

Телеграмм Бот Анонимный

чат

Подготовили:
Василий Белоусов
Дроздов Михаил

Актуальность



Человек не может жить вне социума, но в сегодняшнее время интернет слишком опасен из за огромного количества мошенников. Поэтому мы решили создать анонимного чат бота который поможет необразованным пользователям не бояться за свои данные и общаться без мысли о том, что его данные попадут в плохие руки.

```
from telebot import types
from database import Database
db = Database('db.db')
bot = telebot.TeleBot('')

@bot.message_handler(commands=['start'])
def start(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    item = types.KeyboardButton('поиск собеседника')
    markup.add(item)
    name = f'{message.from_user.first_name} {message.from_user.last_name}, добро пожаловать в анонимный чат! Если хочешь найти собеседника нажми на кнопку поиска'
    bot.send_message(message.chat.id, name, reply_markup=markup)

@bot.message_handler(commands=['menu'])
def menu(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    item1 = types.KeyboardButton('поиск собеседника')
    markup.add(item1)
    bot.send_message(message.chat.id, 'меню', reply_markup=markup)

@bot.message_handler(commands=['stop'])
def stop(message):
    chat_info = db.get_active_chat(message.chat.id)
    if chat_info != False:
        db.delete_chat(chat_info[0])
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
        item = types.KeyboardButton('поиск собеседника')
        markup.add(item)
        bot.send_message(chat_info[1], 'собеседник покинул чат', reply_markup=markup)
        bot.send_message(message.chat.id, 'вы вышли из чата', reply_markup=markup)
    else:
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
        item = types.KeyboardButton('поиск собеседника')
        markup.add(item)
        bot.send_message(message.chat.id, 'вы не начали чат', reply_markup=markup)
```

```
@bot.message_handler(content_types=['text'])
def bot_message(message):
    if message.chat.type == 'private':
