CCI Nîmes

Travaux pratique 

Mise en pratique et réalisation PHP avec Symfony



Réalisé et enseigné par Jérôme Ferratier



**Table des matières**

[1 Introduction 1-3](#_Toc150440916)

[2 Pré requis 2-5](#_Toc150440917)

[3 Symfony – Installation et initialisation 3-7](#_Toc150440918)

[3.1 Création d’un nouveau projet en dernière version minimal 3-7](#_Toc150440919)

[3.2 Création d’un nouveau projet en dernière version minimal LTS. 3-7](#_Toc150440920)

[3.3 Création d’un nouveau projet dans une version spécifique minimal. 3-7](#_Toc150440921)

[3.4 Information projet 3-8](#_Toc150440922)

[3.5 Lancer le serveur WEB Symfony 3-9](#_Toc150440923)

[3.6 Arrêter le serveur WEB Symfony 3-9](#_Toc150440924)

[4 Symfony – Configuration d’un projet 4-10](#_Toc150440925)

[4.1 Vérification des pré requis 4-10](#_Toc150440926)

[4.2 Configuration de l’environnement de développement local 4-11](#_Toc150440927)

[5 Création d’une base de données 5-12](#_Toc150440928)

# Introduction

Le développement de sites ou d’applications web est une tâche longue et complexe. L’utilisation d’un framework, comme **Symfony**, a pour objectif d’aider les développeurs à construire ce type de projet plus qualitativement et rapidement.

Environ 80% des sites web dans le monde sont créés avec le langage de programmation PHP. De nombreux frameworks PHP existent : Lavarel, Codelgniter FuelPHP, etc. Et parmi ces frameworks, Symfony est très apprécié par les développeurs pour ses performances et sa simplicité d’utilisation. L’éditeur de ce Framework est Français !

Conçu pour favoriser la modularité, la réutilisation du code et le respect des bonnes pratiques, Symfony simplifie le processus de création d'applications robustes et évolutives.

Son architecture modulaire permet aux développeurs de construire des projets personnalisés tout en profitant de composants prêts à l'emploi.

Cela, en intégrant des fonctionnalités avancées telles que la gestion des formulaires, la sécurité, et l'ORM (Object-Relational Mapping).

Une image contenant texte, capture d’écran, diagramme, conception

Description générée automatiquement

Ce framework PHP est très apprécié en raison de ses deux avantages technologiques : les bundles et les composants.

Le Bundle est similaire à un plugin. Le principal avantage des bundles est qu'ils sont découplés, ce qui implique qu'ils peuvent être réutilisés et reconfigurés davantage pour de nombreuses applications afin de réduire le coût global de développement.

Les composants aident à réduire les tâches de routine car ils peuvent être utilisés indépendamment en ajoutant vos modules personnalisés sans nuire à l'architecture. Au total, 30 composants Symfony utiles facilitent le [processus de développement Web](https://www.netsolutions.com/insights/web-application-development-guide/) . Les composants du framework Symfony peuvent également être utilisés exclusivement dans d'autres frameworks (par exemple Laravel) ou de simples solutions PHP.

Une image contenant texte, capture d’écran, Police, diagramme

Description générée automatiquement

Une image contenant texte, capture d’écran, diagramme, conception

Description générée automatiquement

Les bundles et les composants aident à éliminer les dépendances rigides de l'architecture en vous donnant la liberté d'être réutilisés. Moins il y a de dépendances, plus il devient facile d'introduire des modifications sans risquer de casser d'autres parties du système. Ainsi, vous pouvez adapter la solution à toutes les exigences et scénarios utilisateur pour créer une application très flexible.

# Pré requis

**Connaissance :**

PHP

***Outils :***

VSCode

***Gestionnaire de décompression : 7zip***

<https://www.7-zip.org/download.html>

***Gestionnaire de dépendance : Composer***

<https://getcomposer.org/>

composer –version

***Gestionnaire de source : GIT  :***

<https://git-scm.com/download/win>

git -v

***Symfony CLI (commande line interface) :***

Recommandé par l’éditeur afin de lancer les différentes commandes d’initialisation et configuration de Symfony

<https://symfony.com/download>

symfony -v

**Remarque :**

Pour plus de facilités penser à renseigner les variables d’environnement systèmes sur le répertoire php8, Symfony ,et git.

