

Программирование на Python

Презентация занятия

Проект: Создание Telegram-бота. Часть 1.

23 занятие



инжинириум[®]

МГТУ им. Н.Э. Баумана

2019

Тема: Проект: Создание Telegram-бота. Часть 1.

1. БОТЫ

1.1 Виды ботов

- Чат-боты
- Игровые боты
- Биржевые боты
- Веб-боты

1.2 Задача ботов

Главная задача бота - имитация действий человека в узкой сфере деятельности.



Тема: Проект: Создание Telegram-бота. Часть 1.

2. ЧАТ-БОТЫ

2.1 Для чего нужны?

2.2 Что неудобно?



Тема: Проект: Создание Telegram-бота. Часть 1.

2. ЧАТ-БОТЫ

2.1 Для чего нужны?

Боты нужны для экономии ресурсов человека и оптимизации работы.

Бот может:

- функционировать 24/7
- обрабатывать сотни запросов одновременно от разных пользователей
- ему не нужно платить зарплату
- его не нужно кормить 😊

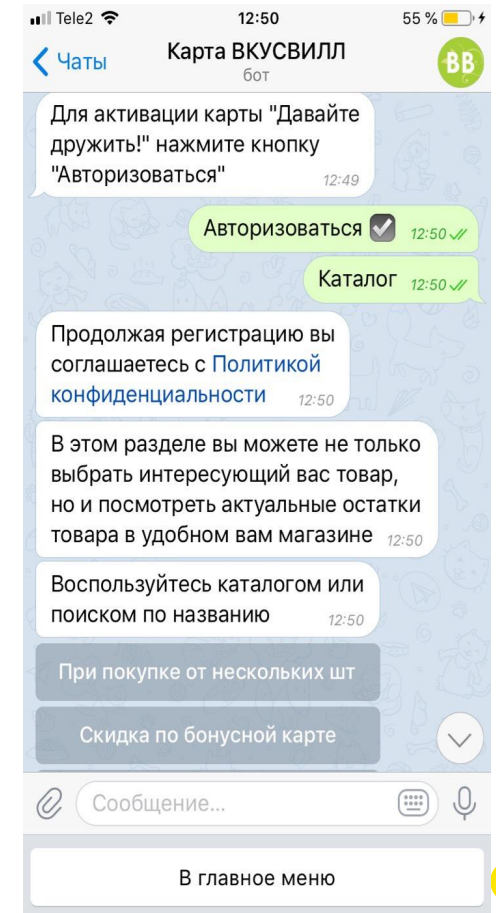
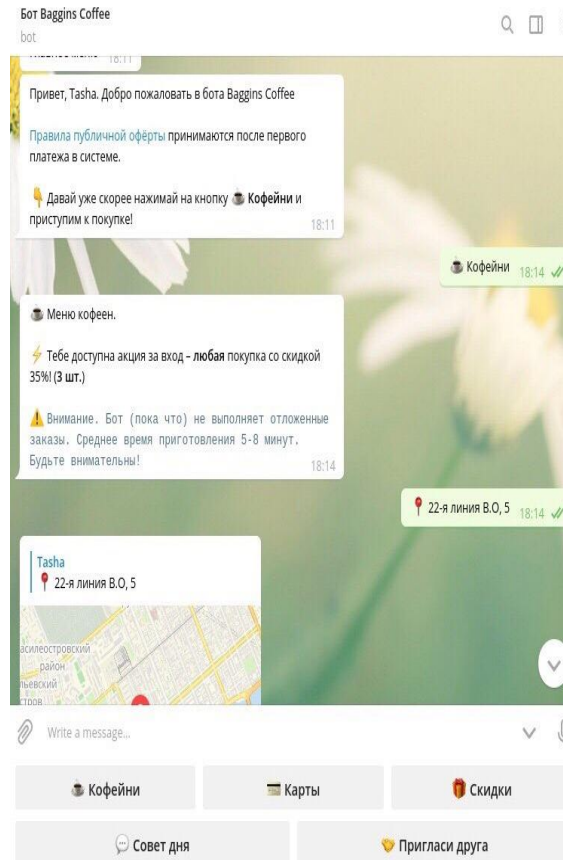
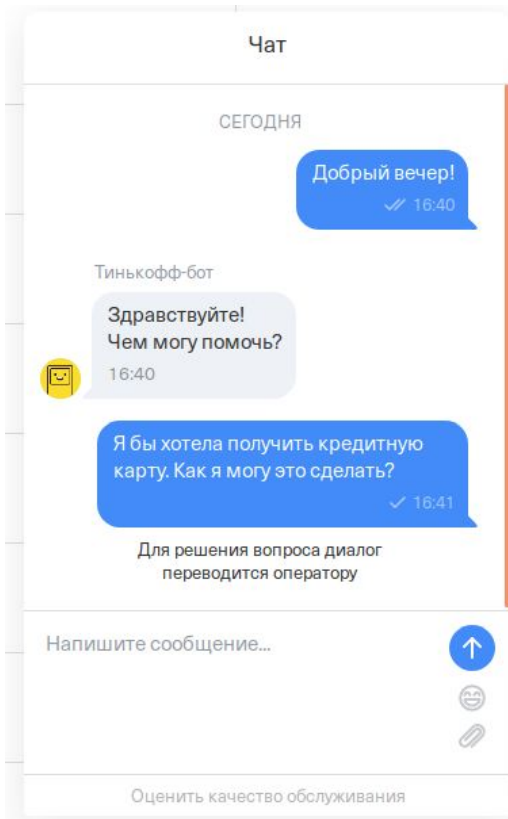
2.2 Что неудобно?

