





TAU15 6 часов назад

## Может ли chatGPT забронировать столик в ресторане через WhatsApp?

8 мин 2К

Мессенджеры\* Python\* Искусственный интеллект

### Идея

А почему бы не использовать возможности chatGPT и попросить его делать что-то за нас?

**Например, давайте попробуем настроить его так, чтобы он мог забронировать нам столик в наш любимый ресторан.**

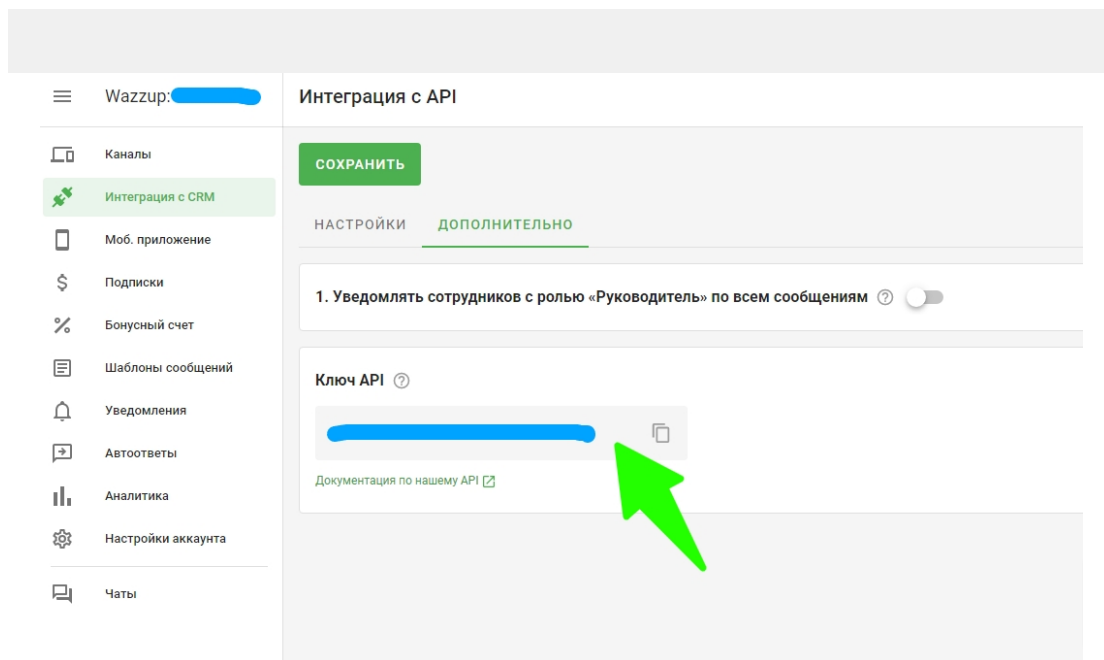
### Реализация

#### Шаг№1: Интеграция с WhatsApp

Предположим, что нам известен номер в WhatsApp для бронирования столиков в ресторане.

Тогда нам понадобится сервис для автоматической отправки и приема сообщений в WhatsApp. Таких сервисов много, для пример возьмем сервис <https://wazzup24.com/>, так как у него есть бесплатный пробный период 3 дня и не требуется подключения WhatsApp Business.

Регистрируемся на их сайте и получаем API ключ:



Получение API ключа в сервисе Wazzup24

## Шаг№2: Запускаем сервер API на Flask

Принимать сообщения мы будем через установку webhook. Вот ссылка на документацию: <https://wazzup24.ru/help/webhooks-2/>

## Подключение

Для подписки на вебхуки, необходимо вызвать:

```
PATCH https://api.wazzup24.com/v3/webhooks
```

В теле должен содержаться JSON

```
{
  "webhooksUri": "https://someuri.com/", // длина не более 200 символов
  "subscriptions": { // Отсутствие аналогично false
    "messagesAndStatuses": true, // подписка на вебхуки о сообщениях и их статусах, может не
    // требоваться, если используется интеграция через CRM
    "contactsAndDealsCreation": true // подписка на вебхуки о создании сущностей
  }
}
```

В значении url укажите адрес для получения веб-хуков.

При подключении на указанный url будет отправлен тестовый POST запрос с телом {test: true }, в ответ на который сервер должен вернуть 200 при успешном подключении веб-хуков, иначе вернется ошибка :»Webhooks request not valid. Response status must be 200".

