

УДК 004.4

DOI: <https://doi.org/10.17721/3041-2323.2024.292-296>

Юрій ПРОКОПЧУК, студ.

ORCID ID: 0009-0008-7363-2080

e-mail: youht432@gmail.com

Київський національний університет імені

Тараса Шевченка, Київ, Україна

ВИКОРИСТАННЯ БІБЛІОТЕКИ REACT ДЛЯ РОЗРОБЛЕННЯ ВЕБЗАСТОСУНКІВ: ПЕРЕВАГИ ТА ВИКЛИКИ

Надано огляд основних характеристик і функціональних можливостей бібліотеки React. Виокремлено сильні та слабкі сторони React, а також продемонстровано доцільність її використання у сучасному фронт-енд розробленні вебзастосунків порівняно з її основними конкурентами.

Ключові слова: веброзроблення, вебдодаток, фронт-енд бібліотека, фронт-енд фреймворк, бібліотека React.

Вступ

Нині є і активно застосовується велика кількість різноманітних інструментів, бібліотек і фреймворків для створення вебзастосунків. Найчастіше, за даними Statista, у 2024 р. розробники використовують такі технології для розроблення front-end частин вебзастосунків: React (39,5 %), Angular (17,1 %), Vue.js (15,4 %) (Most used..., 2024). Як можна побачити з наведеної статистики, React із великим відривом є популярнішим серед програмістів, тому цікавим стає дослідження особливостей використання позитивних і негативних якостей такої технології, яка здобула значне поширення серед розробників.

Результати

React позиціонується розробниками як бібліотека для мови програмування JavaScript для створення користувацьких інтерфейсів (React, n. d.). Однак, варто зазначити, що технологія може працювати також зі строго типізованою TypeScript і таке використання можна зустріти все частіше. Бібліотека React орієнтована на роботу з компонентами – невеликими блоками коду, які можна повторно використовувати (Hutsulyak, 2024). Життєвий цикл кож-

© Прокопчук Юрій, 2024

ного компонента складається з таких фаз: монтування (Mounting), оновлення (Updating) і демонтування (Unmounting). На першій фазі з перерахованих відбувається ініціалізація компонента, його стану, додавання в DOM (Document Object Model). Наступний етап виконують у разі оновлення стану компонента. Остання фаза відповідає за дії з очищення, відміни запитів, підписок при видаленні компонента з DOM. Робота з великою кількістю методів життєвого циклу була поширена при використанні класових компонентів. З появою хуків (hooks) з'явилась можливість працювати зі станом і багатьма іншими можливостями у функціональних компонентах, що значно спостило написання та зрозумілість коду (Retool, 2024).

Бібліотека React початково призначена для створення односторінкових вебзастосунків (Single Page Application) із рендерингом на боці клієнта, що може негативно вплинути на індексування під час роботи пошукових рушіїв. Якщо ця особливість є критичною, то можна розглянути фреймворки, які розв'язують таку проблему, наприклад, Next.js, який, серед іншого, надає можливості рендерингу на боці сервера та дозволяє створювати статичні сайти.

Однією з особливостей React є використання Virtual DOM. DOM є інтерфейсом, що дозволяє динамічно отримувати доступ найчастіше до HTML (Hyper Text Markup Language) документів, які представляються як набір об'єктів деревовидної структури, та мати можливість програмно керувати їхнім вмістом (Lazuardy, & Anggraini, 2022). Маніпуляції з DOM є досить затратними, тому React використовує так званий віртуальний DOM (React DOM), що є полегшеним представленням DOM. Отже, Virtual DOM порівнює зміни із справжнім представленням і вносить їх лише там, де це потрібно, за рахунок чого збільшується продуктивність. Vue також використовує таку концепцію. Angular, однак, взаємодіє з реальним DOM.

Іншою особливістю вартою уваги є використання JSX – синтаксичного розширення, що дозволяє поєднувати код JavaScript із HTML (React, n. d.). Такий підхід у поєднанні з функціональними компонентами дозволяє скоротити час написання коду та може бути простішим для розуміння початківцям. Варто зазначити, що застосування JSX є рекомендованим, хоча й необов'язковим. Vue

також допускає використання такого синтаксичного розширення, проте це не є стандартним підходом.

Активна спільнота, велика кількість навчальних матеріалів, форумів, присвячених розв'язанню проблем та обговоренню перспектив розвитку, є важливими елементами для освоєння й ефективного використання технології. React розроблено компанією "Facebook" ("Meta") й активно нею підтримується. За її допомогою можна з легкістю знайти велику кількість матеріалів для вивчення та пошуку відповідей на поширені запитання.

Оскільки React вважають бібліотекою, це з одного боку є перевагою, а з іншого – недоліком. Фреймворки зобов'язують розробників під час створення застосунків слідувати певним встановленим правилам, бібліотеки ж є вільнішими у виборі використання, надаючи просто набір інструментів для реалізації певної задачі (Levlin, 2020). Фреймворки зазвичай надають основні необхідні інструменти для створення застосунків разом із чіткою структурою розроблення. React, як бібліотека є більш гнучкою у виборі підходів до програмування та підбору інструментів, однак це неминуче призводить до того, що часто не вистачає власних можливостей, зокрема для маршрутизації, роботи з мережею. Ця проблема частково розв'язується великою кількістю додаткових якісних бібліотек, наприклад, React Router. Управління станом також є одним із викликів у створенні вебзастосунків із використанням цієї технології. Якщо в роботі зі станом усередині компонента та з батьківськими компонентами можна скористатись базовими інструментами (hooks, Context API), то для взаємодії із глобальним станом необхідно залучати сторонні бібліотеки, одними з найпопулярніших серед яких є Redux та MobX. Перша використовує деревовидний підхід та єдине сховище. Друга застосовує реактивну концепцію, що відстежує залежності. Такий вільний і широкий вибір часто призводить до невизначеності у підборі конкретної технології для розв'язання проблеми і за зміни команди часто доводиться ознайомлюватись із новим набором інструментів із схожим призначенням або новим стилем програмування та витратити час на додаткове навчання.