Une image contenant texte, Appareils électroniques, capture d’écran, affichage

Description générée automatiquement

# Symfony – Installation et initialisation

## Création d’un nouveau projet en dernière version minimal

Lancer la commande :

T:\wamp\www\ProjetSymfony>symfony new --full « nom projet »

Si la commande s’exécute normalement, une version minimale d’une application symfony est généré.

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

T:\wamp\www\ProjetSymfony>symfony new --full « nom projet »

## Création d’un nouveau projet en dernière version minimal LTS.

Lancer la commande :

T:\wamp\www\ProjetSymfony>symfony new --full « nom projet » --version=lts

LTS signifie long terme support. Ce sont les versions stables déployé et recommandé par les éditeurs pour les environnements de production.

https://symfony.com/releases

## Création d’un nouveau projet dans une version spécifique minimal.

Lancer la commande :

T:\wamp\www\ProjetSymfony>symfony new --full « nom projet » --version=4.2

Ici nous aurons une version symfony 4.2.

## Information projet

La commande suivante vous permettra d’avoir les informations des versions Symfony et Php du projet  :

T:\wamp\www\ProjetSymfony\projet2>symfony console about

Une image contenant texte, capture d’écran

Description générée automatiquement

## Lancer le serveur WEB Symfony

La commande suivante vous permettra ensuite de lancer votre projet depuis le serveur WEB embarqué dans les binaires symfony  :

T:\wamp\www\ProjetSymfony\projet2>symfony serve

Une image contenant texte, capture d’écran, Police, logo

Description générée automatiquement

## Arrêter le serveur WEB Symfony

La commande suivante permet d’arrêter le serveur web symfony et de récupérer son shell.

T:\wamp\www\ProjetSymfony\projet2>symfony server:stop

# Symfony – Configuration d’un projet

## Vérification des pré requis

Lancer les commandes :

>php -v

> composer --version

> mysql -V

> symfony -v

A noter pour faire une verrification complete , faite la commande

symfony check:requirement

php -v

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

mysql -V

Une image contenant Police, capture d’écran, texte, ligne

Description générée automatiquement

symfony -v

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

Composer --version

Une image contenant texte, Police, capture d’écran, Graphique

Description générée automatiquement

symfony check:requirement

Une image contenant texte, Police, Graphique, capture d’écran

Description générée automatiquement

## Configuration de l’environnement de développement local

Depuis l’arborescence de votre projet, copier le fichier .env et coller le contenu dans un nouveau fichier .env.local

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Modifier la ligner configurer la ligne DATABASE\_URL pour lui indiquer votre base de donnée à utiliser.

La config initiale :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

La config cible à avoir :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

# Création d’une base de **données**

## Creation de la base de données

Entrer la commande :

symfony console doctrine:database:create

## Création d’une Entité

Entrer la commande :

symfony console make :user

Symfony vous demande :

* Le nom de la class qui sera associé à la table User
* Si les user doivent etre stoqués en bdd
* D’entrer des champs. ( Par défaut, symfony crérra des chapms par défaut email, esuername, etc… )
* Si les mots de passes doivent être hashé.

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Une fois la commande exécuté, plusieurs fichiers seront créees. Notamment le modele user que l’on retrouve dans src/Entity/User.php

A noter, PHP8 permet d’avoir un système d’annotation avec le caractère « # » afin de spécifier les valeurs. Symfony s’en sert pour caractériser ses variables avec des type, auto incrément, etc…

Nous retrouverons cela automatiquement en BDD par la suite.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

## Modifier une entité

Entrer la commande :

symfony console make :Entity

Symfony vous demande :

* Le nom de la classe à créer ou supprimer
* Le nom de la nouvelle propriété à ajouter
* Le type
* La taille
* Si c’est nullable
* Si vous souhaitez à jouter encore une nouvelle propriété

Une image contenant texte, capture d’écran, logiciel, Système d’exploitation

Description générée automatiquement

## Migration d’entitée

Pour créer une « demande de migration »

Entrer la commande :

symfony console make:migration

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

Une image contenant texte, Police, capture d’écran, Graphique

Description générée automatiquement

Entrer ensuite la commande afin d’exécuter les migrations en attente :

symfony console doctrine:migrations:migrate ou symfony console d:m:m

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Notre base de données est bien mise à jour.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement