DOI: https://doi.org/10.36910/6775-2524-0560-2020-41-08
УДК: 004.62
Лавренчук Світлана Василівна, к.т.н., доцент
https://orcid.org/0000-0002-5453-3924
Чабан Артем Сергійович, студент
Луцький національний технічний університет, м. Луцьк, Україна.

ДОСЛІДЖЕННЯ ЗМІНИ ПОГОДНИХ УМОВ ЗА ДОПОМОГОЮ TELEGRAM BOT API

Лавренчук С.В., Чабан А.С. Дослідження зміни погодних умов за допомогою Telegram Bot API. Розглянуто дослідження зміни погодних умов за допомогою Telegram Bot API. Використано аналітичні типи даних з кількох ресурсів API для виявлення найбільш точного та швидкого доступу до інформації через Telegram Bots із моніторингом більше двох вибірок в одній системі координат. Це дозволить нам вибирати найточніші показники та швидше передавати інформацію кінцевому користувачу. Враховано основні показники температури з точністю до однієї сотої, вологість у відсотках, атмосферний тиск, швидкість вітру, хмарність, індекс ультрафіолету у вибраній системі координат, видимість на вулиці, стан озону в атмосфері.
Ключові слова: Telegram, bot, аналітика, точність даних, моніторинг даних, ввід даних, вивід даних, вибірка, погодні показники, швидкість передачі даних.

Лавренчук С.В., Чабан А.С. Исследование изменения погодных условий с помощью Telegram Bot API. Рассмотрены исследования изменения погодных условий с помощью Telegram Bot API. Использованы аналитические типы данных с нескольких ресурсов API для выявления наиболее точного и быстрого доступа к информации через Telegram Bots с мониторингом более двух выборок в одной системе координат. Это позволит нам выбирать точные показатели и быстрее передавать информацию конечному пользователю. Учтены основные показатели температуры с точностью до одной сотой, влажность в процентах, атмосферное давление, скорость ветра, облачность, индекс ультрафиолета в выбранной системе координат, видимость на улице, состояние озона в атмосфере.
Ключевые слова: Telegram, bot, аналитика, точность данных, мониторинг данных, ввод данных, вывод данных, выборка, погодные показателях, скорость передачи данных.

Lavrenchuk S.V., Chaban A.S. Investigate weather changes using the Telegram Bot API. Studies of changes in weather conditions using the Telegram Bot API are considered. Analytical data types from several API resources were used to identify the most accurate and fast access to information through Telegram Bots with monitoring of more than two samples in one coordinate system. This will allow us to choose accurate indicators and transfer information to the end user faster. The main indicators of temperature are taken into account with an accuracy of one hundredth, humidity as a percentage, atmospheric pressure, wind speed, cloud cover, ultraviolet index in the selected coordinate system, visibility on the street, and ozone in the atmosphere.
Keywords: Telegram, bot, analytics, data accuracy, data monitoring, data entry, data output, sampling, weather indicators, data rate.

Вступ. Телеграм ботами користується все більше людей. Ця, на перший погляд, проста річ дозволяє взаємодіяти з банками, обмінниками, робити онлайн замовлення, керувати Bitcoin-гаманцями, управляти бізнес процесами, має підтримку баз даних, а також може замінити служби підтримки. Тобто цей інструмент слугує для автоматизації процесів, аналітики даних та інших адміністративних процесів.

Ці всі дії користувачі робили раніше за допомого Web-ресурсів. Оскільки при доступі до цієї інформації нам потрібен час для обробки даних, то ми будемо використовувати Telegram API для подальшого зв'язку з нашим ресурсом, від якого будемо отримувати наші вибірки з потрібною інформацією.

До основних переваг отримування даних через Telegram Bot API можна віднести:

- Користувачі швидше отримують потрібну інформацію порівняно з сайтами. Досить багато сайтів і додатків мають подібний інтерфейс, ніби однакові для всіх, але разом з тим не підходять ні для кого. При використанні ботів виникає відчуття, ніби людину обслуговують особисто, і вона може навіть не здогадуватися, що на тому боці знаходиться розумна програма, а не конкретний менеджер. Крім того, персональне обслуговування відкриває нові можливості для просування послуги, додаткових бонусів, персональних акції, що може принести величезну віддачу.
- Текстовий інтерфейс потребує мало інтернет-трафіку, крім того у більшості операторів на месенджери розповсюджується безлімітний інтернет. Це досить важливо, адже близько половини клієнтів виходять в інтернет тільки з мобільних пристроїв, не у всіх є безліміт або висока швидкість. Наприклад, якщо користувачеві потрібно дізнатися прогноз погоди, то він не буде чекати завантаження цілої сторінки у браузері, дивитись на рекламні банери і шукати потрібну інформацію, набагато зручніше написати боту і відразу отримати те, що потрібно.
- Дешеві сервери, щоб розміститись на ресурсі. Не потрібно платити за домен та хостинг. Під наші потреби передачі та для здійснення аналітичного аналізу інформації підходить саме цей варіант.

