

Pourquoi choisir Symfony ?



Sommaire

7 idées reçues sur Symfony Page 3

Un framework : Pour qui ?
Pour quoi ? Page 5

Pourquoi choisir Symfony
(plutôt qu'un autre framework) ? Page 8

Les 6 Commandements de
Symfony Page 12

Un développeur PHP est-il un
développeur Symfony
(et inversement) ? Page 15

Contact Page 18

7 idées reçues sur Symfony

(enfin, 6 et demie)

Symfony est un framework Open Source de développement d'applications Web en PHP. Très bien. Mais encore ?!...

1 Symfony ? Jamais entendu parler... ça vient de sortir ?

Difficile de s'y retrouver parmi la foultitude de frameworks présents sur le marché du développement web ! Pourtant Symfony a déjà quelques belles années d'expériences au compteur - la première version est sortie en 2005, la seconde en 2011. Les téléchargements de Symfony ont passé la barre des 500 millions en 2016, le milliard en 2017 et l'horizon des dix milliards des téléchargements se rapproche.

C'est aujourd'hui un environnement stable et reconnu au plan international. En attestent le nombre et la variété de ses références : Spotify, France Télévisions, Auchan, LaFourchette, ezPublish, BBC News, Drupal, Pepsi...

Sans compter la vaste communauté de contributeurs mobilisés autour de ce framework Open Source. Mais nous y reviendrons plus tard...

2 Les frameworks, c'est pas un peu surfait ?

Vous vous demandez « à quoi bon un framework si j'ai une équipe de développeurs PHP très compétente qui peut très bien faire sans ? ». Vous avez raison... mais en partie seulement ! Parce que ce serait quand même dommage de se priver d'un outil permettant de développer mieux et plus vite, quels que soient d'ailleurs les roxxors que comptent votre équipe.

Le principe de base de Symfony ? Gagner du temps en structurant le code et en réutilisant des modules génériques. Autrement dit, Symfony mâche le travail et fournit nativement des modules tout prêts, réutilisables à votre guise et interopérables. Moins de code à écrire, moins de risque d'erreur, plus de souplesse, plus de qualité et surtout plus de productivité. Pas indispensable, certes, mais quand même rudement pratique !

3 Symfony, ça ne vaut que pour un gros projet ! Moi, je veux juste une petite appli qui tourne bien.

En fait, c'est plutôt une question de besoin que de taille du projet ! Le choix d'une solution de développement (framework, CMS, solution e-commerce, etc.) doit être motivé par les besoins identifiés et la manière dont chacune peut y répondre.

Mais ne nous faites pas dire ce que nous n'avons pas dit : Symfony s'adapte aussi très bien à des « mini-projets » ! Tout simplement parce qu'il est flexible à volonté. Comme toutes les briques logicielles qui le composent sont totalement indépendantes, Symfony est totalement configurable. Une sorte de boîte à outils dans laquelle on se sert en fonction de ses besoins.

Symfony peut être donc utilisé pour le développement d'une application dans son intégralité ou bien en mode « Lego », selon les fonctionnalités voulues.

4 Le jour où on veut faire évoluer l'appli, il faut tout reprendre à zéro.

Et non, justement ! Fle-xi-bi-li-té, on vous dit !

Le fonctionnement en « bundles » (les fameuses briques) permet non seulement d'ajouter des fonctionnalités à la demande, mais aussi de tout changer pour modifier le comportement du framework en fonction de ses besoins. Sans avoir à refaire toute la configuration – grâce notamment à un système de contrats d'interfaces entre les briques.

Et pas question non plus de vous enfermer dans Symfony (on n'est pas comme ça) ! C'est pourquoi Symfony respecte les conventions du PHP : PHPUnit, convention de nommage des classes, etc. Ce qui le rend nativement interopérable avec les standards du marché.

Soit dit en passant, Symfony est tellement interopérable qu'il utilise lui-même des briques logicielles externes dans son noyau (Doctrine, Composer...).

5 Apprendre à coder en Symfony, c'est réapprendre à coder tout court !

On ne vas pas se mentir : même pour un développeur PHP, aborder Symfony peut donner l'impression... de ne plus savoir coder ! Mais n'oublions pas que Symfony reste du PHP. Il ne s'agit donc pas de réapprendre à coder, mais plutôt d'appréhender le code d'une nouvelle manière. Et même si les vieux réflexes ont la vie dure, l'effort s'avère rapidement payant.

