**1. Introduction**

Le but de ce tutoriel est de vous appendre plusieurs méthodes d'installation de Symfony, et de vous apprendre à résoudre les problèmes souvent rencontrés pendant le processus d'installation.

**2. Installer votre application Symfony**

Symfony utilise une application dédiée appelée **Symfony Installer** pour créer facilement vos applications Symfony. Cet installeur exige que PHP 5.4 ou une version supérieure soit installé sur votre système local. Cette installation se fait comme indiqué dans les sections suivantes, selon votre système d'exploitation.

**2-1. Systèmes Linux et Mac OS X**

Il suffit d'ouvrir votre terminal, puis d'exécuter les commandes suivantes :

Sélectionnez

sudo curl -LsS https://symfony.com/installer -o /usr/local/bin/symfony

sudo chmod a+x /usr/local/bin/symfony

Ceci va créer une commande symfony globale dans votre système.

**2-2. Systèmes Windows**

Sous Windows, ouvrez votre console et exécutez les commandes suivantes :

Sélectionnez

php -r "readfile('https://symfony.com/installer');" > symfony

Déplacez ensuite le fichier symfony téléchargé dans un dossier indiqué dans la variable d'environnement PATH ou dans votre dossier de projets :

Sélectionnez

# Par exemple, si WAMP est utilisé...

c:\> move symfony c:\wamp\bin\php

# ... alors, exécutez la commande comme ceci :

c:\> symfony

# Déplacement dans le dossier de vos projets ...

c:\> move symfony c:\projects

# ... alors, exécutez la commande comme ceci

c:\> cd projects

c:\projects\> php symfony

**3. Créer l'application Symfony**

Une fois l'installeur Symfony installé sur votre système local, créez votre première application Symfony avec la nouvelle commande :

Sélectionnez

# Linux, Mac OS X

$ symfony new nom\_de\_mon\_projet

# Windows

c:\> cd projects/

c:\projects\> php symfony new nom\_de\_mon\_projet

Cette commande crée un nouveau dossier nommé nom\_de\_mon\_projet qui contient un tout nouveau projet vide basé sur la version stable la plus récente disponible de Symfony. En plus, l'installeur contrôle si votre système réunit les conditions techniques pour exécuter des applications Symfony. Si ce n'est pas le cas, vous verrez la liste des changements requis.

Si l'installeur ne fonctionne pas chez vous ou n'affiche rien, assurez-vous que l'[extension Phar](http://php.net/manual/fr/intro.phar.php) est installée et activée sur votre ordinateur.

Pour des raisons de sécurité, toutes les versions de Symfony sont signées numériquement avant leur distribution. Si vous voulez vérifier l'intégrité d'une version Symfony, suivez les étapes [expliquées dans ce post](http://fabien.potencier.org/signing-project-releases.html).

**3-1. Baser votre projet sur une version spécifique de Symfony**

Dans le cas où votre projet a besoin d'être basé sur une version spécifique de Symfony, utilisez le second argument optionnel de la commande new :

Sélectionnez

# Utiliser une des versions antérieures de Symfony

$ symfony new my\_project\_name 2.3

$ symfony new my\_project\_name 2.5

$ symfony new my\_project\_name 2.6

$ symfony new my\_project\_name 2.8

$ symfony new my\_project\_name 3.1

# Utiliser une version spécifique d'une version majeure de Symfony

$ symfony new my\_project\_name 2.8.3

$ symfony new my\_project\_name 3.1.5

# Utiliser une version bêta ou une version RC (utile pour tester une nouvelle version de Symfony)

$ symfony new my\_project 2.7.0-BETA1

$ symfony new my\_project 2.7.0-RC1

# Utiliser la version LTS (Long Term Support la plus récente version

$ symfony new my\_project\_name lts

Lisez le [descriptif des releases de Symfony](http://symfony.com/doc/current/contributing/community/releases.html) pour mieux comprendre pourquoi ces différentes versions de Symfony et choisir la meilleure version pour votre projet.

**4. Créer une application Symfony avec l'outil Composer**

Si vous utilisez toujours PHP 5.3, ou que vous ne pouvez pas exécuter l'installeur pour une quelconque raison, vous pouvez créer des applications Symfony en utilisant la méthode d'installation alternative basée sur [Composer](https://getcomposer.org/).

Composer est le manager de dépendances utilisé par les applications PHP modernes et peut aussi être utilisé pour créer de nouvelles applications basées sur le Framework Symfony. Si vous ne l'avez pas, globalement, installé, commencez par lire la section suivante.

**4-1. Installer Composer globalement**

[Composer](https://getcomposer.org/) est le gestionnaire de packages utilisé par les applications PHP modernes. Utilisez celui-ci pour gérer les dépendances dans vos applications Symfony et pour en installer ses composants dans vos projets PHP. Il est recommandé d'installer Composer globalement dans votre système comme expliqué ci-dessous.