- Гнучкість і швидкість відповіді користувачу. На сайті потрібно звертатись у службу підтримки і не зрозуміло коли буде відповідь. Бот сам виправляє свої помилки просто сам перезапуститься через кілька хвилин.

- Захист від небажаних конкурентів. Вся інформація в зашифрованому виді n-p-n знаходиться на пристрої у користувача. Тому зловмисникам неможливо запустити вірусний файл для руйнації проекту.

- Не вимагає установки і авторизації. Щоб скористатися ботом, клієнтові не потрібно завантажувати і встановлювати додаток на телефон або заповнювати форму реєстрації на сайті або в додатку. Як тільки клієнт написав боту – ви відразу знаєте його ім'я і логін, після чого ви можете працювати з користувачем.

Аналіз останніх досліджень. Розглянемо подібні проекти які працюють по даному принципу, проаналізуємо найбільші проекти на телеграм площадці, яка спрямована на світовий ринок.

Weatherman_bot був актуальним кілька років. На даний момент він не працює, оскільки автори не змогли монетизувати проект і втриматись на телеграм площадці.

TheLairBot бот показує детальний прогноз погоди на 5 днів. Він показує температуру, щільність хмар, опади, зливи, сніг, швидкість вітру, пориви вітру, напрямок вітру, тривалість світлового дня. Він питає місто та сам вибирає точку на карті Google maps. Це не дуже зручно, оскільки, якщо користувачеві потрібно вказати координати певного району великого мегаполісу, то актуальність даних втрачається.

WorldWeatherBot – також проект, який на даний момент не працює. На нашу думку, тут така ж проблема, як з *Weatherman_bot* – розробники також не змогли реалізувати свою задумку до логічного кінця.

METAR Info Bot взагалі показує погоду в 3 аеропортах. Це звичайно корисно для клієнтів авіасполучень, але не для загального користувача, який збрається на роботу або планує свій тиждень відповідно до погоди.

Wradar_bot також не працює. По аналізі даного боту незрозуміло чи використовувався програмний код на ньому чи автори просто створили оболонку і наповнили текстом в *BotFather*.

Pogodarlingbot досить цікавий, є ідеї, які можна реалізувати в нашій роботі, але є одна важлива проблема – локалізація під різних користувачів.

Розглянувши основні проекти на Telegram Bot API бачимо що 70% взагалі не працює через незрозумілі причини. *Weatherman_bot* взалі був лідером на ринку з аудиторію понад 1 мільйон користувачів. Проаналізувавши інших ми помітили, що у них дуже незручний дизайн інтерфейсу. Одним з дуже вагомих недоліків є незручне використання та управління. Потрібно натиснути старт і вводити команди вручну такі як */help, /w, /add, /finish, /array, /ide, /choose, /find*. Це зручно та інтуїтивно зрозуміло для людей, які мають або мали якесь відношення до IT-технологій. Звичайний користувач не зрозуміє, як потрібно взаємодіяти з ботом після того, як натисне */start*. Тому слід використовувати *inline keyboard* або *simple keyboard*. Це дозволить конверсію користування проектом на постійній основі збільшити як мінімум на 50%.

Розглянемо основні переваги використання *inline keyboard:*

- Обробка на серверах. Ми ховаємо «страшний код» та обробку даних на серверах і користувач не повинен вводи їх, щоб отримати бажану інформацію.

- Автоматизація. Користувач одним натисканням запускає алгоритм, який пов'язаний з іншими алгоритмами.

- Швидкодія. Обробка інформації швидша ніж на Веб-ресурсах. Ми не будемо очікувати обробку кнопки, ми одразу отримуємо доступ по API. Телеграм не має проблем з падінням серверів, тому втрата швидкості на передачу дуже мінімальна.

- Захист. Користувач може бути впевненим, що його дані ніхто не отримає зі зловмисників (звичайно, якщо адміністратор даного ресурсу не буде зловмисником).

