# modded by @enicaaa & blyat kak suda popal @Yahikoro.

# meta developer of NumMod: @trololo\_1

from .. import loader, utils # noqa

import asyncio

import contextlib

import pytz

import re

import telethon

from telethon.tl.types import MessageEntityTextUrl

import json as JSON

from datetime import datetime, date, time

class NumMod(loader.Module):

"Заражает по реплаю. Moded by @enicaaa , @Yahikoro . по любым вопросам или желаниям так же @enicaaa"

strings = {'name': 'Biomod'}

async def client\_ready(self, client, db):

self.db = db

if not self.db.get("NumMod", "exUsers", False):

self.db.set("NumMod", "exUsers", [])

if not self.db.get("NumMod", "infList", False):

self.db.set("NumMod", "infList", {})

async def зcmd(self, message):

" [arg] [arg] [arg]....\n в качестве аргументов используй числа или первые символы строки."

reply = await message.get\_reply\_message()

a = reply.text

exlist = self.db.get("NumMod", "exUsers")

count\_st = 0

count\_hf = 0

if not reply:

await message.edit('нет реплая.')

return

args = utils.get\_args\_raw(message)

list\_args = []

if not args:

await message.edit('нет аргументов')

return

for i in args.split(' '):

if '-' in i:

ot\_do = i.split('-')

try:

list\_args.extend(str(x) for x in range(int(ot\_do[0]), int(ot\_do[1]) + 1))

except Exception:

await message.respond('используй правильно функцию "от-до"')

return

else:

list\_args.append(i)

lis = a.splitlines()

for start in list\_args:

for x in lis:

if x.lower().startswith(str(start.lower())):

count\_st = 1

if 'href="' in x:

count\_hf = 1

b = x.find('href="') + 6

c = x.find('">')

link = x[b:c]

if link.startswith('tg'):

users = '@' + link.split('=')[1]

if users in exlist:

await message.reply(f'исключение: <code>{users}</code>')

else:

await message.reply(f'заразить {users}')

elif link.startswith('https://t.me'):

a = '@' + str(link.split('/')[3])

if a in exlist:

await message.reply(f'исключение: <code>{a}</code>')

else:

await message.reply(f'заразить {a}')

else:

await message.reply('что за хуета?')

break

await asyncio.sleep(5)

await message.delete()

if not count\_st:

await message.edit('не найдено ни одного совпадения в начале строк с аргументами.')

elif not count\_hf:

await message.edit('не найдено ни одной ссылки.')

elif len(list\_args) >= 3:

await message.respond('<b>заражения успешно завершены.</b>')

async def кcmd(self, message):

"""заражает всех по реплаю."""

reply = await message.get\_reply\_message()

exlist = self.db.get("NumMod", "exUsers")

if not reply:

await message.edit('нет реплая.')

return

json = JSON.loads(reply.to\_json())

for i in range(len(reply.entities)):

try:

link = json["entities"][i]["url"]

if link.startswith('tg'):

users = '@' + link.split('=')[1]

if users in exlist:

await message.reply(f'исключение: <code>{users}</code>')

else:

await message.reply(f'заразить {users}')

elif link.startswith('https://t.me'):

a = '@'

if a in exlist:

await message.reply(f'исключение: <code>{a}</code>')

else:

await message.reply(f'заразить {a}')

else:

await message.reply('что за хуета?')

except Exception:

await message.reply("заразить " + reply.raw\_text[

json["entities"][i]["offset"]:json["entities"][i]["offset"] +

json["entities"][i]["length"]])

await asyncio.sleep(3)

await message.delete()

async def exnumcmd(self, message):

"""добавляет исключения в модуль.\nиспользуй: .exnum {@user/@id}"""

args = utils.get\_args\_raw(message)

exlistGet = self.db.get("NumMod", "exUsers")

exlist = exlistGet.copy()

if not args:

if len(exlist) < 1:

await message.edit('список исключений пуст.')

return

exsms = ''.join(f'<b>{count}.</b> <code>{i}</code>\n' for count, i in enumerate(exlist, start=1))

await utils.answer(message, exsms)

return

if args == 'clear':

exlist.clear()

self.db.set("NumMod", "exUsers", exlist)

await message.edit('список исключений очистен.')

return

if len(args.split(' ')) > 1 or args[0] != '@':

await message.edit(

'количество аргументов <b>больше</b> одного, либо начинается <b>не</b> со знака <code>@</code>'