**4-1-1. Installer Composer sur Linux ou Mac OS X**

Pour installer Composer sous Linux ou Mac OS X, exécutez les deux commandes suivantes :

Sélectionnez

$ curl -sS https://getcomposer.org/installer | php

$ sudo mv composer.phar /usr/local/bin/composer

Si vous n'avez pas curl installé, vous pouvez télécharger le fichier d'installation manuellement à l'adresse : <https://getcomposer.org/installer>, puis lancez :

Sélectionnez

$ php installer

$ sudo mv composer.phar /usr/local/bin/composer

**4-1-2. Installer Composer sur Windows**

Téléchargez l'installeur depuis [getcomposer.org/download](https://getcomposer.org/download) et exécutez-le puis suivez les instructions.

**4-1-3. En apprendre plus**

Lisez la [documentation Composer](https://getcomposer.org/doc/00-intro.md) pour en apprendre plus sur son usage et ses fonctionnalités.

**4-2. Créer une application Symfony avec Composer**

Une fois que Composer a été installé sur votre ordinateur, exécutez la commande create-project pour créer une nouvelle application Symfony basée sur la dernière version stable :

Sélectionnez

$ composer create-project symfony/framework-standard-edition nom\_de\_mon\_projet

Si vous avez besoin de baser votre application sur une version spécifique de Symfony, fournissez cette version en second argument de la commande  :

Sélectionnez

$ composer create-project symfony/framework-standard-edition nom\_de\_mon\_projet "2.3.\*"

Si votre connexion Internet est lente, vous pourriez penser que Composer ne fait rien du tout. Si c'est votre cas, ajoutez le drapeau -vvv à la commande précédente pour afficher une sortie détaillée de tout ce que fait Composer.

**5. Lancer l'application Symfony**

Symfony exploite le serveur Web interne fourni par PHP (depuis la version de PHP 5.4) pour exécuter des applications pendant que vous les programmez. Par conséquent, exécuter une application Symfony consiste à parcourir le dossier de projet et d'exécuter ces commandes :

Sélectionnez

$ cd my\_project\_name/

$ php bin/console server:run

Ouvrez ensuite votre navigateur et accédez à l'URL http://localhost:8000 pour voir la page de bienvenue de Symfony :

|  |
| --- |
| [Image non disponible](https://symfony.developpez.com/documentation/symfony3/part-03-apprendre-installer-configurer/images/3.welcome.png) |

Au lieu de la page de bienvenue, vous pouvez voir une page vierge ou une page d'erreur. Ceci est dû à un problème de configuration de permission sur un dossier. Il y a plusieurs solutions possibles en fonction de votre système d'exploitation. Elles sont toutes expliquées dans le chapitre : [Configurer les permissions](https://symfony.developpez.com/documentation/symfony3/part-03-apprendre-installer-configurer/#L7).

Le serveur Web interne PHP est disponible à partir de la version 5.4. Si vous utilisez toujours l'ancienne version PHP 5.3, vous aurez à configurer un VirtualHost sur votre serveur Web.

La commande server:run est seulement adaptée au développement d'une application. Afin de lancer une application Symfony sur un serveur en production, vous devez configurer votre serveur Web Apache ou Nginx.

Quand vous avez fini de travailler sur votre application Symfony, vous pouvez stopper le serveur avec la commande :

Sélectionnez

$ php bin/console server:stop

Ou avec la combinaison de touches Ctrl+C depuis le terminal ou la console des commandes.

**6. Contrôler l'installation et la configuration d'une application Symfony**

L'installeur de Symfony vérifie si votre système est prêt à exécuter les applications Symfony. Cependant, la configuration de PHP obtenue de la console des commandes peut être différente de celle issue de la configuration Web du PHP. Pour cette raison, Symfony fournit un outil de vérification visuelle de la configuration. Accédez à l'URL suivante pour vérifier votre configuration :

Sélectionnez

1.

http://localhost:8000/config.php

S'il y a des problèmes, corrigez-les maintenant avant de passer à la suite.

**7. Configurer ou régler les problèmes de permissions sur les fichiers**

Un des importants prérequis à l'installation de Symfony est que le dossier var doit être en mode lecture/écriture pour le serveur Web et la ligne de commande utilisateur.

Sur un système Linux ou Mac OS, si le compte utilisateur utilisé par votre serveur Web est différent de votre utilisateur de ligne de commande, vous devez bien configurer les permissions pour éviter des erreurs. Pour le faire, vous pouvez utiliser l'une des méthodes ci-dessous.