Car l'effet directif, contraignant et structurant de Symfony permet finalement de :

- garantir la stabilité, la maintenabilité et l'évolutivité des applications développées, grâce notamment à l'utilisation systématique de bonnes pratiques de programmation ;
- valoriser le travail des développeurs sur les aspects les plus complexes et les plus stratégiques du projet, en les affranchissant du développement de fonctionnalités génériques.

Tellement bien que, croyez-nous ou pas, les développeurs finissent par y prendre goût ! Il faut dire que tout est fait pour leur simplifier la vie : rapidité, fiabilité, flexibilité, sécurité... Et comme on dit : « A développeur heureux... projet fructueux ! »

6 L'Open Source c'est bien, mais question support... ?

Ah l'Open Source ! La liberté, l'ouverture, le partage !... C'est bien beau, mais le jour où vous avez un souci, qui vient à votre secours ? Ne vous inquiétez pas, avec Symfony, vos équipes ne seront pas seules face à leur écran.

Open Source certes, mais créé et supporté par son créateur : SensioLabs. À travers des services d'assistance technique, de conseil, de formations, etc. les experts de SensioLabs pourront vous accompagner et vous aider à vous en sortir !

Et derrière Symfony, une communauté mondiale de développeurs, d'utilisateurs et d'entreprises du monde entier qui contribuent à faire évoluer le framework en y apportant leur expérience et leur savoir-faire. Et en les partageant : [support Open Source sur Slack](#), [Stack Overflow](#), forums... Vous trouverez toujours la réponse à vos questions.

Et parce que « une ligne de code non documentée est une ligne qui n'existe pas », de nombreux ouvrages sont par ailleurs consacrés à Symfony. Le fondateur du projet Symfony, Fabien Potencier, écrit lui-même régulièrement des ouvrages comme « Symfony 5: The Fast Track ». De quoi vous aider tout au long du développement de votre projet.

6,5 What else ?... Symfony, ça fait le café aussi ?

Hélas, non ! En revanche, rendez-vous chez SensioLabs pour discuter de votre projet, on vous en offre un volontiers !

Un framework, Pour qui ? Pour quoi ?

Quand utiliser un framework ? Pour un site web tentaculaire ? Ou une petite application ?

En fait c'est plutôt une question de besoins que de taille du projet ! Le choix d'une solution de développement (framework, CMS, etc.) doit être d'abord motivé par les besoins identifiés et la manière dont chacune peut y répondre.

Mais le débat fait rage entre les adeptes des frameworks et ceux des CMS, opposant volontiers la modularité et l'interopérabilité des uns à la simplicité et la rapidité de déploiement des autres. Alors, comment faire le bon choix ?

Pour le savoir, petit test en 3 questions.

01 Quels sont vos besoins actuels ?

- Rien de compliqué : des fonctionnalités simples, basiques comme du contenu et quelques formulaires.
- ★ Du sur-mesure ! Des fonctionnalités bien spécifiques pour être conformes à nos règles métiers, et qui mixent plusieurs types de services.
- Euh... Je ne sais pas, ils font quoi les autres ?

02 L'application sera-t-elle amenée à évoluer ?

- Non, pas vraiment. On a a priori fait le tour des services qu'on veut proposer sur cette application.
- ★ Oui, certainement. On a plusieurs projets dans le *pipe* qui devront y être intégrés un jour ou l'autre.
- Peut-être... c'est possible. Mais pas forcément. On verra le moment venu.

03 De quelles compétences techniques disposez-vous ?

- Mon assistante maîtrise plutôt bien Dreamweaver pour la mise à jour des actualités sur notre site et la gestion de notre newsletter mensuelle.
- ★ J'ai une bonne équipe de développeurs, avec de solides compétences en programmation web.
- Des compétences techniques ? Rapport à quoi ?

Votre profil :

★ Vous êtes « framework-oriented »

Besoins spécifiques, règles métiers particulières, nécessité de mixer des briques fonctionnelles (contenu + e-commerce, par exemple), évolutions prévues : pour vous, un framework s'impose. Les briques qui le constituent, parce qu'elles sont indépendantes et interoperables, permettent de mettre en œuvre une application sur-mesure, évolutive et maintenable sur le long terme. Et un développeur web n'aura aucune difficulté à s'approprier l'outil. Cela lui permettra même de fournir un code plus structuré, plus propre et conforme aux standards de développement.