Чат-боты требуют четко сформулированного сценарий для выполнения своих действий.



Тема: Проект: Создание Telegram-бота. Часть 1.

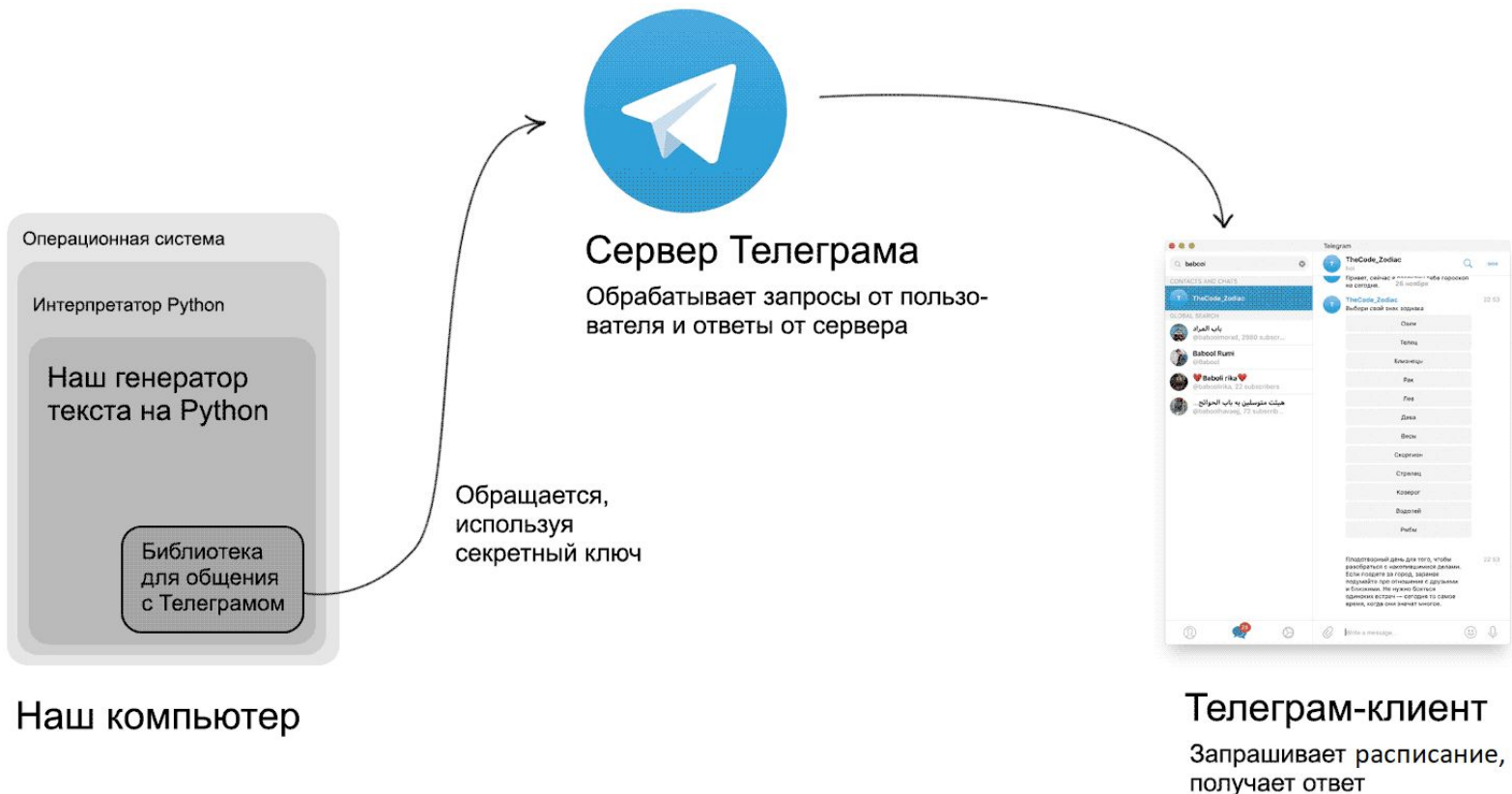
2.3 Виды чат-ботов



Тема: Проект: Создание Telegram-бота. Часть 1.

3. API

3.1 Как все будет работать?



Тема: Проект: Создание Telegram-бота. Часть 1.

3.2 Что это?

API (Application Programming Interface) - однозначно описанные способы, методы и функции, позволяющие одной программе взаимодействовать с другой.

Telegram API Bot – это программный интерфейс, позволяющий программировать собственного бота.

API включает в себя объекты и команды, предназначенные для установки поведения бота Telegram.



Тема: Проект: Создание Telegram-бота. Часть 1.

Важно понимать, что:

- Боты - это сторонние приложения, которые запускаются внутри Telegram
- Пользователи могут взаимодействовать с ботами, отправляя им сообщения, команды и встроенные запросы
- Вы управляете своими ботами, используя HTTPS-запросы к Telegram Bot API
- **GET** — запрашивает контент из сервера.
- **POST** — публикует информацию на сервере.



Тема: Проект: Создание Telegram-бота. Часть 1.

4. ЧТО БУДЕМ ДЕЛАТЬ

План:

1. Регистрируем бота в Telegram
2. Настраиваем бота
