



Symfony

[Sous-titre du document]

Lycée Notre Dame de la Providence /
2e année BTS SIO

THIOUNE Seydina

Sommaire

Fiche technique	2
Introduction.....	3
Qu'est-ce qu'un framework?.....	3
C'est quoi Symfony?	3
Qu'est-ce que MVC?	4
Modèle.....	5
Vue	5
Contrôleur.....	5
Qu'est-ce que ORM?	5
Pourquoi le besoin de créer Symfony?	6
Composer	8
Doctrine	9
Controller.....	9
Routing	10
Avantages et inconvénients de Symfony.....	10
Avantages.....	10
Inconvénients.....	11
Quelques autres frameworks.....	11
Laravel	11
Yii	13
Conclusion	14
Sources:.....	15



Framework Symfony

Fiche technique

Développé par	SensioLabs
Première version	18 octobre 2005
Dernière version	5.0.1 (Décembre 2019)
Dépôt	github.com/symfony/symfony
Écrit en	PHP
Système d'exploitation	Multiplateforme
Environnement	Multiplateforme
Type	Framework Licence MIT
Langues	Multilingue
Site web	www.symfony.com

Introduction

PHP est l'un des langages de programmation les plus populaires au monde, et la récente version de PHP 7 a rendu ce langage de programmation côté serveur meilleur et plus stable que jamais. Cependant sur de grands projet rien ne serait mieux que d'utiliser un framework pour se faciliter la tâche. C'est dans ce contexte que j'ai choisi de parler de framework plus précisément de Symfony.

Qu'est-ce qu'un framework?

Un framework est une application **vierge** générale qui peut être facilement personnalisée en une grande variété d'applications réelles spécifiques. Il peut être une ou plusieurs bibliothèques conçus pour modéliser une architecture d'applications web, de logiciels applicatifs... Un framework fournit du code qui établit une application complète idéale qui sait «**quoi**» faire, mais ne sait pas «**comment**» le faire. Les actions spécifiques (le «**comment**») sont fournies ultérieurement de manière systématique par le programmeur. Il est composé d'un ensemble de composants et d'outils et peut être spécialisé, un domaine particulier, un plateforme spécifique, sur un langage exemple Symfony (fait en Php).

C'est quoi Symfony?

Symfony est un framework MVC (Model View Controller) open-source, écrit en PHP et 100% français.

Destiné aux projets plus complexes, voire haut de gamme, il permet de faciliter le développement et offre une bonne flexibilité. On peut en effet le configurer comme on le souhaite et aussi on peut bénéficier de l'aide d'une grande communauté.

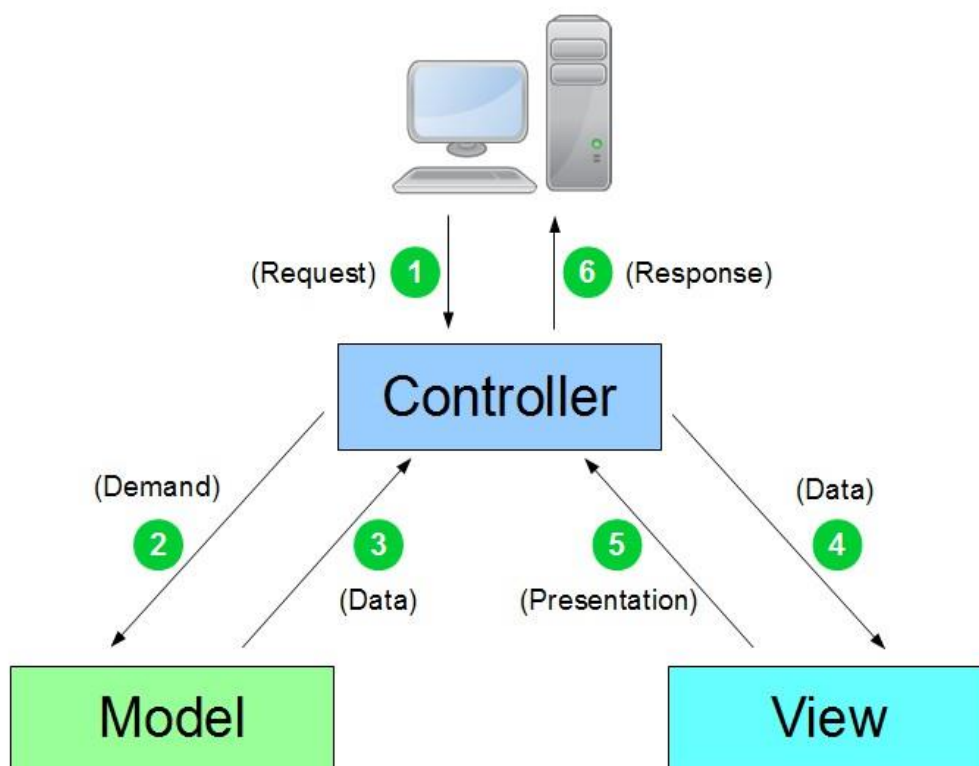
Le framework Symfony dispose d'une barre d'outils de débogage complète et facilite les tests d'où le fait de ne pas avoir besoin d'écrire de script.

