

Symfony

Le présent document aborde l'apprentissage de base du framework Symfony4.

L'[aide en ligne sur ce framework](#), excellente, est la référence incontournable et à jour. Un outil de traduction instantanée de cette aide peut faciliter la compréhension des concepts et techniques.

Installation et configuration de Symfony 4

Avant toutes choses, il faut savoir que vous la possibilité d'utiliser 2 versions de Symfony : - La LTS (Long Term Support, v4.4.7) - La LS (Latest Stable, v5.0.7)

Le choix d'une version ou d'une autre est une histoire de point de vue : - Symfony 5.0, étant la dernière sortie, contient toutes les dernières nouveautés du framework. En contrepartie, elle nécessite des mises à jour fréquentes dû à sa courte durée de maintenance (environ 6 mois). Toutefois, le passage à une version plus récente est facilité. - Symfony 4.4.7 est la dernière LTS et bénéficie d'un support à tout point de vue jusqu'en novembre 2023. Cela permet d'avoir un code stable sans avoir à changer la structure du projet sur une longue période. En contrepartie, vous n'avez pas accès aux dernières nouveautés du framework, et la migration vers la prochaine LTS pourrait être laborieuse.

Plus de précision sur le processus de versions de Symfony [ici](#).

Création d'une application Symfony minimale

Une fois la version de Symfony choisie selon les besoins, nous pouvons initialiser notre projet Symfony. Pour créer une nouvelle application Symfony, s'assurer que les étapes précédentes ont été réalisées. Si vous utilisez PHP directement :

- Créez un dossier (où vous voulez, on l'appellera symfony par exemple) qui contiendra votre/vos projet(s) Symfony.
- Dans ce dossier, ouvrez gitbash et lancez la commande :

```
composer create-project symfony/website-skeleton monProjet v4.4.*
```

Cette commande va créer un nouveau répertoire monProjet, y télécharger des dépendances et même générer les répertoires et fichiers de base nécessaires pour commencer.

