

УДК 004.8:65.05:681.5

DOI: 10.31673/2786-8362.2024.021372

Катков Ю.І., д.т.н.; Березовська Ю.В., PhD;
Саміляк І.М., Почтовик А.Р.

ЗАСТОСУВАННЯ ЗАСОБІВ WEBHOOK У СИСТЕМАХ ОБРОБКИ ТА УПРАВЛІННЯ СПОВІЩЕННЯМ ДЛЯ ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ СЕРВЕРА

Katkov Yu.I., Berezovska Yu.V., Samiliak I.M., Pochtovyk A.R. Implementation of Webhook features in processing systems and management of alerts to improve server productivity. Web application servers are operational tools that facilitate interaction between publishers and subscribers when using various applications to process client requests according to the requirements of receiving information from the applications. Monitoring the performance of the web application server is crucial for quality control. One way to improve the performance of web application servers is by implementing processing and notification management systems for these servers. These systems are an integral part of informing users, engaging them, and ensuring the relevance of information. Traditionally, such systems used the API (Application Programming Interface) of push notification services. The API can be seen as the interaction between publishers and subscribers using requests and responses. However, an API is a program with functions or calls for adding, editing, and retrieving data. Using an API requires handling all the work of processing and managing notifications independently. To automate the process of processing and managing notifications in web application servers, a new approach called Webhook has been developed. Applying Webhook tools in notification management systems to increase server performance is an innovative area that enables computers and programs to perform tasks differently. A Webhook is a method to augment or alter the behavior of a web page or application program on a web server using reverse calls. It can be considered as a notification mechanism for the publisher-subscriber system about events, allowing one system to notify another about specific events in real-time. The article discusses the problem of applying Webhooks. To address this problem, the following tasks are proposed: when to use Webhooks; what advantages application of Webhooks allows obtaining; what disadvantages are associated with using Webhooks; how to configure and use Webhooks; what recommendations can be offered for using Webhooks. Such tasks are timely and relevant.

Keywords: Webhook, management, notifications

Катков Ю.І., Березовська Ю.В., Саміляк І.М., Почтовик А.Р. Застосування засобів Webhook у системах обробки та управління сповіщенням для підвищення продуктивності сервера. Веб-сервер додатків – це робочі засоби, які забезпечують взаємодію видавця-передплатника під час використання різноманітних додатків для обробки навантаження з обслуговування клієнтів відповідно до вимог отримання інформації від додатків. Для контролю за якістю функціонування веб-сервера додатків виконується моніторинг продуктивності цього сервера, що має вирішальне значення. Одним зі способів підвищення продуктивності веб-сервера додатків є застосування систем обробки та управління сповіщеннями цих серверів. Системи обробки та управління сповіщеннями є невід’ємною частиною інформування користувачів, їх залучення та актуальної інформації. Традиційно для роботи таких систем застосовувався інтерфейс API (Application Programming Interface) служби push-повідомлень. Інтерфейс можна розглядати як взаємодію видавця-передплатника один з одним, використовуючи запити та відповіді. Але API – це програма з функціями або дзвінками для додавання, редагування та отримання даних. Використання API вимагає виконувати всю роботу щодо обробки та управління сповіщенням самостійно. Для автоматизації процесу обробки та управління сповіщеннями у веб-сервера додатків розроблено новий підхід, який має назву Webhook. Застосування засобів Webhook у системах управління сповіщень для підвищення продуктивності сервера – це інноваційна галузь, яка створює можливість комп’ютерам і програмам по іншому виконувати завдання. Webhook – це метод щоб збільшити або змінити поведінку веб-сторінки або програми додатку в веб-сервері за допомогою зворотних дзвінків. Можна казати, що це механізм оповіщення системи видавця-передплатника про події, який дозволяє одній системі повідомляти іншу систему про конкретні події в режимі реального часу. В статті розглядається проблема застосування Webhook. Для вирішення цієї проблеми пропонується розглянути такі завдання: коли потрібно використовувати Webhook; які переваги дозволяє отримати застосування Webhook; які недоліки використання Webhook; як виконувати процес налаштування та використання Webhook; які можна запропонувати рекомендації щодо використання Webhook. Дані завдання є своєчасними та актуальними.