Symfony est aussi un ensemble réutilisable de composants autonomes, découplés et cohésifs qui résolvent les problèmes de développement Web courants. Symfony est également un framework web full-stack.

Qu'est-ce que MVC?

Comme son nom l'indique, le Model-View-Controller (MVC) est une structure d'une application composé de trois parties: un **Modèle**, une **Vue** et un **Contrôleur**. Et chacune de ces parties a un rôle spécifique pour le développement spécifique d'une application. MVC est l'un des cadres de développement Web standard les plus utilisés pour créer des projets évolutifs et extensibles

MVC réduit la complexité des applications Web et facilite leur maintenance.



Modèle

C'est au niveau du modèle qu'on retrouve la logique principale et les objets de données qui constituent la fonctionnalité de base de l'application. On peut considérer les modèles comme une représentation des connaissances du monde réel que l'application stocke, traite et livre à l'utilisateur final.

Le modèle se charge d'obtenir les informations à partir d'une base de données, de les emballer dans des objets et de les livrer.

Vue

La vue est l'endroit où les données fournies par le modèle sont présentées à l'utilisateur. Elle supervise les éléments d'interface visuelle. La Vue sélectionne, filtre et organise les informations fournies par le modèle. Une implémentation évidente de plusieurs vues serait de gérer l'affichage des informations sur un navigateur mobile ou de bureau. On peut même dire que c'est la CSS

Contrôleur

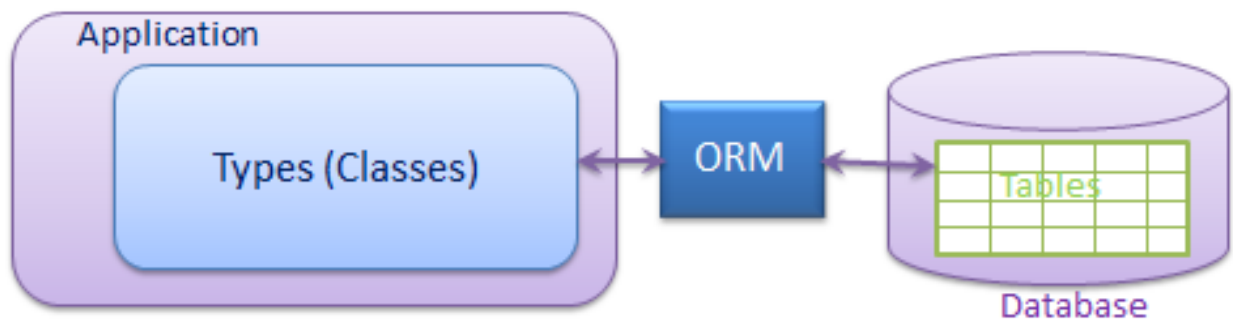
Le contrôleur est la façon dont l'utilisateur interagit avec l'application. Habituellement, le contrôleur prend les demandes des utilisateurs de l'interface (souvent sous la forme de demandes HTTP pour les applications Web) et appelle le modèle, qui récupère et traite les données, avant de les renvoyer au contrôleur. Le contrôleur utilisera alors la vue appropriée pour afficher ces résultats à l'utilisateur.