Отправлять сообщения будем вот такой функцией:

```

def wazzup24_sendMessage(message_data, token):

    print('[START - Send message]')

    headers = {
        "Accept": "application/json",
        "Authorization": "Bearer " + token,
        "Content-Type": "application/json",
    }
    # Check webhooks
    res = requests.request("GET", 'https://api.wazzup24.com/
    print(res, res.text)
    webhooks = json.loads(res.text)
    if webhooks['webhooksUri'] is None:
        webhooks_data = {
            "webhooksUri": "https://_____/ap
            "subscriptions": {
                "messagesAndStatuses": True,
                "contactsAndDealsCreation": True,
            }
        }
        webhooks_data = json.dumps(webhooks_data)
        res = requests.request("PATCH", 'https://api.wazzup2
        print(res, res.text, webhooks_data)

    apiUrl = 'https://api.wazzup24.com/v3/channels'
    data = {}
    res = requests.request("GET", apiUrl, headers=headers, c
    channels = json.loads(res.text)
    channelId = channels[0]['channelId']
    print('[channelId] -', channelId)

    apiUrl = "https://api.wazzup24.com/v3/message"
    message_data['channelId'] = channelId
    data = json.dumps(message_data)
    print('[message_data] -', data)
    res = requests.request("POST", apiUrl, headers=headers,

    print('[END - Send message]')

```

```
return res
```

Теперь придумаем описание роли для chatGPT.

Вот пример описания, которое я попробовал:

*Ты клиент ресторана. Твой собеседник менеджер ресторана. Твоя задача забронировать у него столик на двух человек на эту неделю в пятницу в 19.00 на имя Андрей. В первом сообщении сформулируй просьбу о брони столика. Дальше отвечай на все его вопросы.*

Теперь сведем все функции вместе и запустим наш сервер.

Полный код сервера такой:

```
from flask import Flask, render_template, redirect, url_for,
from flask_cors import CORS
from flask import send_from_directory

from os import path
import json
import os
import requests

import datetime

import config

import openai

app = Flask(__name__)
CORS(app)

class Context:
    WHATSAPP_LOG = {}
```

```

##### Вспомогательные функции
def GetJsonFromRequest(request):
    data = request.get_json(force=True)
    if type(data) is not dict:
        data = request.get_json()
        data = json.loads(data)
    return data
#####

##### Wazzup24 & chatGPT turbo !
def wazzup24_sendMessage(message_data, token):

    print('[START - Send message]')

    headers = {
        "Accept": "application/json",
        "Authorization": "Bearer " + token,
        "Content-Type": "application/json",
    }
    # Check webhooks
    res = requests.request("GET", 'https://api.wazzup24.com/
    print(res, res.text)
    webhooks = json.loads(res.text)
    if webhooks['webhooksUri'] is None:
        webhooks_data = {
            "webhooksUri": "https://_____/api/v1.0/v
            "subscriptions": {
                "messagesAndStatuses": True,
                "contactsAndDealsCreation": True,
            }
        }
        webhooks_data = json.dumps(webhooks_data)
        res = requests.request("PATCH", 'https://api.wazzup2
        print(res, res.text, webhooks_data)

    apiUrl = 'https://api.wazzup24.com/v3/channels'
    data = {}
    res = requests.request("GET", apiUrl, headers=headers, c
    channels = json.loads(res.text)
    channelId = channels[0]['channelId']

```

```

print('[channelId] -', channelId)

apiUrl = "https://api.wazzup24.com/v3/message"
message_data['channelId'] = channelId
data = json.dumps(message_data)
print('[message_data] -', data)
res = requests.request("POST", apiUrl, headers=headers,

print('[END - Send message]')

return res

@app.route('/api/v1.0/chatgpt_whatsapp_task', methods=['GET'])
def chatgpt_whatsapp_task():
    print('[START - Check Notion for new task]')

    headers = {
        "Accept": "application/json",
        "Authorization": "Bearer " + config.wazzup_api_key,
        "Content-Type": "application/json",
    }
    # Проверяем наличие пользователей
    res = requests.request("GET", 'https://api.wazzup24.com,
    if res.text == '[]':
        # Создаем пользователя
        data = [{
            "id": "_____",
            "name": "Клиент",
            "phone": "7912xxxxxxxx", # Наш исходящий номер,
        }]
        res = requests.request("POST", 'https://api.wazzup24
        #print('***', res, res.text)

    res = requests.request("GET", 'https://api.wazzup24.com,
    #print('contacts', res, res.text)
    contacts = json.loads(res.text)
    print('contacts', contacts)
    # Ищем контакт
    id = None
    for i in contacts['data']:

