

GABRIELE ROMANATO

Gli ultimi 5 anni di PHP: I maggiori cambiamenti del linguaggio

Negli ultimi cinque anni, il linguaggio PHP ha continuato la sua evoluzione, affermandosi sempre più come una scelta moderna e robusta per lo sviluppo web. Con ogni nuova versione, PHP ha introdotto funzionalità innovative, miglioramenti delle prestazioni e strumenti per rendere il linguaggio più sicuro, facile da usare e scalabile. Analizziamo i cambiamenti più significativi apportati a PHP negli ultimi anni.

1. Introduzione del tipo `Union` e del tipo `Mixed`

Un passo importante verso un sistema di tipizzazione più rigoroso è stata l'introduzione del supporto per i **Union Types**. Questa funzionalità consente ai programmatori di specificare che un valore può appartenere a più tipi, aumentando la flessibilità e la leggibilità del codice. Parallelamente, il tipo `mixed` è stato introdotto per indicare che una variabile può assumere qualsiasi tipo possibile. Questi cambiamenti hanno migliorato notevolmente il supporto al typing statico.

2. Performance migliorate

Ogni nuova versione di PHP negli ultimi anni ha portato significativi miglioramenti delle prestazioni. La capacità di PHP di gestire carichi di lavoro più elevati, grazie a una migliore ottimizzazione dell'engine Zend, ha reso il linguaggio più competitivo rispetto ad alternative come Node.js e Python. Anche il supporto per il **Just-In-Time Compiler (JIT)** introdotto in PHP 8.0 ha contribuito a migliorare l'esecuzione di codice complesso, anche se i vantaggi principali si osservano nei calcoli numerici e scientifici piuttosto che nei contesti web tradizionali.

3. Aggiunta di funzionalità orientate agli oggetti

Negli ultimi anni, PHP ha continuato a rafforzare il supporto per la programmazione orientata agli oggetti (OOP). Tra le funzionalità di rilievo ci sono:

- **Costanti di classe finali**, che permettono di definire costanti immutabili e non sovrascrivibili nelle classi derivate.
- **Named Arguments**, che consentono ai programmatori di specificare argomenti per nome anziché per posizione, migliorando la leggibilità e la manutenzione del codice.

Questi miglioramenti hanno reso PHP un linguaggio più sofisticato per lo sviluppo di applicazioni complesse.

4. Adozione di un approccio più funzionale

Il linguaggio ha incorporato concetti della programmazione funzionale, come il supporto migliorato per funzioni anonime e callable. Queste funzionalità permettono di scrivere codice più modulare e pulito. La possibilità di utilizzare la funzione `match`, una struttura di controllo simile a `switch` ma più espressiva, ha rappresentato un importante passo in questa direzione.

5. Error Handling migliorato

PHP ha migliorato il modo in cui gestisce gli errori, introducendo un supporto più robusto per eccezioni e avvisi. In PHP 8.0, è stato introdotto il **motore di gestione degli errori uniforme**, che riduce la confusione tra errori fatali, avvisi e notifiche. Questo cambiamento aiuta i programmatori a identificare e risolvere i problemi in modo più rapido ed efficace.

6. Promozione delle proprietà del costruttore

Una delle novità più apprezzate è stata la **promozione delle proprietà del costruttore**. Grazie a questa funzionalità, è possibile dichiarare e inizializzare le proprietà direttamente nel costruttore, riducendo la quantità di codice boilerplate e rendendo le classi più concise.

7. Maggiore attenzione alla sicurezza

PHP ha introdotto molte funzionalità volte a migliorare la sicurezza, come il supporto per **password hashing più robusto** e funzionalità native per evitare vulnerabilità comuni. Ogni nuova versione affronta vulnerabilità segnalate dagli sviluppatori, dimostrando l'impegno della comunità verso un linguaggio sicuro per applicazioni critiche.

8. Deprecazioni e rimozione di vecchie funzionalità

Nel corso degli anni, PHP ha abbandonato molte funzionalità obsolete per favorire uno sviluppo più moderno e conforme alle best practices. Questo processo, pur causando alcune difficoltà nella migrazione, ha permesso al linguaggio di progredire verso una maggiore coerenza e affidabilità.

