

Comment allons nous développer GSB ?

Afin de mener à bien notre projet GSB, nous le développons avec **PHP** et **Symfony**. Effectivement PHP est un langage de programmation ayant de multiples avantages, de plus nous utilisons le puissant framework PHP : Symfony.

PHP est un langage script HTML embarqué et open-source utilisé pour créer des pages web dynamiques. Les avantages de PHP sont :

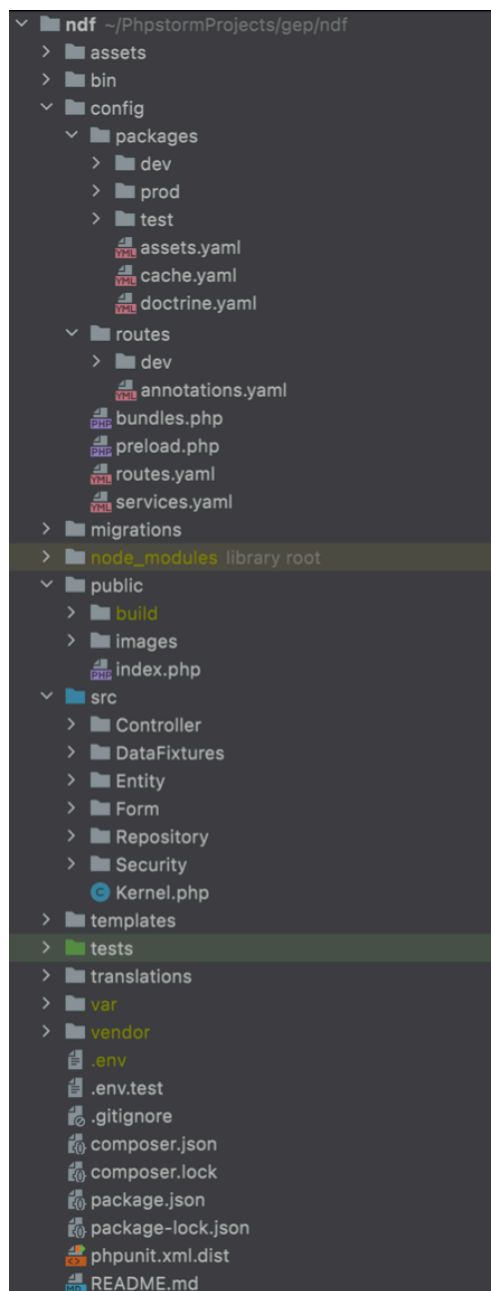
- Sa **flexibilité** et sa grande **compatibilité** avec d'autres bases de données.
- PHP peut être utilisé sur pratiquement n'importe quel système d'exploitation et n'importe quel type de serveur.
- Les applications peuvent être programmées pour s'axer sur les procédures ou sur le projet.
- Les fichiers HTML, les images, les fichiers PDF ou les fichiers Flash peuvent être générés dynamiquement avec PHP.
- PHP prend en charge une grande quantité de bases de données différentes, c'est pourquoi ce langage de script est idéal pour la création de sites web qui se basent sur les données.

En outre, PHP est considéré comme facile à maîtriser. PHP offre un support complet pour la communication du serveur avec d'autres protocoles. De plus, en informatique, un framework se présente sous la forme d'une structure composée de différentes couches indiquant quels types de programmes peuvent ou devraient être construits et la façon dont ils seraient inter-reliés.

Relativement simple d'accès, le framework Symfony est un ensemble de composants (ou « bibliothèques ») qui facilitent le développement web en réduisant de façon considérable le temps et l'effort requis pour créer des composants génériques.

L'utilisation de Symfony dans le développement de projets web à pour avantages, plus de **flexibilité** (complètement configurable), **facilité d'utilisation et grandes performances**, de **l'extensibilité** (bundle complet), **facilité de débogage** (documentation complète), **facilité des tests** (PHPUnit).

Arborescence de GSB



Notre projet est organisé sous forme de différents dossiers.

Assets

Ce dossier contient tous les assets nécessaires pour le front de notre application Symfony. Les assets sont principalement des fichiers .css et .js qui seront par la suite compilés à l'aide du package Webpack Encore.

Bin

Ce dossier contient l'exécutable Symfony et Doctrine dans la console.

Config

Toute la configuration des packages, services et routes se fera dans ce dossier.

Public

C'est le point d'entrée de l'application : chaque requête/demande passe forcément par ce dossier et le fichier index.php.

Étant accessible par tous, il est généralement utilisé pour mettre à disposition des fichiers de ressources, principalement des images.

Src

C'est le cœur du projet ! Il regroupe tout le code PHP de votre application, c'est ici que vous mettrez en place toute la logique de votre application.

Les dossiers qui seront à utiliser pour le fonctionnement de l'application sont :

- **Controller** : dossier regroupant tous les controllers.
- **Entity** : dossier regroupant toutes les entités.
- **Repository** : dossier regroupant tous les repositories
- **Form** : dossier regroupant tous les formulaires.

Templates

Ce dossier contient tous les fichiers vues, Twig.

Tests

Les tests unitaires PHPUnit seront définis ici pour tester notre application.

Translations

Si le package translation de Symfony est installé, les configurations seront dans ce dossier.

Var

Ici seront stockés le cache et les fichiers de log.

Vendor

C'est le dossier qui contient toutes les librairies utilisées par Symfony. Il est généré par Composer à l'installation.

Composer.json

Tous les packages sont enregistrés dans ce fichier. Ils sont installés automatiquement dans le dossier vendors lors de l'initialisation du projet.

Composer.lock

Il contient la version exacte des dépendances utilisées.

.env

C'est un simple fichier de configuration de texte pour contrôler les constantes de votre environnement.

.gitignore

Ce fichier permet de renseigner les fichiers que l'on veut éviter de versionner.