● Vous êtes « CMS-oriented »

Généralement rapide à prendre en main et à déployer, ne nécessitant pas de compétences de développement particulières tout en offrant un certain degré de souplesse, les CMS – ou toute autre solution packagée (CRM, solutions e-commerce, etc.) – semblent tout à fait adaptés à vos besoins. Alors si leurs fonctionnalités natives suffisent à satisfaire vos attentes immédiates et à venir, n'hésitez pas, foncez !

■ Vous êtes « Not so sure-oriented »

Vos besoins ne semblent pas encore bien définis. Prenez le temps de peaufiner votre projet, en spécifiant le plus précisément possible vos besoins actuels et à venir. C'est la première étape permettant de vous orienter vers la solution de développement la plus adaptée à votre projet.



Résumons : un framework est la meilleure solution pour...

- ✓ Répondre à des besoins spécifiques, mettre en œuvre et mixer des fonctionnalités complexes ;
- ✓ Disposer d'une application sur-mesure, évolutive, maintenable et performante sur le long terme ;
- ✓ Faire monter ses équipes de développement en compétences, en les familiarisant avec un outil de programmation structurant.

Pour être sûr de faire le bon choix, contactez-nous !

sales@sensiolabs.com | +33(0)1 86 65 72 00
Plus d'infos sur <https://sensiolabs.com/fr/>

Idée reçue n°23

« Les frameworks, c'est pas un peu surfait ? »

Vous vous demandez « à quoi bon un framework si j'ai une équipe de développeurs PHP très compétente qui peut très bien faire sans ? ». Vous avez raison... mais en partie seulement ! Parce que ce serait quand même dommage de se priver d'un outil permettant de développer mieux et plus vite, quels que soient d'ailleurs les experts que comptent votre équipe.

Le principe de base d'un framework ? Gagner du temps en structurant le code et en réutilisant des modules génériques. Une sorte de boîte à outils dans laquelle on se sert en fonction de ses besoins. Moins de code à écrire, moins de risque d'erreur, plus de souplesse, plus de qualité, et surtout plus de productivité. Pas indispensable, certes, mais quand même rudement pratique !

De plus, utiliser un framework standard (par opposition à « fait maison ») permet d'apporter à une application la structuration standard nécessaire pour offrir à n'importe quel développeur, qu'il ait participé ou non à son élaboration, la capacité de se « l'approprier » facilement.

Pourquoi choisir Symfony ?

(plutôt qu'un autre framework)

Vous avez choisi d'utiliser un framework pour le développement de votre site ou de votre application. Reste à déterminer lequel : difficile de s'y retrouver parmi la foultitude de frameworks présents sur le marché du développement web.

Voici 8 critères à prendre en compte pour bien choisir son framework... et autant de raisons de préférer Symfony à un autre !

1. Sa notoriété et la taille de sa communauté

Si la popularité ne fait pas tout en matière d'outil de développement, c'est tout de même souvent un gage de qualité et de performance. De même que la taille du fan club en révèle généralement le potentiel d'innovation.

Check-list Symfony

- ✓ Une communauté mondiale de plusieurs milliers de contributeurs (intégrateurs, développeurs Symfony mais aussi issus des communautés eZPublish ou Drupal) qui participent activement à la croissance de Symfony (documentation, création de bundles, optimisation du code, support, etc.)
- ✓ De nombreux outils Open Source qui gravitent autour du projet Symfony et en utilisent certains composants (Doctrine, Monolog, Composer...)
- ✓ Une capacité d'adaptation et d'évolution constante
- ✓ L'un des frameworks leaders pour les développements PHP

2. Ses références

Quelle meilleure preuve de la notoriété d'une solution que le nombre, la diversité et la qualité de ses références ?