Ключові слова: Webhook, управління, сповіщення

Вступ

Веб-сервер додатків – це робочі засоби, які забезпечують взаємодію видавця-передплатника під час використання різноманітних додатків для обробки навантаження з обслуговування клієнтів відповідно до вимог отримання інформації від додатків. Для контролю за якістю функціонування веб-сервера додатків виконується моніторинг продуктивності цього сервера, що має вирішальне значення. Одним зі способів підвищення продуктивності веб-сервера додатків є застосування систем обробки та управління сповіщенням цих серверів.

Системи обробки та управління сповіщенням є невід’ємною частиною інформування користувачів, їх залучення та актуальної інформації. Традиційно для роботи таких систем застосовувався інтерфейс API (Application Programming Interface) служби push-повідомлень. Інтерфейс можна розглядати як взаємодію видавця-передплатника один з одним, використовуючи запити та відповіді. Але API – це програма з функціями або дзвінками для додавання, редагування та отримання даних. Використання API вимагає виконувати всю роботу щодо обробки та управління сповіщенням самостійно.

Для автоматизації процесу обробки та управління сповіщенням у веб-сервера додатків розроблено новий підхід, якій має назву Webhook. Webhook – це метод щоб збільшити або змінити поведінку веб-сторінки або програми додатку в веб-сервері за допомогою зворотних дзвінків. Можна казати, що це механізм оповіщення системи видавця-передплатника про події, який дозволяє одній системі повідомляти іншу систему про конкретні події в режимі реального часу.

Тому виникає проблема дослідження Webhook для застосування. Для вирішення цієї проблеми необхідно розглянути наступні завдання: коли потрібно використовувати Webhook; які переваги дозволяє отримати застосування Webhook; які недоліки використання Webhook; як виконувати процес налаштування та використання Webhook; які можна запропонувати рекомендації щодо використання Webhook. Дані завдання є своєчасними та актуальними.

Аналіз останніх наукових досліджень і публікацій. Зростання швидкості інтеграції додатків SaaS з додатками користувача призвела до включення веб-перехоплювачів у більшість програмних додатків інфраструктури бізнес-процесів. Одним зі способів автоматизування бізнесу, оптимізації процесів, які зазвичай залежать від ручного введення, що в кінцевому підсумку заощадить час та енергію, є впровадження технології Webhook. Застосування засобів Webhook у системах управління сповіщення для підвищення продуктивності сервера створює можливість комп’ютерам і програмам по іншому виконувати завдання. Webhook найбільш поширені на платформах SaaS, таких як GitHub, Shopify, Stripe, Twilio та Slack, оскільки вони підтримують різні типи подій залежно від дій, що відбуваються всередині них. Питанню впровадження технології Webhook присвячено багато досліджень та наукових праць [1-5]. Щодо питань опису для адміністраторів-початківців серверів додатків, то не існує загального опису.

Метою роботи є визначення особливостей застосування Webhook у системах обробки та управління сповіщення для підвищення продуктивності сервера.

Виклад основного матеріалу дослідження.

Застосування засобів Webhook у системах обробки та управління сповіщеннями для підвищення продуктивності сервера вимагає ретельного розгляду.

Сервери – це робочі засоби, які керують обчислювальною потужністю цифрової економіки. Існує багато видів серверів. У статті будемо розглядати Web Server (мережевий або веб-сервер додатків), що являє собою засіб (комп’ютер чи програму), призначений для обробки запитів і передачі даних на інші комп’ютери (клієнти) по локальній мережі або через Інтернет. Сервер додатків – сервер, що виконує деякі прикладні програми. Термін також відноситься і до програмного забезпечення, що встановлено на такому сервері і забезпечує виконання прикладного завдання. Тобто допомагає іншим комп’ютерам або пристроям отримувати та обмінюватися інформацією про функціонування додатку в кінцевих пристроях

користувачів [6].

