

“Veille techno backend” - NestJS

I - Comparatif de NestJS & Laravel

Introduction

NestJS et Laravel sont deux frameworks backend populaires pour développer des applications web. Chacun de ces frameworks offre des fonctionnalités puissantes et des conventions qui simplifient le processus de développement backend. Dans ce document, nous présentons donc ces deux frameworks -le premier basé sur Node.js et l'autre basé sur PHP- afin de mettre en évidence leurs avantages et inconvénients respectifs.

NestJS

NestJS est un framework backend pour Node.js qui s'inspire de l'architecture Angular. Il encourage l'utilisation de TypeScript pour ajouter du typage à Javascript.

Avantages

Basé sur Node.js : NestJS utilise Node.js comme base, bénéficiant ainsi de sa rapidité et de sa légèreté, et du fonctionnement côté serveur.

Modularité : NestJS encourage l'utilisation de modules, ce qui facilite la gestion des dépendances et la séparation des tâches dans une application.

Architecture Angular-Like : Inspiré de l'architecture d'Angular, NestJS propose une structure familière pour les développeurs qui travaillent déjà avec des frameworks front-end comme Angular.

TypeScript : NestJS est écrit en TypeScript, offrant ainsi des avantages tels que le typage statique, une lisibilité et maintenance optimale du code, les fonctionnalités ECMAScript. Ces nombreux avantages offrent une meilleure expérience de développement.

Facilité de Test : La structure modulaire de NestJS facilite les tests unitaires et d'intégration, contribuant ainsi à assurer la qualité du code.

Inconvénients

Apprentissage initial : Pour les développeurs non familiers avec Angular, la similarité de l'architecture peut nécessiter un temps d'adaptation.

Communauté moins étendue : Comparée à d'autres frameworks plus anciens, la communauté de Nest peut paraître moins développée, bien qu'elle soit en croissance.

Laravel

Laravel est un framework web PHP qui met l'accent sur l'expressivité, la simplicité et la productivité du développeur.

Avantages

Syntaxe élégante : Laravel propose une syntaxe élégante et expressive, favorisant la lisibilité du code et la productivité du développeur.

Eloquent ORM : Laravel intègre Eloquent, un ORM puissant facilitant la manipulation des bases de données et le mapping entre les objets et les tables.

Blade Templating Engine : Blade, le moteur de templating de Laravel, offre une syntaxe simple et puissante pour la création de vues.

Laravel Mix : Laravel Mix simplifie la gestion des assets, rendant le processus de compilation des fichiers CSS et JavaScript plus facile et plus efficace.

Laravel Forge : Un outil de déploiement et de gestion de serveur, Forge facilite le déploiement d'applications Laravel sur des serveurs cloud.

Inconvénients

Courbe d'apprentissage : Bien que Laravel soit réputé pour sa facilité d'utilisation, la courbe d'apprentissage initiale peut être légèrement plus raide pour les débutants.

Performance : Bien que Laravel soit rapide, il peut ne pas être aussi performant que des frameworks plus légers dans certaines situations.

II - Choix de la technologie

Ce document détaille les raisons du choix de la technologie NestJS comme framework backend pour le développement de notre application Kanban. Cette décision va de pair avec notre choix de Vue.js pour le front du projet.

Le choix de NestJS plutôt que Laravel pour un projet Vue.js se justifie par la synergie naturelle entre les technologies JavaScript/TypeScript et la similarité architecturale entre

NestJS et Angular. Cette cohérence favorise une intégration harmonieuse du back-end avec le front-end, tout en offrant la flexibilité, la scalabilité, et le support nécessaire pour mener à bien le projet de manière efficace.