**7-1. Utiliser le même utilisateur pour l'interface ligne de commande et le serveur Web**

Dans des environnements de développement, il est commun d'utiliser le même utilisateur Unix pour la ligne de commande et le serveur web, car il évite les problèmes de permission lors de la mise en place de nouveaux projets. Ceci peut être fait en éditant la configuration de votre serveur Web (ex. : communément httpd.conf ou apache2.conf pour Apache) et la régler pour être la même que votre interface de ligne de commande (ex. : pour Apache, mettez à jour la valeur User et Group).

Si cette solution est utilisée dans un environnement de production, assurez-vous que ce compte utilisateur a les droits limités à ce qu'il est censé faire (pas d'accès aux données privées ou pas de droits d'exécution de binaires non vérifiés).

**7-2. Utiliser les ACL sur un système supportant chmod +a (MacOS)**

De nombreux systèmes (comme Mac OS) vous permettent d'utiliser la commande chmod +a pour définir un ACL. Utilisez le script ci-dessous pour déterminer le compte utilisateur du serveur Web et lui attribuer les permissions nécessaires:

Sélectionnez

$ rm -rf var/cache/\*

$ rm -rf var/logs/\*

$ HTTPDUSER=`ps axo user,comm | grep -E '[a]pache|[h]ttpd|[\_]www|[w]ww-data|[n]ginx' | grep -v root | head -1 | cut -d\ -f1`

$ sudo chmod -R +a "$HTTPDUSER allow delete,write,append,file\_inherit,directory\_inherit" var

$ sudo chmod -R +a "`whoami` allow delete,write,append,file\_inherit,directory\_inherit" var

**7-3. Utiliser les ACL sur un système qui supporte setfacl (Linux/BSD)**

Plusieurs systèmes Linux et BSD ne supportent pas chmod +a, mais supportent un autre utilitaire nommé setfacl. Vous devrez [activer le support ACL](https://doc.ubuntu-fr.org/acl) sur votre partition et installer setfacl avant de l'utiliser (comme dans le cas d'Ubuntu). Ceci utilise une commande pour essayer de déterminer votre utilisateur serveur Web et l'utiliser avec HTTPDUSER :

Sélectionnez

$ HTTPDUSER=`ps axo user,comm | grep -E '[a]pache|[h]ttpd|[\_]www|[w]ww-data|[n]ginx' | grep -v root | head -1 | cut -d\ -f1`

# if this doesn't work, try adding `-n` option

$ sudo setfacl -R -m u:"$HTTPDUSER":rwX -m u:`whoami`:rwX var

$ sudo setfacl -dR -m u:"$HTTPDUSER":rwX -m u:`whoami`:rwX var

Si ça ne fonctionne pas, essayez d'ajouter l'option -n.

setfacl n'est pas disponible sur les points de montage NFS. Cependant, stocker le cache et les logs par NFS est fortement déconseillé pour des raisons de performance.

**7-4. Sans utiliser les ACL**

Si aucune des précédentes méthodes ne fonctionne pour vous, changez le umask de façon à ce que les dossiers cache et log aient les droits d'écriture pour le groupe et les « autres » (selon que l'utilisateur du serveur Web et celui de la ligne de commande sont dans le même groupe ou non). Pour accomplir ceci, mettez la ligne suivante au début des fichiers bin/console, web/app.php et web/app\_dev.php :

Sélectionnez

umask(0002); // Ceci mettra les permissions à 0775

// ou

umask(0000); // Ceci mettra les permissions à 0777

Notez que l'utilisation des ACL est recommandée quand vous y avez accès sur votre serveur, car changer l'umask n'est pas thread-safe.

**8. Mise à jour des applications Symfony**

À ce point , vous avez créé une application Symfony complètement fonctionnelle dans laquelle vous allez développer votre propre projet.

Chaque application Symfony dépend d'un certain nombre de bibliothèques tierces. Celles-ci sont téléchargées et stockées dans le dossier vendor/ et sont gérées exclusivement par Composer.

Mettre à jour ces bibliothèques tierces fréquemment est une bonne pratique pour éviter les erreurs et les vulnérabilités de sécurité. Exécutez la commande update de Composer pour toutes les mettre à jour en une seule fois (ce processus peut prendre plusieurs minutes selon la complexité de votre projet) :

Sélectionnez

$ cd mon\_nom\_de\_projet/

$ composer update

Symfony fournit une commande pour vérifier que vos dépendances de projet ne contiennent pas de vulnérabilités de sécurité connues.

Sélectionnez

$ php bin/console security:check

Exécuter cette commande régulièrement est une bonne pratique de sécurité pour pouvoir mettre à jour ou remplacer des dépendances compromises aussi vite que possible.

**9. Installer l'application Demo Symfony ou d'autres distributions**

Vous avez déjà téléchargé [l'édition standard de Symfony](https://github.com/symfony/symfony-standard) : le projet de démarrage, par défaut, de toutes les applications Symfony. Vous allez travailler sur ce projet pendant que vous parcourez cette documentation pour construire votre application !

