

## Veille Technologique : Framework Symfony

**Définition :** la veille technologique consiste à s'informer de façon régulière sur une / des technique(s) et leurs mises à disposition. Les informations recueillies peuvent concerner une matière première, un produit, un composant, un procédé.... Cette recherche régulière permet ainsi de rester informé sur les évolutions futures des techniques et de s'en inspirer pour éviter d'éventuels risques.

### Introduction :

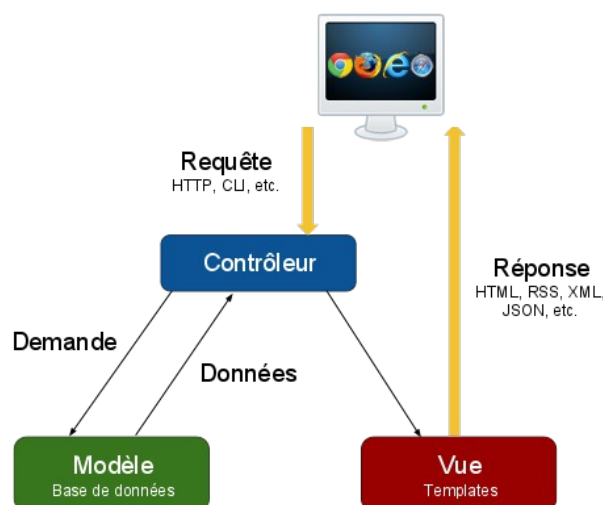
Le framework « Symfony » est un ensemble de composants PHP libre (Open Source) pour les applications web basée sur l'architecture Modèles / Vues / Contrôleurs (MVC). Et est actuellement développé par une société web Française « Sensio Labs » depuis le 18 octobre 2005. Elle a dernièrement sorti la version 4.2.3 du framework, le 3 février 2019.

À l'origine le projet « Symfony » a été créé pour les besoins propres de l'entreprise « SensioLabs » qui créait des fonctionnalités de gestion d'utilisateurs, d'ORM.... Les problématiques étant souvent les mêmes pour d'autres développeurs, le framework a par la suite été partagé avec la communauté des développeurs PHP.

« Symfony » est aujourd'hui utilisé par de nombreuses entreprises, notamment Yahoo, Dailymotion, Drupal, Meetic, Vogue... dû à sa robustesse, sa fiabilité, sa puissance et sa capacité d'adaptation à un très grand nombre de configurations.

### Définition :

Un framework par définition est un environnement de développement où des bibliothèques sont mises en commun pour qu'une base vérifiée soit prête à l'emploi. Il est souvent basé sur l'architecture MVC (Modèles, Vues, Contrôleurs) pour séparer la partie présentation et logique. Cela permet ainsi d'avoir un environnement facile à prendre en main pour des projets d'équipes stables sur le long terme.



## Histoire :

### Version 1 : Symfony 1

À l'origine la première version était nommée **Sensio Framework**, mais suite à la demande du créateur qui voulait garder les initiales de Sensio Framework elle a été renommée SymFony (initiales SF). La dernière version 1 est sortie en novembre 2009 et c'est à partir de novembre 2012 qu'il a été conseillé de migrer vers la version Symfony 2.

### Version 2 : Symfony 2

Suite à l'agrandissement de la communauté (250 contributeurs) le principe des bundles apparaît et remplace celui des plugins. La différence était le poids conséquent d'un plugin car il possède de nombreuses configurations, modules... et peut devenir très lourds en peu de temps. Le bundle réussi à découper l'application en se concentrant sur une seule fonctionnalité.

La version 2 améliore aussi la notion d'autoloading. Passant d'un tableau récupérant toutes les classes PHP à un nouvel outil « Composer » qui gère les dépendances pour PHP.

### Version 3 : Symfony 3

Sortie en Novembre 2015 cette version est très spéciale car elle a supprimée de nombreuses fonctionnalités peu appréciées dans la dernière version 2. Elle a tout de même apportées des nouveautés telles que MicroKernel, DependencyInjection, LDAP connection, Guard Authentification....

