

УДК 004.42

DOI <https://doi.org/10.35546/kntu2078-4481.2024.2.19>**К. П. ГАЗДЮК**

доктор філософії (спеціальність 121), доцент,
доцент кафедри програмного забезпечення комп'ютерних систем
Чернівецький національний університет імені Юрія Федьковича
ORCID: 0000-0002-7568-4422

М. Ю. ГОРБАТЕНКО

кандидат фізико-математичних наук, доцент,
доцент кафедри математичного моделювання
Чернівецький національний університет імені Юрія Федьковича
ORCID: 0000-0002-1702-8785

К. М. ДМИТРАЩУК

асистент кафедри програмного забезпечення комп'ютерних систем
Чернівецький національний університет імені Юрія Федьковича
ORCID: 0009-0004-8188-8618

В. Ю. ВОДЯНЧУК

Middle React/React Native developer
Agiliway
ORCID: 0009-0006-0680-7085

ДОСЛІДЖЕННЯ ІНСТРУМЕНТІВ ЗБІРКИ СУЧАСНИХ ВЕБ ДОДАТКІВ

У статті здійснено аналіз двох сучасних найпопулярніших інструментів для збірки проєктів WebPack та Vite, а також досліджено їх швидкодю. Проведено оцінку функцій та їх відповідність потребам користувачів, аналіз зовнішнього вигляду, інтерфейсу та зручності використання для збірки веб-додатків, можливостей збірників у плані відкритості для розширення функціоналу та їх масштабованість при зростанні обсягів даних та користувацької бази; оцінено рівень захищеності додатків від потенційних загроз інформаційній безпеці та приватності користувачів; проаналізовано швидкість та ефективність виконання різних завдань у веб-додатках; оцінено можливості коректної роботи збірників на різних пристроях (комп'ютерах, планшетах, мобільних телефонах) та у різних браузерах.

Аналізуючи їхні функції та характеристики, виявлено, що кожен з них має свої переваги та обмеження.

WebPack вже довгий час є стандартом для збірки веб-проєктів. Він має потужні можливості конфігурації та розширюваності, що дозволяє налаштовувати його під потреби конкретного проєкту. Однак, із зростанням розміру проєкту та складності конфігурації, може виникати проблема зі збереженням продуктивності.

З іншого боку, Vite зосереджений на швидкості розробки та виконання завдань у реальному часі. Він надає швидкий час перезавантаження під час розробки, що полегшує процес розробки та налагодження. Однак, його можливості розширення та конфігурації можуть бути обмеженими порівняно з WebPack.

Виявлено, що обираючи між WebPack та Vite, команди розробників повинні враховувати специфіку свого проєкту та віддавати перевагу тому інструменту, який найбільше відповідає їхнім потребам у швидкості, гнучкості та продуктивності. Крім того, важливо постійно вивчати та оцінювати нові можливості та інструменти, які з'являються на ринку, для того щоб залишатися в актуальному та конкурентоспроможному стані.

Ключові слова: програмний продукт, браузерне розширення, збірник, вебсторінка, WebPack, Vite.

K. P. HAZDIUK

Doctor of Philosophy (specialty 121), Associate Professor,
Associate Professor at the Computer Systems Software Department
Yuriy Fedkovych Chernivtsi National University
ORCID: 0000-0002-7568-4422

M. YU. GORBATENKO

Candidate of Physical and Mathematical Sciences, Associate Professor,
Associate Professor at the Department of Mathematical Modeling
Yuriy Fedkovych Chernivtsi National University
ORCID: 0000-0002-1702-8785

K. M. DMYTRASHCHUK

Assistant at the Computer Systems Software Department
Yuriy Fedkovych Chernivtsi National University
ORCID: 0009-0004-8188-8618

V. YU. VODIANCHUK

Middle React/React Native developer
Agiliway
ORCID: 0009-0006-0680-7085

INVESTIGATION OF MODERN WEB APPLICATION BUNDLERS

The article provides an analysis of two of the most popular modern tools for building web projects, WebPack and Vite, as well as examines their performance. An evaluation of their features and their alignment with user needs, an analysis of their appearance, interface, and usability for building web applications, the openness of the tools for functionality expansion, and their scalability with increasing data volumes and user base were conducted. Additionally, the security level of the applications against potential threats to information security and user privacy was assessed, along with an analysis of the speed and efficiency of executing various tasks in web applications. Furthermore, the potential for the correct operation of the tools on different devices (computers, tablets, mobile phones) and in different browsers was evaluated. Analyzing their functions and characteristics revealed that each of them has its own advantages and limitations.

