Ressourcesinformatiques

+ QUIZ

Version en ligne
OFFERTE !
pendant 1 an

Apprendre à développer des applications web avec PHP et Symfony

En téléchargement

code source des exemples

Yves ROCAMORA

eni

Table des matières

Les éléments à télécharger sont disponibles à l'adresse suivante :
http://www.editions-eni.fr
Saisissez la référence ENI de l'ouvrage **RIPHSYM** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Chapitre 3
Le langage Objet

Chapitre 5
Les frameworks

Chapitre 6
L'outil Composer

Chapitre 7
Installation de Symfony

Chapitre 8
Configurer une application

Chapitre 9
Une première application

Chapitre 10
Le routage

Chapitre 11
Le moteur de template Twig

Chapitre 16
Les formulaires

Chapitre 17
La sécurité

Chapitre 5
Les frameworks

1. Un framework : pour quoi faire ?

Nous avons vu précédemment, comment créer du code en PHP et construire une page web.

Il est très possible de développer des applications en utilisant directement le langage PHP.

Alors pourquoi utiliser un framework ?

Un framework est une structure de base préétablie dans laquelle vous allez pouvoir développer.

1.1 Quels sont les avantages ?

– La structure du framework vous permet d'aller plus vite dans le développement du code. Un framework contient la plupart des classes dont vous aurez besoin pour mettre au point votre application. Vous n'avez pas besoin de tout développer vous-même !
– Votre code est déjà structuré. Le découpage en sous-dossiers et fichiers est déjà défini. Vous n'avez plus qu'à vous y conformer. De plus, cette structure est optimale.

- Les développeurs qui travailleront sur votre application utiliseront la même structure, les mêmes standards de code. Les bonnes pratiques sont encouragées. Il est plus facile de travailler à plusieurs lorsqu'on adopte la même logique. Il en est fini des codes de développeurs solitaires, qui étaient difficiles à reprendre et à redévelopper pour les autres.
- Il est plus facile pour un développeur connaissant le framework de plonger dans votre application. Il mettra moins de temps à pouvoir travailler dessus. Vous trouverez facilement sur les sites d'annonces d'emplois un candidat qui maîtrisera le framework.
- Les frameworks les plus célèbres possèdent une large communauté. Vous ne serez plus seul face à votre code.
- Vous pourrez récupérer certains composants de votre code pour d'autres applications. L'utilisation du framework entraîne la réutilisabilité du code.
- Un framework évolue et se maintient dans le temps. Vous bénéficierez de cette évolution. Un petit bémol toutefois, quand les nouvelles versions ne sont plus compatibles avec les anciennes, mais cela arrive rarement.

Comme tout, le framework présente aussi des inconvénients.

1.2 Quels sont les inconvénients ?

- Un framework entraîne le chargement de bibliothèques lourdes et dont vous n'avez pas forcément besoin. Cela peut avoir un impact sur la performance et le temps de réponse. Pour pallier ce problème, les frameworks proposent une version light (allégée) qui vous laisse le soin de charger uniquement les bibliothèques dont vous avez besoin de manière dynamique.
- Vous n'avez plus besoin d'écrire votre propre code. Eh oui, c'est un inconvénient. Beaucoup d'utilisateurs de frameworks aujourd'hui n'ont plus le goût, ni même l'aptitude à développer et c'est dommage. Vous aurez tendance à chercher la solution à votre problème dans les outils du framework ou de la communauté au lieu de les inventer vous-même.

– L'utilisation d'un framework implique une période d'apprentissage (d'où cet ouvrage). On ne plonge pas dans un framework sans en connaître les fondamentaux. C'est donc une étape supplémentaire à maîtriser par rapport au langage PHP.
– Un framework évolue sans arrêt. Il se peut que certaines mises à jour mettent à mal vos applications passées. Il faudra faire évoluer vos applications en fonction des mises à jour.
– Réfléchissez bien au framework que vous allez utiliser et ne vous laissez pas séduire par la popularité d'un framework. On a vu dans le passé des frameworks très populaires qui, cinq ans après, sont devenus obsolètes. La mode change, c'est valable aussi pour les frameworks.