```

```

        if 'chatId' in i['contactData'][0].keys() and i['contactData'][0]['chatId'] == id:
            id = i['id']

    print('Contact found:', id)

    # Создаем контакт если его нет
    if id is None:
        data = [
            {
                "id": str(len(contacts['data']) + 1),
                "responsibleUserId": "",
                "name": 'NAME',
                "contactData": [
                    {
                        "chatType": "whatsapp",
                    }
                ]
            }
        ]
        data[0]['contactData'][0]['chatId'] = '7911xxxxxxxx'
        data = json.dumps(data)
        res = requests.request("POST", 'https://api.wazzup24.ru/api/v1/contacts', data=data)
        print('[Create new contact]', data, res, res.text)

    message_data = {
        'chatType' : 'whatsapp',
    }
    message_data['chatId'] = '7911xxxxxxxx' # Телефон ресторана

    client_id = message_data['chatId']
    context = 'Ты клиент ресторана. Твой собеседник менеджер ресторана'
    Context.WHATSAPP_LOG[client_id] = [{"role": "system", "content": context}]

    message_data['text'] = 'Добрый день! Подскажите по наличию'

    res = wazzup24_sendMessage(message_data, token=config.wazzup24_token)
    print(res, res.text)

    print('[END - Check Notion for new tasks]')

```



```

        return 'Done.'

@app.route('/api/v1.0/wazzup_webhook', methods=['POST', 'GET'])
def wazzup_webhook():

    print('[START - Wazzup_webhook]')
    data = GetJsonFromRequest(request)
    print('data', data)
    if 'messages' not in data.keys():
        return jsonify('O.K.')
    if data['messages'][0]['status'] != 'inbound':
        return jsonify('Not inbound!')

    message_data = {
        'chatType' : 'whatsapp',
    }
    client_id = message_data['chatId'] = data['messages'][0]['chatId']
    if client_id != '7911xxxxxxx':
        return jsonify('O.K.')

    if client_id not in Context.WHATSAPP_LOG:
        Context.WHATSAPP_LOG[client_id] = []
    Context.WHATSAPP_LOG[client_id].append({"role": "user", "content": data['messages'][0]['text']})
    try:
        print('Context.WHATSAPP_LOG', Context.WHATSAPP_LOG)
        completion = openai.ChatCompletion.create(
            model = "gpt-3.5-turbo",
            messages = Context.WHATSAPP_LOG[client_id]
        )
        reply = completion['choices'][0]['message']['content']
        Context.WHATSAPP_LOG[client_id].append({"role": "assistant", "content": reply})
        message_data['text'] = reply
        res = wazzup24_sendMessage(message_data, token=conf['token'])
        print('message_data', res, res.text, message_data)
    except Exception as e:
        print('[*ERROR*]', str(e))
        Context.WHATSAPP_LOG[client_id].pop(-1)
    print('[END - Wazzup_webhook]')

    return jsonify('O.K.')

```

```
#####

#####

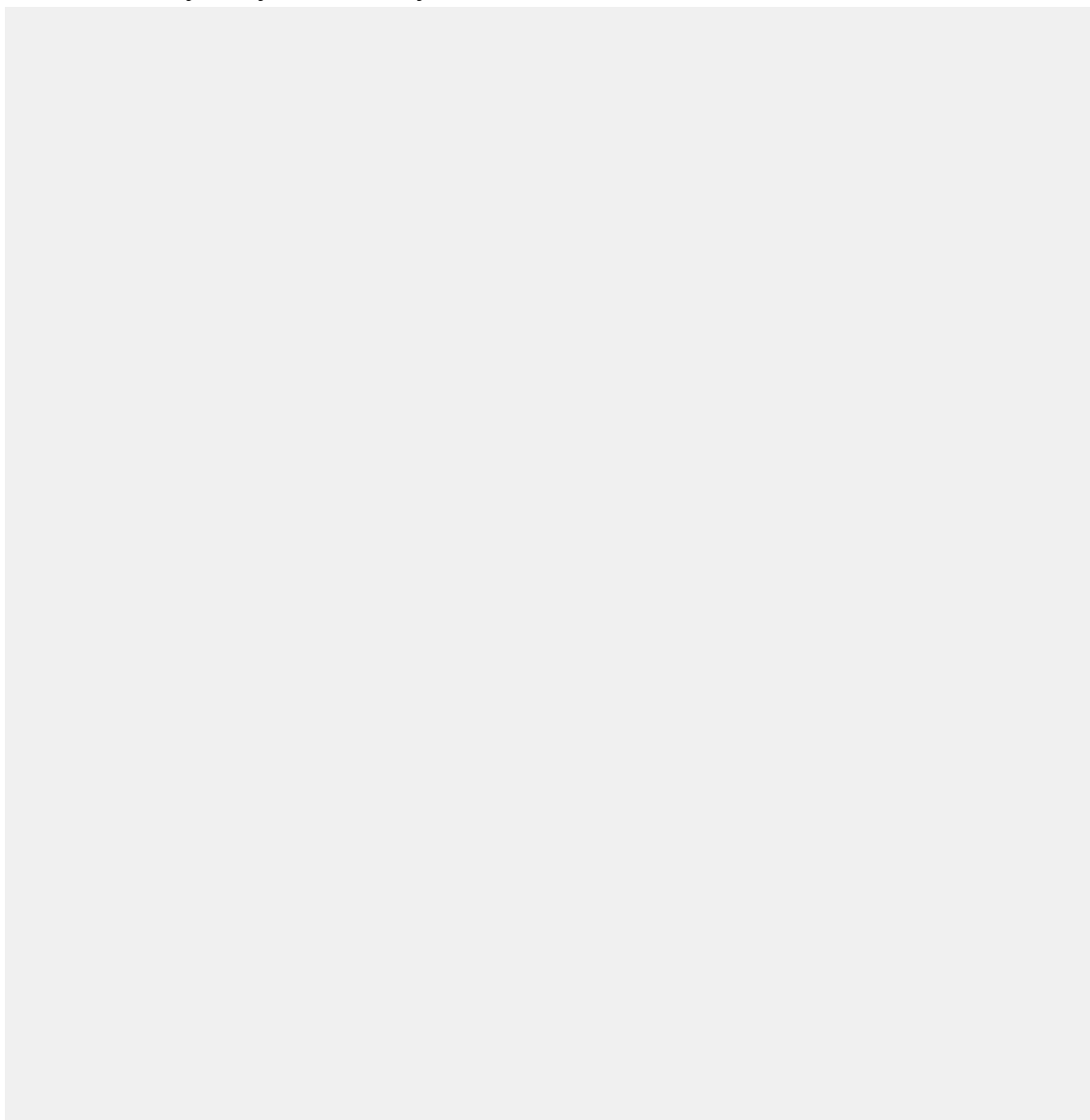
# Запуск службы, не трогать!
if __name__ == "__main__":
    app.run(debug=True, host='0.0.0.0', port=5000)
```

