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](#_Toc151678010)

[2 Pré requis 2-5](#_Toc151678011)

[3 Symfony – Installation et initialisation 3-7](#_Toc151678012)

[3.1 Création d’un nouveau projet en dernière version minimal 3-7](#_Toc151678013)

[3.2 Création d’un nouveau projet en dernière version minimal LTS. 3-7](#_Toc151678014)

[3.3 Création d’un nouveau projet dans une version spécifique minimal. 3-7](#_Toc151678015)

[3.4 Information projet 3-8](#_Toc151678016)

[3.5 Lancer le serveur WEB Symfony 3-9](#_Toc151678017)

[3.6 Arrêter le serveur WEB Symfony 3-9](#_Toc151678018)

[4 Symfony – Configuration d’un projet 4-10](#_Toc151678019)

[4.1 Vérification des pré requis 4-10](#_Toc151678020)

[4.2 Configuration de l’environnement de développement local 4-11](#_Toc151678021)

[5 ORM Symfony 5-12](#_Toc151678022)

[5.1 Création de la base de données 5-12](#_Toc151678023)

[5.2 Création d’une Entité type User 5-12](#_Toc151678024)

[5.3 Créer ou Modifier une entité 5-14](#_Toc151678025)

[5.4 Migration d’entité (envoi en BDD) 5-14](#_Toc151678026)

[5.5 Ajout d’une relation 5-16](#_Toc151678027)

[6 Réalisation 6-17](#_Toc151678028)

[7 Codes utiles 7-18](#_Toc151678029)

[7.1 Alimenter une valeur de date automatiquement 7-18](#_Toc151678030)

# 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 --webapp nom\_du\_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 :

symfony new --webapp nom\_du\_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 --webapp nom\_du\_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 vérification complète , 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 données 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

# ORM Symfony

Nous allons voir comment utiliser l’ORM « Doctrine » de Symfony

## Création de la base de données

Entrer la commande :

symfony console doctrine:database:create

## Création d’une Entité type User

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 exécutée, plusieurs fichiers seront créées. Notamment le modèle 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

## Créer ou 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’entité (envoi en BDD)

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

## Ajout d’une relation

Une relation s’ajoute lors de la création d’un champ. Au moment ou on affecte le type il faut renseigner « relation » et suivre les indications.

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

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

# 1ere réalisation

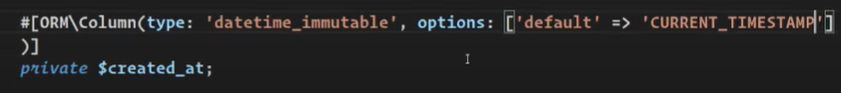
Gestion d’articles par utilisateurs et groupes

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

Description générée automatiquement

# Codes utiles

## Alimenter une valeur de date automatiquement



# Construire un Controller

## **Création d’un Controller**

Symfony console make:controller MainController

Cette commande permet de créer un contrôller et son Template associé. Ici ( MainController ).

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

Description générée automatiquement

Un template est une vue HTML composée d’une class avec une route (mère) et de ses méthodes avec des routes (filles).

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

## Template et Twig

Chaque Template est écrit en HTML et en TWIG.

Twig est un moteur de template open-source pour le langage de programmation PHP. Il est utilisé pour générer des pages en combinant des modèles avec des données.

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

Description générée automatiquement

Un Template peut hériter d’un autre comme ici Base.html.twig

Votre template de base doit accueillir le css et le js global de votre site. Ainsi, tout Template peut hériter de menu / entête / footer en TWIG facilement avec cette syntaxe .

## Intégration de bootstrap

Télécharger Bootstrap depuis l’url :

<https://getbootstrap.com/docs/4.0/getting-started/download/>

Puis dans le répertoire public/assets créer et alimenter les répertoires suivants :

* Css :
  + Bootstrap.min.css
  + Style.css ( style global spécifique )
* Javascript :
  + bootstrap.bundle.js
  + script.js ( script global spécifique ).

Une image contenant texte, Police, logiciel, Logiciel multimédia

Description générée automatiquement

Mettez à jour le Template de base « base.html.twig » comme ci-dessous.