Давайте розглянемо один з незручних для користування телеграм ресурсів, для якого діє правило розробник – розробник (рис.1).

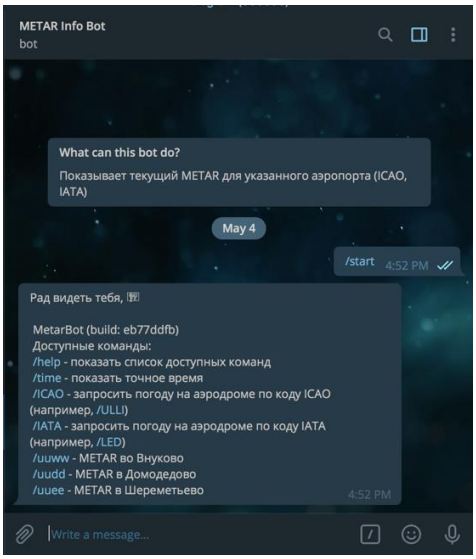

Рис. 1. Взаємодія через інтерфейс методом розробник – розробник

Після того як користувач натиснув старт, то в спрацьовує функція */start*, яка прийняла метод *message* і передає користувачеві набір команд, які незрозумілі для більшості людей є незрозумілими, тому ця зв'язка буде працювати: як розробник – розробник. Перейдемо до сучаснішої версії телеграм-ботів, коли розробник думає про масового користувача: розробник – користувач (рис.2).

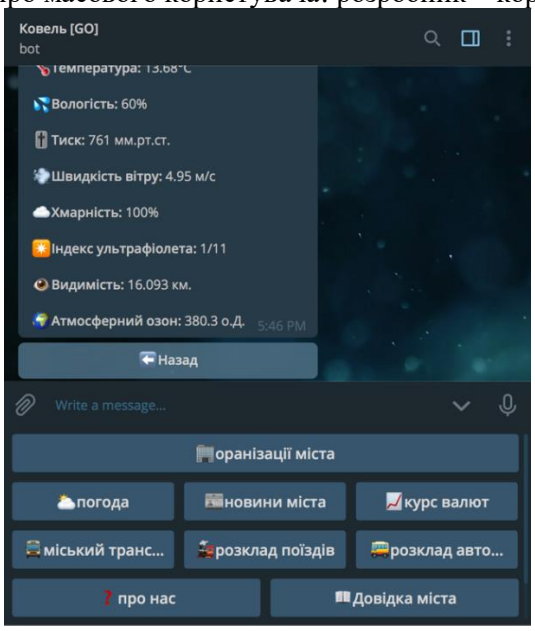

Рис. 2. Взаємодія через інтерфейс методом розробник – користувач.

Постановка проблеми. У роботі потрібно розглянути дослідження зміни погодних умов в різних районах міста за допомогою Telegram Bot API. Порівняти власну розробку з анологічними проектами телеграм та web-ресурсами для передачі даних по API в одній системі координат, використати найточніші показники, а також організувати найшвидшу передачу цих даних.

Результати отриманих досліджень. До основних етапів дослідження зміни погодних умов за допомогою *Telegram Bot API* слід віднести:

- збір інформації з ресурсів: *Weather API, DARK SKY API, OpenWeatherMap, Weatherbit API*;
- аналіз вибірок;
- організація швидкої передачі актуальної інформації користувачам.

Збір інформації відбувається із загальнодоступних ресурсів, потім сортуються масиви інформації для подальшої оптимізації та передачі на сервери *Telegram* для взаємодії з користувачем.

Інформація збирається за допомогою *API*, використовуємо для цього площадку *Telegram* отримавши *id-key-bot*. Багато компаній пропонують *API*, як готовий продукт. Наприклад, *Weather*

Underground продає доступ до свого *API* для отримання метеорологічних даних. Ми обираємо безкоштовні *API*, але якщо нам будуть обмежувати доступ до інформації, то будемо використовувати методи *Web Scaning*.

Сценарій використання. Отримавши *id-key-bot* ми підключаємо його до нашого програмного коду. Після цього ми можемо підключати *API* та передавати загальнодоступну інформацію в наш телеграм бот. Якщо нам потрібно масштабувати наш проект на якісь сторонні Web-ресурси, ми зможемо це зробити за допомогою парсингу нашої інформації.