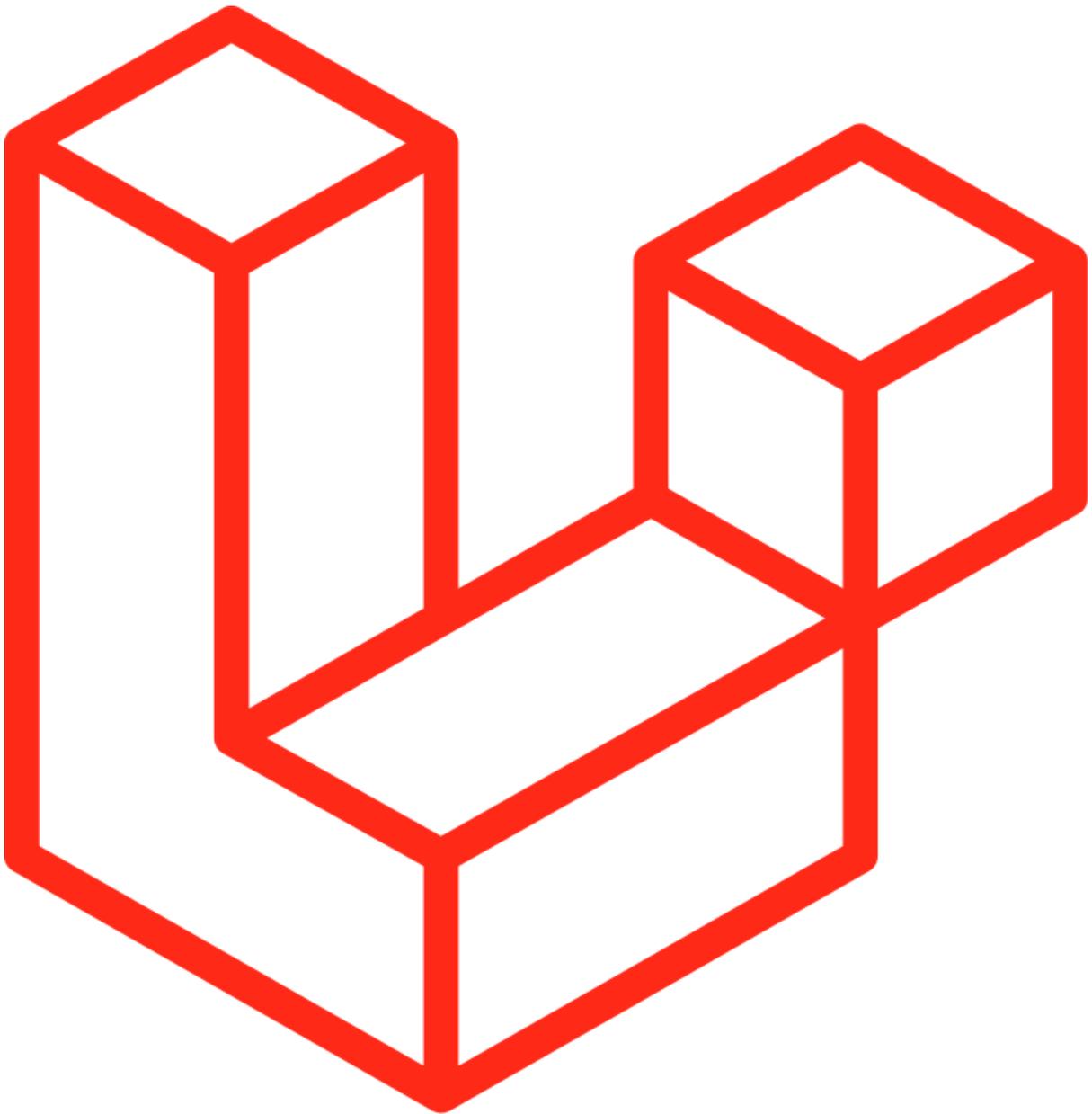
9. Supporto per Fiber e co-routine

Con l'introduzione delle **Fiber**, PHP ha compiuto un passo importante verso un modello di programmazione asincrono. Questo permette agli sviluppatori di gestire operazioni non bloccanti in modo più semplice, migliorando le prestazioni di applicazioni con elevato I/O.

Conclusioni

Negli ultimi cinque anni, PHP si è trasformato in un linguaggio molto più moderno, flessibile e orientato alle performance. Le nuove funzionalità, insieme all'impegno costante della comunità per migliorare il linguaggio, hanno contribuito a mantenere PHP come una delle scelte principali per lo sviluppo di applicazioni web. Con questi cambiamenti, PHP non è solo rimasto rilevante, ma ha consolidato la sua posizione nel panorama dello sviluppo software globale.