        if message.text == 'поиск собеседника':
            markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
            item1 = types.KeyboardButton('остановить поиск')
            markup.add(item1)

            chat_two = db.get_chat()
            if db.create_chat(message.chat.id, chat_two) == False:
                db.add_queue(message.chat.id)
                bot.send_message(message.chat.id, 'поиск собеседника...', reply_markup=markup)
            else:
                mess = 'Собеседник найден! Чтобы остановить диалог, напишите /stop'
                markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
                item1 = types.KeyboardButton('/stop')
                markup.add(item1)
                bot.send_message(message.chat.id, mess, reply_markup=markup)
                bot.send_message(chat_two, mess, reply_markup=markup)

        elif message.text == 'остановить поиск':
            db.delete_queue(message.chat.id)
            markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
            item1 = types.KeyboardButton('/menu')
            markup.add(item1)
            bot.send_message(message.chat.id, 'поиск остановлен, перейдите в меню', reply_markup=markup)
        elif db.get_active_chat(message.chat.id) != False:
            chat_info = db.get_active_chat(message.chat.id)
            bot.send_message(chat_info[1], message.text)
        else:
            bot.send_message(message.chat.id, 'Ты думал здесь что то будет? 0-0-0 не не не. Попробуй ещё раз')

bot.polling(none_stop=True)
```

```
import sqlite3
class Database:
    def __init__(self, database_file):
        self.connection = sqlite3.connect(database_file, check_same_thread = False)
        self.cursor = self.connection.cursor()

    def add_queue(self, chat_id):
        with self.connection:
            return self.cursor.execute("INSERT INTO 'queue' ('chat_id') VALUES (?)", (chat_id,))

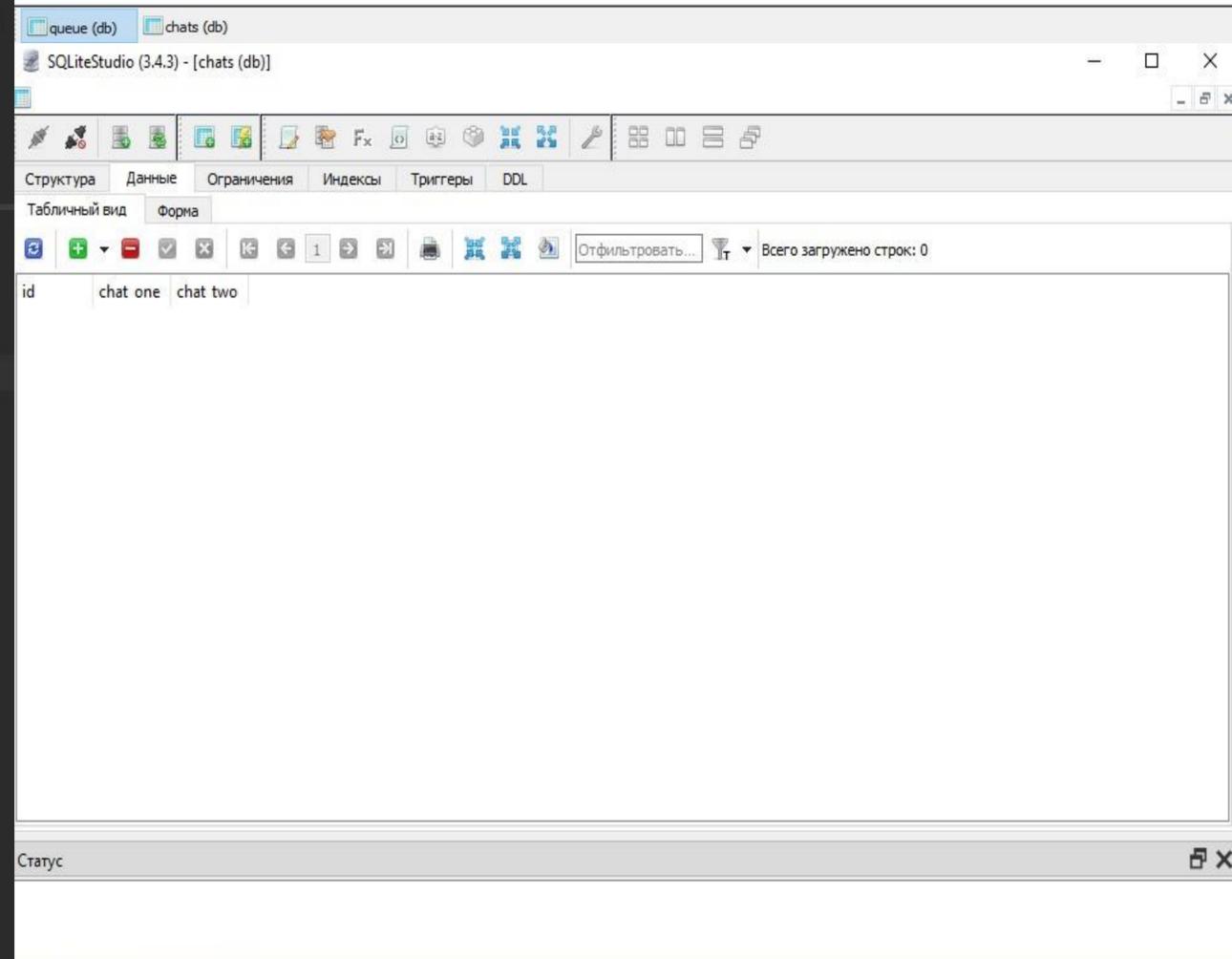
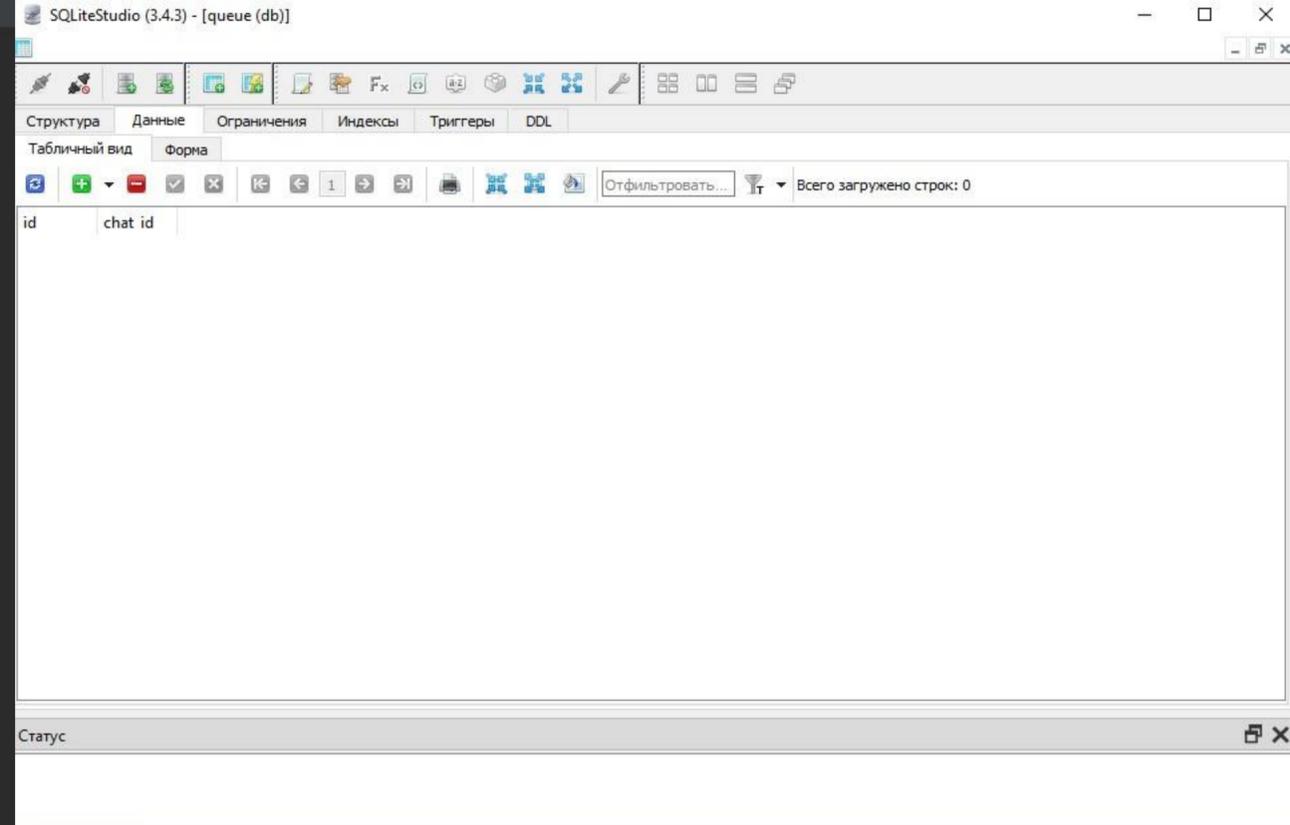
