УДК 004.007

МЕТОДИ ОПТИМІЗАЦІЇ JAVASCRIPT ДЛЯ ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ ВЕБ ЗАСТОСУНКІВ

***А.О. Приліпа*[1]**, *Г.Є. Філатова*[2]

[1] *аспірат кафедри КІП, НТУ «ХПІ», Харків, Україна*
[2] *професор кафедри КІП, доктор техн. наук, НТУ «ХПІ», Харків, Україна*
artem.prylipa@cs.khpi.edu.ua

JavaScript є основним елементом інтерактивності вебзастосунків, проте його необережне використання може значно погіршити швидкість завантаження сторінки та знизити ефективність роботи користувацького інтерфейсу. Для підвищення продуктивності важливо використовувати наступні методи оптимізації JavaScript: мінімізація, використання атрибутів async та defer, розподіл коду, видалення невикористаного коду.

Мінімізація JavaScript передбачає видалення зайвих символів (пробіли, перенос рядків тощо) та скорочення імен змінних. Це дозволяє зменшити розмір файлу, що, у свою чергу, зменшує час завантаження. Стандартні інструменти для мінімізації включають mode production у Webpack, babel-preset-minify для Babel та gulp-uglify для Gulp. За даними досліджень, мінімізація дозволяє знизити об’єм JavaScript-файлів на 30-40% [1].

Атрибути async і defer дозволяють завантажувати JavaScript асинхронно [2]. Використання async вказує браузеру завантажувати і виконувати скрипт без очікування завершення розбору HTML-документа. Атрибут defer, у свою чергу, відкладає виконання скрипту до завершення розбору HTML. Це дозволяє браузеру продовжувати рендеринг сторінки, що значно зменшує час повної завантаженості сторінки.

Розподіл коду дозволяє поділити великий JavaScript-файл на декілька менших, які завантажуються за потребою. Наприклад, діалоги, що рідко використовуються, можуть бути завантажені лише при їх виклику, що значно зменшує початковий об'єм даних для завантаження. Для цього використовуються сучасні бандлери, як-от Webpack, що дозволяють виконувати асинхронне завантаження через функцію import().

Часто залежності містять зайвий код, який не використовується застосунком. Такі бібліотеки містять файли локалізації, які можуть бути непотрібними. Оптимізація цих залежностей дозволяє зменшити загальний розмір JavaScript, що підвищує ефективність завантаження.

Таким чином, застосування зазначених підходів сприяє підвищенню продуктивності вебзастосунків, дозволяючи значно скоротити час завантаження та покращити користувацький досвід.

Список літератури:

1. Qiong, G., & Li, W. (2020). An Optimization Method of Javascript Redundant Code Elimination based On Hybrid Analysis Technique. 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 300-305. https://doi.org/10.1109/ICCWAMTIP51612.2020.9317462.

2. Arteca, E., Tip, F., & Schäfer, M. (2021). Enabling Additional Parallelism in Asynchronous JavaScript Applications. , 7:1-7:28. https://doi.org/10.4230/LIPIcs.ECOOP.2021.7.