Si on ne précise pas la version (ici v4.4.*), Composer prendra automatiquement la dernière version de Symfony.

```
C:\WINDOWS\system32\cmd.exe

X:\symfony>composer create-project symfony/website-skeleton etape0 v4.2.*
Installing symfony/website-skeleton (v4.2.99)
- Installing symfony/website-skeleton (v4.2.99): Loading from cache
Created project in etape0
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
- Installing symfony/flex (v1.4.8): Loading from cache
Symfony operations: 1 recipe (8aa7f4522eb0c9ae06988872bf7c0a89)
- Configuring symfony/flex (>=1.0): From github.com/symfony/recipes:master
Loading composer repositories with package information
Updating dependencies (including require-dev)
Restricting packages listed in "symfony/symfony" to "4.2.*"

- Prefetching 82 packages
- Downloading (100%)

Package operations: 97 installs, 0 updates, 0 removals
- Installing ocramus/package-versions (1.5.1): Loading from cache
- Installing doctrine/lexer (1.2.0): Loading from cache
- Installing doctrine/annotations (v1.8.0): Loading from cache
- Installing doctrine/reflection (v1.0.0): Loading from cache
- Installing doctrine/event-manager (1.1.0): Loading from cache
- Installing doctrine/collections (1.6.4): Loading from cache
- Installing doctrine/cache (1.9.1): Loading from cache
- Installing doctrine/persistence (1.2.0): Loading from cache
- Installing psr/log (1.1.2): Loading from cache
- Installing symfony/debug (v4.2.12): Loading from cache
- Installing symfony/polyfill-mbstring (v1.12.0): Loading from cache
- Installing symfony/http-foundation (v4.2.12): Loading from cache
- Installing psr/container (1.0.0): Loading from cache
- Installing psr/cache (1.0.1): Loading from cache
- Installing symfony/contracts (v1.1.8): Loading from cache
- Installing symfony/event-dispatcher (v4.2.12): Loading from cache
- Installing symfony/http-kernel (v4.2.12): Loading from cache
- Installing symfony/dependency-injection (v4.2.12): Loading from cache
- Installing symfony/routing (v4.2.12): Loading from cache
- Installing symfony/finder (v4.2.12): Loading from cache
- Installing symfony/filesystem (v4.2.12): Loading from cache
- Installing symfony/config (v4.2.12): Loading from cache
- Installing symfony/var-exporter (v4.2.12): Loading from cache
- Installing psr/simple-cache (1.0.1): Loading from cache
- Installing symfony/cache (v4.2.12): Loading from cache
- Installing symfony/framework-bundle (v4.2.12): Loading from cache
- Installing sensio/framework-extra-bundle (v5.3.1): Loading from cache
- Installing symfony/asset (v4.2.12): Loading from cache
- Installing symfony/dotenv (v4.2.12): Loading from cache
- Installing symfony/expression-language (v4.2.12): Loading from cache
- Installing symfony/inflector (v4.2.12): Loading from cache
- Installing symfony/property-access (v4.2.12): Loading from cache