3. Устанавливаем Python-библиотеку для работы с API Telegram и VPN
4. Учим программу реагировать на сообщения в чате
5. Пишем меню для выбора дня недели
6. Сделаем так, чтобы при нажатии на кнопку отображалось соответствующее расписание на день
7. Пишем inline-кнопки для отображения времени занятий



Тема: Проект: Создание Telegram-бота. Часть 1.

5. СОЗДАНИЕ БОТА

5.1 РЕГИСТРАЦИЯ БОТА

- В поиске telegram находим **@BotFather** и начинаем работу с помощью команды **/start**
- С помощью команды **/newbot** создаем своего бота

- Вводим **name** и **username**.

Обратите внимание, что username должен оканчиваться на bot!

- В ответ мы получим сообщение с токеном бота и ссылкой на документацию.

Токен нужно сохранить, желательно надёжно, так как это единственный ключ для авторизации бота и взаимодействия с ним.



Тема: Проект: Создание Telegram-бота. Часть 1.

5.2 НАСТРОЙКА БОТА

- В этом же диалоге мы можем поменять все необходимые настройки для нашего бота
- Мы можем настроить внешний вид:
 - Поставить главную фотографию
 - Задать описание и т.д.
- При любом вопросе можно набрать команду **/help**

Далее переходим к нашему боту и видим, что первая команда – **/start**
Хм, но пока он что-то молчит...



Тема: Проект: Создание Telegram-бота. Часть 1.