C'est un peu le capitaine de l'équipe.

Qu'est-ce que ORM?

Un ORM (Object-Relational Mapping) est une technique qui permet d'interroger et de manipuler les données d'une base de données à l'aide d'un paradigme orienté objet. Lorsqu'on parle d'ORM, la plupart des gens font référence à une bibliothèque qui implémente la technique de mappage relationnel-objet.

La bonne partie des ORM, c'est qu'ils facilitent l'écriture de code une fois qu'on les maîtrise.



Pourquoi le besoin de créer Symfony?

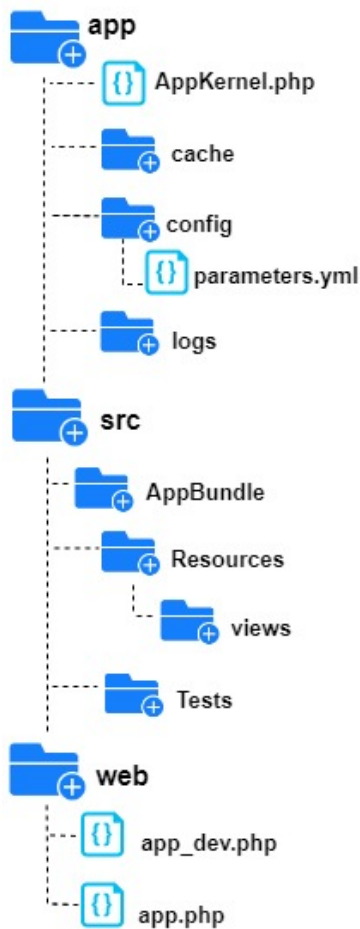
L'agence web française **SensioLabs** est à l'origine du framework **Sensio Framework**. À force de toujours recréer les mêmes fonctionnalités de gestion d'utilisateurs, gestion ORM, etc., elle a développé ce framework pour ses propres besoins. Comme ces problématiques étaient souvent les mêmes pour d'autres développeurs, le code a été par la suite partagé avec la communauté des développeurs PHP.

Le projet est alors devenu Symfony (conformément à la volonté du créateur de conserver les initiales S et F de Sensio Framework).

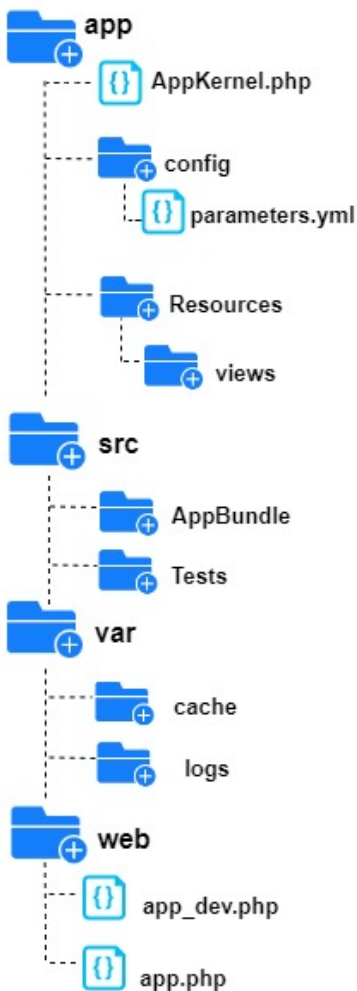
Depuis sa création en octobre 2005, Symfony est actuellement à sa 5e version ce décembre 2019. Mais la version 4.3.* reste encore plus prudent le temps de se familiariser avec cette nouvelle version.

Le 5 septembre 2017, Symfony passe la barre du milliard de téléchargements ce qui témoigne de sa popularité.

