

Денис Лесишак, Світлана Яцишин*

¹ НЛТУ України, Львів, Україна, 23lesyshak.d@nltu.lviv.ua

² НЛТУ України, Львів, Україна. ORCID: 0000-0001-5200-4837,
yatsyshyn@nltu.edu.ua

Принцип роботи HTTP проксі сервера з використанням Service Worker

Анотація. У цій роботі здійснено дослідження механізму роботи проксі-сервера на базі технології Service Worker. На основі цієї архітектури реалізовано вебсервіс, що дає змогу користувачам отримувати доступ до інтернет-ресурсів через проксі-сервер за посиланням, без потреби в налаштуванні комп'ютера чи браузера. Тестування показало високу сумісність при відкритті сайтів через проксі.

Ключові слова – проксі, Service Worker, HTTP, браузер, .NET

У сучасному цифровому світі доступ до інформації є критично важливим, однак інтернет-цензура та обмеження доступу до певних вебресурсів залишаються поширеними проблемами. Щоб обійти ці обмеження, користувачі часто вдаються до проксі-сервісів або VPN, які дають змогу приховувати IP-адресу, місцезнаходження та трафік, забезпечуючи анонімний і безпечний доступ до інтернет-ресурсів [1].

Кожен спосіб проксування трафіку має свої переваги й недоліки. Проксі-сервіси зазвичай не шифрують трафік, обмежені роботою з HTTP-протоколом, але їх легко налаштувати в браузері, і вони відносно прості в реалізації порівняно з VPN. Сьогодні користувачам доступний широкий вибір технологій, що дає змогу обрати оптимальний варіант залежно від потреб.

Сучасні вебтехнології відкривають нові можливості для створення проксі-серверів, які значно гнучкіші та ефективніші, ніж традиційні HTTP- чи SOCKS-проксі. Однією з таких технологій є Service Worker, яка дає змогу організувати повноцінний HTTP-проксі безпосередньо в браузері користувача.

Дана робота присвячена розробці вебзастосунку HTTP-проксі-сервера з використанням Service Worker API для проксування HTTP-запитів. Пропонований підхід дає змогу обійтися без додаткового налаштування на стороні клієн-

та, забезпечуючи безпечний та надійний доступ до інтернету з підтримкою сучасних вебсайтів.

Service Worker API. Service worker – це «проксі», що знаходяться між вебзастосунками, браузером та мережею (коли вона доступна). Він призначений, зокрема, для створення ефективного досвіду роботи в режимі офлайн, виконання відповідних дій залежно від наявності мережі, кешування, а також оновлення ресурсів, які зберігаються на сервері [2]. Це фоновий процес, який не має доступу до DOM безпосередньо та представляє собою JavaScript-файл, що може контролювати вебсторінку або сайт.

Основною функцією Service Worker API, що є ключовою для реалізації проксі-сервісу безпосередньо в браузері, є можливість перехоплення мережеских запитів. Це дає змогу нам перехоплювати всі запити, які ініціює користувач через веббраузер, і перенаправляти їх на власний вебсервіс. В свою чергу вебсервіс діє як проксі-сервер, самостійно відправляючи запити до потрібних ресурсів та повертаючи результати користувачу.

Щоб почати використовувати Service Worker, спершу потрібно переконатися, що вебсайт працює через захищений протокол HTTPS, оскільки Service Workers працюють лише в захищеному середовищі [3]. Його реєстрація відбувається у JavaScript-коді сторінки, після чого браузер завантажує та встановлює Service Worker. Для оновлення Service Worker необхідно перевстановити його, оскільки браузер спочатку перевіряє, чи змінився файл, і лише тоді виконує оновлення.

Варто зауважити, що існують також Web Workers, які як і Service Workers, працюють в окремому потоці, але вони активуються тільки тоді, коли вебсторінка відкрита. Web Worker використовується для тривалих або ресурсомістких завдань і не мають доступу до DOM чи об'єкта window. Основна різниця полягає в тому, що вебворкери завжди залежать від відкритої сторінки, тоді як сервісний воркер може працювати у фоновому режимі та реагувати на події, навіть коли сторінка закрита [4].

Принцип роботи. Розглянемо, як працює наш проксі-сервер, що базується на Service Worker. Спочатку, коли користувач відкриває вебзастосунок, браузер надсилає запит на проксі-сервер для завантаження домашньої сторінки, яку він отримує та відображає. Далі, якщо користувач вводить посилання і переходить на нову сторінку, проксі-сервер надає сторінку з JavaScript-кодом, що ініціює реєстрацію Service Worker у браузері. Це є важливим моментом, адже зареєстрований Service Worker перехоплює всі подальші мережеві запити користувача.

Після реєстрації Service Worker починає обробляти кожен запит користувача, серіалізуючи його у формат TompHTTP і надсилаючи на наш проксі-сервер. Проксі-сервер, водночас, десеріалізує запит, надсилає його до інтернету та отримує відповідь, яку знову серіалізує у формат TompHTTP і повертає Service Worker.

Отримавши відповідь, Service Worker здійснює модифікацію даних, щоб сторінка виглядала як справжній ресурс, водночас забезпечуючи доступ до необхідних URL через проксі. Зрештою, браузер користувача відображає модифіковану сторінку.

Отже, Service Worker, працюючи спільно з проксі-сервером, забезпечує анонімний та безпечний доступ до інтернет-ресурсів, перехоплюючи та обробляючи всі запити у браузері користувача без потреби додаткових налаштувань з його боку.