)

return

if args in exlist:

exlist.remove(args)

self.db.set("NumMod", "exUsers", exlist)

await message.edit(f'пользователь <code>{args}</code> исключён.')

return

exlist.append(args)

self.db.set("NumMod", "exUsers", exlist)

await message.edit(f'пользователь <code>{args}</code> добавлен.')

async def zarlistcmd(self, message):

""" лист ваших заражений.\n.zarlist {@id/user} {count} {args}\nдля удаления: .zarlist {@id/user}\nаргументы:\n-k -- добавить букву k(тысяч) к числу.\n-f -- поиск по ид'у/юзеру.\n-r -- добавлению в список по реплаю. """

args = utils.get\_args\_raw(message)

infList = self.db.get("NumMod", "infList")

timezone = "Europe/Kiev"

vremya = datetime.now(pytz.timezone(timezone)).strftime("%d.%m")

with contextlib.suppress(Exception):

args\_list = args.split(' ')

if not args:

if not infList:

await utils.answer(message, "лист заражений <b>пуст</b>.")

return

sms = ''.join(

f'<b>• <code>{key}</code> -- <code>{value[0]}</code> [<i>{value[1]}</i>]</b>\n' for key, value in

infList.items())

await utils.answer(message, sms)

return

if '-r' in args.lower():

reply = await message.get\_reply\_message()

if not reply:

return await utils.answer(message, 'реплай должен быть на смс ириса "<b>...подверг заражению...</b>"')

elif reply.sender\_id != 707693258 and 'подверг заражению' not in reply.text:

return await utils.answer(message, 'реплай должен быть на смс ириса "<b>...подверг заражению...</b>"')

else: # ☣

text = reply.text

x = text.index('☣') + 4

count = text[x:].split(' ', maxsplit=1)[0]

x = text.index('user?id=') + 8

user = '@' + text[x:].split('"', maxsplit=1)[0]

infList[user] = [str(count), vremya]

self.db.set("NumMod", "infList", infList)

await utils.answer(

message,

f"пользователь <code>{user}</code> добавлен в список заражений.\n"

f"число: <code>{count}</code>\n"

f"дата: <b>{vremya}</b>"

)

elif args\_list[0] == "clear":

infList.clear()

self.db.set("NumMod", "infList", infList)

await utils.answer(message, "лист заражений <b>очищен</b>.")

elif args\_list[0] in infList and '-f' in args.lower():

user = infList[args\_list[0]]

await utils.answer(message, f"<b>• <code>{args\_list[0]}</code> -- {user[0]} [<i>{user[1]}</i>]</b>")

elif len(args\_list) == 1 and args\_list[0] in infList:

infList.pop(args\_list[0])

self.db.set("NumMod", "infList", infList)

await utils.answer(message, f"пользователь <code>{args}</code> удалён из списка.")

elif args\_list[0][0] != '@':

await utils.answer(message, 'это не <b>@ид/юзер</b>.')

else:

try:

user, count = str(args\_list[0]), float(args\_list[1])

except Exception:

await utils.answer(message, "данные были введены не корректно")

return

k = ''

if '-k' in args.lower():

k += 'k'

infList[user] = [str(count) + k, vremya]

self.db.set("NumMod", "infList", infList)

await utils.answer(

message,

f"пользователь <code>{user}</code> добавлен в список заражений.\n"

f"число: <code>{count}</code>{k}\n"

f"дата: <b>{vremya}</b>"

)

async def numfiltercmd(self, message):

""" .numfilter {args1} {args2 OR reply} \nвызови команду, чтобы просмотреть аргументы."""

args = utils.get\_args\_raw(message)

reply = await message.get\_reply\_message()

filter\_and\_users = self.db.get("NumMod", "numfilter", {'users': [], 'filter': None, 'status': False})

if not args:

return await utils.answer(

message,

f"-sU --- добавить|удалить юзеров(не больше 200), на которых будет триггериться фильтр(ид|реплай).\n"

f"[{', '.join([f'<code>{i}</code>' for i in filter\_and\_users['users']])}]\n"

f"-sF --- установить фильтр. допустим один.\n"

f"<code>{filter\_and\_users['filter'] or '❌не установлен.'}</code>\n"

f"-t --- запустить|остановить.\n"

f"<b>{'✅запущен' if filter\_and\_users['status'] else '❌остановлен'}.</b>\n\n"

f"<b>работает так:</b>\n"