Vous avez, a priori, choisi Symfony et c'est un bon choix au regard de ses possibilités, de son utilisation et de son évolution. Vous n'aurez donc pas ce genre de déconvenues dans l'avenir.

2. Les frameworks PHP

À l'heure où ces lignes sont écrites, il existe trois grands frameworks PHP qui se partagent le marché :

– Symfony, bien sûr, qui est incontestablement le framework le plus apprécié, surtout en Europe. Il possède une très grosse communauté. C'est un framework français (développé par SensioLabs). Son utilisation est grandissante, notamment pour de gros projets comme Drupal 8, eZ Publish 5, Dailymotion, BlablaCar…
– Laravel, le plus gros concurrent de Symfony. Il reprend beaucoup de composants issus de Symfony, comme le système de routage, la gestion des formulaires, les classes de requêtes et de réponses... Il est très utilisé aux États-Unis.
– CodeIgniter, qui est le plus simple des frameworks. On peut l'appréhender en moins d'une heure. C'est ce qui fait sa popularité. Il est utile pour ceux qui veulent avoir une structure de base de framework et rien de plus. Mais ses fonctionnalités sont beaucoup moins développées que celles des autres.

Il existe d'autres frameworks, comme Zend, Yii ou CakePhp qui sont beaucoup moins utilisés aujourd'hui.

3. Le framework Symfony

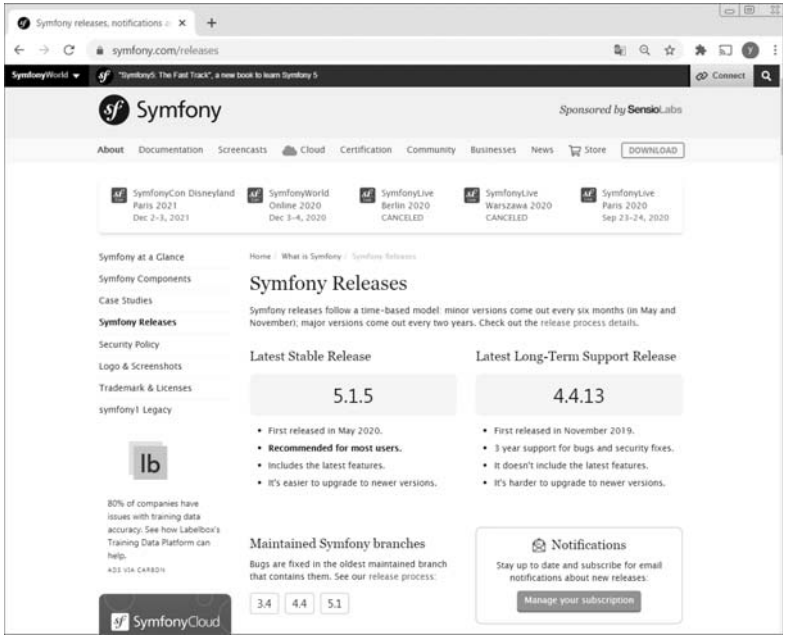
3.1 Présentation du framework

Le framework Symfony est celui qui nous intéresse particulièrement. Sa grande communauté, son adaptabilité, sa longévité en font l'un des frameworks les plus fiables aujourd'hui.