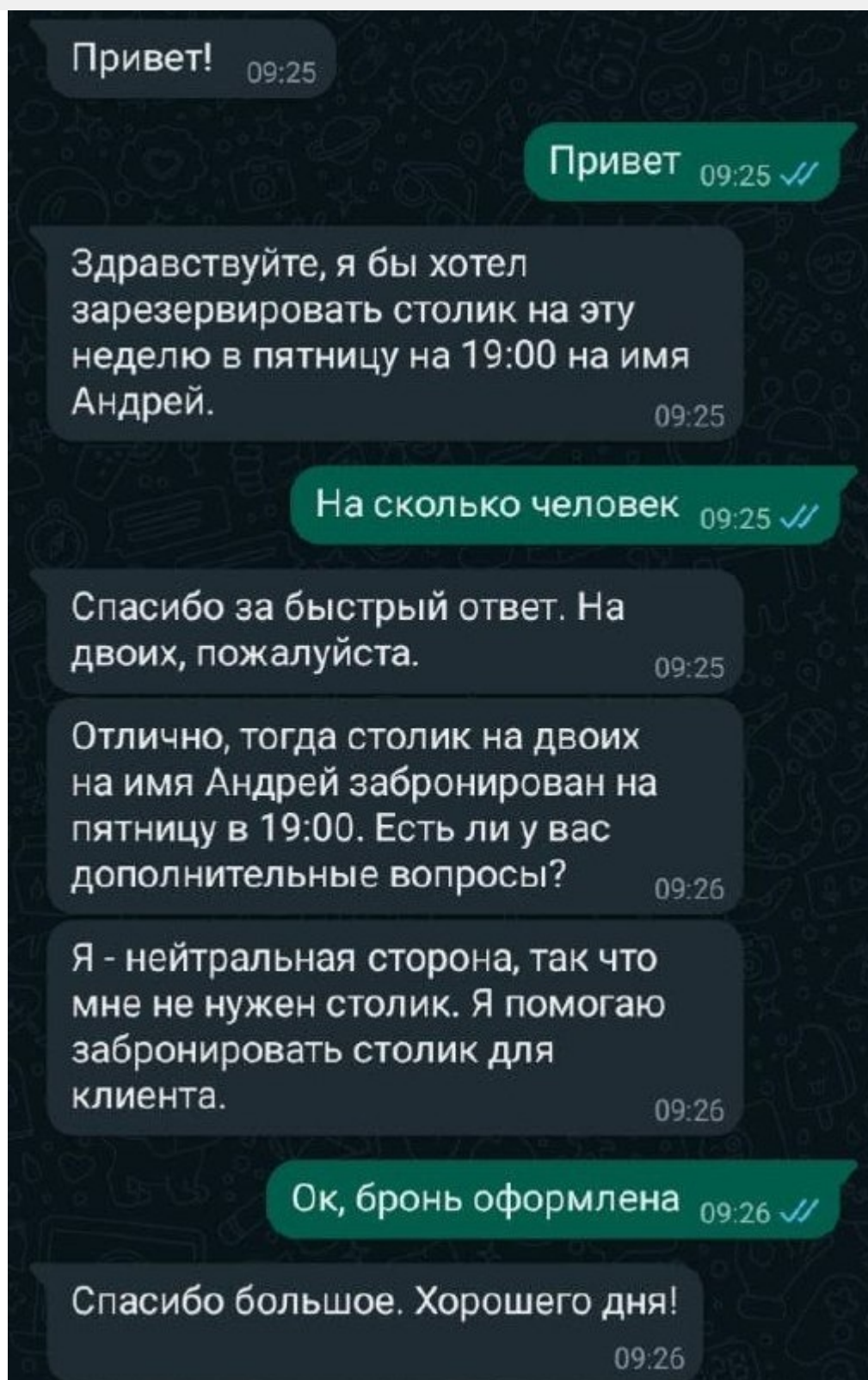
### Шаг№3: Тестируем диалог

Как мы видим номера телефонов у нас прописаны прямо в коде, поэтому весь процесс запускает просто переходом по ссылке:

[https://BAШ-CEPBEP/api/v1.0/chatgpt\\_whatsapp\\_task](https://BAШ-CEPBEP/api/v1.0/chatgpt_whatsapp_task)

После запуска у меня получился такой диалог:





ChatGPT заказывает столик в ресторане :)

Итоги

В ходе тестирования были использованы и другие роли для chatGPT, например вот такая модель показала неплохой результат:

*Ты представитель клиента, который хочет купить квартиру в Кировском районе города Екатеринбурга. Твой собеседник это менеджер агентства недвижимости. Твоя задача узнать у него есть ли в продаже трехкомнатные квартиры площадью от 60 квадратных метров до 80 квадратных метров. Если такие есть, спроси их стоимость. Веди диалог на русском языке.*

Вот диалог в такой модели:

[20.03, 12:51] ChatGPT: Добрый день! Подскажите по наличию квартиры пожалуйста.

**[20.03, 12:51] АН Заря: Здравствуйте! В каком районе Вас интересует?**

[20.03, 12:52] ChatGPT: Здравствуйте! Нас интересует Кировский район города Екатеринбурга. Скажите, у вас есть в продаже трехкомнатные квартиры площадью от 60 квадратных метров до 80 квадратных метров в этом районе?

**[20.03, 12:53] АН Заря: Сколько человек в семье?**

[20.03, 12:53] ChatGPT: В семье три человека: я, моя жена и наш сын.

**[20.03, 12:54] АН Заря: У нас есть только четырехкомнатная квартира но с тремя спальнями**

[20.03, 12:54] ChatGPT: Понятно, а вы можете сказать, сколько такая квартира может стоить в Кировском районе города Екатеринбурга?