Принцип роботи веб-сервера додатків полягає у тому, що він приймає HTTP-запити від клієнтів (веббраузерів) та видає їм HTTP-відповіді, зазвичай разом з HTML-сторінкою, зображенням, файлом, медіа-потокком або іншими даними. Тобто, веб-сервери додатків зазвичай конфігуруються для обробки навантаження з обслуговування клієнтів відповідно до вимог отримання інформації від додатків. Протокол передачі гіпертексту (Hypertext Transfer Protocol, HTTP) призначений для забезпечення зв'язку між клієнтами та серверами. HTTP працює як протокол запиту-відповіді між клієнтом та сервером. Веб-браузер може бути клієнтом, а додаток на комп'ютері, на якому розміщений веб-сайт, може бути сервером.

Для контролю за якістю функціонування веб-сервер додатків виконується моніторинг продуктивності цього сервера, що має вирішальне значення. Тому на показники продуктивності веб-сервера додатків слід завжди звертати увагу. Моніторинг веб-сервера додатків вирішує це завдання. Він дає уявлення про його продуктивність, що є життєво важливим для запобігання збоїв у бізнес-операціях. Одним зі способів підвищення продуктивності веб-сервера додатків є застосування систем обробки та управління сповіщеннями цих серверів.

Система обробки та управління сповіщеннями допомагає забезпечити точну доставку важливої інформації потрібній людині з додатку в потрібний час, не залежно від дій співробітників. Системи обробки та управління сповіщеннями є невід'ємною частиною інформування користувачів, їх залучення та актуальної інформації. Користуючись даною системою можна негайно повідомити про підтвердження замовлення, замовлення доставки або зміни специфікацій. Такі системи відрізняються за складністю та масштабом залежно від конкретних вимог додатків або послуг, які вони обслуговують, а також від місця розташування. У статті розглядаються системи обробки та управління сповіщеннями у веб-сервера додатків.

Системи обробки та управління сповіщеннями в серверах працюють наступним чином. Веб-сервер додатків відправляє вміст сповіщення та токени пристроїв в API (Application Programming Interface) служби push-повідомлень. API означає інтерфейс прикладного програмування. Програмний інтерфейс – містить важливі підпрограми, протоколи взаємодій та особливості для створення програмного забезпечення. У контексті API слово «Додаток» стосується будь-якого програмного забезпечення з певною функцією. Інтерфейс можна розглядати як контракт обслуговування між двома програмами. Цей контракт визначає, як вони взаємодіють один з одним, використовуючи запити та відповіді. Тобто, як тільки сервер відправить запит до служби push-повідомлень, служба відправить повідомлення на пристрої користувачів. API – це програма з функціями або дзвінками для додавання, редагування та отримання даних. Використання API вимагає виконувати всю роботу щодо обробки та управління сповіщенням самостійно.

Служба push-повідомлень – технологія, яку використовують для надсилання повідомлень користувачам. Спливаючі повідомлення або оповіщення програм на мобільному пристрої – це так звані сигнали зв'язку від сервера додатків. Сервер, а не користувач, ініціює обмін push-повідомленнями. Програми використовують push-сповіщення, щоб інформувати користувачів про майбутні угоди, події в додатку або іншу інформацію, яка може їх зацікавити. Хоча мобільні push-сповіщення є найбільш поширеним типом, існують також push-сповіщення через Інтернет, переносні push-сповіщення та голосові push-сповіщення. Push-повідомлення доставляються із завантаженої програми на мобільний телефон [7].

Для автоматизації процесу обробки та управління сповіщенням у веб-сервер додатків розроблено новий підхід, який має назву Webhook.

Webhook – це метод збільшити або змінити поведінку веб-сторінки або ж програми додатку в веб-сервері за допомогою зворотних дзвінків. Можна казати, що це механізм оповіщення системи видавця-передплатника про події, який дозволяє одній системі повідомляти іншу систему про конкретні події в режимі реального часу. Для розуміння як працює Webhook необхідно в першу чергу визначити що це таке. Назва webhook є простою

комбінацією слова *web*, що відноситься до його зв'язку на основі HTTP, і функції *hook* програмування перехоплення (веб-перехоплювач), яка дозволяє програмам перехоплювати виклики або інші події, які можуть представляти інтерес. Прикладом повідомлення, яке надсилається до зовнішньої системи можуть бути: CRM, інтернет-магазин, кол-центр тощо. Тобто виглядає Webhook як посилання: *http://mysite.ru/webhook?id={id}&type={type}*. Це програмний код, що зазвичай складається з двох частин – змінної та самих даних.