- Installing symfony/options-resolver (v4.2.12): Loading from cache
- Installing symfony/polyfill-intl-icu (v1.12.0): Loading from cache
- Installing symfony/intl (v4.2.12): Loading from cache
- Installing symfony/form (v4.2.12): Loading from cache
- Installing symfony/stopwatch (v4.2.12): Loading from cache
- Installing symfony/console (v4.2.12): Loading from cache

- Configuring symfony/routing (>=4.2): From github.com/symfony/recipes:master
- Configuring sensio/framework-extra-bundle (>=5.2): From github.com/symfony/recipes:master
- Configuring symfony/console (>=3.3): From github.com/symfony/recipes:master
- Configuring doctrine/doctrine-cache-bundle (>=1.3.5): From auto-generated recipe
- Configuring doctrine/doctrine-bundle (>=1.6): From github.com/symfony/recipes:master
- Configuring doctrine/doctrine-migrations-bundle (>=1.2): From github.com/symfony/recipes:master
- Configuring symfony/security-bundle (>=3.3): From github.com/symfony/recipes:master
- Configuring symfony/swiftmailer-bundle (>=2.5): From github.com/symfony/recipes:master
- Configuring symfony/translation (>=3.3): From github.com/symfony/recipes:master
- Configuring symfony/validator (>=4.1): From github.com/symfony/recipes:master
- Configuring symfony/twig-bundle (>=3.3): From github.com/symfony/recipes:master
- Configuring symfony/web-profiler-bundle (>=3.3): From github.com/symfony/recipes:master
- Configuring easycorp/easy-log-handler (>=1.0): From github.com/symfony/recipes:master
- Configuring symfony/monolog-bundle (>=3.3): From github.com/symfony/recipes:master
- Configuring symfony/debug-bundle (>=4.1): From github.com/symfony/recipes:master
- Configuring symfony/maker-bundle (>=1.0): From github.com/symfony/recipes:master
- Configuring symfony/phpunit-bridge (>=4.3): From github.com/symfony/recipes:master
- Configuring symfony/web-server-bundle (>=3.3): From github.com/symfony/recipes:master
Executing script cache:clear [OK]
Executing script assets:install public [OK]

Some files may have been created or updated to configure your new packages.
Please review, edit and commit them: these files are yours.

Some files may have been created or updated to configure your new packages.
Please review, edit and commit them: these files are yours.

What's next?

* Run your application:
  1. Go to the project directory
  2. Create your code repository with the git init command
  3. Download the Symfony CLI at https://symfony.com/download to install a development web server

* Read the documentation at https://symfony.com/doc

Database Configuration

* Modify your DATABASE_URL config in .env

* Configure the driver (mysql) and
  server_version (5.7) in config/packages/doctrine.yaml

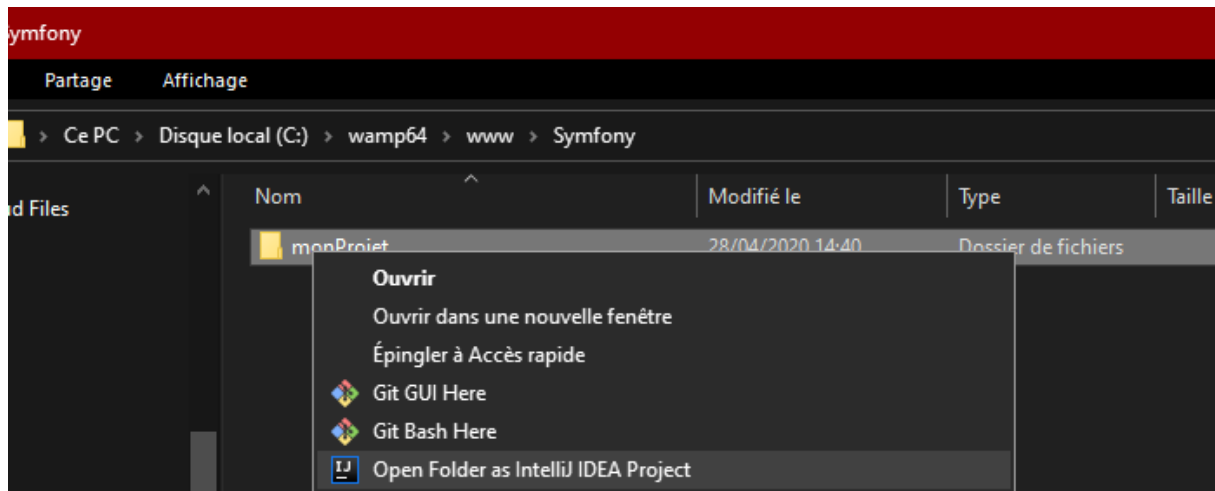
How to test?

* Write test cases in the tests/ folder
* Run php bin/phpunit

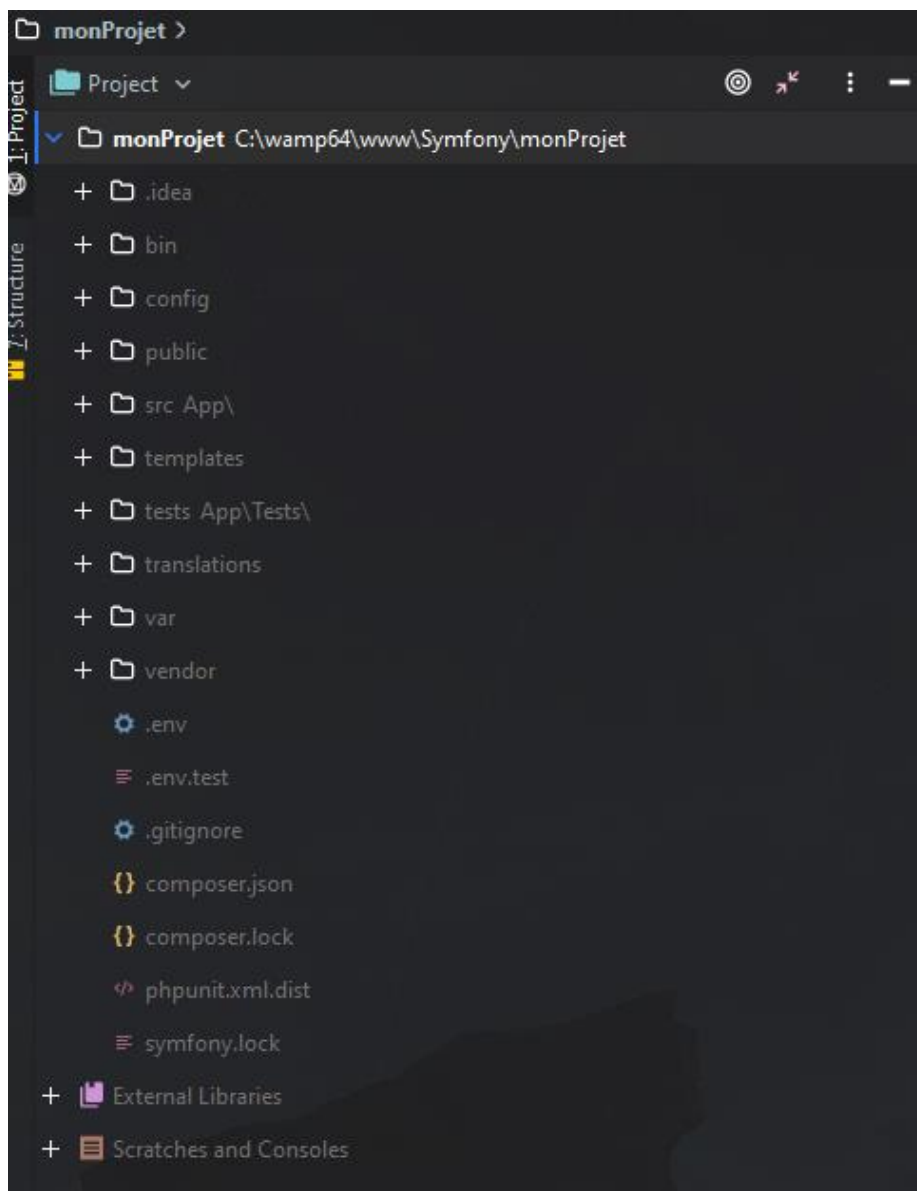
X:\symfony>
```