### Symfony 3 - Nouvelles fonctionnalités

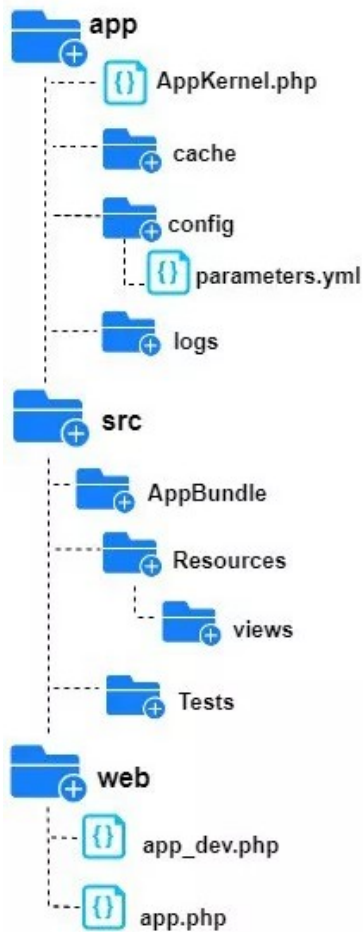
### Version 4 : Symfony 4

Sortie en Novembre 2017 cette version majeure apporte son lot de nouveauté. Notamment le plugin Composer Symfony Flex pour gérer l'installation d'une application Symfony. Elle devient plus légère (70% plus légère qu'une nouvelle application faite avec Symfony 3). Elle introduit de nouvelles commandes de bases et des packages utiles.

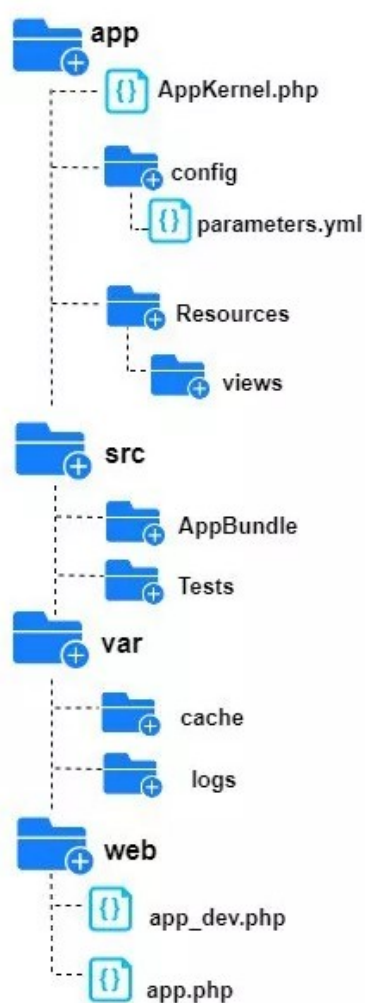
Ils ont par exemple remarqué que les variables d'environnement déployées dans les applications fuitées trop souvent et qu'il fallait par conséquent les stockées dans des fichiers virtuels exposés par des plateformes comme Docker. Ils ont par conséquent développés une fonctionnalité permettant de lire dynamiquement les fichiers à l'exécution.

Ils ont aussi utilisé la version 7.1 de PHP ce qui n'était pas le cas des versions précédentes.

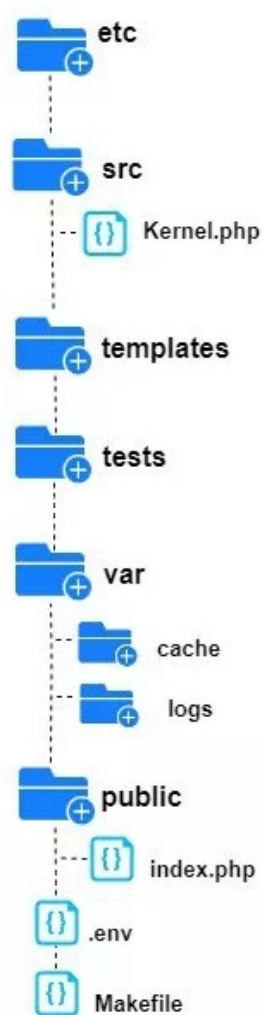
## Symfony 2



## Symfony 3



## Symfony 4



Comparaison des structures de dossiers entre Symfony 2, 3 et 4

### Utilisation :

Comme tous les frameworks, « Symfony » possède une librairie normalisée avec un ensemble d'éléments et de composants intégrables rapidement ainsi qu'une méthodologie et une architecture propre. Ces composants permettent ainsi de faire moins d'erreur, d'avoir une meilleure gestion des fichiers et par conséquent d'être plus productif.