Порівняння Webhook/API. Webhook схожий на API, але простіший. Якщо є вимога дзвінка для додавання, редагування та отримання даних, доводиться виконувати всю роботу самостійно, то застосувавши Webhook, забезпечується створення автоматичного повідомлення, яке надсилається з веб-застосунку (веб-додатку) іншим веб-застосунком. Звідси Webhook має функцію зворотного виклику. Функція зворотного виклику створює код, що аналогічний коду, який реалізує Callback-функцію (або зворотний виклик). Код передає повідомлення запит в іншу функцію як аргумент, а потім викликається після завершення будь-якої дії результат.

Технічно Webhook є веб-перехоплювачем, що будується за допомогою протоколу HTTP.

HTTP – протокол передачі даних, що використовується в комп'ютерних мережах. Назва скорочена від *HyperText Transfer Protocol*, протокол передачі гіпертекстових документів HTTP належить до протоколів моделі OSI 7-го прикладного рівня. Веб-перехоплювач забезпечує спрощений керований подіями зворотний зв'язок між двома інтерфейсами видавця-передплатника, що визначаються користувачем.

Веб-перехоплювачі – це дані та виконувані команди, що надсилаються з однієї програми до іншої через HTTP, а не через командний рядок на комп'ютері, у форматі XML, JSON або серіалізації з кодуванням форми. Їх називають веб-перехоплювачами, оскільки вони є програмними перехоплювачами – або функціями, які запускаються, коли щось відбувається і працюють через Інтернет. Вони зазвичай захищені за рахунок скритності – кожен користувач програми отримує унікальну випадкову URL-адресу для відправки даних веб-перехоплювача – хоча за бажанням їх можна захистити за допомогою ключа або підпису.

Для розуміння як працює Webhook розглянемо приклад. Припустимо, необхідно знати щоразу, коли погода опускається нижче 0 градусів за Цельсієм. Для цього необхідно, щоб метеорологічна служба передавала цю інформацію через інтеграцію веб-перехоплювача. Створюється URL-адреса веб-перехоплювача у своїй системі, яка під час виклику повідомляє, що температура впала нижче 0 градусів за Цельсієм. Після реєстрації цієї URL-адреси у метеорологічній службі, дана служба викликає автоматично URL-адресу кожного разу, коли температура падає нижче 0 градусів за Цельсієм. Тобто запит веб-перехоплювача – це тип дзвінка HTTP, який визначається користувачем. По суті, це повідомлення про подію через HTTP POST. Коли у вихідній програмі відбувається певна подія, вона запускає веб-перехоплювач, який потім відправляє повідомлення, зазвичай у форматі JSON або XML, на вказану URL-адресу – URL-адресу веб-перехоплювача. Це повідомлення містить інформацію про подію, яка щойно відбулася. Звідси Webhook – це інструмент, який використовується у веб-розробці або розробці API, що дозволяє програмам надавати інформацію в реальному часі іншим програмам при виникненні певної події. У них є корисне навантаження, яке відправляється на унікальну URL-адресу, надаючи програмам простий спосіб надсилати дані один одному. Тобто Webhook – це URL-адреса, яка надається іншій системі, яку система викликає, коли відбувається якась подія, про яку необхідно знати.

Використовувати Webhook для відстеження на стороні сервера просто. За допомогою веб-перехоплювачів дані можуть автоматично відправлятися в серверний контейнер диспетчера тегів Google щоразу, коли на веб-сайті відбувається будь-яка подія або дія. Найбільш поширеним варіантом використання веб-перехоплювача при додаванні тегів на стороні сервера є надсилання веб-перехоплювачів із CRM або CMS. Наприклад, кожного разу, коли створюється, оновлюється, розміщується замовлення тощо можна надсилати ці дані на сервер GTM.