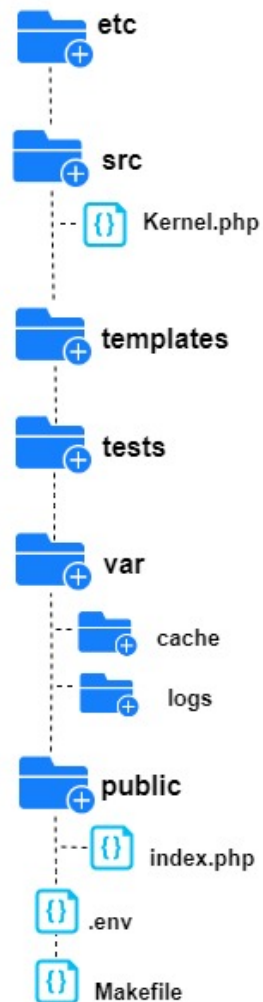
Symfony 2



Symfony 3



Symfony 4

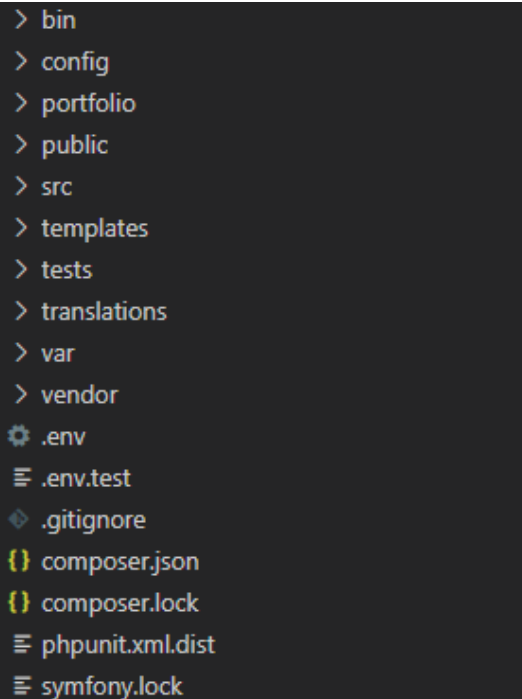


Structure de Symfony et son fonctionnement

Tout d'abord, il faut savoir que Symfony travaille avec Php mais surtout il faut utiliser la version la plus récente possible. Avec un environnement et étant débutant, pour notre projet de créer une application web, suffit de taper au niveau de votre ligne de commande

```
“composer create-project symfony/website-skeleton nom_du_projet”
```

Et vous aurez une arborescence comme suit:



```
> bin
> config
> portfolio
> public
> src
> templates
> tests
> translations
> var
> vendor
⚙ .env
≡ .env.test
💎 .gitignore
{} composer.json
{} composer.lock
≡ phpunit.xml.dist
≡ symfony.lock
```

Structure d'un projet d'application avec Symfony 4

On peut remarquer sur notre ligne de commande qui nous a permis de créer notre squelette "Composer". Mais aussi Symfony utilise Doctrine comme ORM. Les modèles dans Symfony sont créés avec Twig: un moteur de modèle flexible, rapide et sécurisé. Lorsque notre application reçoit une demande, elle exécute une *action de contrôleur* pour générer la réponse. La configuration de routage définit l'action à exécuter pour chaque URL entrante. Il fournit également d'autres fonctionnalités utiles, telles que la génération d'URL optimisées pour le référencement.

Composer

Composer est un outil de gestion des dépendances en PHP. Il permet de déclarer les bibliothèques dont un projet dépend et il les gère (installe / met à jour). Il traite des "packages" ou des bibliothèques, mais il les gère par projet, en les installant dans un répertoire (par exemple **vendor**) à l'intérieur du projet.

Doctrine

Doctrine est l'ORM pour Php par défaut du framework Symfony (depuis la version 1.3). Cependant son utilisation dans le cadre d'un projet développé avec Symfony est optionnelle. De plus, il peut être utilisé avec de nombreux autres frameworks.

