



Atelier Framework coté serveur

Au : 2024-2025

Classe : L2DSI

Enseignante : Nidhal Cherif



Atelier N°1

Installation et configuration du Symfony 7

Objectifs :

- Utiliser l'invite des commandes cmd pour exécuter un script php
- Installer composer
- Installer et configurer Symfony en environnement Windows
- Comprendre l'arborescence d'un projet Symfony
- Mettre en place VS code ou PHP Storm pour travailler

1. Exigences techniques



La création d'un projet Symfony de version 7.2 exige l'installation de :
PHP 8.2 ou supérieur et composer qui est utilisé pour l'installation des packages

- Télécharger Xampp
- Configurer php sous Windows : rajouter le chemin où a été installé php dans votre variable d'environnement path.



Voici un article qui explique comment faire :

<https://www.forevolve.com/en/articles/2016/10/27/how-to-add-your-php-runtime-directory-to-your-windows-10-path-environment-variable/>

- Installer **composer** à partir du site : <https://getcomposer.org/download>
- Installer git : <https://git-scm.com/download/win>
- Installer Symfony CLI à partir du site <https://symfony.com/download>

Cela crée un binaire appelé Symfony qui fournit tous les outils dont vous avez besoin pour développer et exécuter votre application Symfony localement.

Ou faire l'installation avec composer en tapant la commande suivante:

```
>composer require symfony/console
```

2. Installation du Framework Symfony

- sous l'invite des commandes cmd, écrire la commande suivante pour la création d'un projet nommé MyProject complet (--webapp) avec une version 7.2 de symfony :

```
> symfony new MyProject --version="7.2.*" --webapp
```

- Maintenant, placez-vous dans le répertoire **MyProject** en utilisant la commande cd

```
> cd MyProject
```

👉 Vous pouvez vous rendre dans votre répertoire **MyProject** via l'explorateur de fichiers et de faire **un clic droit** en appuyant en même temps **sur la touche Shift et ctrl** de votre clavier. Dans le menu contextuel, choisissez **Ouvrir une fenêtre de commandes ici** (ou PowerShell ici)

- Lancer le serveur avec la commande :

```
MyProject>symfony server:start
```

👉 le serveur personnalisé ne sera lancé que lorsqu'on veut développer (en mod dev) son port est 8000 et non pas 80 comme Apache ou Nginx le jour ou on passera le site en mode production on n'en aura plus besoin

- Tester l'url : <http://localhost:8000/>

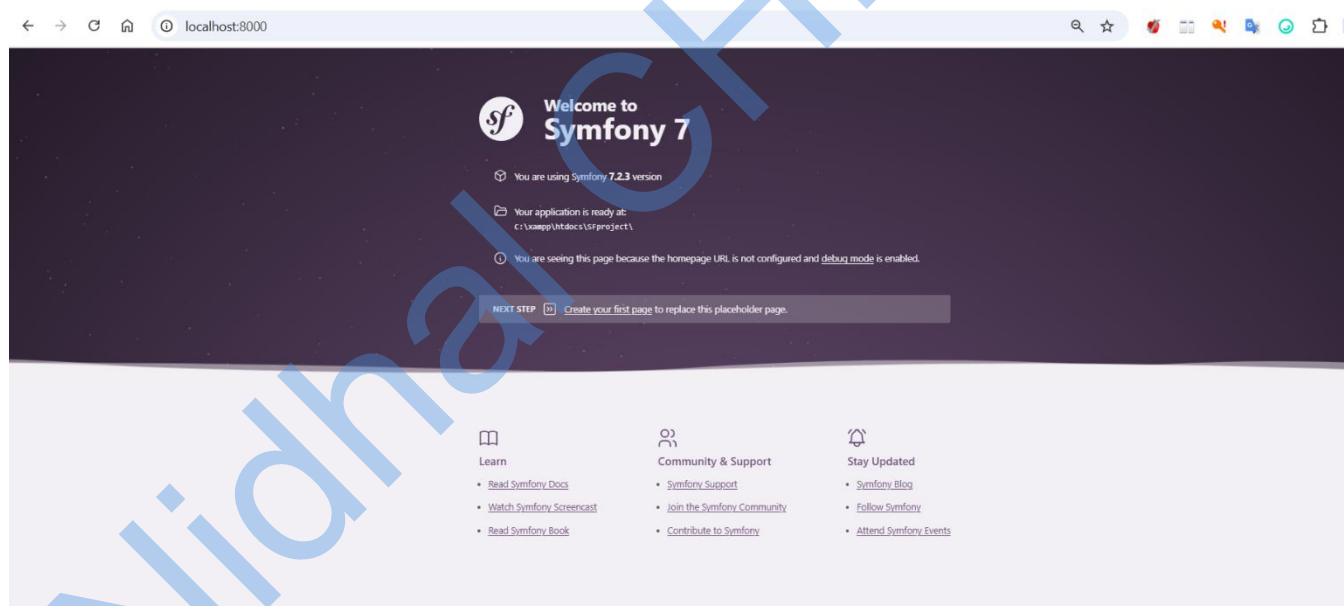


Figure1 : La page d'accueil de Symfony è

Pour arrêter le serveur, taper :

```
MyProject>symfony server:stop
```

- Activation de TLS

Il est important de parcourir localement la version sécurisée de vos applications pour détecter rapidement les problèmes de contenu mixte et pour exécuter des bibliothèques qui ne s'exécutent qu'en HTTPS.