    def delete_queue(self, chat_id):
        with self.connection:
            return self.cursor.execute("DELETE from 'queue' where chat_id = ?", (chat_id,))

    def delete_chat(self, id_chat):
        with self.connection:
            return self.cursor.execute("DELETE from 'chats' where id = ? ", (id_chat,))

    def get_chat(self):
        with self.connection:
            chat = self.cursor.execute("SELECT * from 'queue'",()).fetchmany(1)
            if bool(len(chat)):
                for row in chat:
                    return row[1]
            else:
                return False

    def create_chat(self, chat_one, chat_two):
        with self.connection:
            chat_two != False
            #print(chat_two)
            try:
                if chat_two != False:
                    self.cursor.execute("DELETE from 'queue' WHERE chat_id = ?", (chat_two,))
                    self.cursor.execute("INSERT INTO 'chats' ('chat_one', 'chat_two') VALUES (?,?)", (chat_one, chat_two,))
                    return True
                else:
                    return False
            except:
                return False
```

```
def get_active_chat(self, chat_id):
    with self.connection:
        chat = self.cursor.execute("SELECT * from 'chats' where chat_one=?", (chat_id,))
        id_chat = 0
        for row in chat:
            id_chat = row[0]
            chat_info = [row[0], row[2]]
        if id_chat == 0:
            chat = self.cursor.execute("SELECT * from 'chats' where chat_two=?", (chat_id,))
            for row in chat:
                id_chat = row[0]
                chat_info = [row[0], row[1]]
            if id_chat == 0:
                return False
            else:
                return chat_info
        else:
            return chat_info
```



Чтобы мы добавили?



01

Улучшить систему
очереди

03

Поиск по общим темам

02

Возможность
использовать
стикеры

04

Добавление мини-игр
между
пользователями

**Спасибо за
внимание**

@vaca2004bot

