

Rapport de veille back

1. NestJs / Laravel.....	1
1.1 NestJs.....	1
1.1.1 Les avantages du framework.....	1
1.1.1 Les désavantages de NestJs.....	2
1.2 Laravel.....	2

1. NestJs / Laravel

1.1 NestJs

NestJs est un framework Node Js reposant sur TypeScript. Il a été développé au cours de l'année 2017, initié par Kamil Mysliwiec.

Au fil des années suivant son apparition, il est devenu le framework avec la croissance de popularité la plus haute, notamment sur Github. Le projet a reçu un franc succès et de très nombreux likes en peu de temps.

Il se base sur un modèle d'architecture stricte se rapprochant de celui d'Angular intégrant un MVC.

Ce dernier point introduit le suivant, sur les différents avantages et inconvénients du Framework.

1.1.1 Les avantages du framework

Parmis les principaux avantages du framework, on retrouve notamment :

- Son architecture suggérée est optimale. En effet, le framework utilise un CLI proche de celui de Symfony, Laravel ou Ruby permettant de mettre en place un MVC organisé autour des modules de l'application.
- Sa structure modulaire, permettant de simplifier la division du projet en différents blocs indépendants.
- Les tests sont simplifiés de part la prise en charge de fonctionnalités telles que les modules DI. Il propose des programmes de tests simples et fiables.

- Le framework est léger et complet. Il dispose d'une multitude de modules, GraphQL, socket...

1.1.1 Les désavantages de NestJs

- Le framework est relativement bien adapté au projet de grosse envergure, mais peut s'avérer verbeux puisqu'il multiplie les fichiers dans le but de maintenir une structure organisée.
- La forte utilisation des décorateurs, presque abusive, nuit à la lisibilité du code. De plus, certains décorateurs changent la façon d'accéder à certaines données. Par exemple, `request.body` est remplacé par `@Body()`.
- NestJS encourage implicitement le monolith : bien qu'un package microservices soit proposé, le framework est très orienté vers les applications monolithiques par ses choix et fonctionnalités.
- Il peut être difficile et long à prendre en main pour des développeurs novices, ce qui peut poser problème lors de la réalisation de projets à courte durée et nécessitant une phase d'apprentissage.

1.2 Laravel

Laravel est un framework développé en 2011 par Taylor Otwell. Il est de loin, le framework PHP le plus utilisé au monde actuellement, devant Symfony, préféré par les développeurs Français.

Tout comme NestJs, il se base sur une architecture MVC offrant une bonne courbe d'apprentissage. Il met en avant d'ailleurs sa qualité de "Rapid Application Development".

1.2.1 Les avantages du framework

Parmi les avantages principaux de Laravel, on peut noter la souplesse et la propreté du code généré. A l'aide notamment de sa documentation très fournie et claire, une durée d'apprentissage contenue. Le code généré reste accessible de par sa simplicité et sa concision.

Pour en revenir à la documentation, Laravel dispose d'un grand nombre de ressources disponibles sur internet, forum, tutoriels mais également sur la documentation officielle. La documentation Laravel est même considérée comme une des meilleures dans le monde des frameworks PHP.

Sa communauté est déjà très vaste et ne cesse de s'agrandir, laissant derrière elle un nombre croissant de discussions, forums, et ressources à nouveau.

Il met en avant une intégration rapide, permettant le lancement d'un produit plus rapidement.

Il est performant et sécurisé.

Il est modulable et propose un grand nombre de paquets prêts à l'emploi et un service de messagerie.

Parmi les fonctionnalités primaires du framework, on peut relever :

- des progiciels intégrés tels que Dusk, Scout, Horizon, Passport, Telescope...
- Le moteur de template Blade permettant d'utiliser des fonctions PHP directement dans sa logique.
- L'interface Artisan, CLI permettant de générer du code, migrer les modèles de base de données..
- Le routage inverse est une fonctionnalité unique qui prend en charge la propagation automatique des modifications apportées aux routes dans les liens pertinents. En d'autres termes, il crée automatiquement des identificateurs de ressources uniformes (URI). Le développeur peut alors définir très facilement les urls d'un site web ou d'une application web, et rediriger les requêtes http vers les contrôleurs correspondants.
- L'ORM Eloquent utilise le modèle Active Record qui permet de travailler avec des bases de données relationnelles. Notamment, en utilisant des objets PHP plutôt que des requêtes SQL brutes.
Il permet, par exemple, de faire facilement un CRUD (Create, Read, Update, Delete).
- Les middlewares
- Les tests unitaires ...

1.2.1 Les inconvénients

- Pas de prise en charge de la fonction de paiement, il faut donc implémenter un module externe.
- Des éventuels problèmes de compatibilité entre les différentes versions du framework.
- Le manque de continuité entre les versions : Il n'y a pas de transition entre les versions 4 et 5 par exemple. Cela peut causer des désagréments aux développeurs qui souhaitent mettre leur code à jour.

2. Choix technique

Pour réaliser ce projet, mon choix technique s'est porté sur Laravel pour plusieurs raisons.

Premièrement, l'équipe de développement est entièrement formée au PHP et au modèle MVC, c'est donc un paramètre important à prendre en compte concernant la durée d'apprentissage du framework.

Il permet la mise en place et la modération simplifiée et rapide des bases de données. Cela facilite donc la mise en place de cette dernière et rend le processus moins long.

La structure qu'offre Laravel est optimale en termes de sécurité, permettant de mettre en place des systèmes robustes en peu de temps.

Le front-end de l'application étant en Vue Js, il est possible d'intégrer une application Vue directement au cœur de Laravel.