- Activer la version sécurisée en local avec la commande suivante :

```
>symfony server:ca:install
```

- Démarrer le serveur :

```
>symfony serve -d
```

- Comprendre la structure du Symfony

Le dossier **MyProject** existe maintenant et l'arborescence d'un projet Symfony est en place comme suit :

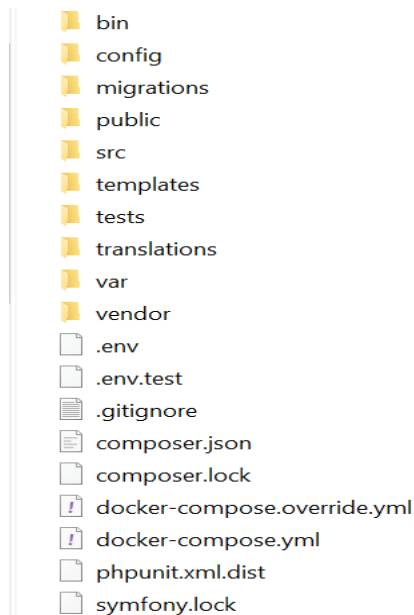


Figure 2 : Arborescence du projet Symfony 6

bin/ : Ce dossier contient les exécutables disponibles dans le projet. La plus importante est la console qui permet d'exécuter des commandes Symfony.

Config / : Il contient toute la configuration de votre application.

public/ : il contient les fichiers publiquement accessible comme le css, les images, Javascript.....

templates / : il contient les vues de votre applications

src/ : il contient votre code PHP (Contrôleur et Modèle)

test/ : Le test automatique de votre application est ici.

var/ : C'est ici que sont créés automatiquement les fichiers comme le cache et les logs.

Template/ : c'est le dossier vue de l'architecture MVC il contient les templates

3. Mise en place Visual studio Code pour travailler

- Installer Visual studio code à partir du site : <https://code.visualstudio.com>
- Installer Les extensions suivante dans VScode :

- **Extension PHP DocBlocker** : Essentiel pour documenter correctement le code.
- **Extension Twig** : Elle permet la coloration syntaxique au sein des fichiers .
- **Extension PHP Namespace Resolver** : Essentielle lorsqu'on travaille avec le système des espaces de noms de PHP (les Namespaces), elle vous permettra de retrouver aisément dans quel espace de noms se trouve telle ou telle classe de Symfony.

4. Mise en place de PhpStorm de JetBrains

4.1 Création d'un compte JetBrains

Pour pouvoir télécharger l'IDE PhpStorm, vous devez commencer par créer un compte student avec votre compte institutionnel name@rades.r-iset.tn via cette url :

<https://www.jetbrains.com/shop/eform/students>

Produits JetBrains pour l'apprentissage

Avant de vous inscrire, veuillez consulter [la FAQ et les conditions de l'abonnement éducatif](#).

Figure 3 : Création d'un compte JetBrains

4.2 Téléchargement de PhpStorm

Une fois votre compte est confirmé, connectez vous pour pouvoir accéder aux différents produits de JetBrains. Et télécharger PhpStorm