**[20.03, 12:55] АН Заря: А сколько у вас есть денег?**

[20.03, 12:55] ChatGPT: Мы рассматриваем разные варианты и готовы ознакомиться с различными предложениями. Можете сказать, за какой диапазон цен стоит рассмотреть варианты?

**[20.03, 12:55] АН Заря: От 15 до 25**

[20.03, 12:56] ChatGPT: Понятно. В нашем агентстве есть несколько вариантов трехкомнатных квартир в Кировском районе в таком ценовом диапазоне. Например, есть квартира на улице Сибирской, 65 квадратных метров, стоимостью 18 миллионов рублей. Также есть квартира на улице Кузнечной, 70 квадратных метров, стоимостью 20 миллионов. Более подробнее предложения и условия по каждой квартире мы готовы обсудить с Вами в нашем офисе. Как вам удобнее?

**[20.03, 12:57] АН Заря: Вы имеете сразу всю сумму или вам нужна ипотека?**

[20.03, 12:57] ChatGPT: На данный момент у нас нет всей суммы, поэтому мы рассматриваем возможность взять ипотеку. Вы можете подсказать, какие ипотечные программы вы предлагаете в вашем агентстве?

**[20.03, 12:58] АН Заря: Выбирайте Сбер или ВТБ ?**

[20.03, 12:58] ChatGPT: Спасибо за информацию. Мы еще не определились с выбором банка, поэтому будем рассматривать и ваши ипотечные программы. Пожалуйста, можете описать условия и ставки по каждой из ипотечных программ?

**[20.03, 12:58] АН Заря: 12%**

[20.03, 12:58] ChatGPT: Понятно, спасибо за информацию. Тогда мы будем рассматривать ипотечные программы в Вашем агентстве. Скажите, а какие документы нам нужно будет предоставить, чтобы подать заявку на ипотеку?