Check-list Symfony

- ✓ Des millions de sites et d'applications développés à travers le monde
- ✓ Intranets, sites grand public, réseaux sociaux, sites communautaires, applications de gestion et de workflow...
- ✓ Des références prestigieuses parmi lesquelles [Spotify](#), [France Télévisions](#), [Auchan](#), [LaFourchette](#), [ezPublish](#), [BBC News](#), [Drupal](#), [Pepsi](#), [American Express](#), [New York Times](#), [M6Web](#), [TF1](#)...

3. Sa philosophie

Derrière l'origine d'un projet se cache avant tout la réponse à un besoin initial. Ici, celui de capitaliser sur les connaissances et les outils à long terme afin de concentrer ses efforts sur les vrais enjeux métiers sans avoir à réinventer la roue à chaque nouveau projet.

Check-list Symfony

- ✓ Un outil d'abord imaginé pour les propres besoins de son créateur, SensioLabs, qui continue à l'utiliser au quotidien pour le développement des projets de ses clients
- ✓ Un outil développé par des professionnels pour des professionnels, pour répondre de manière pragmatique à un besoin concret (CQFD !)

4. Sa licence d'exploitation

Bien appréhender la notion de licence peut vous éviter quelques mauvaises surprises ! Elle peut en effet avoir un impact non négligeable sur votre application : conditions d'exécution et d'usage, droit de modification du code source, etc.

Check-list Symfony

- ✓ Distribué sous licence Open Source MIT
- ✓ Modification autorisée de tout ou partie du code source
- ✓ Aucune obligation de viralité, autrement dit de redistribution du code à l'Open Source
- ✓ Usage autorisé dans des applications commerciales propriétaires

5. Sa stabilité et sa pérennité

Choisir une solution immédiatement opérationnelle, c'est bien. Qui tienne la route sur le long terme, c'est mieux ! Un framework doit garantir la longévité de votre application, en termes de maintenance comme d'évolutions.

Check-list Symfony

- ✓ Un code structuré respectant les bonnes pratiques de programmation
- ✓ Une interopérabilité de fait avec les standards PHP
- ✓ Une structure modulaire pour faciliter l'ajout, la modification ou la suppression de fonctionnalités
- ✓ Une roadmap clairement établie (2 versions mineures par an, 1 version LTS tous les 2 ans)
- ✓ Une version LTS (Long Term Support) supportée pendant 4 ans (4 ans pour les failles de sécurité et 3 ans pour la correction de bugs)
- ✓ Une compatibilité garantie lors des montées en version

6. Les ressources humaines et techniques nécessaires

Derrière une application web, il doit y avoir des gens (compétents, productifs) et une infrastructure (solide, performante). Et il vaut mieux s'en assurer avant la fin du projet !

Check-list Symfony

- ✓ Rapidité de prise en main de l'outil grâce à une documentation abondante et un support disponible
- ✓ Montée en compétence simplifiée des développeurs, grâce à la promotion systématique des bonnes pratiques de développement et à l'adaptation de concepts éprouvés d'autres technologies (Java, Python, Ruby...)
- ✓ Valorisation du travail des développeurs sur les aspects les plus stratégiques du projet, en les affranchissant du développement de fonctionnalités génériques
- ✓ Augmentation constante du nombre de développeurs Symfony, y compris certifiés
- ✓ Plaisir de développer avec Symfony

7. Le support et la documentation

On ne vas pas se mentir : même pour un développeur aguerri, aborder un nouveau framework peut donner l'impression de ne plus savoir coder et l'exposer à de grands moments de solitude devant son écran...

Savoir s'entourer des bonnes compétences et des savoir-faire adéquats peut lui sauver la mise.

Check-list Symfony

- ✓ Une devise : « Une ligne non documentée est une ligne qui n'existe pas »
- ✓ Une littérature abondante consacrée à Symfony, traduite en plusieurs langues et mise à jour quotidiennement par toute la communauté : des centaines de blogs, des tutoriels pour tous les niveaux, etc.
- ✓ Un support par SensioLabs (conseils, formations, assistance technique...) et par la communauté Symfony ([support Open Source sur Slack](#), [Stack Overflow](#), forums ...)

8. La sécurité

Toute application est potentiellement exposée à des attaques malveillantes, même celle des plus « grands ».

Pour minimiser les risques, il est toujours préférable de choisir un framework capable de prendre en compte et contrer les principales menaces.