Клієнт-серверна комунікація сервісу працює по специфікації TOMP (Too Many Proxies) – уніфіковану специфікацію для вебпроксі, побудованих на базі Service Workers. TOMP забезпечує стандартизований протокол для обміну даними між браузером і проксі-сервером, що сприяє уніфікації та спрощенню реалізації таких рішень. [5]

Проведемо тестування створеного функціоналу. Для цього відкриємо домашню сторінку вебпроксі-сервісу та введемо інтернет-адресу ресурсу для переходу – наприклад, на сторінку Всеукраїнської науково-практичної конференції (cmit.nltu.edu.ua) (рис. 1).

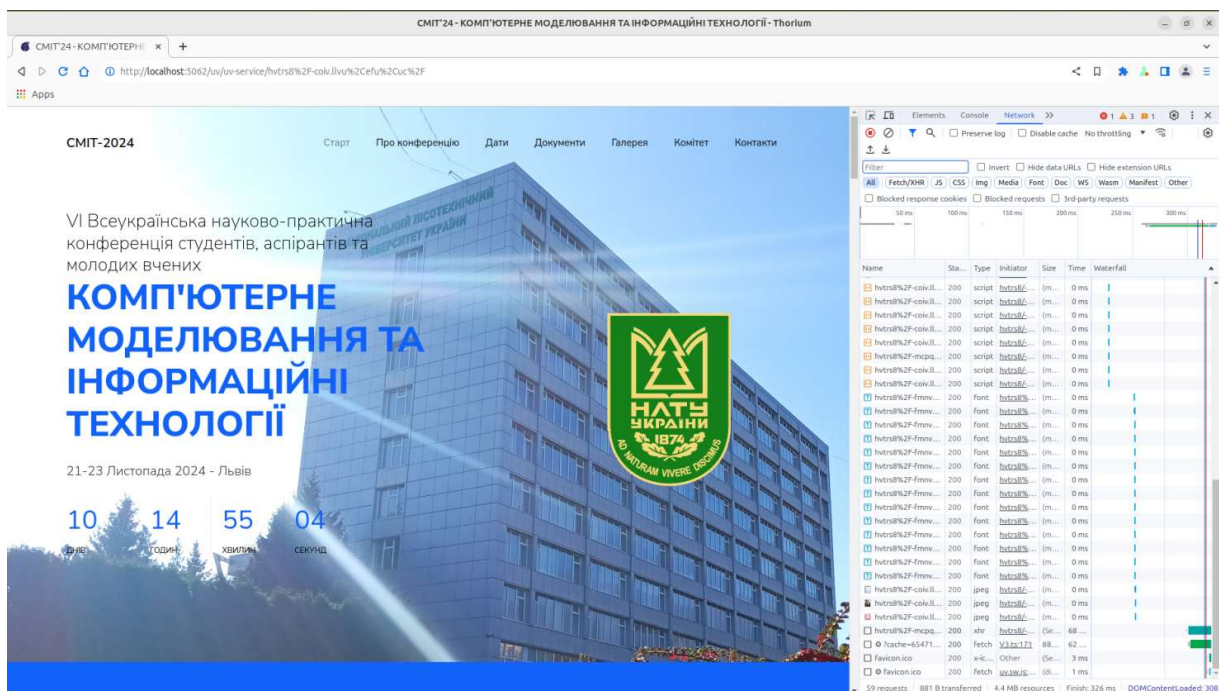


Рисунок 1. Сайт smit.nltu.edu.ua відкритий через проксі сервіс

Як бачимо на рисунку, відкривається сторінка ресурсу, який ми вказали, і вона працює коректно, хоча адреса в адресному рядку браузера відрізняється. У вкладці Network також спостерігаємо, що всі мережеві запити надсилаються через наш вебсервіс, а не напряду. Це підтверджує, що розроблений функціонал працює коректно.

Висновок. За допомогою Service Worker вдалося створити проксі-сервер, який надає анонімний і безпечний доступ до інтернет-ресурсів, приховуючи трафік користувача та обробляючи всі мережеві запити безпосередньо в браузері. Запропонований підхід значно спрощує процес проксування, оскільки сервіс працює в захищеному середовищі HTTPS та не потребує встановлення додаткового програмного забезпечення.

Тестування показало, що розроблений функціонал успішно виконує свої задачі, надаючи користувачеві доступ до потрібних ресурсів через проксі-сервіс та приховуючи справжню адресу ресурсу. Це підтверджує ефективність розробленого рішення для реалізації анонімного доступу та обходу інтернет-цензури в сучасних умовах.

Список використаних літературних джерел

- [1] Proxies As A Tool Against Internet Censorship [Електронний ресурс]
<https://proxybros.com/circumvent-internet-censorship-with-proxies>
- [2] Service Worker API – Web APIs | MDN [Електронний ресурс] –
https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API
- [3] Introduction to Service Workers [Електронний ресурс] –
<https://medium.com/blibliidotcom-techblog/introduction-to-the-service-worker-3c986d9a996d>
- [4] Difference Between a Service Worker and a Web Worker [Електронний ресурс]
–<https://love2dev.com/blog/what-is-a-service-worker>
- [5] Specifications for TompHTTP software/scripts. [Електронний ресурс] –
<https://github.com/tomphhttp/specifications>

Development of a Web Application for an HTTP Proxy Server on the.NET Platform Using Service Worker

Denys Lesyshak, Svitlana Yatsyshyn

This study investigates the mechanism of an HTTP proxy server based on Service Worker technology. An architecture is proposed that enables the creation of a standalone proxy server library for further use in other.NET applications. Based on this architecture, a web service has been implemented, allowing users to access internet resources through the proxy server via a link without requiring any configuration of the computer or browser. Testing demonstrated high compatibility when accessing websites through the proxy.