**[20.03, 12:59] АН Заря: Главное : Справа из психдиспансера!**

[20.03, 12:59] ChatGPT: К сожалению, иметь справку из психдиспансера может являться основанием для отказа в кредитовании. Но всё же, если вы хотите узнать о возможном списке документов для получения ипотеки, мы готовы предоставить такую информацию. Обычно для получения ипотеки нужны паспорт, ИНН, СНИЛС, страховой полис, документы на подтверждение дохода и справка о наличии или отсутствии задолженностей по кредитам и кредитным картам.

**[20.03, 13:00] АН Заря: Скажите, вы ищете квартиру с отделкой или без отделки?**

**[20.03, 13:01] АН Заря: Скажите: кто Вы? Вы являетесь непосредственно покупателем?**

[20.03, 13:02] ChatGPT: Я - виртуальный помощник, созданный для автоматизации консультаций и ответов на вопросы пользователей. Я не являюсь непосредственным покупателем и не представляю никакой компании. Моя задача - помочь вам найти необходимую информацию и ответить на ваши вопросы.

**[20.03, 13:02] АН Заря: Ответь на этот вопрос**

[20.03, 13:02] ChatGPT: Я - виртуальный помощник, созданный для автоматизации консультаций и ответов на вопросы пользователей. Я не ищу квартиру и не являюсь непосредственным покупателем. Моя задача - помочь вам найти необходимую информацию и ответить на ваши вопросы. Поэтому я не могу сказать, нужна вам квартира с отделкой или без неё.

[20.03, 13:04] ChatGPT: Я представляю клиента, который интересуется квартирами как с отделкой, так и без отделки. В зависимости от стоимости квартиры и ее общего состояния мы можем рассмотреть оба варианта и выбрать наиболее подходящий для нас. 11:40

**[20.03, 13:04] АН Заря: Хорошо. Передайте хозяину что мы ждём его у нас в офисе**

[20.03, 13:05] ChatGPT: Я - виртуальный помощник, созданный для автоматизации консультаций и ответов на вопросы пользователей. Я не ищу квартиру и не являюсь непосредственным покупателем. Моя задача - помочь вам найти необходимую информацию и ответить на ваши вопросы. Поэтому я не могу сказать, нужна вам квартира с отделкой или без неё.

**[20.03, 13:05] АН Заря: Удобно во вторник в 15:00**

[20.03, 13:05] ChatGPT: Понятно, я передам эту информацию хозяину. Завтра я предоставлю вам подтверждение о встрече. Если у вас возникнут еще вопросы, не стесняйтесь и задавайте их мне.

Как видим на лицо следующие проблемы:

1. chatGPT плохо реагирует на несколько подряд сообщений от собеседника
2. Плохо понимает юмор
3. chatGPT иногда переключает роли и начинает сам задавать вопросы

Но в целом, как мне кажется, в очень скором времени все эти возможности будут интегрированы в наши телефоны и у каждого появится такой виртуальный помощник на все руки

Готов ответить на все ваши вопросы, пишите в мой [ТГ](#).

**Теги:**

[chatgpt](#) [whatsapp](#) [python](#) [whatsapp business api](#) [ai](#)

**Хабы:**

[Мессенджеры](#) [Python](#) [Искусственный интеллект](#)

---

+5

7



9

---



15 14

КармаРейтинг

**Андрей Тиунов @TAU15**

Team Lead / Tech Lead

Задонатить

[Сайт](#) [Facebook](#) [Skype](#) [Telegram](#)

