

```
msg = 'Enter message'
msg_out = bot.send_message(message.chat.id, msg)
```

Третья строка отвечает за функцию, которую мы вызываем. Ей на вход идет сообщение пользователя. Она будет реагировать на него. Функция `ask_who` будет описана ниже.

```
msg = 'Enter message'
msg_out = bot.send_message(message.chat.id, msg)
bot.register_next_step_handler(msg_out, ask_who)
```



8. Опишем функцию `ask_who` (данная функция была создана для бота шифрования сообщений, однако по примеру вы можете создать любую функцию для вашего бота):

```
67 def ask_who(message):
68     try:
69         mes = message.text
70         if mes == '/Cancel' or mes == '/cancel':
71             msg = 'Sending canceled'
72             msg_out = bot.send_message(message.chat.id, msg)
73         else:
74             global message_text
75             message_text = mes
76             msg = 'Enter recipient'
77             msg_out = bot.send_message(message.chat.id, msg)
78             bot.register_next_step_handler(msg_out, ask_code)
79     except:
80         bot.send_sticker(message.chat.id,
81                         'CAACAgIAAxkBAAEC6pFgXFtHscdGzzVeKp7UIh-kVr6aMgACugADwPsIAAGSbw-9i6NJSx4E')
```

- строка 67 - на вход наша функция получает сообщение пользователя
- строка 69 - в переменную `mes` заносим текст сообщения.
- строка 70 - проверяем, хочет ли отменить отправку сообщения
- строка 78 - `register_next_handler` нужна для вызова другой функции

9. `bot.polling()` всегда должна быть в конце главного файла (`bot.py`), она будет зацикливать весь код, что выше нее. Благодаря ней бот не будет выключаться.

```
217 # always the last
218 bot.polling()
```