Check-list Symfony

- ✓ Une protection « par défaut » où le principe de précaution prévaut (tout est verrouillé... sauf les éléments nécessitant un déblocage)
- ✓ La détection et la correction de failles de sécurité plus rapides que pour un code propriétaire
- ✓ La gestion des vulnérabilités les plus courantes (XSS, CSRF et injection SQL)
- ✓ La mise à disposition de l'outil gratuit Symfony Security Checker pour vérifier si votre application utilise des dépendances présentant des failles de sécurité connues

Lancez-vous !

Téléchargez Symfony dès maintenant sur
<https://symfony.com/>

Ou contactez-nous :
sales@sensiolabs.com
+33(0)1 86 65 72 00



Idée reçue n°11

« Apprendre à coder en Symfony, c'est réapprendre à coder tout court ! »

N'oublions pas que Symfony reste du PHP. Il ne s'agit donc pas de réapprendre à coder, mais plutôt d'appréhender le code d'une nouvelle manière. Et même si les vieux réflexes ont la vie dure, l'effort s'avère rapidement payant.

Car l'effet structurant de Symfony permet finalement de :

- garantir la stabilité, la maintenabilité et l'évolutivité des applications développées, grâce notamment à l'utilisation systématique de bonnes pratiques de programmation ;
- valoriser le travail des développeurs sur les aspects les plus complexes et les plus stratégiques du projet, en les affranchissant du développement de fonctionnalités génériques.

Tellement bien que, croyez-nous ou pas, les développeurs finissent par y prendre goût ! Il faut dire que tout est fait pour leur simplifier la vie : rapidité, fiabilité, flexibilité, sécurité... Et comme on dit : « A développeur heureux... projet fructueux ! »

Les 6 commandements de Symfony

Simple, performant, stable, flexible, évolutif... Plus qu'un simple outil, Symfony c'est avant tout une solution orientée vers les développeurs d'une part, et vers les applications développées d'autre part. Retour sur les 6 commandements à l'origine du framework.

Principe n°1 : Rapidité et performance tu faciliteras

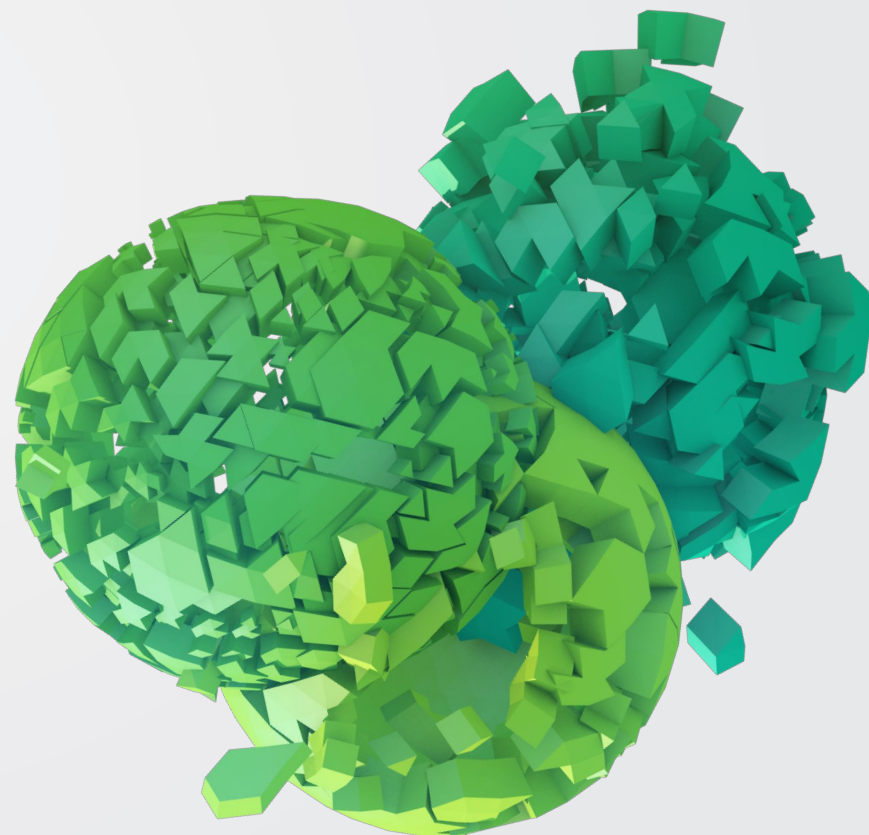
Symfony intègre nativement un certain nombre de mécanismes et d'outils destinés à optimiser les performances d'une application. En fonction du contexte du projet et des performances visées, les développeurs peuvent par exemple s'appuyer sur le Profiler qui collecte des données techniques et des métriques sur chaque requête (quantité de mémoire consommée, temps de génération de la page...). Ou encore sur le conteneur d'injection de dépendances qui permet d'optimiser l'utilisation de la mémoire au niveau du PHP. Par ailleurs, de par son architecture, Symfony est capable de s'adapter aux contraintes de n'importe quel environnement technique.