Remarque : Nous avons créé un projet de type website-skeleton, il est optimisé pour les applications web traditionnelles. En revanche, pour construire des microservices, des applications console ou des API, il est préférable d'utiliser un projet de type skeleton. Il faudra alors importer manuellement toutes les dépendances dont on pourrait avoir besoin.

Cliquez droit sur le dossier nouvellement créé et sélectionnez "Ouvrir avec 'votre IDE' ".



Dans cet IDE, l'arborescence initiale du projet monProjet apparaît comme suit



Exécution d'une application Symfony

En environnement de production, il est courant d'utiliser un serveur Web comme Apache ou Nginx (voir [la configuration d'un serveur Web pour exécuter symfony](#)).

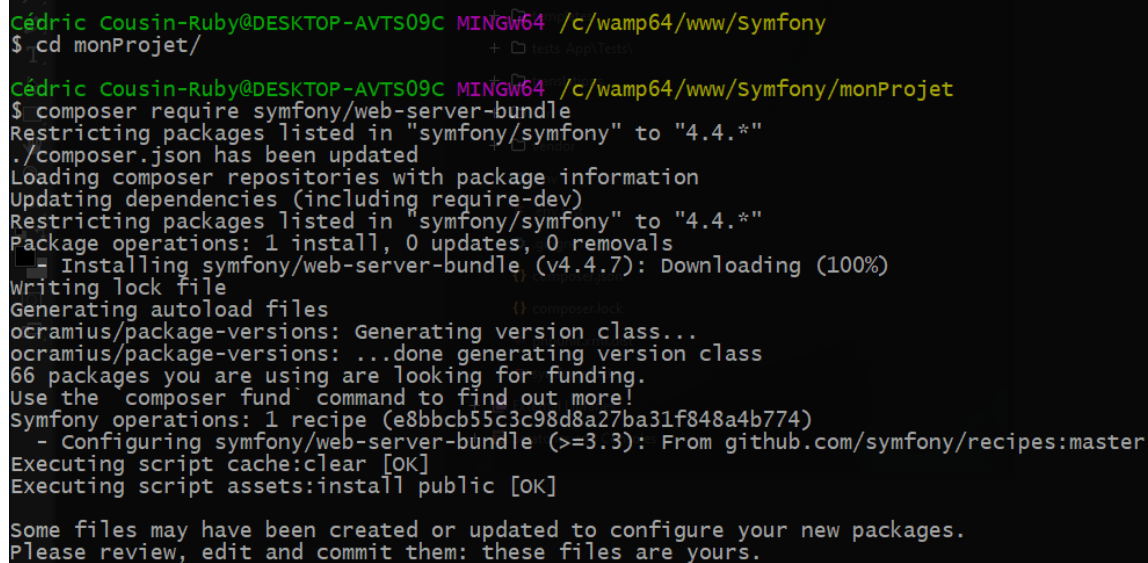
Mais pour le développement, il est possible d'utiliser le serveur web PHP Symfony, ou d'utiliser Wampserver par exemple.

