

АНАЛІЗ СУЧАСНИХ МЕТОДІВ ПРИСКОРЕННЯ ЗАВАНТАЖЕННЯ ВЕБ-САЙТІВ

Азаренков В.І., доцент, кафедра САІТ, НТУ«ХПІ»
Миросниченко Є.Є., студент, кафедра САІТ, НТУ «ХПІ»

Abstract. *This work is devoted to the modern website acceleration techniques. These techniques make websites faster, more efficient, and more competitive in the market. The techniques discussed include debounce and throttling, code minification and GZip compression, and lazy loading. Additionally, specific coding recommendations are provided to further improve website loading speed.*

Keywords: FRONT-END, BACKEND, LAZY LOADING, API, GZIP.

Швидкість завантаження сторінок – дуже важливий параметр для користувачів та пошукових систем. Вважається, що якщо для завантаження сторінки сайту потрібно більше 3 секунд, користувач покине цей сайт [1]. Розглянемо методи, які допоможуть прискорити завантаження веб-сторінок.

1. Використання фреймворків – з їх допомогою, наприклад Vue.js, React.js для Front-end розробки або Express.js, NestJS для Backend-розробки можна краще оптимізувати сайт, бо вони:

- використовують усталені паттерни програмування, що дозволяє зменшити ймовірність виникнення надлишкових операцій в коді;
- часто використовують модульний підхід до розробки (код розбивається на компоненти), тому на сайті завантажуються лише ті модулі, які потрібні на поточній сторінці;
- сприяють повторному використанню коду, що зменшує його загальний обсяг.

2. Використання меншої кількості бібліотек – через те, що кожна бібліотека має свій розмір, загальний обсяг даних збільшується в рази, що суттєво впливає на швидкість завантаження сайту.

Ось декілька порад, для того, як використовувати бібліотеки з найбільшої оптимізацією для сайтів:

- потрібно вибирати лише найнеобхідніші бібліотеки, для простих завдань краще використовувати власноруч написаний код;
- за допомогою сервісу bundlephobia.com та подібних можна проаналізувати розмір різних пакетів, дізнатися скільки часу вони потребують для завантаження та приймати подальше рішення щодо використання цього пакету або бібліотеки в проєкті;
- замість імпорту цілої бібліотеки імпортувати лише ті модулі або функції, які потрібні;
- для бібліотек, які не використовуються одразу при завантаженні сторінки, краще використовувати відкладене завантаження (Lazy loading).

3. Використання техніки програмування debounce – вона використовується для обмеження частоти виконання певної функції. Найпростішим прикладом використання може бути введення тексту в поле пошуку. Замість того, щоб при введенні кожного

символа відправляти запит на сервер, можна відправити запит тільки тоді, коли користувач закінчить введення тексту або через певний період часу.

4. Створення окремих версій вебсайту для десктопних і мобільних пристроїв – застосовується не дуже часто, але також може підвищити оптимізацію сторінки. Цей підхід полягає в створенні декількох версій для різних типів пристроїв. Коли користувач заходить на сайт, визначається тип пристрою, і згідно цього показує потрібну версію, а інші версії не завантажуються.

Але важливо зазначити, що цей підхід містить великий недолік – розробникам потрібно більше часу для написання коду, що робить вартість розробки більш високою.

5. Використання throttling – техніки, яка обмежує частоту викликів функції протягом певного періоду часу. Це дуже корисно для покращення продуктивності та швидкості реагування веб-сторінок, які мають прослуховувачі подій, які запускають важкі або довгі операції, такі як анімація, прокручування, зміна розміру, отримання даних тощо.

Наприклад, для функції, яка отримує деякі дані з API щоразу, коли користувач прокручує сторінку, можна відправляти лише один запит щосекунди, замість того, щоб робити сотні запитів, коли користувач прокручує сторінку. Таким чином можна уникнути перевантаження сервера або браузера непотрібними запитами [2].

6. Застосування мінімізації коду – це процес видалення непотрібних символів (пробілів, коментарів, табуляції) з коду для значного зменшення його розміру, що зменшує й швидкість завантаження веб-сайту [3].

7. Застосування Lazy Loading (відкладене завантаження) – завдяки цьому контент завантажуються не відразу, а поступово, в процесі того, як користувач переміщається по сайту (прокручує сторінки вниз або переходить на інші розділи). Це техніка значно зменшує перший рендеринг сторінки, навантаження на сервер та мобільний трафік користувачів [1].

8. Використання GZip для стиснення даних – це метод стиснення, який дозволяє значно зменшити розмір файлів, що передаються, шляхом усунення надмірної інформації без втрат [4]. Цей метод аналізує файли на наявність повторюваних шаблонів і замінює їх на більш компактні представлення.

Отже, сучасні методи прискорення завантаження веб-сайтів дозволяють значно зменшити час очікування користувача і підвищити загальну ефективність веб-проєкту. Комплексне впровадження цих підходів є ключовим кроком до створення швидкого, зручного та конкурентоспроможного веб-сайту.

Література.

1. Lazy loading для SEO. <https://ideadigital.agency/blog/lazy-loading-dlya-seo/>.
2. Throttling in JavaScript. Easiest Explanation. <https://dev.to/jeeetvora331/throttling-in-javascript-easiest-explanation-1081>.
3. Мініфікація. <https://www.vpnunlimited.com/ua/help/cybersecurity/minification>.
4. Як увімкнути кешування та GZip стиснення. <https://thehost.ua/ua/wiki/administration/ssh/gzip>.