La syntaxe de Twig est basée sur ces trois constructions:

- `{{ ... }}`, utilisé pour afficher le contenu d'une variable ou le résultat de l'évaluation d'une expression;
- `{% ... %}`, utilisé pour exécuter une logique, telle qu'une conditionnelle ou une boucle;
- `{# ... #}`, utilisé pour ajouter des commentaires au modèle (contrairement aux commentaires HTML, ces commentaires ne sont pas inclus dans la page rendue).

Controller

Un contrôleur est une fonction PHP qu'on crée, il lit les informations de la requête et renvoie une réponse. La réponse pourrait être une page HTML, JSON, XML, un téléchargement de fichier, une redirection, une erreur 404 ou autre chose. Le contrôleur exécute la logique arbitraire dont *notre application a* besoin pour afficher le contenu d'une page.

Twig

Le langage de création de modèles Twig permet d'écrire des modèles concis et lisibles qui sont plus conviviaux pour les concepteurs Web et, de plusieurs manières, plus puissants que les modèles PHP. Et ainsi nous permettant de modéliser nos pages web.

Routing

Les routes peuvent être configurées en YAML, XML, PHP ou en utilisant des annotations. Tous les formats offrent les mêmes fonctionnalités et performances, alors choisissez votre favori. Symfony recommande des annotations car il est pratique de placer la route et le contrôleur au même endroit.

Exemple:

```
1  <?php
2
3  namespace App\Controller;
4
5  use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
6  use Symfony\Component\Routing\Annotation\Route;
7
8  class DefaultController extends AbstractController
9  {
10     /**
11      * @Route("/", name="portfolio")
12      */
13     public function index()
14     {
15         return $this->render('default/index.html.twig', [
16             'controller_name' => 'DefaultController',
17         ]);
18     }
19 }
```

Avantages et inconvénients de Symfony

Avantages

L'utilisation d'un framework n'est pas indispensable, mais c'est une bonne pratique pour simplifier la vie du codeur. Les avantages pour pousser à utiliser Symfony sont:

- **Haute flexibilité** : avec les avantages des bundles et des composants, il s'agit de l'un des frameworks PHP les plus riches en fonctionnalités.
- **Prise en charge commerciale** : c'est l'un des rares cadres pris en charge commercialement. SensioLabs fournit des tutoriels et certifications officiels.

- **Fiabilité** : Symfony s'est avéré être l'un des frameworks les plus fiables. Peu de grandes marques ont fait confiance à symfony avec leurs sites Web.
- **Documentation bien gérée** : la documentation Symfony est considérée comme l'une des meilleures par rapport aux autres frameworks.
- **Une grande communauté** : C'est depuis 2017 que Symfony a atteint le milliard de téléchargement témoignant de cette communauté.
- **Longévité** : stable et bien testé avec des mises à jour régulières. Les versions les plus récentes restent prises en charge à long terme et compatibles.

Inconvénients

Difficile de trouver une chose parfait donc Symfony aussi a des inconvénients:

- C'est une pratique courante de Symfony d'utiliser des modèles de conception, c'est-à-dire d'appliquer une fois le code créé pour d'autres parties du système. Cela facilite les tests et la prise en charge, mais peut réduire la vitesse de développement et retarder la date de lancement.
- Symfony nécessite beaucoup plus d'efforts pour apprendre que, par exemple, Laravel et Yii, qui sont plus intuitivement compréhensibles.

Quelques autres frameworks

Dans cette partie nous allons essentiellement parlé de deux autres frameworks Php comme Symfony à savoir Laravel et Yii.

Laravel

Laravel est un framework web open-source écrit en PHP respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet. Laravel est distribué sous licence MIT, avec ses sources hébergées sur GitHub.

Laravel reste pourtant basé sur son Symfony, pour au moins 30 % de ses lignes (utilisation de "Symfony component").