Utilisation du serveur web Symfony

Pour cela, ouvrir un terminal sur votre ide (ou gitbash dans le dossier 'monProjet'), et utilisez les commandes suivantes :

Installation du serveur local :

```
composer require symfony/web-server-bundle
```

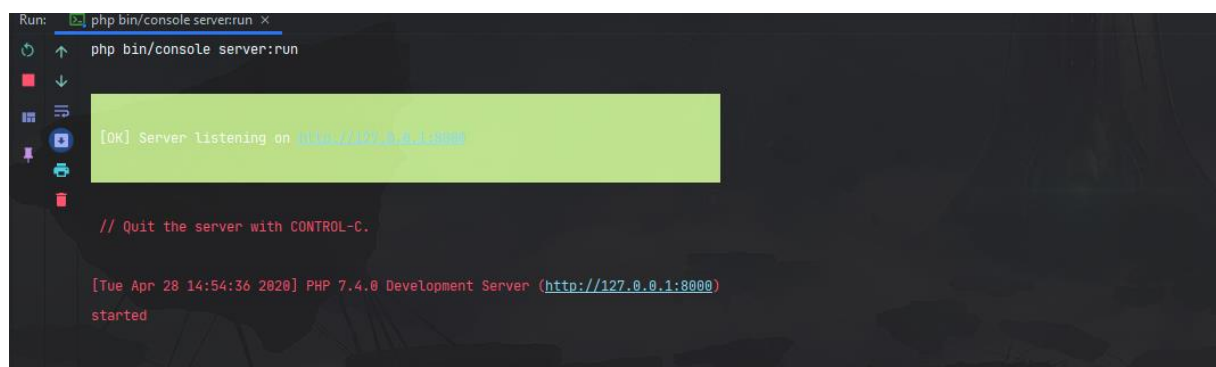


```
Cédric Cousin-Ruby@DESKTOP-AVTS09C MINGW64 /c/wamp64/www/Symfony
$ cd monProjet/
Cédric Cousin-Ruby@DESKTOP-AVTS09C MINGW64 /c/wamp64/www/Symfony/monProjet
$ composer require symfony/web-server-bundle
Restricting packages listed in "symfony/symfony" to "4.4.*"
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Restricting packages listed in "symfony/symfony" to "4.4.*"
Package operations: 1 install, 0 updates, 0 removals
  - Installing symfony/web-server-bundle (v4.4.7): Downloading (100%)
  Writing lock file
  Generating autoload files
  ocrapius/package-versions: Generating version class...
  ocrapius/package-versions: ...done generating version class
  66 packages you are using are looking for funding.
  Use the composer fund command to find out more!
  Symfony operations: 1 recipe (e8bbcb55c3c98d8a27ba31f848a4b774)
    - Configuring symfony/web-server-bundle (>=3.3): From github.com/symfony/recipes:master
  Executing script cache:clear [OK]
  Executing script assets:install public [OK]

Some files may have been created or updated to configure your new packages.
Please review, edit and commit them: these files are yours.
```