Функція програмування перехоплення в Webhook, дозволяє створювати веб-перехоплювачі, які перехоплюють подію, що відбувається в серверній програмі, і пропонують серверу відправити корисне навантаження клієнту через Інтернет. Веб-перехоплювачі використовуються безліччю веб-додатків для отримання невеликих обсягів даних від інших програм, але веб-перехоплювачі також можна використовувати для запуску робочих процесів автоматизації в середовищах GitOps. GitOps – це операційне середовище, яке використовує найкращі практики DevOps, що використовуються для розробки програм, такі як контроль версій, спільна робота, відповідність вимогам та CI/CD, та застосовує їх для автоматизації інфраструктури [8].

Таким чином, веб-перехоплювачі діють як віртуальні повідомлення між програмами, які запускаються якоюсь подією, наприклад, необхідно отримувати повідомлення кожного разу, коли з'являється новий передплатник. Працюючи в режимі реального часу, Webhook дозволяють передавати дані з одного додатку (застосунку) до іншого і можуть допомогти уточнити дії, які здійснюють передплатники. Можна казати що, це новий підхід до архітектури програмного забезпечення, який дозволяє програмам і службам надсилати щоразу веб-повідомлення іншим програмам.

Необхідно зазначити, що Webhook потужний інструмент, який дозволяє розробникам інформувати кілька сервісів про оновлений контент або дані користувача. Його можна використовувати практично для чого завгодно, включаючи push-повідомлення про відправку товару або відправку платіжної інформації під час здійснення покупки. Він забезпечує простоту та зручність без необхідності ручного введення або проміжного програмного забезпечення, що робить веб-розробку більш простою, ніж будь-коли. Це дозволяє веб-сторінкам залишатися синхронізованими, оскільки веб-перехоплювачі дозволяють розробникам швидко підключати два веб-сервіси та автоматично передавати інформацію між ними в режимі реального часу. Простий спосіб, за допомогою якого онлайн-акаунти можуть "спілкуватися" один з одним і автоматично отримувати повідомлення, коли відбувається щось нове.

Webhook зазвичай використовуються для з'єднання двох різних програм. Коли в програмі-тригері відбувається подія, вона серіалізує дані про цю подію і відправляє їх на URL-адресу веб-перехоплювача з програми дії – тієї, в якій необхідно щось виконати на основі даних з першої програми. Прикладами різноманітних програм застосування Webhook є Телеграм, HABR, TimePad, PayPal, Twilio, WooCommerce тощо.

Webhook в Телеграм (багатоплатформовий месенджер) – це URL, який ви надаєте телеграму, і за яким телеграм буде відправляти оновлення у форматі JSON щоразу, коли вашому боту надходить повідомлення або відбувається інша подія.

Webhook в HABR (вебсайт, який поєднує ознаки соціальної мережі й колективного блогу) – це механізм оповіщення системи про події. Припустимо, є певний сервіс, який повинен сповіщати про події, коли вони відбуваються. Як варіант постійно запитувати нові дані на сервері, мінус буде швидше в швидкості отримання даних, якщо на сервері наприклад стоїть затримка на отримання нових оповіщень.

Webhook в TimePad (сервіс, який допомагає організаторам проводити заходи, а учасникам знаходити кращі події) – механізм отримання повідомлень про певні події на TimePad (в основному про дії користувачів) на свій власний сайт. Дозволяє відслідковувати зміни статусів реєстрації.

Webhook в PayPal (міжнародна електронна платіжна система), повідомляє ваш бухгалтерський додаток, коли ваші клієнти платять вам.

Webhook в Twilio (американська компанія, яка надає інструменти для здійснення та отримання телефонних дзвінків, надсилання та отримання текстових повідомлень перенаправляє телефонні дзвінки на ваш номер) і WooCommerce (плагін електронної комерції для WordPress) може повідомляти вас про нові замовлення в Slack (корпоративний месенджер).

