



المعهد العالي للدراسات التكنولوجية بمدنين

Institut Supérieur des Etudes Technologiques de Médenine

## Atelier Framework Coté Serveur

### TP1

### Mise en place de l'environnement du travail (Symfony 6)

Enseignante : K.MECHLOUCH

Classe : L2DSI

## 1. Notion d'un Framework

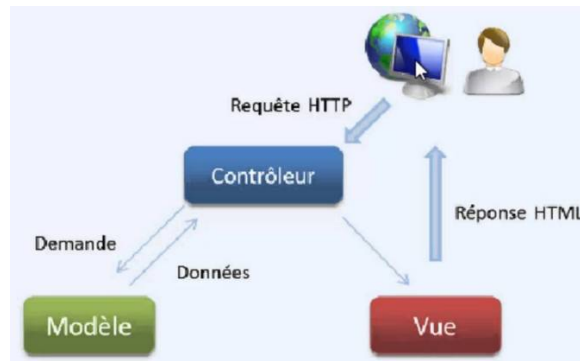
Un **framework** (ou **infrastructure logicielle** en français) désigne un ensemble d'outils et de composants logiciels à la base d'un logiciel ou d'une application. Le framework, encore appelé structure logicielle, canevas ou socle d'applications en français, établit les fondations d'un logiciel ou son squelette applicatif.

L'objectif du framework est de simplifier et d'uniformiser le travail des développeurs. Il fonctionne comme un cadre ou un patron, mais son maniement suppose d'avoir déjà un profil expérimenté. En général, une infrastructure logicielle est associée spécifiquement à un langage de script ou de programmation. Par exemple, Hibernate est un framework pour JavaScript, Django pour Python et Symfony pour PHP.

## 2. Modèle MVC

**Modèle-vue-contrôleur** ou **MVC** est un motif d'architecture logicielle destiné aux interfaces graphiques lancé en 1978 et très populaire pour les applications web. Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs (cf. figure1).

- Un modèle (Model) contient les données à afficher.
- Une vue (View) contient la présentation de l'interface graphique.
- Un contrôleur (Controller) contient la logique concernant les actions effectuées par l'utilisateur.



**Figure 1 : Architecture MVC**

Ce motif est utilisé par de nombreux frameworks pour applications web tels que Ruby on Rails, Grails, Spring, Struts, Symfony, Laravel et Django.

### 3. Présentation de Symfony

**Symfony** est un ensemble de composants PHP ainsi qu'un framework MVC libre écrit en PHP.

L'agence web française SensioLabs est à l'origine du framework *Sensio Framework*. À force de toujours recréer les mêmes fonctionnalités de gestion d'utilisateurs, gestion ORM, etc., elle a développé ce framework pour ses propres besoins. Comme ces problématiques étaient souvent les mêmes pour d'autres développeurs, le code a été par la suite partagé avec la communauté des développeurs PHP.

### 4. Installation de Symfony

#### ➤ Outils à utiliser

Avant de créer votre premier projet Symfony vous devez :

#### a. Installer un serveur web

Vous avez besoins d'installer **PHP 8.0.2** ou supérieur et un SGBD comme MySQL. On vous demande alors d'installer **XamppServer** comme exemple.

Nous aurons parfois besoin d'exécuter des commandes PHP via la console pour générer du code ou gérer la base de données dans le framework Symfony. Ce sont des commandes qui vont nous faire gagner du temps. Vérifions donc sous Windows que PHP est bien disponible en console :

- Lancez l'invite de commandes (cmd).
- Tapez php -v.

```

C:\Users\mechl>php -v
Microsoft Windows [version 10.0.19042.1466]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\mechl>php -v
PHP 8.1.2 (cli) (built: Jan 19 2022 10:18:23) (ZTS Visual C++ 2019 x64)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
  
```

**Figure 2 : Vérification de l'installation de PHP**

En cas d'erreur => Windows ne sait pas où trouver PHP ; il faut juste lui montrer le chemin :

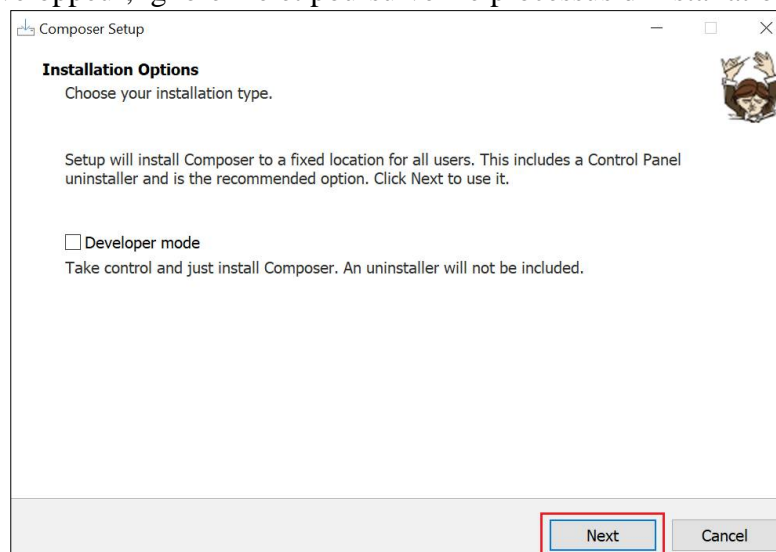
- Aller au répertoire xampp et copier l'emplacement de votre interpréteur php (C:\xampp\php) ;
- Accéder aux paramètres système avancés : Panneau de configuration - Système et sécurité – Système - Paramètres système avancés - Cliquez sur le bouton Variables d'environnement ;
- Dans le panneau Variables système, trouvez l'entrée Path (vous devriez avoir à faire descendre l'ascenseur pour la trouver) ;
- Entrez votre répertoire PHP à la fin et Confirmez en cliquant sur OK.
- Vous devez ensuite redémarrer l'invite de commandes pour prendre en compte les changements.

## b. Installer Composer

Composer est un gestionnaire de dépendances pour PHP. Il sert de gestionnaire de dépendance entre applications et bibliothèques. Il permet à ses utilisateurs de déclarer et d'installer les bibliothèques dont le projet principal a besoin et permet de créer (par duplication) un projet à partir d'un modèle donné.

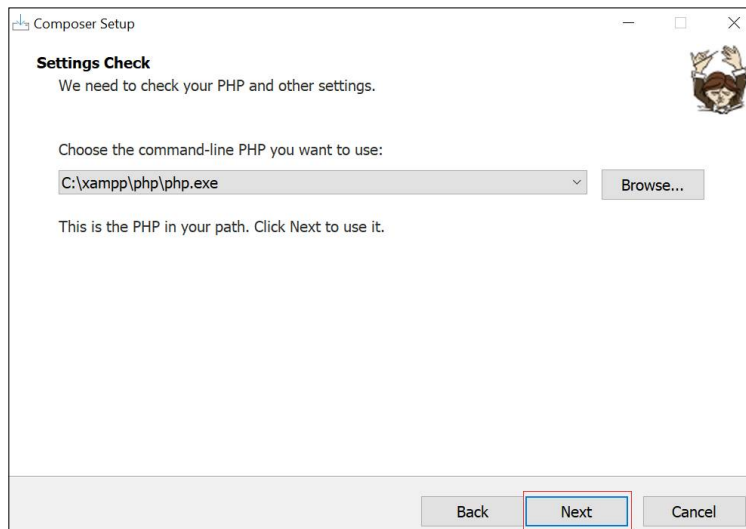
Pour l'installer, il suffit de suivre les étapes suivantes :

- Lancez l'assistant d'installation de **Composer**. Lorsqu'il vous demande d'activer le mode développeur, ignorez-le et poursuivez le processus d'installation.



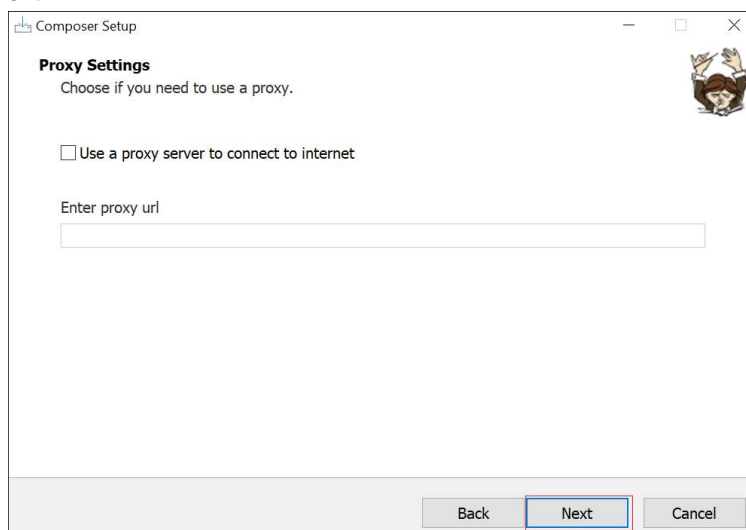
**Figure 3 : Installation de Composer (1)**

- Une autre fenêtre s'ouvre et vous demande de localiser la ligne de commande PHP. Par défaut, elle se trouve dans **C:/xampp/php/php.exe**. Après avoir spécifié le chemin, cliquez sur **Suivant**.



**Figure 4 : Installation de Composer (2)**

- Vous serez invité à entrer les **paramètres du proxy**. Laissez la case non cochée et sautez cette partie en cliquant sur **Suivant**. Ensuite, dans la dernière fenêtre, cliquez sur **Installer**.



**Figure 5 : Installation de Composer (3)**

- Une fois l'installation terminée, ouvrez l'invite de commande et tapez la commande suivante : `composer` :  
En cas d'erreur :

```
C:\Users\mechl>composer
'composer' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commandes.
```

**Figure 6 : Vérification de l'installation de Composer**

Il faut ajouter le chemin de Composer dans les variables d'environnement système (Path) :  
C:\ProgramData\ComposerSetup.

### c. Installer Symfony CLI

En option, vous pouvez également installer Symfony CLI. Cela crée un binaire appelé `symfony` qui fournit tous les outils dont vous avez besoin pour développer et exécuter votre application Symfony localement.

Si vous utilisez le fichier binaire (créer un dossier c:\symfony et y placer le fichier)

#### d. Installer Visual Studio Code

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS.

##### ➤ Création du projet

- Créer un dossier « symfony » sous xampp/htdocs.
- Si vous allez créer une application web :

```
symfony new my_project_directory --webapp
```

- Si vous allez créer un microservice, une application console ou une API :

```
symfony new my_project_directory
```

Si vous n'utilisez pas l'outil Symfony, exécutez ces commandes pour créer la nouvelle application Symfony à l'aide de Composer :

- Si vous allez créer une application web :

```
composer create-project symfony/skeleton my_project_directory
```

```
$ cd my_project_directory
```

```
$ composer require webapp
```

- Si vous allez créer un microservice, une application console ou une API :

```
$ composer create-project symfony/skeleton my_project_directory
```

**N.B :** Par défaut, les commandes précédentes permettent de créer une application qui utilise la dernière version. Si vous souhaitez utiliser une version spécifique, ajoutez l'option --version :

```
symfony new my_project_directory --version="6.2.* "
```

ou

```
composer create-project symfony/skeleton:"6.2.*" my_project_directory
```

##### ➤ Exécution

- Une fois l'exécution est terminée, se positionner dans le dossier de votre projet «TP1» par la commande :

```
cd TP1
```

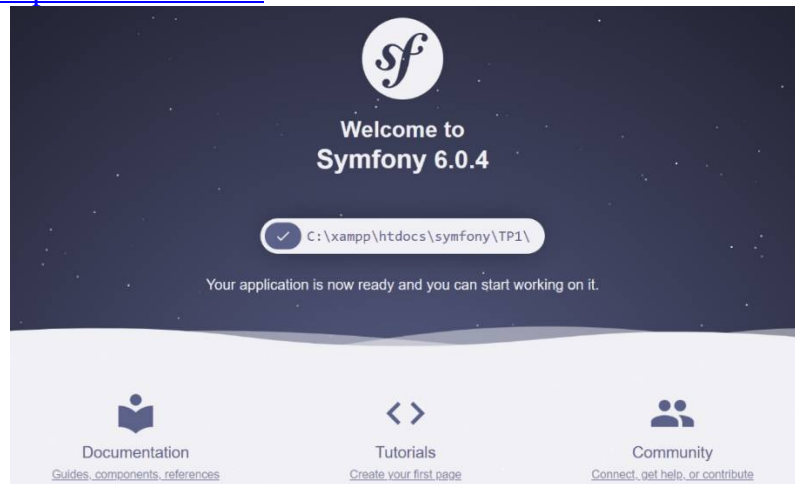
- Lancer le serveur local en exécutant la commande suivante :

```
symfony server:start
```

ou

```
php -s localhost:8000 -t public
```

- Lancer un navigateur et accéder à l'URL suivante pour vérifier l'installation de ton projet : <http://localhost:8000>:



**Figure 7 : Page d'accueil**

- Pour arrêter le serveur tapez Ctrl+C depuis votre terminal.

## **5. Structure d'un projet Symfony**

Un projet Symfony est basé sur les dossiers suivants (cf. figure 8) :

**config/**

Dans ce dossier, vous configurerez les routes, les services et les packages.

**src/**

Ce dossier contient le code PHP.

**templates/**

Ce dossier contient les templates Twig.

**bin/**

Ce dossier contient le fichier bin/console (et d'autres fichiers exécutables moins importants).

**var/**

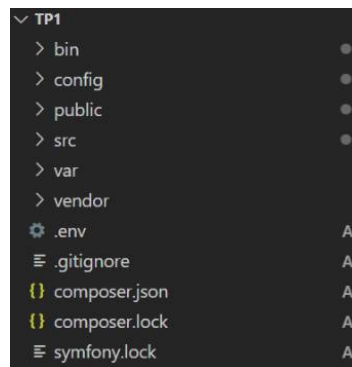
Ce dossier contient les fichiers créés automatiquement, comme les fichiers de cache (var/cache/) et les journaux (var/log/).

**vendor/**

Ce dossier contient les bibliothèques tierces qui sont téléchargés via le gestionnaire de packages Composer.

**public/**

Il s'agit de la racine du document pour votre projet : vous placez ici tous les fichiers accessibles au public.



**Figure 8 : Structure d'un projet Symfony**