Principe n°2 : Souplesse et flexibilité tu favoriseras

Ce qui rend Symfony si flexible ? Sa capacité à être totalement configurable, surchargeable et extensible à volonté. Chacune des briques qui le constituent étant indépendante, Symfony permet ainsi de :

- développer, en mode Full Stack, une application complexe dans son intégralité, avec de nombreuses fonctionnalités ;
- construire un framework brique à brique, selon les fonctionnalités dont vous avez besoin ;
- utiliser un *microframework* pour le développement d'une application précise, sans avoir à installer l'ensemble du framework.

Symfony est donc de fait adapté aux développements en méthode agile.



Principe n°3 : D'évoluer tu permettras

Du plus petit composant au noyau même de Symfony, tout est présenté sous forme de bundles (ou plugin). Aujourd'hui, Symfony en compte plus de 2 000, offerts par la communauté pour répondre à des besoins variés : gestion de contenus, gestion des utilisateurs, manipulation d'images, indexation de contenus, interfaçage avec des bases de données NoSQL, intégration simplifiée avec les réseaux sociaux, etc. Résultat : vous pouvez ajouter des fonctionnalités au framework au gré de vos besoins, mais aussi modifier complètement le comportement du framework en agissant sur son noyau sans avoir besoin de tout reconfigurer – grâce notamment à un système de contrats d'interfaces entre les briques. Sans compter que les bundles peuvent être également réutilisés tels quels sur un autre projet. Et depuis l'arrivée de Flex, Symfony permet d'automatiser complètement sa configuration pour l'usage commun grâce aux conventions.

Principe n°4 : Stabilité et pérennité tu garantiras

Que se passe-t-il le jour où une nouvelle version de Symfony est disponible ? C'est simple : vous ne vous retrouvez pas tout seul : chaque version LTS supporte pendant 3 ans les versions majeures de Symfony, et pendant 4 ans les éventuels problèmes de sécurité. Le contrat et l'interface de Symfony bénéficient en outre d'une garantie sur les versions mineures : la compatibilité entre toutes les versions mineures est assurée sur l'API définie par les interfaces publiques. Ce qui permet par ailleurs de garantir la compatibilité avec l'ensemble des *bundles*.

Principe n°5 : Aux développeurs tu penseras

En concevant Symfony d'abord comme son propre outil de travail, SensioLabs a aussi pensé au confort (et à la productivité) de ses développeurs en leur fournissant un environnement de développement particulièrement propice à l'efficacité. D'abord en les affranchissant des tâches rébarbatives (développement de fonctionnalités mineures, par exemple) pour leur permettre de se concentrer sur des aspects plus stratégiques. Ensuite en leur fournissant des outils d'aide au développement, comme la désormais célèbre Web Debug Toolbar qui liste automatiquement un ensemble d'informations particulièrement utiles – requêtes en base de données, temps de réponse, etc. Enfin, en intégrant un support (natif ou via des plugins communautaires) dans les EDI et éditeurs de texte du marché comme PhpStorm ou VSCode.