WebPack has long been the standard for building web projects. It offers powerful configuration and extensibility capabilities, allowing it to be tailored to the needs of a specific project. However, as project size and configuration complexity grow, there may be challenges in maintaining productivity.

On the other hand, Vite focuses on rapid development and real-time task execution. It provides quick reloading during development, simplifying the development and debugging process. However, its extensibility and configuration capabilities may be limited compared to WebPack.

It was found that when choosing between WebPack and Vite, development teams should consider the specifics of their project and prefer the tool that best meets their needs in terms of speed, flexibility, and productivity. Additionally, it is important to continuously explore and evaluate new opportunities and tools emerging in the market to remain current and competitive.

Key words: software product, browser extension, bundler, web page, WebPack, Vite.

Постановка проблеми

Bundler або збірник, є на сьогодні незамінним інструментом у сучасній веб розробці [1]. Це дозволяє розробникам без проблем працювати з кодом на мовах, які не підтримуються браузером за замовчуванням (TypeScript, SCSS). Також він відповідає, за адаптацію програми для середовища development або production. У випадку створення власного проєкту, bundler не лише допоможе максимально оптимізувати код, а й зробить програму більш ефективною.

На даний момент, найпоширенішим засобом збірки проєктів є Webpack [2], його домінування нерозривно пов'язане із значною популярністю, оскільки при виникненні питань стосовно застосування цього продукту можна з легкістю знайти відповіді в Інтернеті, завдяки потужній спільноті користувачів. Щотижня збірник Webpack завантажується з репозиторію NPM (Node Package Manager) [3] близько двадцяти чотирьох мільйонів разів, що свідчить про його вражаючу популярність. Проте, на ринку існує тенденція до уникнення монополій, і тому поступово відбувається введення нового учасника – Vite [4]. Останній спрямований на вирішення однієї з проблем Webpack, а саме – часу збірки проєкту. Далі буде розглянуто принципи роботи обох інструментів збірки та проаналізовано фактори, які забезпечують Vite його високу швидкодію.

На даний момент постає проблема порівняння двох найпоширеніших засобів збірки проєктів, Webpack і Vite, щоб визначити їхні переваги та недоліки у контексті використання веб-розробки.

Аналіз останніх досліджень і публікацій

Аналіз сучасних досліджень в області розробки веб застосунків показує, що великою перевагою є використання таких інструментів як React, Node.js, TypeScript, Python, Django, Webpack і Docker [5, 6]. Дослідження інструментів збірки таких як Webpack та Vite має важливе значення, оскільки він входить до складу потужного стеку технологій, який дозволяє розробникам створювати програми з високою продуктивністю та ефективністю. Подальше їх вивчення дозволить розширити знання про оптимізацію та організацію коду для розгортання програм, що є важливим аспектом у сучасній веб-розробці.

Перед тим як перейти безпосередньо до порівняння двох найпопулярніших збірників веб дооатків, розглянемо принципи їх роботи. WebPack на даний час, є одним з найбільш часто використовуваних пакетів для вебпроєктів, але він набагато повільніший ніж Vite. Webpack збирає всі файли в проєкті до того, як сервер розробки буде готовий.

Власне, на Рис. 1 можна побачити головну причину такої повільної роботи webpack, особливо у великих проєктах. З кожним розширенням проєкт збільшується та обсяг коду, що має обробити збірник, стає все більшим, власне як і процес збирання проєкту. Коли додаються нові модулі у проєкт, WebPack потребує додаткової обробки, щоб почати з ними працювати.

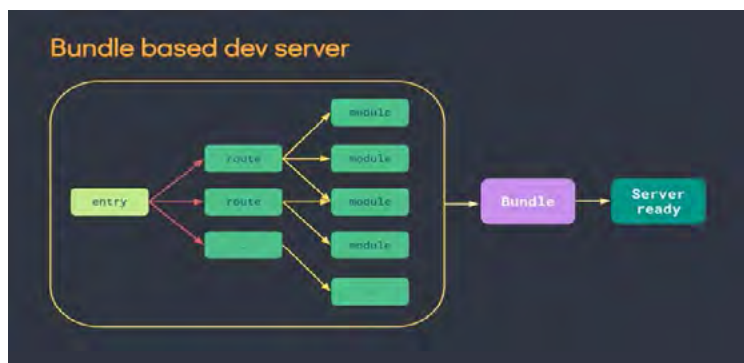


Рис. 1. Принцип роботи Webpack

Після встановлення зазначених пакетів, Webpack потрібно налаштувати. Для цього створюється файл webpack.config.js, що експортує об'єкт, який містить параметри Webpack-у.