Важливо зазначити, що React використовує одностороннє зв'язування між постачальником і споживачем даних, тобто дані рухаються в одному напрямку від моделі до інтерфейсу, хоча в написанні додаткової логіки чи використанні додаткових інстру-

ментів можна досягти двостороннього. Для Angular стандартним є двостороннє зв'язування, але за потреби можна зробити й одностороннє. Вважають, останнє більш прозоре та передбачуване, його легше контролювати, однак часто зустрічаються ситуації, коли необхідно скористатись першим.

Якщо звернутись до тестів швидкодії, наведених у джерелі (React, n. d.), то можна побачити, що Vue виявляється швидшим за всіх конкурентів майже в усіх тестах. Angular і React показують сумірні результати. Перший показує кращі результати в частковому оновленні, виборі та видаленні рядка, створенні великої кількості рядків, у той час як другий є кращим при створенні рядка, заміні всіх рядків, перестановці й очищенні деякої їхньої кількості, приєднанні рядка до таблиці. У тестуванні обсягів використання пам'яті, знову найкращий результат показує Vue, далі з помітним відривом йде React і найгірший результат демонструє Angular. Варто зазначити, що за винятком кількох тестів, результати переважно відрізняються не надто сильно.

Дискусія і висновки

React є конкурентоспроможною технологією, яка попри деякі труднощі у роботі, при детальному вивченні особливостей використання, має велику кількість беззаперечних переваг, що вирізняє її серед інших і робить цікавим інструментом для створення front-end частин вебзастосунків.

Список використаних джерел

- Hutsulyak, O. (2024). *Why Use React for Web Development: 10 Reasons To Apply*. TechMagic Blog. <https://www.techmagic.co/blog/why-we-use-react-js-in-the-development/>
- Lazuardy Fariz Syah, M., & Anggraini, D. (2022). Modern front-end web architectures with React.Js and Next.Js. *International Research Journal of Advanced Engineering and Science*, 7(1), 132–141. <https://www.semanticscholar.org/paper/Modern-Front-End-Web-Architectures-with-React.Js-Fariz-Lazuardy/9df33541508d8ee667593bb630532f4fc9a52af6>
- Levlin, M. (2020). *DOM benchmark comparison of the front-end JavaScript frameworks React, Angular, Vue, and Svelte: Master's Thesis in Computer Science*. Åbo. <https://www.doria.fi/handle/10024/177433>
- Most used web frameworks among developers 2024. (2024). *Statista*. <https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>
- React. (n. d.). *React Documentation*. <https://react.dev/>
- Retool Blog. (2024). The React lifecycle: Methods and hooks explained. *Retool. A better way to build custom software*. <https://retool.com/blog/the-react-lifecycle-methods-and-hooks-explained>

References

- Hutsulyak, O. (2024). *Why Use React for Web Development: 10 Reasons To Apply*. TechMagic Blog. <https://www.techmagic.co/blog/why-we-use-react-js-in-the-development/>
- Lazuardy Fariz Syah, M., & Anggraini, D. (2022). Modern front-end web architectures with React.Js and Next.Js. *International Research Journal of Advanced Engineering and Science*, 7(1), 132–141. <https://www.semanticscholar.org/paper/Modern-Front-End-Web-Architectures-with-React.Js-Fariz-Lazuardy/9df33541508d8ee667593bb630532f4fc9a52af6>
- Levlin, M. (2020). *DOM benchmark comparison of the front-end JavaScript frameworks React, Angular, Vue, and Svelte: Master's Thesis in Computer Science*. Åbo. <https://www.doria.fi/handle/10024/177433>
- Most used web frameworks among developers 2024. (2024). *Statista*. <https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>
- React. (n. d.). *React Documentation*. <https://react.dev/>
- Retool Blog. (2024). The React lifecycle: Methods and hooks explained. *Retool | A better way to build custom software*. <https://retool.com/blog/the-react-lifecycle-methods-and-hooks-explained>

Отримано редакцією журналу / Received: 16.09.24

Прорецензовано / Revised: 26.09.24

Схвалено до друку / Accepted: 01.10.24

Yurii PROKOPCHUK, Student

ORCID ID: 0009-0008-7363-2080

e-mail: youht432@gmail.com

Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

USING THE REACT LIBRARY FOR WEB APPLICATION DEVELOPMENT: ADVANTAGES AND CHALLENGES

In this article, an overview of the React library's main characteristics and functional capabilities is given. Strong and weak sides of React are highlighted and the expediency of its usage in modern front-end development of web applications in comparison with its main competitors is shown.

Keywords: *Web development, web application, front-end library, front-end framework, React library.*

Автор заявляє про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The author declares no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.