5.3 ПОДГОТОВКА К НАПИСАНИЮ КОДА

1. Устанавливаем виртуальное окружение

```
>> virtualenv PRG1  
>> PRG1\Scripts\activate.bat
```

2. Устанавливаем библиотеку pytelegrambotapi

```
>> pip install pytelegrambotapi
```

3. Устанавливаем VPN

VPN Betternet (например)



Тема: Проект: Создание Telegram-бота. Часть 1.

5.3 ПОДГОТОВКА К НАПИСАНИЮ КОДА

Первое, что нужно сделать это импортировать нашу библиотеку и подключить токен бота:

```
import telebot  
bot = telebot.TeleBot('%ваш токен%')
```



Тема: Проект: Создание Telegram-бота. Часть 1.

5.4 РЕАГИРУЕМ НА СООБЩЕНИЯ Обработчик команды start

```
import telebot

bot = telebot.TeleBot('%ВАШ ТОКЕН%')

@bot.message_handler(commands=['start'])
def welcome(message):
    bot.reply_to(message, message.text)
    # bot.send_message(message.chat.id, "Привет, я могу показывать расписание!")
```

И последнее, что нам осталось сделать до запуска, — добавить после метода такую строчку:

```
bot.polling(none_stop=True)
```



Тема: Проект: Создание Telegram-бота. Часть 1.

Реагируем на разные типы сообщений

Здесь мы **слушаем** текстовые данные

```
@bot.message_handler(content_types=['text'])
def welcome(message):
    bot.send_message(message.chat.id, message.text)
```

Здесь мы **слушаем** сообщения, где есть документы или голосовые сообщения

```
@bot.message_handler(content_types=['document', 'voice'])
def welcome(message):
    bot.send_message(message.chat.id, "Я не понимаю, что ты хочешь сказать")
```



Тема: Проект: Создание Telegram-бота. Часть 1.

Реагируем на разные типы сообщений

Значения `content_type`:

The Message object also has a `content_type` attribute, which defines the type of the Message. `content_type` can be one of the following strings: `text`, `audio`, `document`, `photo`, `sticker`, `video`, `video_note`, `voice`, `location`, `contact`, `new_chat_members`, `left_chat_member`, `new_chat_title`, `new_chat_photo`, `delete_chat_photo`, `group_chat_created`, `supergroup_chat_created`, `channel_chat_created`, `migrate_to_chat_id`, `migrate_from_chat_id`, `pinned_message`.

Ссылка на документацию:

<https://github.com/eternnoir/pyTelegramBotAPI/>



Тема: Проект: Создание Telegram-бота. Часть 1.

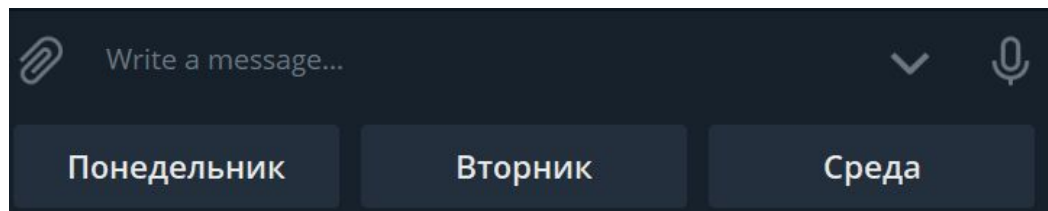
5.5 ДОБАВЛЯЕМ КНОПКИ

1. Меню снизу

Добавление

```
from telebot import types
# resize_keyboard' => true, клавиатура будет сжата в размерах
@bot.message_handler(commands=['start'])
def welcome(message):
    # keyboard bottom
    menu = types.ReplyKeyboardMarkup(resize_keyboard=True)
    button1 = types.KeyboardButton("Понедельник")
    button2 = types.KeyboardButton("Вторник")
    button3 = types.KeyboardButton("Среда")

    menu.add(button1, button2, button3)
    bot.send_message(message.chat.id, "Привет, я могу показывать расписание! "
    | "Выбери день недели.", reply_markup=menu)
```



Тема: Проект: Создание Telegram-бота. Часть 1.