Symfony simplifie aussi le code à l'aide de couches d'abstractions entre les différents éléments de l'application grâce aux multiples drivers de bases de données qu'il possède (Doctrine, ORM, PDO...).

### Installation :

*Éléments requis : PHP version 7.13 (ou plus), les extensions PHP : CType, ICONV, JSON, PCRE, Session, SimpleXML, Tokenizer un environnement de développement fonctionnel (serveur APACHE, PHP, MYSQL, éditeur de texte).*

- Vérifier les éléments requis :

```
$ cd your-project/  
$ composer remove symfony/requirements-checker
```

- Installer en ligne de commande :

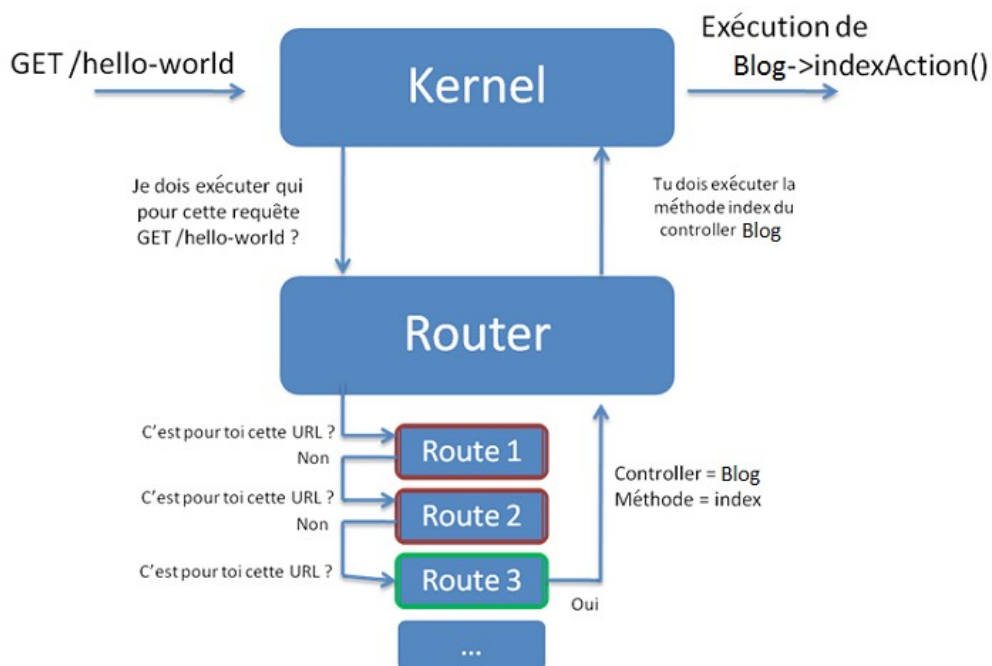
```
C:\> php -r "file_put_contents('symfony', file_get_contents(https://symfony.com/installer));"
```

- Créer un projet en ligne de commande :

```
C:\> php symfony new my_project
```

### Fonctionnement :

Pour fonctionner « Symfony » utilise un système de route (définie dans le fichier `/app/config/routing.yml`) et de bundles. Chaque bundle possède ses propres contrôleurs et ses vues qui sont gérés par le système de route.



Contrairement au fonctionnement du modèle MVC, « Symfony » ne fonctionne pas par répertoire mais par appel de fonction. Le contrôleur va parcourir les différentes URL jusqu'à tomber sur la bonne route puis va stopper sa recherche une fois trouvée.

1/ Dossier **APP** : contient toute la configuration du projet (variables globales, configuration sécurité, configuration des bundles...);

2/ Dossier **VENDOR** : contient toutes les librairies du projet sous forme de regroupement de fonction et de classe.

3/ Dossier **WEB** : contient la partie publique de l'application web (fichiers JS, CSS, images...).

4/ Dossier **SRC** : contient l'application décomposée en plusieurs bundles qui eux-même possèdent au minimum des contrôleurs, des vues, des informations de configurations et des routes.

Le point de départ de l'application est donc le fichier routing.yml qui contient toutes les routes auquel le bundle peut répondre par une action d'un contrôleur.