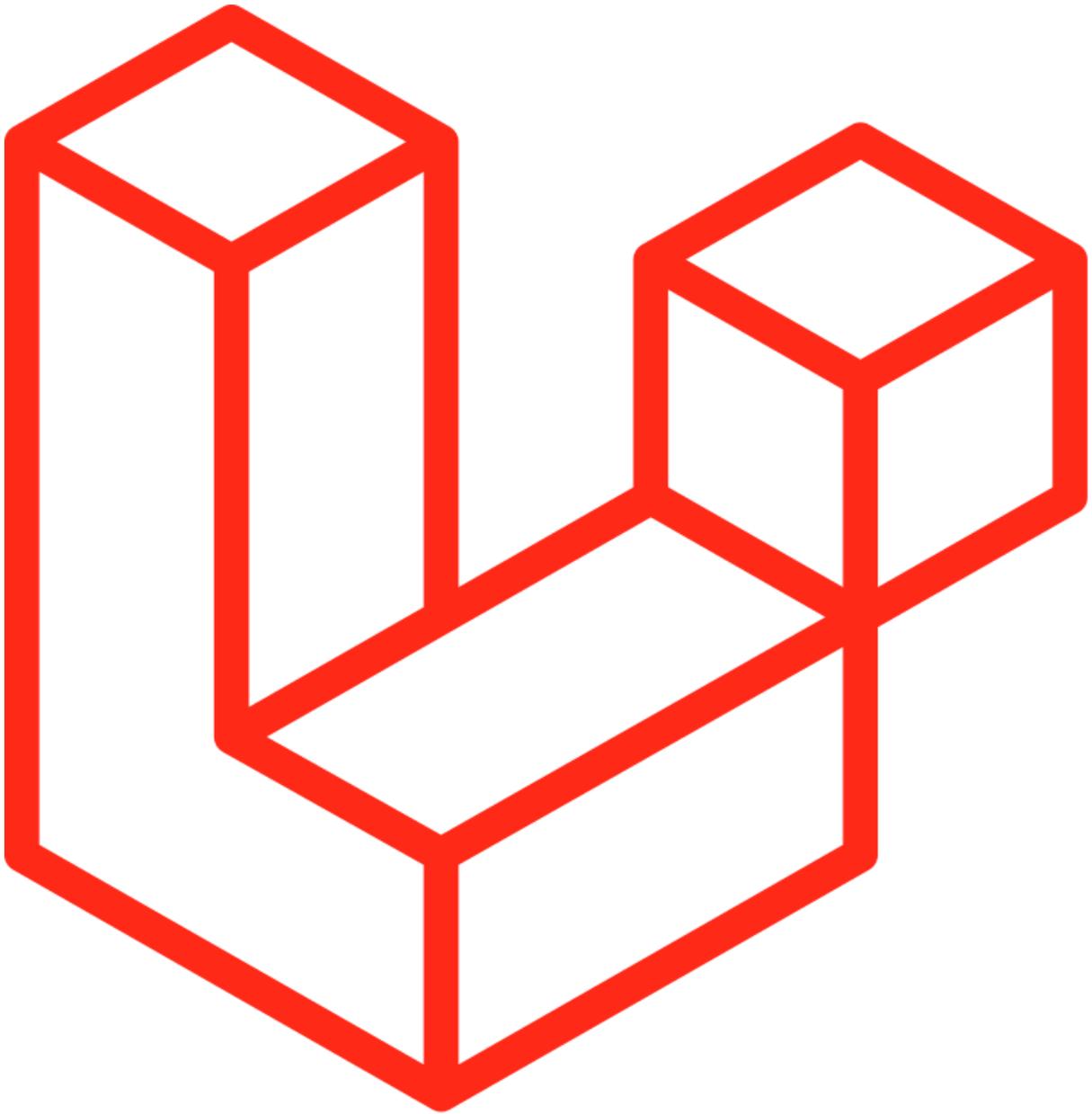
Applicazioni Correlate



-

Laravel Placeholder Image

Applicazione in Laravel per creare immagini segnaposto.
Docker Docker Compose Composer PHP Laravel JavaScript

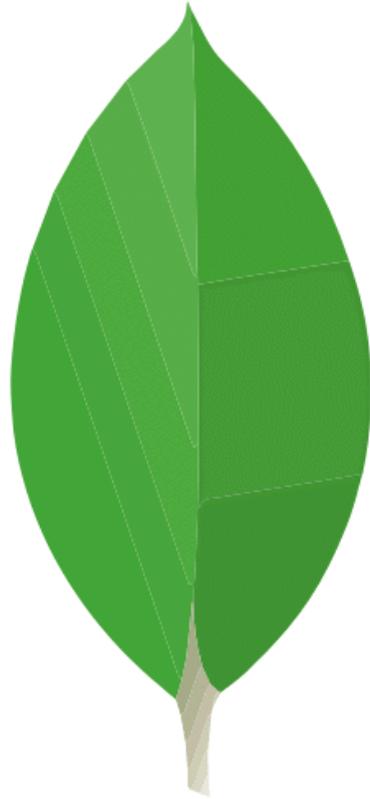


•

Laravel Shopping Cart

Applicazione che fa parte del progetto Laravel E-commerce e implementa la gestione del carrello in Laravel.

Docker Docker Compose Composer PHP Laravel



-

PHP MongoDB App

Applicazione basata su MongoDB con il driver PHP ufficiale.
Docker Docker Compose Composer PHP MongoDB