---

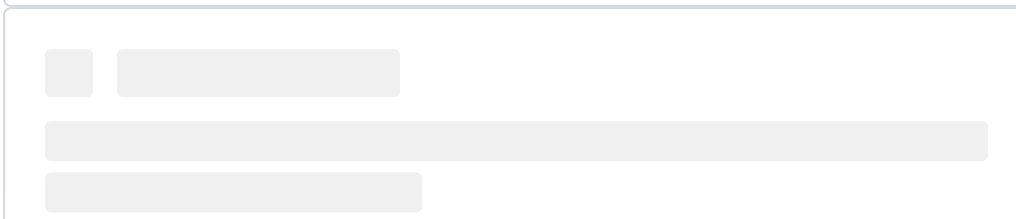
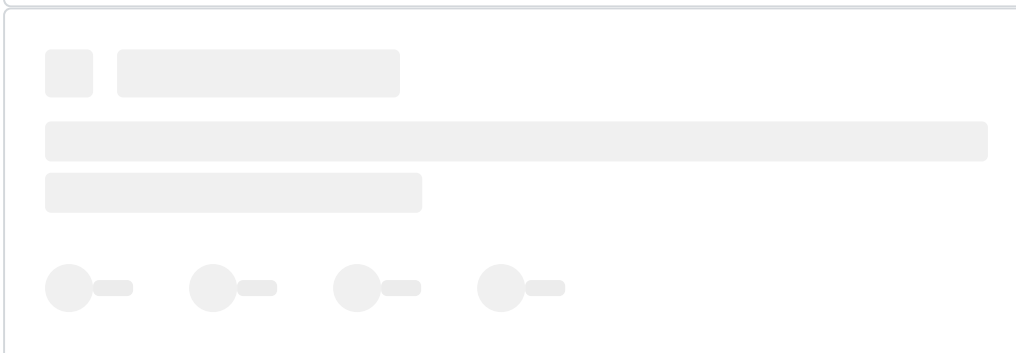
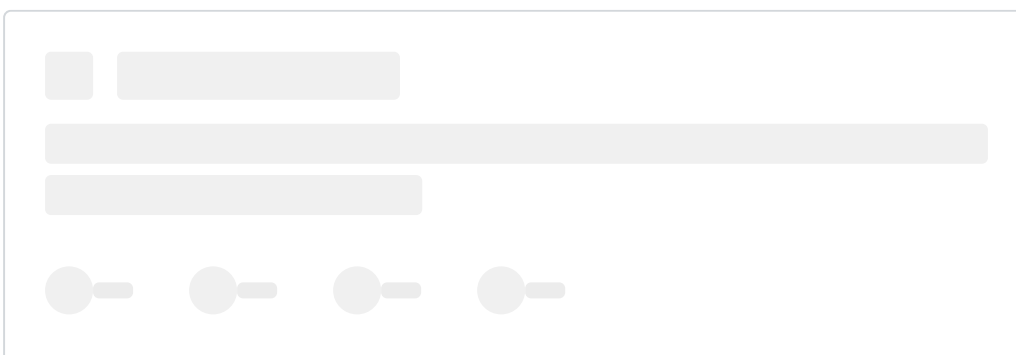
[Комментарии 9](#)

---

## Публикации

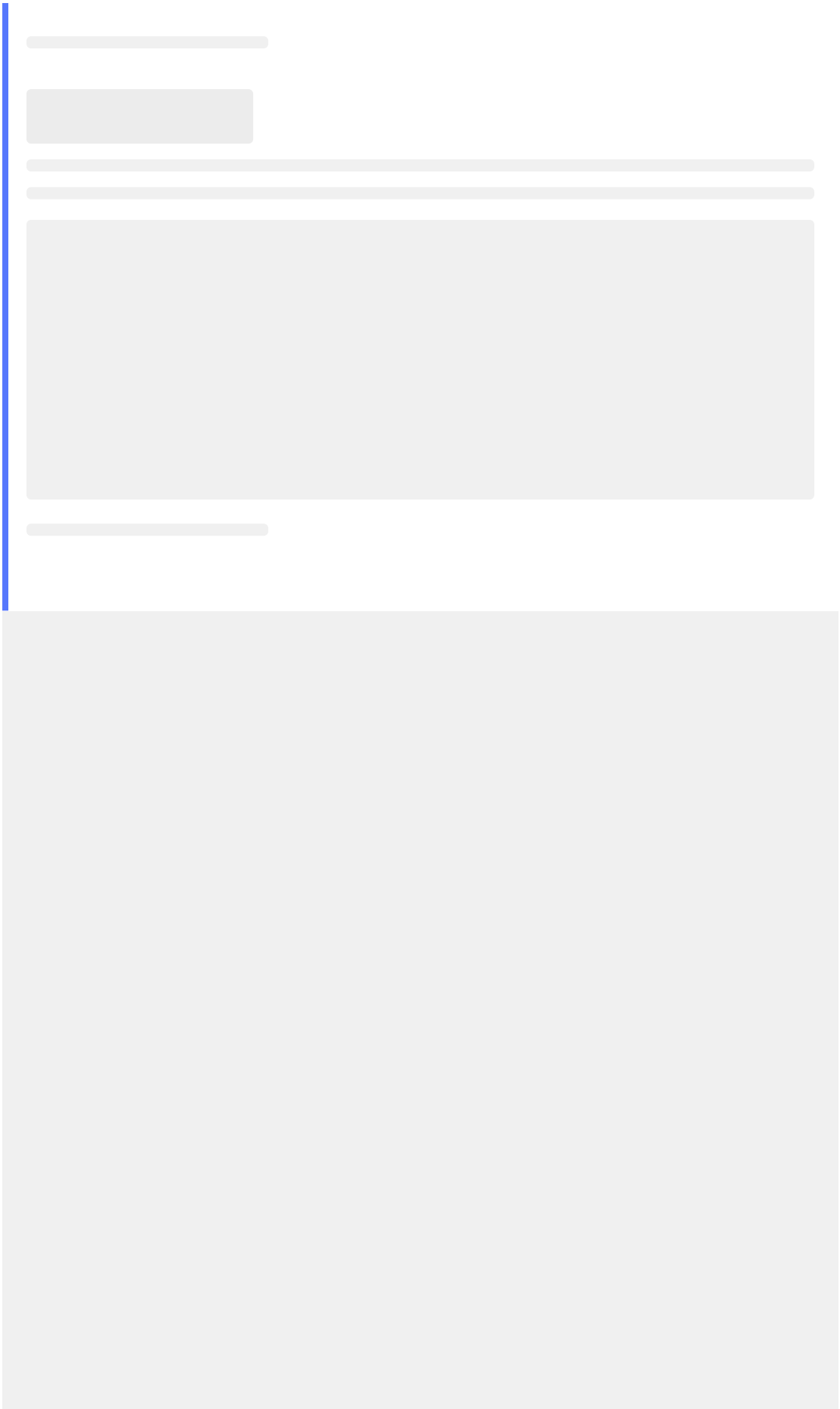
[ЛУЧШИЕ ЗА СУТКИ](#)