*Exemple routing.yml :*

```
1 page_show:
2   path: /{name}
3   defaults: { _controller: SupPageBundle:Page:show, name: index}
4   requirements:
5     name: "[a-zA-Z0-9 ]+"
6
7 page_add:
8   path: admin/page/add
9   defaults: { _controller: SupPageBundle:Page:add}
10
11 page_update:
12   path: admin/page/update/{name}
13   defaults: { _controller: SupPageBundle:Page:update}
14   requirements:
15     name: "[a-zA-Z0-9 ]+"
16
17 page_remove:
18   path: admin/page/remove/{name}
19   defaults: { _controller: SupPageBundle:Page:remove}
20   requirements:
21     name: "[a-zA-Z0-9 ]+"
22
```

Chaque route possède un nom « page » et un chemin « /page / » ainsi qu'un paramètre « {param} ». On fait correspondre à cela un contrôleur avec la syntaxe « NomBundle :NomContrôleur :NomAction ».

## **Avantages :**

### **Gain de temps**

Pour un développeur « Symfony » est un gain de temps de qualité et optimisé. Via l'utilisation des composants déjà existants on peut se concentrer sur les tâches importantes et complexes ce qui nous garanti un projet durable et solide. Un développeur ayant été formé sur Symfony pourra ainsi reprendre en main n'importe qu'elle solution sans avoir participé à son élaboration. En effet, comme les fichiers doivent respecter une syntaxe particulière ainsi qu'une arborescence précise cela garantit une facilité de maintenance sur le long terme et les développeurs savent rapidement dans quel fichier il faut aller pour apporter des modifications. L'architecture MVC séparant la logique métier de l'application et la présentation permet à un intégrateur web, voir un webdesigner d'intervenir sur la présentation (vues) du projet sans avoir à intervenir sur les fichiers PHP complexes.

### **Sécurité**

« Symfony » intègre des mesures de sécurité préventives pour lutter contre les failles de sécurités importantes (injections SQL...). Il apporte de plus une facilité de débogage via un profileur intégré et une barre d'outil extensible et fournit également des détails exacts comme le temps nécessaire au rendu d'une vue, la quantité de mémoire utilisée....

### **Optimisation**

Divers environnements de développement sont livrés avec un panel d'outil qui aident à automatiser les tâches d'ingénierie logicielle ordinaire, avec par exemple des outils de génération de code pour le prototypage et l'administration du back-end en un seul clic....

### **Open Source**

De plus « Symfony » est un framework libre, open source, et n'impose aucunes contraintes et permet aussi de développer des solutions propriétaires.

### **Communauté importante**

La communauté de développeurs contribuent aussi au succès du framework. Il se présente dans le top 3 des frameworks mondiaux PHP Open Source. Sa communauté assure donc une documentation complète et à jour, un dépannage rapide en cas de problème via des plateformes comme StackOverflow, Slack et tout autre forums de développeurs. De nombreux articles présentant des astuces et des retours d'expériences sont aussi en ligne, ce qui permet de se faire un avis au préalable pour faire les bons choix.

### **Développée sur le marché professionnel**

Enfin « Symfony » est un framework très implanté sur le marché professionnel, connaître cette technologie ouvre donc de nombreuses portes d'emploi intéressantes sur des projets complexes et intéressants. Notamment car l'utilisation de ce framework est plutôt portée sur des projets comportants de nombreuses requêtes et une gestion des utilisateurs. Il est par conséquent moins adaptés aux petits projets, comme les sites vitrines par exemple.

## **Inconvénients :**

### **Temps d'apprentissage**

Le temps d'apprentissage est bien plus élevé que pour un langage lui-même. Il est nécessaire d'apprendre le fonctionnement d'autres technologies telles que YAM ou Doctrine qui sont des outils complémentaires indispensables.

### **Pas idéal pour les petits projets**

Le framework étant assez lourd il n'est pas idéal pour les petits projets tels que les sites vitrines. Il faut de plus utiliser des lignes de commandes pour l'installer, pour la gestion des bases de données ou encore pour l'installation de bundles.