Principe n°6 : Simple et accessible tu te rendras

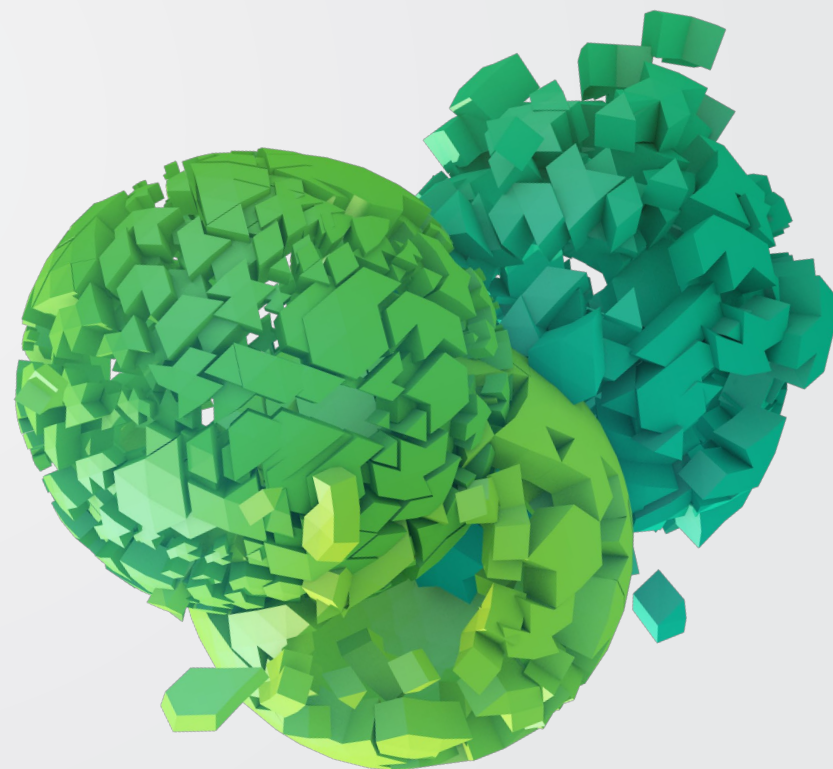
Soutenu par une communauté active, assorti d'une documentation précise et abondante et consolidé par un support professionnel, Symfony est l'un des frameworks les plus accessibles du marché.

Y compris pour les débutants à qui il offre toutes les ressources nécessaires à une prise en main simple et rapide.

Et si nous parlions de votre projet ?

sales@sensiolabs.com
+33(0)1 86 65 72 00

Plus d'infos sur <https://sensiolabs.com/fr/>



Idée reçue n°42

«Le jour où on veut faire évoluer l'appli, il faut tout reprendre à zéro ! »»

Le fonctionnement en « bundles » permet non seulement d'ajouter des fonctionnalités à la demande, mais aussi de tout changer pour modifier le comportement du framework en fonction de ses besoins. Sans avoir à refaire toute la configuration – grâce notamment à un système de contrats d'interfaces entre les briques.

Et pas question non plus de vous enfermer dans Symfony (on n'est pas comme ça) ! C'est pourquoi Symfony respecte les conventions de PHP : PHPUnit, convention de nommage des classes, etc. Ce qui le rend nativement interopérable avec les standards du marché.

Soit dit en passant, Symfony est tellement interopérable qu'il utilise lui-même des briques logicielles externes dans son noyau : Doctrine, Twig, Monolog, Composer...

Un développeur PHP est-il un développeur Symfony ?

(et inversement)



Choisir un framework pour le développement de son projet, c'est aussi se poser la question des ressources humaines qu'il faudra lui consacrer. Or Symfony a beau reposer sur PHP, on entend souvent dire qu'apprendre à coder en Symfony, c'est réapprendre à coder tout court... Le jeu en vaut-il alors vraiment la chandelle ?

Si l'on en croit les principaux concernés, l'effort – s'il en est – s'avère rapidement payant. Pour preuve, observons le comportement d'un développeur Symfony dans son environnement naturel...

Il n'est jamais seul face à son écran

Ne vous fiez pas à son apparente solitude dans l'adversité ! Les ressources disponibles autour de Symfony (documentation abondante, support par SensioLabs et par une communauté mondiale de plusieurs milliers de contributeurs sur laquelle s'appuyer) permettent au développeur, même débutant, de toujours trouver la réponse à ses questions. En ce sens, Symfony n'est finalement pas si compliqué que ça à prendre en main...

Il est à la fois rapide, efficace et serein

Symfony a été conçu comme une boîte à outils, composée de briques logicielles indépendantes, configurables, flexibles et réutilisables. Moins de code à écrire, moins de tâches rébarbatives à prendre en charge et surtout moins de risque d'erreurs : le développeur gagne non seulement en productivité mais aussi en sérénité. Sans parler des bonnes pratiques « embarquées » dans le framework qui sont nativement appliquées sans avoir besoin de les connaître ni de les comprendre... On constate d'ailleurs que le développeur Symfony réduit, de fait, sa dépendance à la caféine.