Застосування API. API – це, в першу чергу, інтерфейс, який дозволяє розробникам використовувати готові блоки динамічної інформації для побудови програми. При використанні веб-додатків API може передавати інформацію у відмінному від стандартного *HTML* форматі, завдяки чому ним зручно користуватися при написанні власних програм. Сторонні загальнодоступні *API* найчастіше віддають дані в одному з двох форматів: *XML* або *JSON*. *JSON* набагато лаконічніший і простіший в читанні, ніж *XML*. Сервіси, що надають доступ до даних в *XML*-форматі, поступово відмовляються від нього.

Розглянемо один із прикладів *Weather API*:

```
{"coord":{"lon":-0.13,"lat":51.51},
"weather":[{"id":300,"main":"Drizzle","description":"light intensity
drizzle","icon":"09d"}],
"base":"stations",
"main":{"temp":280.32,"pressure":1012,"humidity":81,"temp_min":279.1
5,"temp_max":281},
"visibility":10000,
"wind":{"speed":4.1,"deg":80},
"clouds":{"all":90},
"dt":1485789600,
"sys":{"type":1,"id":5091,"message":0.0103,"country":"GB","sunrise":
1485762037,"sunset":1485794875},
"id":2643743,
"name":"London","cod":200}
```

Ми бачимо перед собою набір даних, за допомогою яких обираємо потрібну нам інформацію та в режимі *live* її відразу отримує кінцевий користувач.

Після цього переходимо на наступний етап – аналіз вибірок. Розглянемо алгоритм аналізу:

- Вибираємо основу вибірки, відкидаємо зайву інформацію. Упорядковуємо порядок списку одиниць відбору (опади, температура, швидкість вітру, вологість, хмарність та рекомендації одягу залежно від погодних умов з точністю до годин).
- Для кожної вибірки формуємо варіаційний ряд. Варіаційний ряд складаємо за однаковими ознаками та ділимо їх на кілька груп для детальнішого аналізу.
- Для того, щоб бачити наочно, побудуємо графіки та гістограми абсолютних частот.
- Проведемо аналіз по кожному району міста.
- Порівнюємо всі вибірки між собою та загальними даними по місту і вибираємо найкращий варіант для нашого дослідження.
- Проведемо паралелі по районах, порекомендуємо оптимальний одяг для переміщення об'єкта з точки А в точку Б, залежно від погодних умов.

Розглянемо швидкодію передачі даних. Після того, як в нас визначилась вибірка з найточнішими результатами, нам потрібно передати всю інформацію без втрат кінцевому користувачу. Для цього звичайно потрібна хороша мережа 4G і вище або високошвидкісний доступ до інтернету зі швидкістю більше 10 мбіт/сек. Також потрібно написати програмний алгоритм для швидшої передачі даних ніж O(n), а також бажано оптимізувати пам'ять під сервери Telegram.

Оскільки швидкість обробки даних за певну одиницю часу нам важливіша, ніж ресурси пам'яті, яких зараз достатньо, то саме до оцінки O(n) ми будемо прагнути оптимізувати наш алгоритм для більш швидкої передачі даних.

Після оптимізації скрипта до O(n) швидкість передачі даних в середньому становить 0.23 секунди (рис. 3), тобто користувач отримує інформацію менше, ніж за пів секунди. Ресурсів пам'яті також задіяно не багато (18.9 кб), оскільки ми не зберігаємо дані на сервері, наш бот слугує містком між інформацією на віддалених серверах або веб ресурсах до телеграм клієнта, яким користується споживач інформації. Виявилось, що pooling (звернення користувача до телеграм боту)

використовувати краще, оскільки взаємодія з нашим сервером відбувається швидше. Якби нам потрібно було б зашифрувати інформацію ми б використовували б webhook при цьому втрачали більше часу на орацювання запитів користувачів.