### **Manipulations importantes**

Comme il a été mentionné précédemment l'utilisation des routes prend du temps à être maîtrisée et quand bien même cela est fait il n'en reste pas difficile de maîtriser la technologie entièrement. Ce qui demande parfois de manipuler et de bidouiller les lignes de codes pour obtenir ce que l'on veut. Si bien que le côté « optimisé » peu parfois être un désavantage.

## **Outils du framework :**

### *1/ Doctrine :*

« Doctrine » est un ORM (Object Relation Manager), son but est de permettre à l'utilisateur de communiquer avec une base de données. C'est donc un outil indispensable mais optionnel de Symfony, et permet ainsi de faire persister les objets PHP dans la base de données mais aussi de les récupérer via un travail de mapping. Il permet aussi de créer des entités qui pourront être utilisées comme objet dans PHP et comme base dans la base de données).

### *2/ Composer :*

« Composer » est un gestionnaire de version des bundles de Symfony. Il est composé d'une bibliothèque de gestion des dépendances pour PHP qui permet d'installer des librairies à nos projets. En cas de dépendances dans les librairies « Composer » va télécharger les autres librairies nécessaires et les installer.

### *3/ Twig*

« Twig » est un moteur de template permettant de séparer les données PHP et le code HTML. Dans la gestion d'équipe cela permet de distinguer le travail d'un graphiste de celui des développeurs. Il a donc pour avantage d'être rapide, sécurisé et flexible.

### Frameworks concurrents :

- **Le Zend Framework** : est un framework pour PHP 5 créé en Mars 2006 par Zend Technologie. Il a pour but de simplifier le développement web tout en recommandant les bonnes pratiques orientées objets en offrant des outils aux développeurs via des librairies. Il est aussi basé sur l'architecture MVC sans l'obliger. Zend Framework est ainsi compatible avec PHP 5, possède une conception totalement orientée objet. Il a une faible dépendance des composants entre eux, possède une communauté importante et est adapté à l'environnement professionnel.
- **Code Igniter** : est un framework PHP qui permet d'améliorer le développement des projets en fournissant un ensemble de bibliothèques prenant en charge les fonctions répétitives via une interface simple et une structure logique. Ainsi CodeIgniter est très léger et stable, son apprentissage est rapide et la communauté est active. Il répond néanmoins seulement aux applications à petite échelle et n'est pas compatible avec Ajax.



2009

Mise en ligne de la version 1 de  
Symfony.

- Sortie de Symfony 3 en 2015 (mise à jour  
des documentations sur symfony.com)
- Présentation de Symfony dans un article  
de SupInfo par Solène Limousin (2016)

2018

Choix du sujet de veille  
technologique. Début de  
recherche d'informations diverses  
sur le sujet Symfony.

- Article de Raphel Bellon sur le framework  
Symfony
- Veille technologique d'Alexis Saludo
- Veille technologique de Benoit Sauvage

- Migration vers Symfony 2 en 2012

- Article de Olivier Pons sur les avantages et  
inconvénients de Symfony

2017

Ouverture du compte Medium  
(réseau social professionnel) du  
créateur de Symfony : Fabien  
Potencier

- Sortie de Symfony 4 en 2017 (mise à jour  
des documentations sur symfony.com)

- Article de Novaway et son expertise sur  
l'utilisation du framework en entreprise  
(2018)

2019

Rédaction de ma veille  
technologique sur Symfony



## **Technologies mise en places pour récupérer les informations :**

*Suivi du profil du fondateur du projet « Symfony » :*

<https://medium.com/@fabpot>

*Article sur l'histoire de Symfony :*

<https://medium.com/slickteam/le-framework-symfony-7ecca18a95a9>

*Flux RSS*

## **Sources :**

<http://symfony.com/doc/current/index.html>

<https://openclassrooms.com/courses/developpez-votre-site-web-avec-le-framework-symfony2/symfony2-un-framework-php>

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

<https://raphaelbellon.wordpress.com/veille-technologique/>

<https://alexisjsaludo.wordpress.com/symfony-2/>

<https://benoitsauvage.wordpress.com/veille-technologique/>

<https://docplayer.fr/15966944-Veille-technologique-symfony-2.html>

<https://www.supinfo.com/articles/single/1685-framework-symfony>

<https://www.novaway.fr/notre-expertise/symfony>

<https://www.olivierpons.fr/2014/03/22/symfony-2-avantages-et-inconvenients/>