Особливості під час застосування Webhook. По-перше, розглянемо коли необхідно використовувати Webhook:

1. Односторонній зв'язок у реальному часі (від джерела до місця призначення);
2. Постійне з'єднання між зв'язком двох систем;
3. Необхідно негайно відреагувати на подію з SaaS, яка підтримує веб-перехоплювачі;
4. Є потреба використовувати модель push для негайного надсилання оновлень.

По-друге, розглянемо які переваги дозволяє отримати застосування Webhook:

1. *Усуває потребу опитування:* ефективні та швидкі, оскільки усувають необхідність опитування чи оновлення для перевірки нових даних. Опитування – веб-програма неодноразово запитує дані в іншій веб-програмі через певні проміжки часу, що може споживати смугу пропускання та ресурси. Оновлення – користувач перезавантажує веб-сторінку для перегляду нових даних, що може бути незручно та повільно. З іншого боку, веб-перехоплювачі передають дані передплатнику, як тільки подія відбувається, що зменшує затримку та покращує взаємодію з користувачем. Це заощаджує ресурси клієнтської програми;

2. *Швидко налаштовуються:* якщо програма підтримує веб-перехоплювачі, їх легко налаштувати через інтерфейс серверної програми. Клієнт вводить URL-адресу веб-перехоплювача своєї програми та встановлює деякі основні параметри, наприклад, яка подія його цікавить;

3. *Автоматизує передачу даних:* корисне навантаження відправляється, коли вказана подія відбувається в серверній програмі. Цей обмін ініціює подія, тому він відбувається так швидко, як тільки дані можуть бути передані з сервера на пристрій клієнта – у режимі реального часу, наскільки це можливо при будь-якій передачі даних;

4. *Забезпечує обмін у формі коротких сповіщень про специфічні корисні навантаження:* Webhook покладаються на те, що сервер визначає обсяг даних, який надсилається, залишаючи клієнту можливість інтерпретувати корисне навантаження і використовувати його продуктивним чином. Оскільки клієнт не контролює точний час або розмір передачі, веб-перехоплювачі обробляють невеликі обсяги інформації між двома кінцевими точками, часто у формі сповіщення;

5. *Оновлення в режимі реального часу:* веб-перехоплювачі забезпечують миттєві повідомлення, на відміну від механізмів опитування, які потребують регулярної перевірки оновлень;

6. *Ефективність:* знижують навантаження на сервер і підвищують продуктивність, оскільки усувають необхідність частого опитування. Прикладом є пакетна обробка: веб-перехоплювач може використовуватися для повідомлення про завершення завдання;

7. *Гнучкими для налаштування:* можна визначити логіку для керування тим, які події запускають запити веб-перехоплювача та які дані надсилаються. Іноді необхідна лише конкретна інформація. Веб-перехоплювачі також можуть підтримувати кілька передплатників на ту саму подію, що забезпечує інтеграцію та автоматизацію різних веб-додатків. Наприклад, у Stripe є чудова пропозиція веб-перехоплювачів. Замість того, щоб отримувати інформацію про кожну подію, що відбувається в Stripe, а їх дуже багато, можна зареєструватися на конкретні події, які цікавлять;

8. *Можуть видаляти повідомлення:* Webhook може надсилати повідомлення до текстового каналу без необхідності входу в систему як бот. Вони також можуть отримувати, редагувати та видаляти власні повідомлення.

По-третє, розглянемо недоліки використання Webhook:

1. *Не гарантують доставку або підтвердження:* дані або повідомлення не завжди можуть бути надіслані до цільової системи. Це означає, що видавець не знає, чи передплатник успішно отримав і обробив веб-перехоплювач. Це може призвести до втрати, неузгодженості або дублювання даних. Якщо вихідна система простоює, вона може не захопити відповідні дані або сповіщення; коли це станеться, повідомлення або дані не будуть надіслані до цільової системи;

2. Доставка *Webhook* може завершитися невдачею з кількох причин: наприклад, якщо сервер не працює або на відповідь необхідно більше 10 секунд, GitHub зареєструє доставку як збій. GitHub не виконує автоматичну повторну доставку невдалих поставок;

