\*\*\*BOT.py\*\*\*: Отключить для языка: английский

import discord

import asyncio

import random

import time

import os

from copy import copy

from html.parser import HTMLParser

from discord.errors import ClientException, NoMoreItems

from discord.player import FFmpegPCMAudio

TOKEN = ‘токен бота'

client = discord.Client()

CMSG, CGUILD, VCL, PREVSRC = None, None, None, ''

class VKLastPostsParser(HTMLParser):

    def reset(self):

        HTMLParser.reset(self)

        self.nnn = 0

        self.anek\_i = 0

        self.anek = ['', '', '', '']

    def handle\_starttag(self, tag, attrs):

        pass

    def handle\_endtag(self, tag):

        if tag == 'div' and self.nnn == 3:

            self.nnn = 1

            self.anek\_i += 1

    def handle\_data(self, data):

        if self.get\_starttag\_text() == '<div class="wall\_item">' and self.nnn < 2:

            self.nnn += 1

        if self.nnn == 2 and self.get\_starttag\_text() == '<div class="pi\_text">' or self.nnn == 3:

            self.anek[self.anek\_i] += str(data) + '\n'

            self.nnn = 3

def GetMemberID(from\_string):

    for sym in '<@!>': from\_string = from\_string.replace(sym, ' ')

    return from\_string

def GetMember(uid):

    global client

    return client.guilds[0].get\_member(int(uid))

def GetMembersGuilds(uid):

    global client

    l = []

    for g in client.guilds:

        l.append(g.get\_member(int(uid)))

    return l

def \_GetHighestRole(roles):

    h = 0

    for r in roles:

        if r.position > h:

            h = r.position

    return h

def GetHighestRole(member):

    return \_GetHighestRole(member.roles)

def IsHighestRole(num, guild):

    return num == \_GetHighestRole(guild.roles)

class Bot:

    kbreasons = [

        '111',

    ]

    games\_short = {

        'LOL': 'League of Legends',

        'Over': 'Overwatch',

        'Mine': 'Minecraft',

        'Ter': 'Terraria'

    }

    class Command:

        class RS:

            def \_\_init\_\_(self, text='', funcs=[]):

                self.text = text

                self.funcs = funcs

        prefix = 'свой префикс'

        def \_\_init\_\_(self, name, mask, command, helpText, restricted\_only=[], aliases=[]):

[self.name](http://self.name/) = name

            self.mask = mask

            self.command = command

            self.helpText = helpText

            self.restricted\_only = restricted\_only

            self.aliases = aliases

        def Run(self, rargs):

            global CMSG

            troles = [str(r) for r in CMSG.author.roles]

            f = False

            for r in self.restricted\_only:

                if r in troles:

                    f = True

                    break

            if len(self.restricted\_only) > 0 and not f:

                return [Bot.Command.RS](http://bot.command.rs/)(text=' еще не вырос. Скажешь, как подрастет.')

            if rargs.startswith('/?'):

                return [Bot.Command.RS](http://bot.command.rs/)(text=self.helpText)

            if self.mask == '':

                return self.command({})

            strs = []

            T = rargs

            while T.find('"') != -1:

                T = T[T.find('"') + 1:]

                zz = T[:T.find('"')]

                strs.append(zz)

                rargs = rargs.replace('"' + zz +'"', '%%cc')

                T = T[T.find('"') + 1:] if len(T) > T.find('"') + 1 else ''

            rargs = rargs.split(' ')

            mask = self.mask.split(' ')

            res = {}

            total = 0

            rq = 0

            for par in mask:

                required = par.find('(') != -1

                for c in '([])':

                    par = par.replace(c, '')

                par = par.split(':')

                name = par[0]

                typ = par[1] if len(par) > 1 else ''

                defVal = par[2] if len(par) > 2 else ''

                if required: rq += 1

                if len(rargs) > total:

                    rarg = rargs[total]

                    if defVal != '':

                        if rarg == defVal:

                            res[name] = True

                        else:

                            res[name] = False

                    else:

                        try:

                            if rarg == '%%cc' and len(strs) > 0:

                                rarg = strs[0]

                                strs.pop(0)

                            if typ == 'INT':

                                res[name] = int(rarg)

                            elif typ == 'FLOAT':

                                res[name] = float(rarg)

                            else:

                                if name.find('member') >= 0:

                                    tr = GetHighestRole(GetMember(GetMemberID(str(rarg))))

                                    mr = GetHighestRole(CMSG.author)

                                    if tr >= mr and not IsHighestRole(mr, CMSG.author.guild):

                                        return [Bot.Command.RS](http://bot.command.rs/)(text=' еще не вырос. Скажешь, как подрастет.')