f"[фильтр] (еби|еб|бей|кусь|кусай|уеби|зарази|заразить) (1-10) ((@id|user)|link(даже полный линк ид'а))\n"

f"[фильтр] вакцин[ау]|лечись|ва[ккц]|хи[лльсяйинг]\n"

f"[фильтр] жертвы|ежа\n"

f"[фильтр] болезни|бол\n"

f"[фильтр] лаб[уа]|статы\n"

f"например:[фильтр] квалификация 4 (улучшает квалификацию учённых на 4 ур.\n"

f"или [фильтр] [чек] квалификация [чек] 4 (проверяет цену квалификации учённых на 4 ур.\n"

f"доступные навыки:\n"

f"Игнор регистра!!"

)

args = args.split(' ', maxsplit=1)

if len(args) == 1 and not reply and args[0] != '-t':

return await utils.answer(message, '❌ нет 2 аргумента и реплая.')

elif args[0] == '-sU':

try:

user\_id = args[1]

if not user\_id.isdigit():

return await utils.answer(message, 'это не ид.')

except Exception:

user\_id = str(reply.sender\_id)

if user\_id in filter\_and\_users['users']:

filter\_and\_users['users'].remove(user\_id)

await utils.answer(message, f"✅ ид <code>{user\_id}</code> удалён.")

elif len(filter\_and\_users['users']) <= 200:

filter\_and\_users['users'].append(user\_id)

await utils.answer(message, f"✅ ид <code>{user\_id}</code> добавлен.")

else:

return await utils.answer(message, '❌ Превышен лимит в 200 юзеров.')

return self.db.set("NumMod", "numfilter", filter\_and\_users)

elif args[0] == '-sF':

try:

filter\_and\_users['filter'] = args[1].lower().strip()

self.db.set("NumMod", "numfilter", filter\_and\_users)

return await utils.answer(message, f"✅ фильтр ~~~ <code>{args[1]}</code> ~~~ успешно установлен!")

except Exception:

return await utils.answer(message, "где 2 аргумент❓")

elif args[0] == '-t':

if filter\_and\_users['status']:

filter\_and\_users['status'] = False

self.db.set("NumMod", "numfilter", filter\_and\_users)

return await utils.answer(message, "❌ дов(фильтр) остановлен.")

else:

filter\_and\_users['status'] = True

self.db.set("NumMod", "numfilter", filter\_and\_users)

return await utils.answer(message, "✅ дов(фильтр) запущен.")

else:

return await utils.answer(message, "❌ неизвестный аргумент.")

async def watcher(self, message):

if not isinstance(message, telethon.tl.types.Message): return

filter\_and\_users = self.db.get("NumMod", "numfilter", {'users': [], 'filter': None, 'status': False})

user\_id = str(message.sender\_id)

if not filter\_and\_users['filter'] or not filter\_and\_users['status'] or user\_id not in filter\_and\_users[

'users'] or message.is\_private: return

text = message.raw\_text.lower()

if not text.startswith(filter\_and\_users['filter']): return

if send\_mesа := re.search(

r"(?P<z>бей\s|кусь\s|кусай\s|зарази\s|заразить\s|еб\s|еби\s|уеби\s{,2}\s)(?P<lvl>[1-9]?[0]?\s)?(?P<link>@[0-9a-z\_]+|(?:https?://)?t\.me/[0-9a-z\_]+|tg://openmessage\?user\_id=(?P<id>[0-9]+))",

text):

send\_mesа = send\_mesа.groupdict()

send\_mesа['link'], send\_mesа['id'] = '@' + send\_mesа['id'] if send\_mesа['id'] else send\_mesа['link'], ''

send\_mesа['z'] = 'заразить '

send\_mesа['lvl'] = send\_mesа['lvl'] or ''

mes = ''.join(send\_mesа.values())

await message.respond(mes)

elif send\_mesz := re.search(r"(?P<zar>заразность\s|зз\s)(?P<lvl>[0-5]+)", text):

send\_mesz = send\_mesz.groupdict()

send\_mesz['zar'] = '++заразность '

send\_mesz['lvl'] = send\_mesz['lvl'] or ''

mes = ''.join(send\_mesz.values())

await message.respond(mes)

elif send\_mesz := re.search(r"(?P<zar>заразность чек\s|зз чек\s|чек зз\s|чек заразность\s)(?P<lvl>[0-5]+)", text):

send\_mesz = send\_mesz.groupdict()