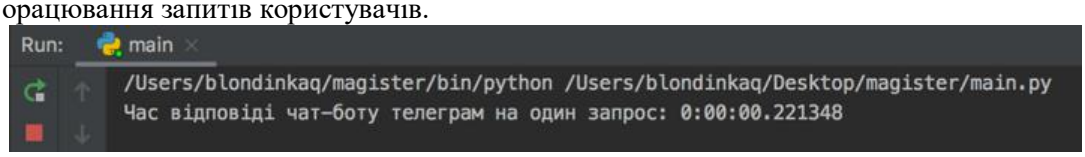

Рис. 3. Час передачі телеграм за допомогою Pooling

Висновки. У роботі розглянуто алгоритм дослідження зміни погодних умов за допомогою Telegram Bot API. Показано, що доступні сервіси телеграм зараз не працюють або вони не інтуїтивно зрозумілі для звичайного користувача. Розглянуто основні проблеми вирішення локалізації, вибір координатних даних, вибірка точніших та актуальніших даних, оптимізація передачі даних на стороні користувача та на стороні сервера, а також оптимізація алгоритму та програмного коду O(n). Локалізація слугує для того, щоб вміст продукту був наповнений всіма мовами для зручності користувачів, орієнтуючись не тільки на локальний ринок, а й на інші міжнародні ринки, проводячи аналіз інших досліджень, ця функція взагалі не була включена. Вибір координат – ще одна проблема інших продуктів, враховуючи великі мегаполіси або просто великі міста такі як: Лондон, Варшава, Київ, Стенфорд, не можна просто взяти одну точку координат та вказати в центральному парку, оскільки люди з окружних районів, яким до цієї точки 30 км будуть отримувати недостовірні дані, які можуть вплинути на плани та подальше користування продуктом користувача, тому нами прийнято рішення, що користувач сам повинен обирати точні дані для доступу до інформації. Вибірка точніших та актуальних даних за допомогою відкидання неточної інформації потрібна для того, щоб інформація не втрачала актуальності, оскільки дані в цій предметній області змінюються динамічно. Оптимізація передачі даних на стороні користувача не залежить від розробника, на це впливає апаратна частина користувача та операційні системи, на яких підтримуються додаток Telegram: Androin, IOS, Windows PC, LinuxOS, MacOs, а також доступ до інтернет мережі зі швидкістю більше 1 мбіт/сек.

Спираючись на низку фактів, можемо зробити висновок про те що, описаний вище алгоритм дасть змогу досягти швидшого та точнішого результату. Досліджуючи зміни погодних умов за допомогою *Telegram Bot API*, будемо передавати актуальну інформацію користувачам. Ми пропонуємо зручний інтерфейс, який збільшить кількість користувачів, після чого буде можливість масштабувати проект на інші *API* такі як *Facebook, Instagram, Viber, Discord* та інші.

Список бібліографічного опису

1. Янг А. Node.js в действии / А. Янг, М. Брэдли. – Питер, 2015. – 448 с.
2. Бхаргава М. Грокаем алгоритмы / Манджул Бхаргава. – Питер, 2017. – 288 с.
3. Васильєв О. Програмування мовою Python / Олексій Васильєв. – Львів: Богдан, 2019. – 504 с.
4. Пишемо telegram-бота на python за допомогою бібліотеки telebot частина 1 [Електронний ресурс] – Режим доступу до ресурсу: https://habr.com/ru/post/448310/
5. Node js документація [Електронний ресурс] – Режим доступу до ресурсу: https://nodejs.org/en/.
6. Фреймворк для работы с Telegraph API [Електронний ресурс] – Режим доступу до ресурсу: https://telegraf.js.org/#/.
7. Telegram Bot API [Електронний ресурс] – Режим доступу до ресурсу: https://core.telegram.org/bots/api.
8. Telegram API [Електронний ресурс] – Режим доступу до ресурсу: https://core.telegram.org/.
9. Создаем Telegram бота на API.AI [Електронний ресурс] – Режим доступу до ресурсу: https://habr.com/ru/post/336668/.
10. JSON в Python [Електронний ресурс] – Режим доступу до ресурсу: https://python-scripts.com/json.

References

1. Young A. Node.js in action / A. Young, M. Bradley. – Peter, 2015. – 448 p.
2. Bhargava M. Grokaem algorithms / Manjul Bhargava. – Peter, 2017. – 288 p.
3. Vasiliev O. Programming in Python / Alexei Vasiliev. – Lviv: Bogdan, 2019. – 504 p.
4. Write a telegram-bot in python using the telebot library part 1 [Electronic resource] – Mode of access to the resource: https://habr.com/ru/post/448310/
5. Node js documentation [Electronic resource] – Mode of access to the resource: https://nodejs.org/en/.
6. Framework for working with the Telegraph API [Electronic resource] – Resource access mode: https://telegraf.js.org/#/.
7. Telegram Bot API [Electronic resource] – Mode of access to the resource: https://core.telegram.org/bots/api.
8. Telegram API [Electronic resource] – Resource access mode: https://core.telegram.org/.
9. Create a Telegram bot on API.AI [Electronic resource] – Mode of access to the resource: https://habr.com/ru/post/336668/.
10. JSON in Python [Electronic resource] – Mode of access to the resource: https://python-scripts.com/json.