Il est soucieux de ses congénères

Une ancienne application à maintenir ou à faire évoluer ? Un développeur connaît trop bien le piège du code legacy, qui entraîne sa meute dans les affres d'un décorticage de code chronophage et contre-productif. Un développeur Symfony sait comment apporter à une application la structuration nécessaire pour éviter cet écueil et offrir à n'importe quel développeur, qu'il ait participé ou non à son élaboration, la capacité de se « l'approprier » facilement. Et ainsi de la maintenir dans le temps et de la faire évoluer rapidement.

Il ne renie pas ses racines

Un développeur initié à Symfony n'en oublie pas pour autant les pratiques élémentaires de PHP, bien au contraire. D'une part, parce que PHP, c'est comme le vélo, ça ne s'oublie pas. D'autre part, parce que Symfony reste du PHP et un langage de programmation 100% orienté objet. Il ne s'agit donc pas de réapprendre à coder, mais plutôt d'appréhender le code d'une nouvelle manière : structuration du code, utilisation systématique de bonnes pratiques de développement... Il est donc non seulement capable de développer en PHP, mais en plus de manière plus efficace et plus propre.

Il valorise sa horde

Un développeur Symfony, pour peu qu'il soit certifié comme tel, est un atout majeur dans une équipe de développement. La Certification Symfony, en validant ses compétences opérationnelles sur le framework, permet au développeur de gagner en légitimité et en crédibilité auprès de ses potentiels recruteurs. Et pour les recruteurs, c'est la garantie non seulement d'avoir dans leur équipe un profil expert mais aussi de valoriser des savoir-faire et des qualifications reconnus auprès de leurs propres clients et prospects.

Il est heureux

D'abord parce que Symfony, en l'affranchissant du développement de fonctionnalités mineures, lui permet de se concentrer sur les vrais enjeux du projet et de valoriser pleinement son travail. Ensuite, parce que Symfony lui facilite considérablement la vie : la légendaire Web Debug Toolbar, le support natif des environnements de développement, les pages d'erreurs détaillées ou encore la sécurité prise en charge nativement. Enfin parce que Symfony, en promouvant un ensemble de bonnes pratiques (tests, documentation, automatisation), favorise le bon déroulement d'un projet. Autrement dit, un confort de développement inégalé qui profite à toutes les parties prenantes du projet.

Formations, support, ressources : découvrez nos services pour bien démarrer avec Symfony.

sales@sensiolabs.com | +33(0)1 86 65 72 00 | Plus d'infos sur <https://sensiolabs.com/fr/>



Idée reçue n°7

« L'Open Source c'est bien, mais question support... ? »

Ah l'Open Source ! La liberté, l'ouverture, le partage !... C'est bien beau, mais le jour où vous avez un souci, qui vient à votre secours ? Ne vous inquiétez pas, avec Symfony, vos équipes ne seront pas seules face à leur écran.

Open Source certes, mais créé et supporté par son créateur : SensioLabs. A travers des services d'assistance technique, de conseils, de formations, etc., les experts de SensioLabs pourront vous accompagner et vous aider à vous en sortir !

Et derrière Symfony, une communauté mondiale de développeurs, intégrateurs, utilisateurs et entreprises du monde entier qui contribuent à faire évoluer le framework en y apportant leur expérience et leur savoir-faire. Et en les partageant : [support Open Source sur Slack](#), [Stack Overflow](#), forums ... Vous trouverez toujours la réponse à vos questions.

Et parce qu' « une ligne de code non documentée est une ligne qui n'existe pas », de nombreux ouvrages sont par ailleurs consacrés à Symfony. Le fondateur du projet Symfony, Fabien Potencier, écrit lui-même régulièrement des ouvrages comme « Symfony 5: The Fast Track ». De quoi vous aider tout au long du développement de votre projet.



Contactez SensioLabs

92-98, Boulevard Victor Hugo
92110 Clichy Cedex · France

+33 1 86 65 72 00
sales@sensiolabs.com