Symfony fournit également d'autres projets et des squelettes que vous pouvez utiliser pour démarrer.

**9-1. L'application Demo de Symfony**

L'application Demo de Symfony est une application entièrement fonctionnelle qui présente la procédure recommandée pour développer des applications Symfony.

L'application a été conçue comme un outil d'apprentissage pour les débutants et ses codes source contiennent des tonnes de commentaires et de notes pouvant servir.

Afin de télécharger l'application Demo Symfony, exécutez la commande demo de l'installeur Symfony n'importe où dans votre système :

Sélectionnez

# Linux, Mac OS X

$ symfony demo

# Windows

c:\projects\> php symfony demo

Une fois téléchargé, entrez dans le dossier symfony\_demo/ et lancez le serveur Web interne de PHP en exécutant la commande bin/console server:run.

Accédez à l'URL http://localhost:8000 dans votre navigateur pour commencer à utiliser l'application.

**9-2. Installer une distribution Symfony**

Les packages de projets Symfony sont des « distributions », lesquelles sont des applications entièrement fonctionnelles qui incluent les bibliothèques noyau de Symfony, une sélection de bundles utiles, une structure relativement stable et une configuration par défaut. En fait, quand vous avez créé une application Symfony dans les paragraphes précédents, vous avez en réalité téléchargé la distribution par défaut fournie par Symfony, qui est appelée « Symfony Standard Edition ».

La « Symfony Standard Edition » est de loin la distribution la plus populaire, et aussi le meilleur choix pour les développeurs démarrant avec Symfony. Cependant, la communauté Symfony a publié d'autres distributions populaires que vous pouvez utiliser dans vos applications :

* la « [Symfony CMF Standard Edition](https://github.com/symfony-cmf/symfony-cmf-standard) » est la meilleure distribution pour démarrer avec le projet [Symfony CMF](http://cmf.symfony.com/), qui est un projet rendant plus facile aux développeurs l'ajout de fonctionnalités CMS aux applications construites avec le framework ;
* la « [Symfony REST Edition](https://github.com/gimler/symfony-rest-edition) » montre comment construire une application qui fournit une API REST complète en utilisant le bundle [FOSRestBundle](https://github.com/FriendsOfSymfony/FOSRestBundle) et plusieurs autres bundles en relation.

**10. Installer une application Symfony existante**

En travaillant en collaboration dans une application Symfony, il est rare de créer une nouvelle application Symfony comme expliqué dans les sections précédentes.

Généralement, quelqu'un d'autre l'a déjà créée et soumise dans un référentiel partagé.

Il est recommandé de ne pas soumettre certains fichiers (parameters.yml) et répertoires (vendor/, cache, logs) au référentiel, donc vous devez exécuter ce qui suit pour installer une application Symfony existante :

Sélectionnez

# Clone le projet pour télécharger son contenu

$ cd projects/

$ git clone ...

# Faites en sorte que Composer installe les dépendances du projet dans le répertoire vendor/

$ cd my\_project\_name/

$ composer install

# Maintenant Composer va vous demander les valeurs pour les paramètres qui ne sont pas définis

$ ...

**11. Utiliser le contrôle de source**

Si vous utilisez un gestionnaire de versions tel que [Git](http://git-scm.com/), vous pouvez de façon sûre effectuer un « commit » de tous vos codes de projet. Les applications Symfony contiennent déjà un fichier .gitignore spécialement préparé pour Symfony.

Il existe d'autres instructions spécifiques à la façon de configurer votre projet pour être stocké dans Git.

**11-1. Vérification d'une application Symfony versionnée**

En utilisant Composer pour gérer vos dépendances d'application, il est recommandé d'ignorer entièrement le dossier Vendor/ avant d'effectuer un « commit » dans le dépôt. Ceci signifie que si vous vérifiez une application Symfony depuis un dépôt Git, il n'y aura pas de dossier Vendor/ et l'application ne fonctionnera pas hors du dépôt.

De façon à la faire fonctionner, vérifiez votre application Symfony, puis exécutez la commande Composer install de façon à télécharger toutes les dépendances requises :

Sélectionnez

$ cd mon\_nom\_de\_projet/

$ composer install

Comment Composer sait quelles dépendances spécifiques il doit installer ? Quand une application Symfony est chargée dans le dépôt, les fichiers composer.json et composer.lock sont aussi chargés. Ce sont ces fichiers qui informent Composer des dépendances à installer (avec leur version spécifique).

**12. Commencer le développement**

Maintenant que vous avez une application Symfony entièrement fonctionnelle, vous pouvez commencer le développement. Votre distribution doit contenir du code exemple. Consultez le fichier README.md inclus avec la distribution (ouvrez-le comme un fichier texte) pour en apprendre plus sur le code exemple inclus avec votre distribution.