Основним завданням Webpack-а є аналіз модулів, їх вибіркоче перетворення та інтелектуальне об'єднання в одну або більше колекцій (bundles), тому Webpack-у потрібно знати три речі:

1. Точка входу програми.
2. Перетворення, які потрібно виконати.
3. Місце, в яке слід помістити сформовану колекцію.

Особливість Vite полягає в тому, що він використовує переваги модулів ES (ECMAScript). У середовищі розробки Vite запускає сервер, який використовується для компіляції та обслуговування на льоту будь-яких необхідних залежностей через модулі ES (Рис. 2). Такий підхід дозволяє Vite обробляти та надавати лише необхідний на цей момент код. Таким чином, Vite потрібно обробляти набагато менше коду під час запуску сервера та оновлення коду.

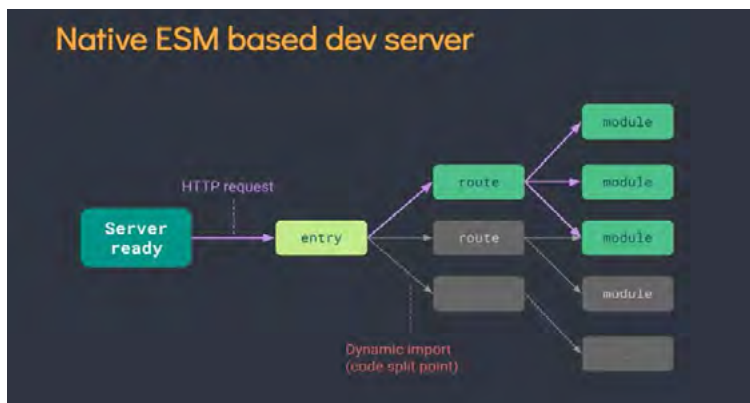


Рис. 2. Принцип роботи Vite

Ще однією особливістю є те, що він використовує Esbuild для попередніх групувань залежностей під час розробки. Esbuild — це надзвичайно швидкий (Рис. 3) bundler JavaScript, написаний мовою Go.

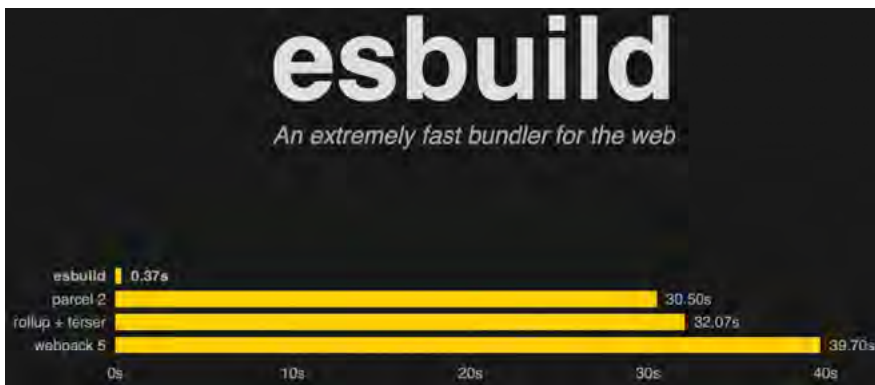


Рис. 3. Швидкість збірки проекту різними збірниками

Формулювання мети дослідження

Метою дослідження є порівняння двох найпопулярніших інструментів збірки веб-проектів, Webpack і Vite, з метою визначення їхніх переваг та недоліків у контексті швидкості збірки, продуктивності розробки та загальної ефективності. Дослідження спрямоване на виявлення потенційних переваг кожного інструменту та визначення оптимального вибору для розробки веб-додатків з урахуванням специфіки проектів та потреб користувачів.

Викладення основного матеріалу дослідження

Проведемо порівняльну характеристику даних збірників за такими параметрами як кількість скачувань, вбудовані інструменти, система модулів, що використовується ними, підтримка плагінів, доцільність використання, спільнота та розмір вихідного файлу. Варто зазначити, що найважливішим параметром для порівняння є швидкість інструментів, її ми розглянемо в на основі реального проекту.