3. Основною вразливістю будь-якого сервісу веб-перехоплювачів є підrobка запитів на стороні сервера (*SSRF*): *SSRF* – коли зловмисник змушує службу виконати внутрішній ненавмисний запит у власній мережі. *Webhook* – ідеальна мета для цього;

4. *Webhook* не надійні та небезпечні, а це означає, що вони схильні до збоїв, які можуть виникнути з кількох причин, наприклад: збій мережі, погані запити, помилка автентифікації. Якщо один з них перебуває в автономному режимі, перевантажений або скомпрометований, *Webhook* може вийти з ладу, затриматися або бути перехопленим;

5. *Webhook* складні і ними важко керувати, оскільки вони вимагають від видавця та передплатника налаштування та підтримки кінцевих точок веб-перехоплювачів, автентифікації та обробки помилок. Веб-перехоплювачі також збільшують зв'язок і залежність між веб-додатками, що може ускладнити їх масштабування, оновлення або налагодження.

Процес налаштування та використання *Webhook*. Процес налаштування та використання *Webhook* включає кілька етапів: конфігурація, подія-тригер, повідомлення та дія.

Конфігурація. Застосування засобів *Webhook* у системах обробки та управління сповіщенням для підвищення продуктивності сервера має сценарії відносно наступної схеми: сервер веб-додатків/панель моніторингу веб-додатків. У цьому сценарії веб-додаток має інтерфейс панель та сервер. Наприклад, крок 1 – можна налаштувати URL-адресу веб-перехоплювача на сервері для отримання повідомлення про успішну публікацію запланованого повідомлення на Facebook; крок 2 – *Webhook* сервера отримує це повідомлення; крок 3 – потім запускає процес оновлення деяких даних та панелі керування користувача; крок 4 – кінцевий користувач побачить на панелі керування нові дані, що повідомлення було успішно опубліковано.

Подія-тригер. Тригерні сповіщення – це *push*-сповіщення, які автоматично активуються, коли на веб-сайті відбувається певна подія користувача. Події – це взаємодія користувача з контентом на сторінці, яка може викликати автоматичне *push*-сповіщення. Щоб розпочати роботу з веб-перехоплювачами на стороні сервера, необхідно спочатку вирішити, яку програму (додаток) ви хочете запускати і яка програма буде їх отримувати. Зазвичай, надсилання додатку називається «джерелом», а приймання додатку – «призначенням». Після того, як обрано обидві програми, наступним кроком буде налаштування параметрів веб-перехоплювача кожного з них.

Повідомлення та дія. *Webhook* спрацьовує, коли на сайті, у CRM, чат-боті чи інших системах відбувається подія. Запускається тригер *Webhook* – це автоматичний тип тригера, який прослуховує певний тип даних, подібно до тригерів подій. У той час як тригери подій використовуються для активації тригера на основі внутрішньої активності, веб-перехоплювачі замість цього використовуються при активації тригера на основі зовнішньої активності. Наприклад, людина написала коментар або додала новий товар до системи обліку товарів. Коли ця подія відбувається, сервер створює HTTP-запит і надсилає його на адресу, вказану клієнтом, для отримання веб-перехоплювачів. Клієнт вчасно отримує нові дані – клієнт задоволений. Користувач може налаштувати веб-перехоплювачі таким чином, щоб події на деяких сайтах запускали дії на інших сайтах. Наприклад, людина створює замовлення в інтернет-магазині → система відправляє *Webhook* у додаток власника → додаток повідомляє власника та відправляє йому оцінку.

Налаштовування веб-перехоплювачів. У випадку з веб-перехоплювачами процес налаштування зазвичай складається з трьох етапів:

1. Отримання URL-адреси веб-перехоплювача з програми, до якої потрібно надіслати дані;
2. Використання даної URL-адреси в розділі веб-перехоплювача програми, з якої необхідно отримувати дані;
3. Обрати тип подій, про які необхідно, щоб повідомляла програма.

Після налаштування, коли додається новий контакт, програма автоматично передає дані на URL-адресу веб-перехоплювачів іншої програми. Це просте з'єднання один до одного, яке запускається автоматично.