Lancement du serveur local :

```
php bin/console server:run
```



```
Run: php bin/console server:run
php bin/console server:run

[OK] Server listening on http://127.0.0.1:8000

// Quit the server with CONTROL-C.

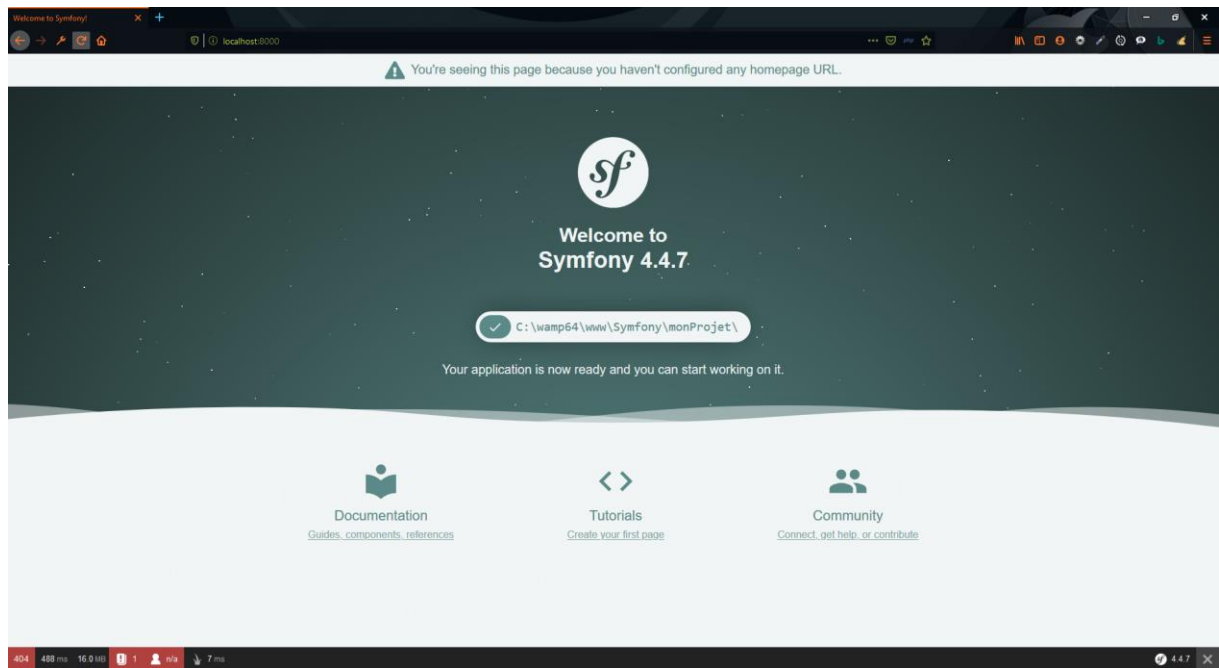
[Tue Apr 28 14:54:36 2020] PHP 7.4.0 Development Server (http://127.0.0.1:8000)
started
```

Depuis la version 4.4, il est conseillé d'installer l'utilitaire symfony CLI que vous pouvez télécharger à cette adresse <https://symfony.com/download>.

Une fois l'installation terminée, vous pouvez lancer le serveur de développement en lançant la commande suivante:

```
symfony server:start
```

Ouvrir un navigateur et accéder à <http://localhost:8000/>. Si tout fonctionne, une page d'accueil s'affiche et confirme que l'environnement de développement est opérationnel.



Pour arrêter le processus du serveur, appuyer sur **Ctrl C** à partir du terminal.

Lancement du projet sur un serveur local

Sur un serveur local, l'installation d'un projet est identique, sauf qu'elle se fera dans le répertoire 'www' de wamp. Ainsi, pour lancer le projet dans notre cas, il suffit juste de démarrer le serveur de manière classique, de se rendre sur votre navigateur et de taper dans la barre d'URL : **localhost/Symfony/monProjet/public/index.php**

Toutes les urls du projet partiront de celle-ci.