<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>{% block title %}Welcome!{% endblock %}</title>

        <link rel="icon" href="data:image/svg+xml,<svg xmlns=%22http://www.w3.org/2000/svg%22 viewBox=%220 0 128 128%22><text y=%221.2em%22 font-size=%2296%22>⚫️</text></svg>">

        {# Run `composer require symfony/webpack-encore-bundle` to start using Symfony UX #}

        <!--Feuille de style

        Le mot clé "asset" permet : d'aller chercher des ressources dans le fichiers public

        Le mot clé "{ { } }" permet : de faire l'equivalent d'une echo en twig

        -->

        <link rel="stylesheet" href="{{ asset('assets/css/bootstrap.min.css')}}">

        <link rel="stylesheet" href="{{ asset('assets/css/style.css')}}">

        {% block stylesheets %}

        {% endblock %}

        <!--mon code js de style

        le mot clé defer permet : de charger le contenu js du fichier juste avant le chargement du dom content loaded

        -->

        <script src="{{ asset('assets/js/bootstrap.bundle.js')}}" defer></script>

        <script src="{{ asset('assets/js/script.js')}}" defer></script>

        {% block javascripts %}

        {% endblock %}

    </head>

    <body>

        <div class="container">

        {% block body %}{% endblock %}

        </div>

    </body>

</html>

# Construire un formulaire

## Création d’un formulaire

On lance la commande suivante pour créer un formulaire sur notre entité Groupes.

php bin/console make:form Groupes

Celle-ci créera un formulaire dans app\form

## Paramétrage d’un contrôleur pour joindre le formulaire

On ajoute les lignes ci-dessous dans notre contrôleur main

Il faut dans le code du contrôleur souhaité appeler :

* Le formulaire
* La bibliothèque Request de http Foundation pour les GET, POST, PUT, DELETE
* Doctrine pour l’ORM.

use App\Form\GroupeTypes;

use Symfony\Component\HttpFoundation\Request;

use Doctrine\ORM\EntityManagerInterface;

La méthode ci-dessous permet de récupérer et enregistrer en base de donnée les informations du formulaire crée.

Informations récupérées sont idGroup et nameGroup.

#[Route('/groupeNew', name: 'groupeNew')]

    public function groupeNew(Request $request, EntityManagerInterface $entityManager): Response

    {

        $groupes = new Groupes();

        $form = $this->createForm(GroupesType::class, $groupes);

        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {

            // Enregistrez l'entité dans la base de données

            $groupes->setIdGroup($form->get('idGroup')->getData());

            $groupes->setNameGroup($form->get('nameGroup')->getData());

            $entityManager->persist($groupes);

            $entityManager->flush();

            return $this->redirect('/main/groupes');

        }

        return $this->render('main/new.html.twig', [

            'form' => $form->createView(),

        ]);

    }

## Paramétrage d’une vue pour joindre le formulaire

Puis on créer enfin une vue TWIG pour notre formulaire.

{% extends 'base.html.twig' %}

{% block title %}Hello MainController!{% endblock %}

{% block body %}

<style>

    .example-wrapper { margin: 1em auto; max-width: 800px; width: 95%; font: 18px/1.5 sans-serif; }

    .example-wrapper code { background: #F5F5F5; padding: 2px 6px; }

</style>

<div class="example-wrapper">

    <h1>Create a new Groupe</h1>

    {{ form\_start(form) }}

    {{ form\_widget(form) }}

    <button type="submit">Create</button>

    {{ form\_end(form) }}

</div>

{% endblock %}

# Construire une authentification

## Création de l’autentification

T:\wamp5\www\ProjetSymfony \projet01> symfony console make:auth

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

Description générée automatiquement

Plusieurs fichiers vont être générés dont le securityController.

Dans ce fichier, la méthode login contient une fonction commentée qui permet de gérer si une personne est bien connecté ou non. A réutiliser.

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

Description générée automatiquement

## Paramétrage de l’authentification

En cas de succès d’authentification nous redirigerons l’utilisateur vers nous page « Main »

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

Description générée automatiquement

## Création d’un formulaire d’authentification

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

Description générée automatiquement

# 2eme réalisation

Pour chacune de nos tables précédemment crée, construire également pour chacune d’entre elles des formulaires de saisie ainsi que des vues.

L’objectif et d’avoir un CRUD pour nos tables.

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

Description générée automatiquement