                                res[name] = str(rarg)

                        except:

                            res[name] = 0

                    total += 1

                else:

                    res[name] = False

            if total < rq:

                raise Exception("Слишком мало аргументов")

            return self.command(res)

        @staticmethod

        def RemoveThisMessage(args):

            global CMSG

            res = [Bot.Command.RS](http://bot.command.rs/)()

            res.text = 'Deleted, ' + args['suqa'] + '!'

            res.funcs = [lambda: CMSG.channel.delete\_messages([CMSG])]

            return res

        @staticmethod

        def Kick(args):

            res = [Bot.Command.RS](http://bot.command.rs/)()

            reason = Bot.kbreasons[args['reason']]

            print(type(args['member']).\_\_name\_\_, type(args['91']).\_\_name\_\_, type(reason).\_\_name\_\_)

            res.text = args['member'] + ' кикнут' + (' 99' if args['91'] else '') + ' (' + reason + ')!'

            uid = GetMemberID(args['member'])

            members = GetMembersGuilds(uid)

            if not args['91'] and len(members) > 0:

                members = [members[0]]

            #print(type(member).\_\_name\_\_, member)

            res.funcs = []

            for member in members:

                res.funcs.append(lambda: member.kick(reason=reason))

            return res

        @staticmethod

        def Ban(args):

            res = [Bot.Command.RS](http://bot.command.rs/)()

            reason = Bot.kbreasons[args['reason']]

            res.text = args['member'] + ' забанен' + (' 99' if args['91'] else '') + ' (' + reason + ')' + '!'

            uid = GetMemberID(args['member'])

            members = GetMembersGuilds(uid)

            if not args['91'] and len(members) > 0:

                members = [members[0]]

            #print(type(member).\_\_name\_\_, member)

            res.funcs = []

            for member in members:

                res.funcs.append(lambda: member.ban(delete\_message\_days=(7 if args['91'] else 0), reason=reason))

            return res

        @staticmethod

        def Anek(args):

            href = '<https://vk.com/jumoreski>'

            h = requests.get(href).text

            p = VKLastPostsParser()

            p.feed(h)

            p.close()

            r = random.randint(0, 2)

            return [Bot.Command.RS](http://bot.command.rs/)(text = 'Анекдот из ' + href + ' (' + str(r) + '):\n' + p.anek[r])

        @staticmethod

        def Purge(args):

            global CMSG

            res = [Bot.Command.RS](http://bot.command.rs/)()

            res.funcs.append(lambda: CMSG.channel.purge(limit=10))

            return res

        @staticmethod

        def WhoIsIn(args):

            global CMSG, client

            res = [Bot.Command.RS](http://bot.command.rs/)()

            game = ''

            try: game = Bot.games\_short[args['game']]

            except: game = args['game']

            r2 = ''

            total = 0

            for guild in client.guilds:

                for member in guild.members:

                    if member == None: continue

                    act = member.activity

                    actname = [act.name](http://act.name/) if act != None else 'None'

                    if actname.lower() == game.lower():

                        r2 += member.display\_name + '\n'

                        total += 1

            res.text = 'В ' + game + ' сейчас играют (' + str(total) + '):\n' + (r2 if total > 0 else ':(')

            return res

        @staticmethod

        async def StopVoice():

            global VCL, PREVSRC

            try:

                await VCL.disconnect()

            except: pass

            finally: PREVSRC = '\_\_SKIP\_\_'

        @staticmethod

        async def StartVoice(nsrc):

            global CMSG, VCL

            try:

                c =  CMSG.author.voice.channel

                VCL = await c.connect()

            except: pass

            while VCL == None:

                time.sleep(0.5)

            #try:

            global PREVSRC#os.path.exists(nsrc)

            if PREVSRC != '\_\_SKIP\_\_' and nsrc != PREVSRC:

                if VCL.is\_playing():

                    VCL.source = FFmpegPCMAudio(nsrc, executable='D:\\ffmpeg\\bin\\ffmpeg.exe')

                    PREVSRC = nsrc

                    #VCL.stop()

                else:

                    print(nsrc, PREVSRC, VCL.is\_playing())

                    VCL.play(FFmpegPCMAudio(nsrc, executable='D:\\ffmpeg\\bin\\ffmpeg.exe'))

            print('zaloop started')

            #except: pass

        @staticmethod

        def Play(args):

            global VCL, PREVSRC

            res = [Bot.Command.RS](http://bot.command.rs/)()

            if PREVSRC == '\_\_SKIP\_\_': PREVSRC = ''

            res.funcs.append(lambda: Bot.Command.StartVoice(args['src']))

            res.text = 'Сейчас играет: ' + args['src']

            return res

        @staticmethod

        def StopPlaying(args):

            global PREVSRC, VCL

            res = [Bot.Command.RS](http://bot.command.rs/)()

            res.funcs = [Bot.Command.StopVoice]

            res.text = 'Плеер остановлен.'

            return res

        @staticmethod

        def SkipPlaying(args):

            if VCL != None:

                PREVSRC = '\_\_SKIP\_\_'

                VCL.stop()

            return [Bot.Command.RS](http://bot.command.rs/)(text='Композиция пропущена.')

