

## **ПОРІВНЯННЯ ОСОБЛИВОСТЕЙ JAVASCRIPT РАНТАЙМІВ**

JavaScript-оточення (runtime) відіграють ключову роль у функціонуванні JavaScript-коду на стороні сервера. Основними конкурентами в цьому напрямку є Node.js, Deno та Bun. Кожен з них пропонує унікальні можливості, показники продуктивності та філософію дизайну. У цьому порівнянні розглядаються їхні архітектури, ефективність, моделі безпеки, модульні системи, та інструментарій.

Розглянемо особливості архітектури JavaScript-оточення:

- Node.js. Побудований на JavaScript-движку V8 від Google. Він використовує модель вводу/виводу, керовану подіями, що не блокується, тому робить його ефективним для масштабованих мережевих додатків. Реалізований він в основному на мовах C++;
- Deno. Також, як і Node.js побудований на движку V8 і написаний на мові Rust. Він робить акцент на безпеці, сучасних можливостях JavaScript та спрощеній системі модулів;
- Bun. Побудований на движку JavaScriptCore (розроблений Apple для Safari) і написаний на Zig. Він фокусується на продуктивності, прагнучи стати швидким універсальним оточенням виконання JavaScript з вбудованими інструментами для компонування.

Ефективність є критичним фактором при виборі рантайму, особливо для додатків, що вимагають високої пропускну здатності та низьких затримок [1]. Хоча Node.js є галузевим стандартом для серверного JavaScript, бенчмарки показують, що він відстає від нових середовищ виконання за основними показниками продуктивності. При обробці HTTP-запитів він обробляє приблизно 13254 запити в секунду. Deno пропонує покращену продуктивність порівняно з Node.js, обробляючи близько 22286 запитів на секунду в аналогічних тестах. Це покращення пояснюється оптимізацією архітектури. Bun перевершує як Node.js, так і Deno, досягаючи приблизно 52479 запитів на секунду в HTTP-бенчмарках. Використання движка JavaScriptCore та оптимізації в його дизайні сприяють такій високій продуктивності.

Не менш важливим фактором є забезпечення безпеки. За замовчуванням Node.js не накладає обмежень на доступ до файлової системи або мережі, покладаючи відповідальність за реалізацію заходів безпеки на розробників. Deno використовує підхід, що ставить безпеку на перше місце, вимагаючи явних прапорів дозволу для операцій, пов'язаних з доступом до файлової системи, мережевих запитів або змінних оточення. Bun працює подібно до Node.js, де розробники відповідають за управління аспектами безпеки.

На початку Node.js використовував систему модулів CommonJS. Згодом було впроваджено підтримку ESM-модулів за допомогою синтаксису імпорту та експорту, хоча це вимагає певних конфігурацій. Deno підтримує тільки ESM, дозволяючи прямий імпорт з URL-адрес або локальних файлів. Bun пропонує підтримку як ESM, так і модулів CommonJS без додаткової конфігурації. Така гнучкість полегшує міграцію існуючих проєктів Node.js та інтеграцію з npm-пакетами.

Node.js може похвалитися розвинутою екосистемою з широким спектром бібліотек та фреймворків. Однак для таких завдань, як компонування та тестування, вона покладається на зовнішні інструменти. Deno надає набір вбудованих інструментів, включаючи лінтер, форматер тощо. Він також пропонує власну підтримку TypeScript, усуваючи потребу у зовнішніх транскрипторах. В свою чергу Bun має на меті бути універсальним набором інструментів з вбудованою підтримкою пакувальника, транскрипції та тестування. Він також включає швидкий менеджер пакунків і пропонує власну підтримку TypeScript і JSX.

Висновки. Розглянуті оточення мають свої переваги та недоліки. Node.js пропонує розвинуту екосистему та широке розповсюдження, що робить його надійним вибором для багатьох додатків. Deno пропонує розширені функції безпеки, що приваблює розробників, для яких ці аспекти є пріоритетними. Bun забезпечує виняткову продуктивність та інтегрований інструментарій, що робить його привабливим для критично важливих до продуктивності додатків.

### **Список використаних джерел**

1. Ulili S. Node.js vs Deno vs Bun. Radically better observability stack. URL: <https://betterstack.com/community/guides/scaling-nodejs/nodejs-vs-deno-vs-bun/> (date of access: 17.11.2024).
2. Documentation. Node.js – Run JavaScript Everywhere. URL: <https://nodejs.org/docs/latest/api/> (date of access: 17.11.2024).
3. Deno. URL: <https://docs.deno.com/> (date of access: 17.11.2024).
4. Bun. Bun – A fast all-in-one JavaScript runtime. URL: <https://bun.sh/docs> (date of access: 17.11.2024).