

Node.js è un'alternativa a WordPress e PHP?

In questo articolo valuteremo Node.js come alternativa a WordPress nell'ambito dei CMS.

Con l'avvento di Ghost, Node.js è di fatto divenuto un'alternativa più che valida nel variegato mondo dei CMS. WordPress è sempre saldo al comando ma, dopo l'avvio di Calypso, ha dovuto riconsiderare il design del suo codice alla luce di quello che è diventato JavaScript oggi.

I problemi di WordPress

I problemi di WordPress non derivano da PHP. WordPress e il suo team hanno dimostrato come si possa trasformare un semplice CMS in un vero framework per la gestione di contenuti che spesso vanno ben al di là dei semplici post.

I problemi di WordPress derivano spesso dall'assoluta superficialità e sciatteria con cui molti autori di temi e plugin scrivono il loro codice.

Plugin e temi di scarsa qualità riducono la performance di WordPress e possono causare seri problemi di sicurezza.

E i problemi di performance non derivano dalla scelta di MySQL (si possono risolvere con un tuning sul server e usando strumenti di sostegno come Redis o con drop-in come Percona) ma dall'uso improprio che spesso ne fanno molti plugin.

In definitiva, molti problemi di WordPress sono gli stessi di tutte le community open-source: il problema principale è il controllo sulla qualità del codice, problema che purtroppo non può essere risolto senza sacrificare molto di quello spirito di partecipazione libero che anima queste community.

JavaScript oggi

Per anni abbiamo usato JavaScript solo lato client e lo abbiamo odiato a causa delle incompatibilità tra browser. Poi sono nati framework e librerie come jQuery, AngularJS, React e abbiamo cominciato ad amarlo.

Intanto alcuni browser-dinosauri erano spariti e le nuove leve come Chrome ed Edge prendevano sul serio gli standard e ci siamo accorti che potevamo usare JavaScript direttamente senza intermediari. ECMAScript nel frattempo è arrivato alla versione 6 e ora abbiamo `class` che per noi amanti dell'OOP in PHP ci ripaga degli anni passati a cercare di capire come mai un analogo del nostro `__construct()` in JavaScript sembrasse blasfemo.

Cominciamo a ragionare.

node -v

Node.js non è l'alternativa, ma una tra le molte alternative possibili. Ogni progetto ha le sue esigenze e non si possono dare diktat autoritari senza tenere conto del fatto che anche ciascun sviluppatore ha le sue esigenze e le sue preferenze nonché le sue skill.

Se conoscete bene JavaScript, Node.js è la vostra alternativa. Se invece usate JavaScript senza una buona padronanza del linguaggio in sè, allora è come voler creare un CMS con Zend o Symfony dopo un corso di PHP di 24 ore.

Node.js è veloce, performante, event-driven, basato sul non-blocking I/O e sull'engine V8 ecc. ecc. Questo è il mantra che ripetono tutti, ed è tutto vero. Ma è JavaScript al 100%.

Il rischio è quello di commettere errori nel design del codice, ad esempio creando routine ignorando che esiste già un metodo JavaScript che la

implementa già o producendo un codice da far venire la labirinte perché non si sa come evitare il callback hell.

Quindi JavaScript first, poi Node. Come del resto vale per WordPress: PHP first, poi WordPress.

Conclusione

Node.js non è un antagonista di WordPress: bisogna pensare in termini di alternative possibili. Solo così facendo possiamo dotarci di nuovi strumenti e progredire nella nostra formazione professionale.