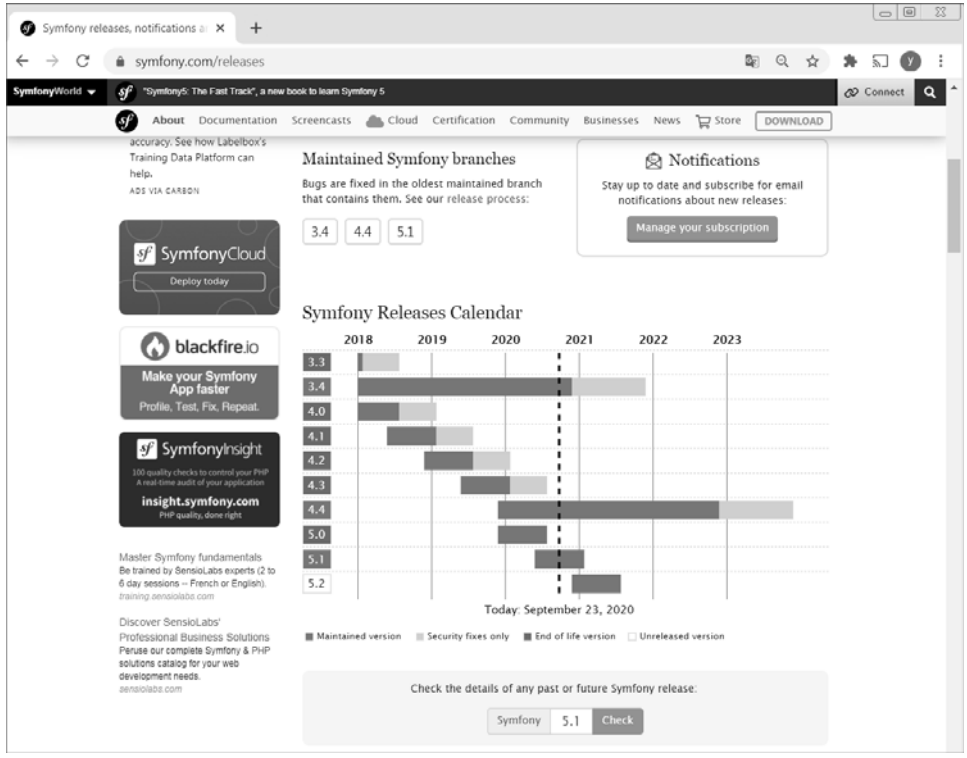
Symfony est en open source, ce qui signifie que vous pouvez le télécharger et l'utiliser gratuitement même pour des applications commerciales. Sa première version est sortie en 2005.

Rendons-nous sur le site de Symfony : https://symfony.com

Si vous cliquez sur le menu **About** dans la barre de navigation supérieure, puis sur **Symfony Releases** dans le menu vertical gauche, vous découvrez la version actuelle de Symfony (*Latest Stable Release*) et la dernière version de support à long terme (*Latest Long-Term Support Release*) :



Si vous scrollez sur la page, vous découvrez la *roadmap*, c'est-à-dire le calendrier des mises à jour de chaque version :



On s'aperçoit que deux versions continuent à être supportées longuement : la version 3.4 et la version 4.4. Ceci s'explique par le fait qu'un changement important de la structure de base de Symfony a été effectué à partir de la version 4.

