



**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”**

**Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем**

**Лабораторна робота №2 з дисципліни “Бази
даних. Частина 2” тема “Практика
використання сервера Redis”**

Варіант 1

Виконав

студент III курсу

групи КП-82

Анікєєв Ігор Анатолійович

Посилання на репозиторій: <https://github.com/flain1/dblabs2>

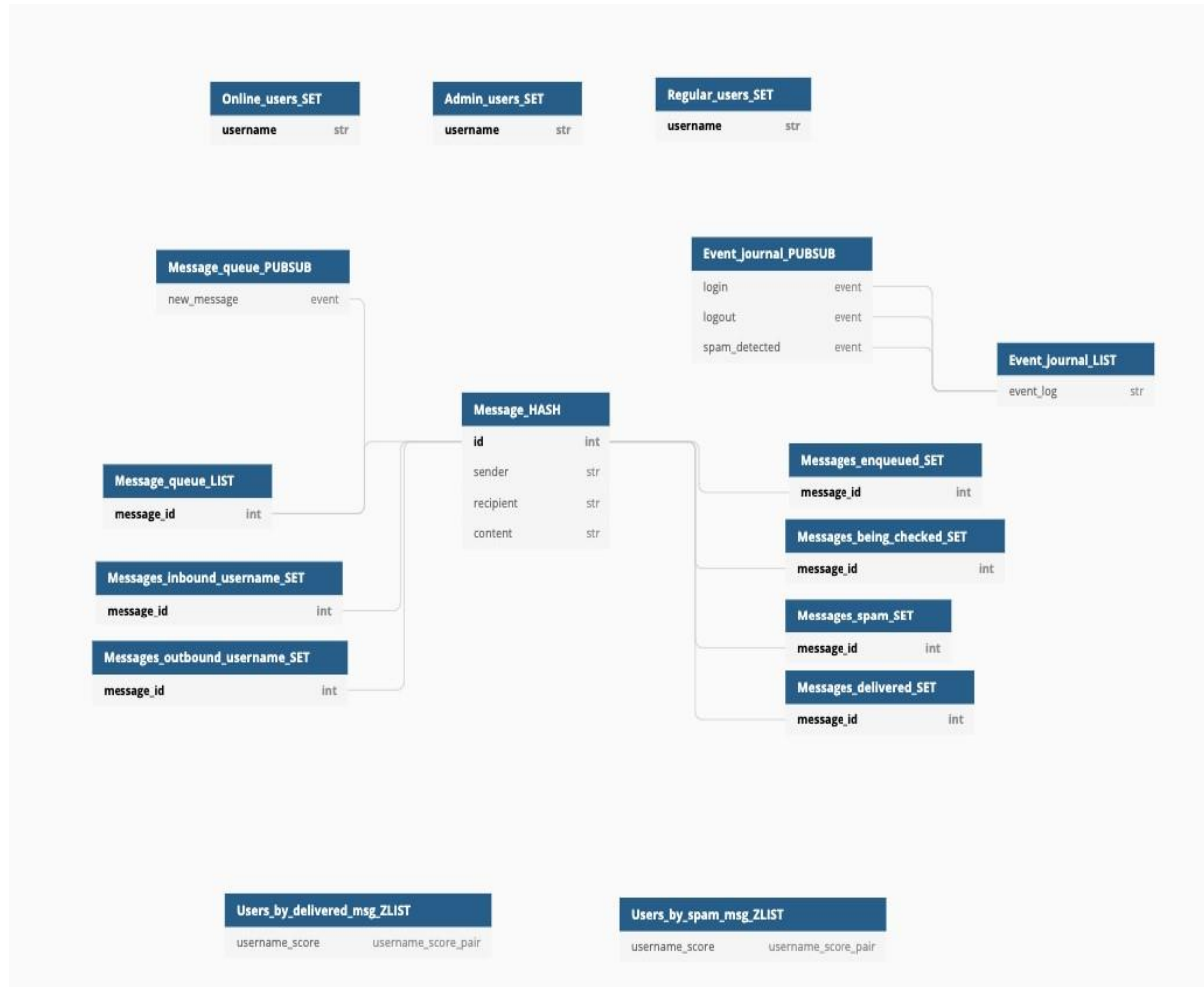
Мета

Метою роботи є здобуття практичних навичок створення ефективних програм, орієнтованих на використання сервера Redis за допомогою мови Python.