1. Меню снизу Обработка ответов

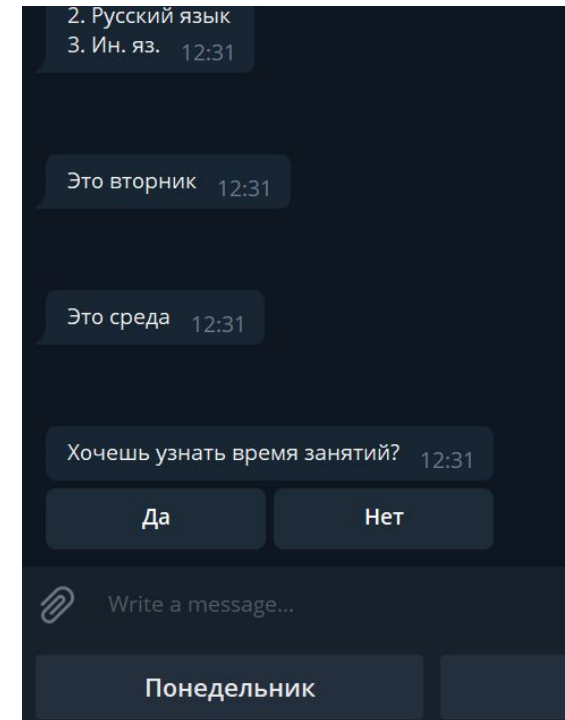
```
@bot.message_handler(content_types=['text', 'document', 'audio'])
def get_messages(message):
    if message.chat.type == 'private':
        if (message.text == 'Понедельник'):
            bot.send_message(message.chat.id, '1. Математика \n2. Русский язык \n3. Ин. яз.')
        elif (message.text == 'Вторник'):
            bot.send_message(message.chat.id, 'Это вторник')
        elif (message.text == 'Среда'):
            bot.send_message(message.chat.id, 'Это среда')
        else:
            bot.send_message(message.chat.id, 'Выбери день недели.')
```



Тема: Проект: Создание Telegram-бота. Часть 1.

2. Inline кнопки Добавление

```
elif (message.text == 'Среда'):  
    bot.send_message(message.chat.id, 'Это среда')  
else:  
    inline = types.InlineKeyboardMarkup(row_width=2)  
    button1 = types.InlineKeyboardButton("Да", callback_data="yes")  
    button2 = types.InlineKeyboardButton("Нет", callback_data="no")  
  
    inline.add(button1, button2)  
  
    bot.send_message(message.chat.id, "Хочешь узнать время занятий?",  
                    reply_markup=inline)
```



Если мы сейчас запустим программу, кнопки появятся, но ответа никакого не последует

Тема: Проект: Создание Telegram-бота. Часть 1.

2. Inline кнопки Обработка ответов

Здесь мы будем слушать и отвечать на все кнопки inline:

```
@bot.callback_query_handler(func=lambda call: True)
def callback_inline(call):
    if call.data == 'yes':
        bot.send_message(call.message.chat.id, "1. 8:00 - 8:45 \n2. 9:00 - 9:45")
    else:
        bot.send_message(call.message.chat.id, "Ну ладно(")
```

А здесь ответим на кнопку с конкретным значением:

```
@bot.callback_query_handler(func=lambda call: call.data=='yes')
def callback_inline(call):
    bot.send_message(call.message.chat.id, "1. 8:00 - 8:45 \n2. 9:00 - 9:45")
```



Тема: Проект: Создание Telegram-бота. Часть 1.

2. Inline кнопки

Удаление кнопки после ответа

```
@bot.callback_query_handler(func=lambda call: True)
def callback_inline(call):
    if call.data == 'yes':
        bot.send_message(call.message.chat.id, "1. 8:00 - 8:45 \n2. 9:00 - 9:45")
    else:
        bot.send_message(call.message.chat.id, "Ну ладно(")

    #remove inline button
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.message_id,
                          text='А тут поменялся текст', reply_markup=None)
```



Тема: Проект: Создание Telegram-бота. Часть 1.

Рефлексия

1. Чему мы сегодня научились?
2. Что больше всего понравилось?
3. Для чего существует API?
4. Про существование каких запросов мы сегодня узнали?
5. Что было сложнее всего?
6. Что было легче всего?