Всі отримані дані представлено в таблиці 1. Можна зробити висновки, що Webpack популярніший за Vite. Кількість користувачів Webpack перевищує в 3 рази, проте це пояснюється його роком випуску. Даний інструмент був випущений в 2012 році, а Vite тільки в 2020. Останній має вбудовану підтримку TypeScript, style loading, hot reloading, а також пропонує шаблони для збірки проектів на React, Vue, JS, TS та інших технологіях. Це означає, що налаштування не потребує великих зусиль програміста і це автоматизовано в збірнику Vite. Webpack в свою чергу складніший для конфігурації, він вимагає підключень плагінів вручну.

В двох збірників активна спільнота, яка постійно збільшується. Проте, в першого вона більша. Зазвичай Vite використовують для невеликих нових проектів, а Webpack надають перевагу при розробці ентерпрайсних продуктів.

Таблиця 1

Порівняльна характеристика Vite та Webpack

Характеристика	Vite	Webpack
Кількість скачувань	8,828,073	26,043,383
Вбудовані інструменти	Вбудована підтримка TS, style loading, hot reloading	Додаткова конфігурація плагінів
Система модулів	ES modules	ES modules, Common JS
Підтримка плагінів	Пропонує шаблони для React, Vue, JS, TS, etc.	Налаштовується вручну
Використання	Найкраще підходить для нових невеликих проектів	Використовується як для невеликих так і для масштабних проектів
Спільнота	Зростаюча спільнота	Велика, активна спільнота
Розмір вихідного файлу	Невеликий, оптимізовано під production	Більший розмір, потрібна додаткова оптимізація

Швидкодію збірників вебдодатків було перевірено на основі реального невеликого проекту. Даний проект містить 400+ файлів. Це комплексна система, яка має багато модулів, частину з яких наведено на рис. 4.

Проект було написано за допомогою збірника Webpack та оптимізовано під development та production, підключено плагіни для обробки стилів та підтримки TypeScript. Оскільки Vite опрацьовує це автоматично, нам не потрібно нічого додавати вручну.

На рис. 5 наведено частину конфігураційного коду проекту за допомогою Webpack.

Далі додано файл vite.config.ts, в якому підключено плагін react для роботи збірки проекту на даній технології.

На рис. 6 наведено частину конфігурацій проекту за допомогою Vite.

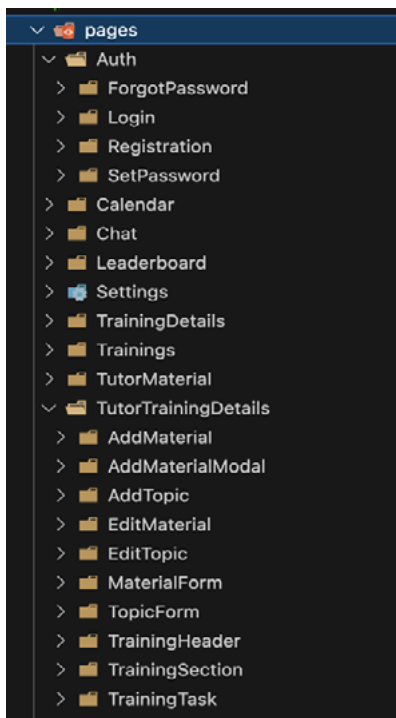


Рис. 4. Зміст модулів проекту

```
module: {
  rules: [
    {
      test: /\.?(ts|js)x?$/i,
      exclude: /node_modules/,
      use: [
        {
          loader: 'babel-loader',
          options: {
            presets: [
              '@babel/preset-env',
              '@babel/preset-react',
              '@babel/preset-typescript'
            ]
          }
        }
      ]
    },
    {
      test: /\.?(css|scss)$/,
      use: ['style-loader', 'css-loader', 'sass-loader']
    }
  ]
}
```

Рис. 5. Конфігурації проекту за допомогою Webpack

```
import { defineConfig } from "vite";
import react from "@vitejs/plugin-react";
import path from "path";

export default defineConfig({
  plugins: [react()],
  resolve: {
    alias: {
      shared: path.resolve(__dirname, "src/shared"),
      pages: path.resolve(__dirname, "src/pages"),
    },
  },
});
```

Рис. 6. Конфігурації проекту за допомогою Vite

Переконавшись, що обидва проєкти працюють однаково, з підгрузкою зображень та стилів, було проведено дослідження на швидкодію згаданих збірників. Запущено код в режимі development та режимі production 20 разів. Взято середнє арифметичне результатів та занесено в табл. 2.