Постановка завдання

Реалізувати можливості обміну повідомленнями між користувачами у оффлайн та онлайн режимах із можливістю фільтрації спам-повідомлень.

Структура Redis



SET - Використовується для зберігання унікальних, неупорядкованих даних (імена користувачів). Використовується для зберігання статусу повідомлень для простішого знаходження повідомлень за їх статусом.

HASH - Використовується для зберігання пар ключ-значення, де значення зазвичай об'єкт, що моделює якусь структуру даних. Використовується для зберігання даних про повідомлення до доступу до них за id-ключем.

LIST - Двустороння черга, що дозволяє впорядкований доступ до даних. Використовується для черги повідомлень, де нові додаються у кінець, а

worker обробляє їх, беручи з початку черги. Також використовується для збереження подій доданих до журналу у хронологічному порядку.

ZLIST - Дозволяє ранжувати записи за нумеричним ключем.

Використовується для зберігання пар

“користувач->кількість_надісланих_повідомлень” та

“користувач->кількість_спам_повідомлень”. Це дозволяє легко знайти найактивніших та найшкідливіших користувачів.

PUB/SUB - Дозволяє створити канал повідомлень куди можна надсилати та з якого можна слухати повідомлення. Використовується для журналювання подій логін/логаут/знайдення_спам_повідомлення. Також використовується як триггер для сповіщення worker-а про нові повідомлення у черзі та початку їх опрацювання.

Запуск функціоналу

Замість консольного інтерфейсу було створено API з аналогічним функціоналом.

Доступні наступні запити:

- **POST /login** - Авторизувати користувача за ім'ям. Позначити його як online у Redis
- **POST /logout** - Позначити користувача як offline
- **POST /message** - Надіслати повідомлення користувачу recipient від sender з вмістом content.
- **GET /message/inbound** - Дістати вхідні повідомлення для користувача з username
- **GET /user-stats** - Дістати кількість повідомлень для username, що були доставлені/в черзі/перевіряються/у спамі
- **GET /spammer-stats** - Дістати список користувачів ранжований за найбільшою кількістю спам повідомлень що вони надіслали
- **GET /chatter-stats** - Дістати список користувачів ранжований за найбільшою кількістю успішно надісланих повідомлень
- **GET /online-users** - Дістати список користувачів, що на даний момент онлайн
- **GET /event-journal** - Дістати список подій у хронологічному порядку

Скріншоти роботи програми

Приклад відправки повідомлення

Приклад отримання статистики повідомлень

The screenshot shows a REST client interface. The left pane displays the request: `GET Lab2 BASE_URL /event-journal`. The right pane shows the response status `200 OK` and the JSON body:

```
{
  "chatters": [
    { "index": 0, "type": "LOGIN", "value": "Dizzzmas" },
    { "index": 1, "type": "LOGOUT", "value": "Dizzzmas" },
    { "index": 2, "type": "LOGIN", "value": "Dizzzmas" },
    { "index": 3, "type": "LOGOUT", "value": "Dizzzmas" },
    { "index": 4, "type": "SPAM", "value": "message with id 6 by Diz..." },
    { "index": 5, "type": "SPAM", "value": "message with id 4 by Diz..." },
    { "index": 6, "type": "SPAM", "value": "message with id 3 by Diz..." },
    { "index": 7, "type": "LOGIN", "value": "Dizzzmas" }
  ]
}
```

The screenshot shows a REST client interface. The left pane displays the request: `POST Lab2 BASE_URL /message`. The right pane shows the response status `200 OK` and the JSON body:

```
{
  "content": "My message",
  "id": 6,
  "recipient": "Ilya",
  "sender": "Dizzzmas"
}
```

Приклад перегляду журналу подій

The screenshot shows a REST client interface. The left pane displays the request: `GET Lab2 BASE_URL /user-stats` with the URL parameter `username=Dizzzmas`. The right pane shows the response status `200 OK` and the JSON body:

```
{
  "being_spam_checked": 0,
  "delivered": 3,
  "enqueued": 0,
  "marked_as_spam": 3
}
```

Висновок

У ході лабораторної роботи було здобуто навички створення ефективних програм, орієнтованих на використання сервера Redis за допомогою мови

Python. Отримано досвід використання основних структур та команд Redis.