

python-asyncio 3.4.3

python-aiogram 3.2.0

Запуск через терминал в центральной папке через:

python circles_bot.py

Структура:

- data
алгоритмы для работы с базой данных, саму базу данных [*.db]
- handlers
хэндлеры для обработки пользовательских запросов
- keyboards
клавиатуры для взаимодействия с пользователем
- utils
утилиты. Скрипты для всего-всего: поисковая машина, поиск городов по координате, удаление предыдущих сообщений
- circles_bot.py
управляющий файл

Столбцы базы данных:

id строки

used_id – уникальный номер пользователя (по сути номер чата с пользователем)

likes – кого пользователь лайкнул. строка формата "id1, id2, id2"

liked – кем пользователь был лайкнут (likes и liked взаимозависимы. liked нужен только для ускорения). строка формата "id1, id2, id2"

already_saw – кого пользователь уже просмотрел. строка формата "id1, id2, id2"

already_seen_by – кем пользователь был просмотрен (already_saw и already_seen_by взаимозависимы. already_seen_by нужен только для ускорения). строка формата "id1, id2, id2"

waiting – режим ожидания. Ждет ли пользователь лайков. Если мэтч уже был, то не ждет. Формат: да/нет

active – режим активности. Активен ли профиль. Если профиль истек по давности, то false. Формат: да/нет

time – время и дата создания профиля

name – имя

latitude – широта координаты

longitude – долгота координаты

target – категория пользователя. Создатель или гость события

gender – гендер пользователя

look_for – какой гендер ищет пользователя. 3 значения – м/ж/оба

circle – адрес кружочка. он хранится в переписке с пользователем

username – юзернейм. для отправки ссылки на пользователя при мэтче

msgs_to_delete – сообщения на удаление. Некоторые сообщения удаляются сразу, без записи в БД, прямо внутри проги (например это касается сообщений об удалении или создании профиля)

Пользовательский путь:

Создается профиль. После создания профиля (сразу в скрипте создания!) происходит рассылка (если создатель) всем ждущим и активным или получение ближайшего события (если гость).

После каждого действия происходит проверка на мэтчи и на актуальность профиля. Проверка на мэтчи: совпадение значения likes и liked.

У пользователя всегда есть возможность пожаловаться на другого пользователя. В таком случае в профиль объекта жалобы добавится id субъекта в столбец already_seen_by и в столбец already_saw. Так же и для субъекта жалобы.

Примечание: при записи изменений в БД некоторые значения требуют перевода в формат текста через str(value)