Caractéristiques principales de laravel:

- **Pagination automatique:** La fonction de pagination automatique dans le framework PHP laravel s'occupe des paramètres généraux.
- **Systèmes de routage:** le framework PHP Laravel fournit une référence au développeur laravel pour sélectionner la route à déclencher sur l'application. Les routes dans le framework PHP Laravel sont définies dans l'application, HTTP, routes ou fichier PHP. Le fichier est automatiquement téléchargé par le framework Laravel. La méthodologie de définition de l'itinéraire est simple et expressive qui reconnaît une URL et une fermeture. L'utilisateur n'a qu'à saisir le nom de l'itinéraire et la bonne URL sera insérée automatiquement.
- **Test unitaire:** Le test de code est facilité par le framework PHP Laravel avec l'utilitaire de ligne de commande artisan. En utilisant cette fonctionnalité, les développeurs peuvent facilement écrire leurs tests unitaires.
- **Moteur de modèles:** créez des mises en page époustouflantes à l'aide de modèles Laravel légers avec amorçage de contenu dynamique. Les widgets structurels CSS et JS Code sont également là pour aider.
- **Eloquent ORM:** qui est plus rapide que les autres Frameworks PHP.
- **Bibliothèques:** les bibliothèques orientées objet préinstallées ne sont accessibles que dans Laravel et dans aucun autre framework PHP. La bibliothèque d'authentification est sans problème à mettre en œuvre, et elle dispose également de fonctionnalités telles que la protection contre la contrefaçon de requêtes intersites, la vérification des utilisateurs actifs, le hachage Bcrypt, le chiffrement ainsi que la réinitialisation du mot de passe.
- **Modulaire:** Les vingt bibliothèques diverses sur lesquelles Laravel est basé sont aliénées en modules de détachement qui permettent aux développeurs de laravel de créer des applications Web réactives, modulaires et utiles.

Yii

Créé par Qiang Xue en 2008, Yii est un framework de développement d'applications / web sécurisé, rapide et performant. Yii utilise le gestionnaire de dépendances Composer pour PHP pour gérer différentes dépendances et installations (plus d'informations plus loin). Yii est également le framework PHP le plus rapide, grâce à la technique de chargement paresseux.

Une autre grande fonctionnalité de Yii est l'intégration de jQuery. L'intégration permet aux développeurs frontaux d'embrasser rapidement le framework et utilise des échafaudages pour générer du code. Semblable à Symfony, Yii utilise également des composants pour permettre le développement rapide d'applications.

Caractéristiques clés de Yii 2:

- Yii aide à créer une application plus sûre en utilisant les fonctionnalités d'un composant d'application de sécurité .
- Il exploite les fonctionnalités modernisées de PHP comme la liaison statique, les classes SPL, les interfaces et les fonctions anonymes.
- L'installation de Yii est gérée par le compositeur, ce qui rend son installation très simple.
- Tous les éléments de cache de Yii sont étendus à partir de Yii / caching / Cache, ce qui accorde le système de cache nécessaire lorsqu'une API commune est utilisée.
- Il s'agit généralement du système crud le plus couramment utilisé et possède des fonctionnalités qui évitent le codage répétitif et les validations jquery inutiles



Conclusion

Enfin, dire qu'un **framework** est, dans certaines situations, appelé **moteur**. Il appelle les fonctions qu'on doit fournir, lorsqu'elles sont déterminées.

Juste pour dire qu'il n'y a probablement pas de bonne réponse sur quelle solution convient le mieux à un projet. On peut lire des tonnes de recherches et ne pas recevoir une vision claire. Tout outil peut être également bon ou mauvais selon les personnes qui l'utilisent.

Un framework ne veut rien dire sans un développeur qui est bon en lui. Si vous trouvez un excellent spécialiste, peu importe le framework à utiliser dans la mesure où il en est compétent.

Sources:

Référence de cours

<https://fr.wikipedia.org/wiki/Symfony>

<https://symfony.com>

<https://www.youtube.com/watch?reload=9&v=UTusmVpwJXo>

<https://openclassrooms.com/fr/courses/3619856-developpez-votre-site-web-avec-le-framework-symfony>

<https://www.grafikart.fr/formations/symfony-4-pratique>