Таблиця 2

Швидкісна характеристика Vite та Webpack

	Vite	Webpack
Development	356ms	31.55s
Production	12.89s	40.30s

Результати дослідження показали, що Vite запустив сервер за 356 мс в режимі development, в той час збірник Webpack потрібно було 31.55 с. Різниця вражаюча, Vite зробив це швидше у 88 разів. Щодо production, Vite запустив сервер за 12.89 с, а Webpack за 40.30 с.

Висновки

Розглянуто принцип роботи бандлерів Vite та Webpack та проаналізовано їх, а саме Vite запускає сервер, а потім збирає всі файли, але тільки ті, які потрібні користувачу на вебсторінці, яку він проглядає. Webpack в свою чергу спочатку збирає файли всього проєкту, а потім запускає сервер, що робить вихідний файл більш об'ємним і сповільнює запуск проєкту.

Проаналізовано принципи роботи збірників Vite та Webpack. Vite запускає сервер, а потім збирає всі файли, але тільки ті, які потрібні користувачу на вебсторінці, яку він безпосередньо переглядає. Webpack, на відміну від Vite, спочатку збирає файли всього проєкту, а потім запускає сервер, що робить вихідний файл більш об'ємним і сповільнює запуск проєкту.

Проведено дослідження цих інструментів збірки за такими характеристиками: кількість скачувань, вбудовані інструменти, система модулів, що використовується ними, підтримка плагінів, доцільність використання, спільнота та розмір вихідного файлу. З використанням реального проєкту, оцінено швидкодію даних інструментів, результат якої показав, що Vite працює в 88 разів швидше для режиму development, що полегшує процес розробки програмного продукту та економить час програміста.

Варто зазначити, що обираючи між WebPack та Vite, команди розробників повинні враховувати специфіку свого проєкту та надавати перевагу тому інструменту, який найбільше відповідає їхнім потребам у швидкості, гнучкості та продуктивності. Крім того, важливо постійно вивчати та оцінювати нові можливості та інструменти, які з'являються на ринку, для того щоб залишатися в актуальному та конкурентоспроможному стані.

Список використаної літератури

1. Rack, J., Staicu, C.-A. Jack-in-the-box: An Empirical Study of JavaScript Bundling on the Web and its Security Implications [Електронний ресурс]. Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. November 2023. Pp. 3198–3212. <https://doi.org/10.1145/3576915.3623140>
2. Webpack. URL: <https://webpack.js.org/>
3. Duldulao, D.B., Cabagnet, R.J.L. Getting Started with the Node Package Manager [Електронний ресурс] // Practical Enterprise React. Apress, Berkeley, CA, 2021. Pp. 11–19. URL: https://doi.org/10.1007/978-1-4842-6975-6_2
4. Vite. Next Generation Frontend Tooling. URL: <https://vitejs.dev/>
5. Zammetti, F. Modern Full-Stack Development: Using TypeScript, React, Node.js, Webpack, and Docker . 2020. URL: <https://doi.org/10.1007/978-1-4842-5738-8>
6. Zammetti, F. Tying It Up in a Bow: Webpack. 2022. URL: https://doi.org/10.1007/978-1-4842-8811-5_7
7. Fusco, D. Large Scale Apps with Vue, Vite and TypeScript. 2023. URL: <https://www.damianofusco.com/book-sample-vue-typescript>

References

1. Rack, J., & Staicu, C. A. (2023). Jack-in-the-box: An Empirical Study of JavaScript Bundling on the Web and its Security Implications. In Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23) (pp. 3198–3212). URL: <https://doi.org/10.1145/3576915.3623140>
2. Webpack. (n.d.). Available at: <https://webpack.js.org/>
3. Duldulao, D. B., & Cabagnet, R. J. L. (2021). Getting Started with the Node Package Manager. In F. Zammetti (Ed.), Practical Enterprise React (pp. 11–19). Apress. URL: https://doi.org/10.1007/978-1-4842-6975-6_2
4. Vite | Next Generation Frontend Tooling. (n.d.). Available at: <https://vitejs.dev/>
5. Zammetti, F. (2020). Modern Full-Stack Development: Using TypeScript, React, Node.js, Webpack, and Docker. URL: <https://doi.org/10.1007/978-1-4842-5738-8>
6. Zammetti, F. (2022). Tying It Up in a Bow: Webpack. URL: https://doi.org/10.1007/978-1-4842-8811-5_7
7. Fusco, D. (2023). Large Scale Apps with Vue, Vite and TypeScript. Available at: <https://www.damianofusco.com/book-sample-vue-typescript>