send\_mesz['zar'] = '+заразность '

send\_mesz['lvl'] = send\_mesz['lvl'] or ''

mes = ''.join(send\_mesz.values())

await message.respond(mes)

elif send\_mesp := re.search(r"(?P<pat>патоген\s|паты\s)(?P<lvl>[0-5]+)", text):

send\_mesp = send\_mesp.groupdict()

send\_mesp['pat'] = '++патоген '

send\_mesp['lvl'] = send\_mesp['lvl'] or ''

mes = ''.join(send\_mesp.values())

await message.respond(mes)

elif send\_mesp := re.search(r"(?P<pat>патоген чек\s|паты чек\s|чек паты\s|чек патоген\s)(?P<lvl>[0-5]+)", text):

send\_mesp = send\_mesp.groupdict()

send\_mesp['pat'] = '+патоген '

send\_mesp['lvl'] = send\_mesp['lvl'] or ''

mes = ''.join(send\_mesp.values())

await message.respond(mes)

elif send\_mesl := re.search(r"(?P<let>летальность\s|летал\s)(?P<lvl>[1-5]+)", text):

send\_mesl = send\_mesl.groupdict()

send\_mesl['let'] = '++летальность '

send\_mesl['lvl'] = send\_mesl['lvl'] or ''

mes = ''.join(send\_mesl.values())

await message.respond(mes)

elif send\_mesl := re.search(r"(?P<let>летальность чек\s|летал чек\s|чек летал\s|чек летальность\s)(?P<lvl>[1-5]+)", text):

send\_mesl = send\_mesl.groupdict()

send\_mesl['let'] = '+летальность '

send\_mesl['lvl'] = send\_mesl['lvl'] or ''

mes = ''.join(send\_mesl.values())

await message.respond(mes)

elif send\_mesk := re.search(r"(?P<kvala>квала\s|квалификация\s)(?P<lvl>[0-5]+)", text):

send\_mesk = send\_mesk.groupdict()

send\_mesk['kvala'] = '++квалификация '

send\_mesk['lvl'] = send\_mesk['lvl'] or ''

mes = ''.join(send\_mesk.values())

await message.respond(mes)

elif send\_mesk := re.search(r"(?P<kvala>квала чек\s|квалификация чек\s|чек квалификация \s|чек квала\s)(?P<lvl>[0-5]+)", text):

send\_mesk = send\_mesk.groupdict()

send\_mesk['kvala'] = '+квалификация '

send\_mesk['lvl'] = send\_mesk['lvl'] or ''

mes = ''.join(send\_mesk.values())

await message.respond(mes)

elif send\_mesb := re.search(r"(?P<sb>сб\s|безопасность\s)(?P<lvl>[0-5]+)", text):

send\_mesb = send\_mesb.groupdict()

send\_mesb['sb'] = '++безопасность '

send\_mesb['lvl'] = send\_mesb['lvl'] or ''

mes = ''.join(send\_mesb.values())

await message.respond(mes)

elif send\_mesb := re.search(r"(?P<sb>сб чек\s|безопасность чек\s|чек безопасность\s|чек сб\s)(?P<lvl>[0-5]+)", text):

send\_mesb = send\_mesb.groupdict()

send\_mesb['sb'] = '+безопасность '

send\_mesb['lvl'] = send\_mesb['lvl'] or ''

mes = ''.join(send\_mesb.values())

await message.respond(mes)

if re.search(r"болезни|бол", text):

await message.respond('мои болезни')

if re.search(r"жерт[выа]|ежа", text):

await message.respond('мои жертвы')

if re.search(r"статы|лаб[уа]{,2}", text):

await message.respond('.лаб')

if re.search(r"цен[аз]{,2}", text):

await message.respond('купить вакцину')

elif re.search(r"вакцин[ау]|ва[ккц]|лечись|хи[лльсяйинг]{,2}", text):

await message.respond('.купить вакцину')

async def идcmd(self, message):

"""показывает ID юзера."""

args = utils.get\_args\_raw(message)

reply = await message.get\_reply\_message()

try:

if args:

user = await message.client.get\_entity(

args if not args.isdigit() else int(args)

)

else:

user = await message.client.get\_entity(reply.sender\_id)

except ValueError:

user = await message.client.get\_entity(message.sender\_id)

await message.edit(

f"<b>Имя:</b> <code>{user.first\_name}</code>\n"

f"<b>ID:</b> <code>{user.id}</code>"

)