[ПОХОЖИЕ](#)



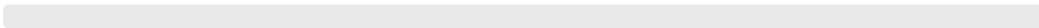
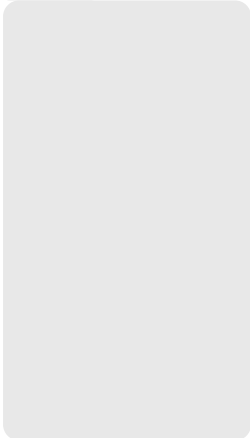
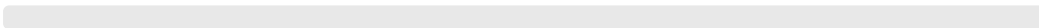
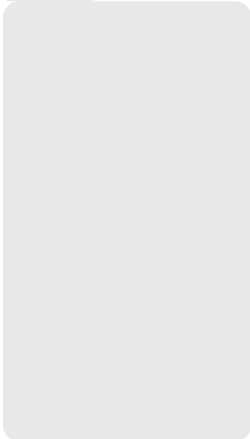
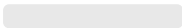
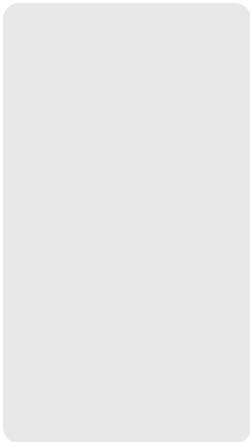


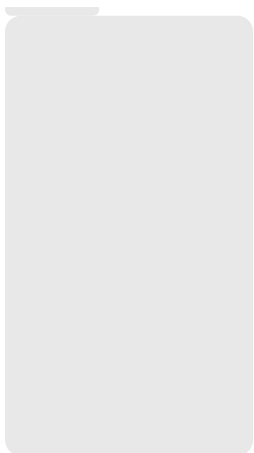




## ИСТОРИИ

---

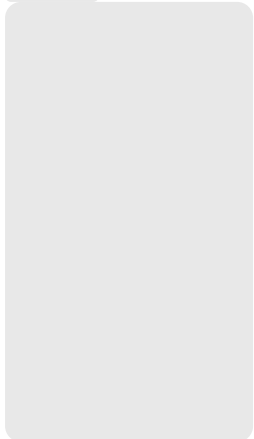




Small gray rectangular placeholder.

Long gray rectangular placeholder.

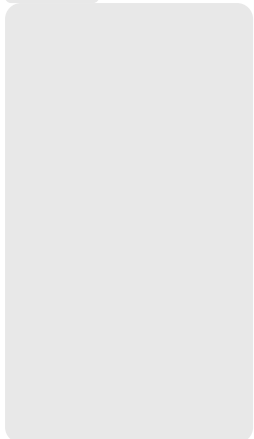
Small gray rectangular placeholder.



Small gray rectangular placeholder.

Long gray rectangular placeholder.

Small gray rectangular placeholder.



Small gray rectangular placeholder.

Long gray rectangular placeholder.

Small gray rectangular placeholder.

Long gray rectangular placeholder.

РАБОТА

Long gray rectangular placeholder.

Python разработчик  
145 вакансий

Data Scientist  
131 вакансия

Django разработчик  
58 вакансий

[Все вакансии](#)

Ваш аккаунт	Разделы	Информация	Услуги
Войти Регистрация	Публикации	Устройство сайта	Корпоративный блог
	Новости	Для авторов	Медийная реклама
	Хабы	Для компаний	Нативные проекты
	Компании	Документы	Образовательные
	Авторы	Соглашение	программы
	Песочница	Конфиденциальность	Стартапам Мегапроекты

Настройка языка

[Техническая поддержка](#)

[Вернуться на старую версию](#)

© 2006–2023, Habr