        @staticmethod

        def Howl(args):

            return [Bot.Command.RS](http://bot.command.rs/)(funcs=[lambda: Bot.Command.StartVoice('Howl.mp3')], text='\*Вой вам в голосовой\*')

        @staticmethod

        def Whine(args):

            return [Bot.Command.RS](http://bot.command.rs/)(funcs=[lambda: Bot.Command.StartVoice('Whine.mp3')], text='\*Скулеж\*')

    def Help(self, args):

        res = [Bot.Command.RS](http://bot.command.rs/)()

        res.text = 'Помощь:\n'

        for c in self.commands:

            res.text += [c.name](http://c.name/) + str(c.aliases) + ' ' + c.mask + ': ' + c.helpText + '\n'

            res.text += '\n'

        return res

    commands = []

    def CheckCmds(self, com, msg):

        global CMSG

        CMSG = msg

        CGUILD = msg.channel.guild

        result = [Bot.Command.RS](http://bot.command.rs/)()

        success = False

        #try:

        for cmd in self.commands:

            tr = copy(cmd.aliases)

            tr.append([cmd.name](http://cmd.name/" \t "_blank))

            for t in tr:

                if com.lower().startswith(Bot.Command.prefix + t):

                    result = cmd.Run(com[len(Bot.Command.prefix + t) + 1:])

                    success = True

        #except Exception as e:

            #result.text = str(e)

            #print(e)

        #finally:

        if result is str:

            result = [Bot.Command.RS](http://bot.command.rs/)()

        return result, success

    def \_\_init\_\_(self):

        self.commands.append(Bot.Command('removethis', '(suqa:)', Bot.Command.RemoveThisMessage, ''))

        h = ' (@юзер) (причина\_число) [99]\nЕсли написать 99, то действие выполнится на всех серверах с ботом.\nПричины:\n'

        for i in range(len(Bot.kbreasons)):

            h += str(i) + ' - ' + Bot.kbreasons[i] + '\n'

        self.commands.append(Bot.Command('забань', '(member:) (reason:INT) [91::99]', Bot.Command.Ban, 'забань' + h))

        self.commands.append(Bot.Command('кикни', '(member:) (reason:INT) [91::99]', Bot.Command.Kick, 'кикни' + h))

        self.commands.append(Bot.Command('чистка', '', Bot.Command.Purge, 'Чистит канал', aliases = ['смыв', 'смой', 'убери это 99', 'спуск']))

        self.commands.append(Bot.Command('повой', '', Bot.Command.Howl, 'Вытье вам в ухо', aliases=['повыть', 'вой']))

        self.commands.append(Bot.Command('поскули', '', Bot.Command.Whine, 'Скулеж вам в ухо', aliases=['поскулить', 'скули']))

        self.commands.append(Bot.Command('анекдот', '', Bot.Command.Anek, 'Анеки', aliases = ['анек', 'лол', 'lol']))

        self.commands.append(Bot.Command('play', '(src:)', Bot.Command.Play, 'Проигрывает музыку с компа одмена, или из url (должна быть ссылка на файл!)', aliases = ['!', 'p']))

        self.commands.append(Bot.Command('stop', '(src:)', Bot.Command.StopPlaying, 'Остановить проигрыватель', aliases = ['/', 's']))

        self.commands.append(Bot.Command('skip', '(src:)', Bot.Command.SkipPlaying, 'Пропустить композицию', aliases = ['-', 's']))

        h = 'кто в ("name of game"). Список участников, играющих в game.\nСписок кратких названий:\n'

        for k in Bot.games\_short:

            h += k + ' = ' + Bot.games\_short[k] + '\n'

        self.commands.append(Bot.Command('кто в', '(game:)', Bot.Command.WhoIsIn, h))

        self.commands.append(Bot.Command('памаги', '', self.Help, '', aliases = ['?', 'help', 'h', 'помогите', 'памагити']))

\*\*\*MAIN.PY\*\*\*:   
import discord

from discord.ext import commands

bot = commands.Bot(command\_prefix = settings['w1'])

@bot.command() # Не передаём аргумент pass\_context, так как он был нужен в старых версиях.

async def hello(ctx): # Создаём функцию и передаём аргумент ctx.

    author = ctx.message.author # Объявляем переменную author и записываем туда информацию об авторе.

    await ctx.send(f'Hello, {author.mention}!') # Выводим сообщение с упоминанием автора, обращаясь к переменной author.

bot.run(settings['NzgzNjkxMzU2NjQzMDAwMzYw.X8ebcw.mU2zG2jJz\_7zHboogpAXTqtpr2w']) # Обращаемся к словарю settings с ключом token, для получения токена

import json

import requests

@bot.command()

async def fox(ctx):

    response = requests.get('<https://some-random-api.ml/img/fox>') # Get-запрос

    json\_data = json.loads(response.text) # Извлекаем JSON

    embed = discord.Embed(color = 0xff9900, title = 'Random Fox') # Создание Embed'a

    embed.set\_image(url = json\_data['link']) # Устанавливаем картинку Embed'a

    await ctx.send(embed = embed) # Отправляем Embed