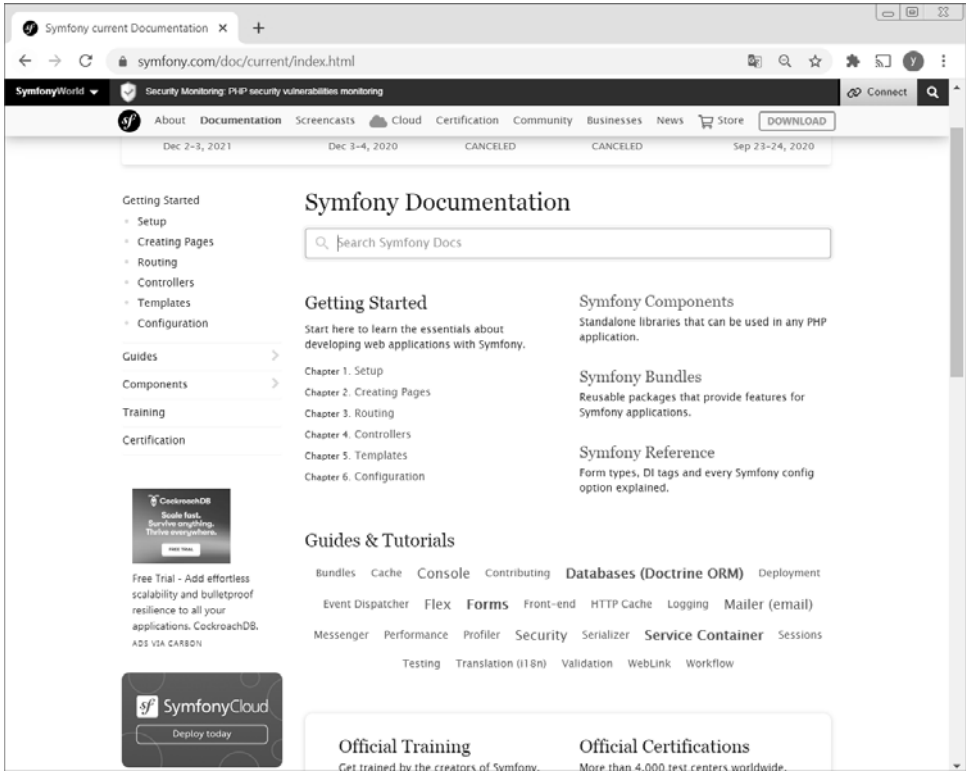
Vos applications en version 3.4 ou antérieure doivent être remaniées pour pouvoir fonctionner dans la version 4. Symfony continue donc à supporter la version 3.4, afin de laisser un peu de temps aux entreprises pour leur permettre de faire leurs migrations (jusqu'en 2022).

Nous allons utiliser la version actuelle, la version 5, qui ne présente pas de modifications majeures par rapport à la version 4.

3.2 La documentation

La documentation est très bien faite en Symfony. Nous vous engageons vivement à vous y référer le plus souvent possible. Le premier réflexe lorsqu'on a besoin d'une solution en Symfony (avant de faire une recherche sur Internet) est de consulter la documentation.

- Cliquez sur l'onglet **Documentation**. Vous avez une zone de recherche qui vous permet de rechercher à l'aide de mots-clés des informations dans la documentation.



Vous disposez de plusieurs sortes de documentation :

Le Quick Tour : vous le trouvez en cliquant sur l'onglet **About** et en scrollant vers le bas. C'est un tour rapide d'utilisation de Symfony. Vous y apprendrez comment installer Symfony et les fondamentaux pour son utilisation basique. Il vous permettra de vous faire une opinion du framework sans trop vous impliquer. Mais cela reste une découverte très rapide du framework.

Le Getting Started : vous le trouvez en cliquant sur l'onglet **Documentation**. Avec ses six chapitres, il vous permet d'approfondir votre connaissance du framework pour découvrir l'essentiel du développement d'applications web.

Le Symfony Components : vous le trouvez en cliquant sur l'onglet **Documentation**. Symfony est un framework composé de différents composants indépendants (les components). Il est possible d'installer un seul composant, ponctuellement, dans un script PHP, sans installer la totalité du framework. Cette documentation vous montre comment installer chaque composant de manière isolée et comment l'utiliser.

Le Symfony Reference : vous le trouvez en cliquant sur l'onglet **Documentation**. C'est une documentation beaucoup plus technique et poussée. Elle vous donnera des informations détaillées sur la configuration du framework et des détails sur certains points, comme les formulaires, les services…Elle s'adresse à des développeurs qui maîtrisent déjà le framework.

Le Symfony Framework Best Practices : vous le trouvez en cliquant sur l'onglet **Documentation**. Comme son nom l'indique, ce sont les bonnes pratiques d'utilisation du framework. C'est intéressant de connaître les bases du bon développement du framework telles que les ont imaginées les créateurs (en l'occurrence, Fabrice Potentier, créateur de Symfony).

Le Symfony CMF : vous le trouvez en cliquant sur l'onglet **Documentation** en bas de la page.

Symfony CMF est un CMS conçu par la communauté à partir de Symfony. Un CMS est un gestionnaire de contenu utilisable en mode graphique sans avoir à développer du code (comme WordPress, par exemple). C'est intéressant par curiosité, mais inutile si vous voulez développer votre propre application Symfony.