У п'ятих, розглянемо рекомендації щодо використання Webhook. Щоб подолати деякі недоліки веб-перехоплювачів і скористатися їхніми перевагами, наведені нижче рекомендації щодо використання веб-перехоплювачів для доставки даних у реальному часі:

1. Необхідно використовувати HTTPS та шифрування, щоб забезпечити зв'язок через веб-перехоплювач та захистити дані від несанкціонованого доступу чи зміни;
2. Використовувати повтори та підтвердження, щоб забезпечити доставку та обробку веб-перехоплювачів, а також коректно обробляти помилки та збої;
3. Використовувати фільтри та перехоплювачі, щоб вказати події й дані, які необхідно надсилати та отримувати, а також уникати надсилання непотрібних або конфіденційних даних;
4. Використовувати журнали і моніторинг для відстеження й аналізу активності та продуктивності веб-перехоплювачів, а також для виявлення й усунення будь-яких проблем та аномалій.

Висновки

Таким чином, Webhook пропонують простий та зручний спосіб отримувати оновлення повідомлень про дії в Інтернеті в режимі реального часу, дозволяючи швидко реагувати на ці повідомлення та діяти відповідно до них з мінімальними зусиллями. Крім того, веб-перехоплювачі є безпечним способом обміну інформацією з іншими веб-сервісами, такими як маркетингові або аналітичні платформи, оскільки вони не вимагають від розробників або програмістів надання користувачам облікових даних для доступу або конфіденційних даних. Визначення цього моменту, коли відбувається подія, яка запускає інтеграцію веб-перехоплювача, що має вплив на підвищення продуктивності сервера, може бути визначена автоматично за рахунок спеціальних команд.

Webhook має великі перспективи використання у багатьох випадках, про що свідчить цінність використання Webhook на платформах електронної комерції: події веб-перехоплювача використовуються для оновлення статусу замовлення, відстеження постачання та управління запасами.

Список використаної літератури:

1. What is a Webhook? How They Work With Examples URL: <https://www.ayrshare.com/what-is-a-webhook-how-they-work-with-examples/>.
2. What is a Webhook? Understanding the difference and how to use them URL: <https://cyclr.com/blog/what-is-a-webhook>.
3. What is a webhook? URL: <https://www.redhat.com/en/topics/automation/what-is-a-webhook>.
4. What are webhooks and their use in server-side tracking Examples URL: <https://stape.io/blog/what-are-webhooks-and-their-use-in-server-side-tracking>.
5. How To Effectively Monitor Server Performance Examples URL: <https://theqlead.com/test-management/server-performance-monitoring/>.
6. Callback function Examples URL:

https://developer.mozilla.org/ru/docs/Glossary/Callback_function.

7. What is a Push Notification Service? URL: <https://aws.amazon.com/what-is/push-notification-service/>.

8. How to Take Control of Your Webhook Reliability Examples URL: <https://hookdeck.com/webhooks/guides/taking-control-of-your-webhook-reliability>.

9. What are the benefits and drawbacks of webhooks for real-time data delivery? Examples URL: <https://www.linkedin.com/advice/0/what-benefits-drawbacks-webhooks-real-time-data>.

Автори статті

Катков Юрій – доктор технічних наук, доцент, професор, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0009-0007-1194-4014

Березовська Юлія – PhD, доцент, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0000-0002-9973-0497

Саміляк Іван – аспірант, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0009-0008-2157-1263

Почтовик Аліна – студентка, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0009-0008-7774-8945

Authors of the article

Katkov Yuriy – Doctor of Science (technic), Associate Professor, State University of Information and Communication Technologies, Kyiv, Ukraine.

ORCID: 0009-0007-1194-4014

Berezovska Yuliia – PhD, Associate Professor, State University of Information and Communication Technologies, Kyiv, Ukraine.

ORCID: 0000-0002-9973-0497

Samiliak Ivan – postgraduate, State University of Information and Communication Technologies, Kyiv, Ukraine.

ORCID: 0009-0008-2157-1263

Pochtovyk Alina – student, State University of Information and Communication Technologies, Kyiv, Ukraine.

ORCID: 0009-0008-7774-8945