1 Student Pack License

Figure 4 : Les produits de JetBrains

4.3 Configuration de PhpStorm

- Créer un nouveau projet php

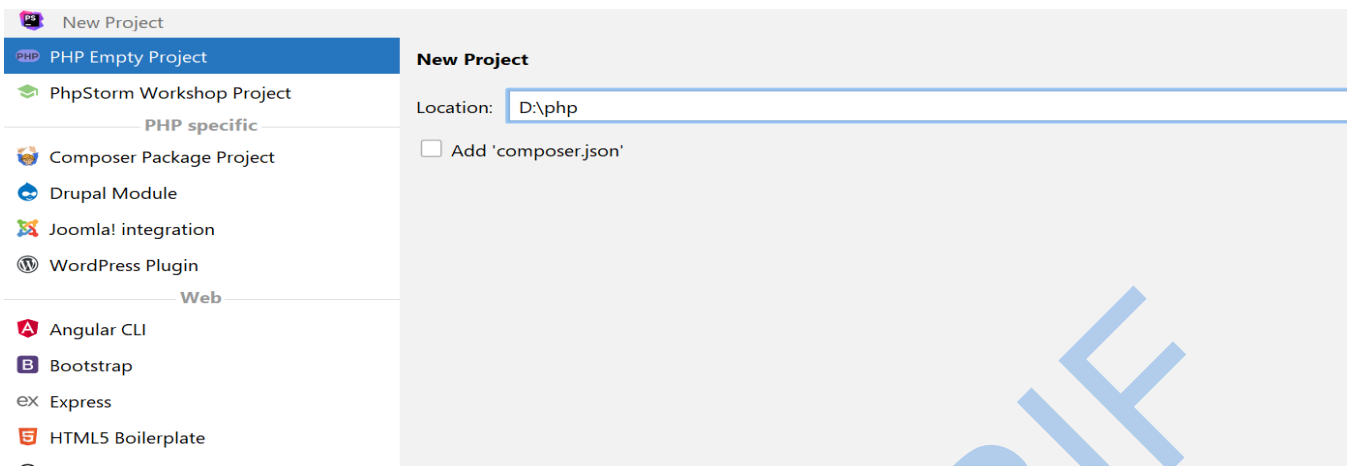


Figure 5 : Création d'un nouveau projet php sous PhpStorm

- Créer un nouveau fichier php : hello.php

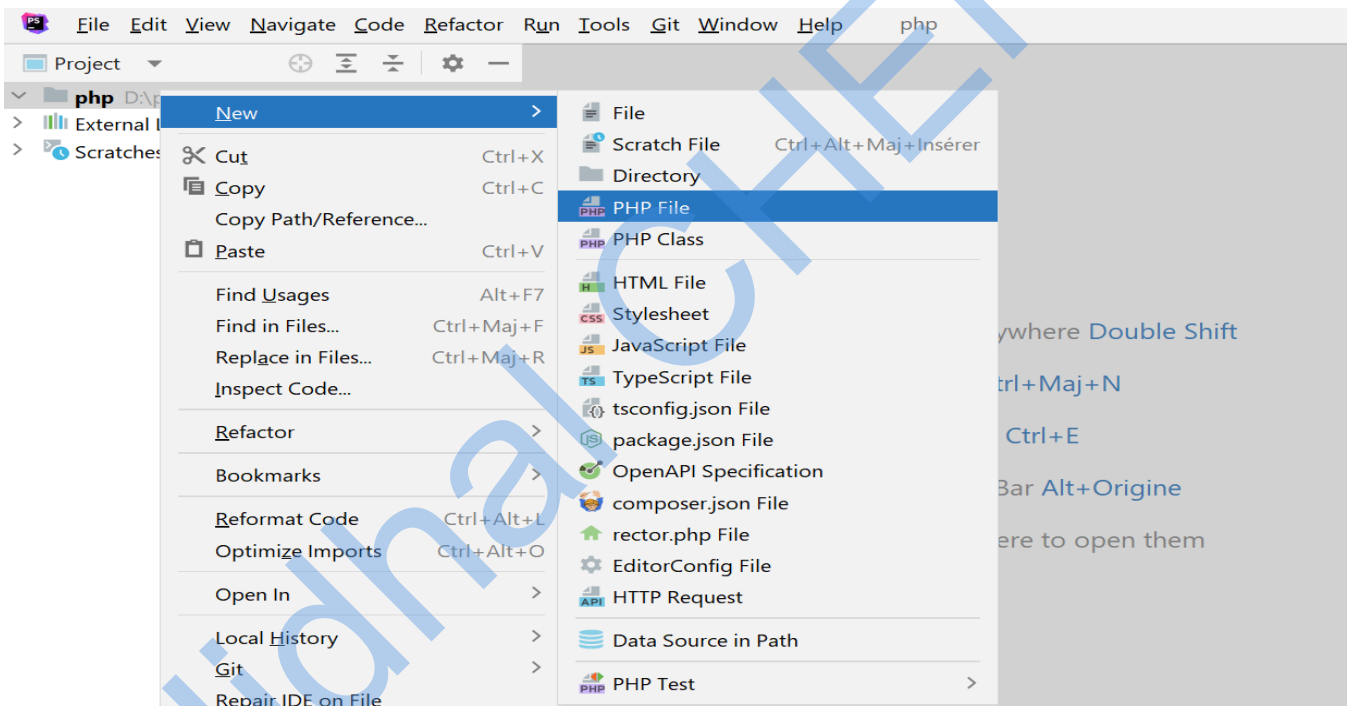


Figure 6 : Création d'un nouveau fichier php

- Saisir le code php suivant et exécuter le avec Run

```
<?php
echo "Hello everyone!";
```

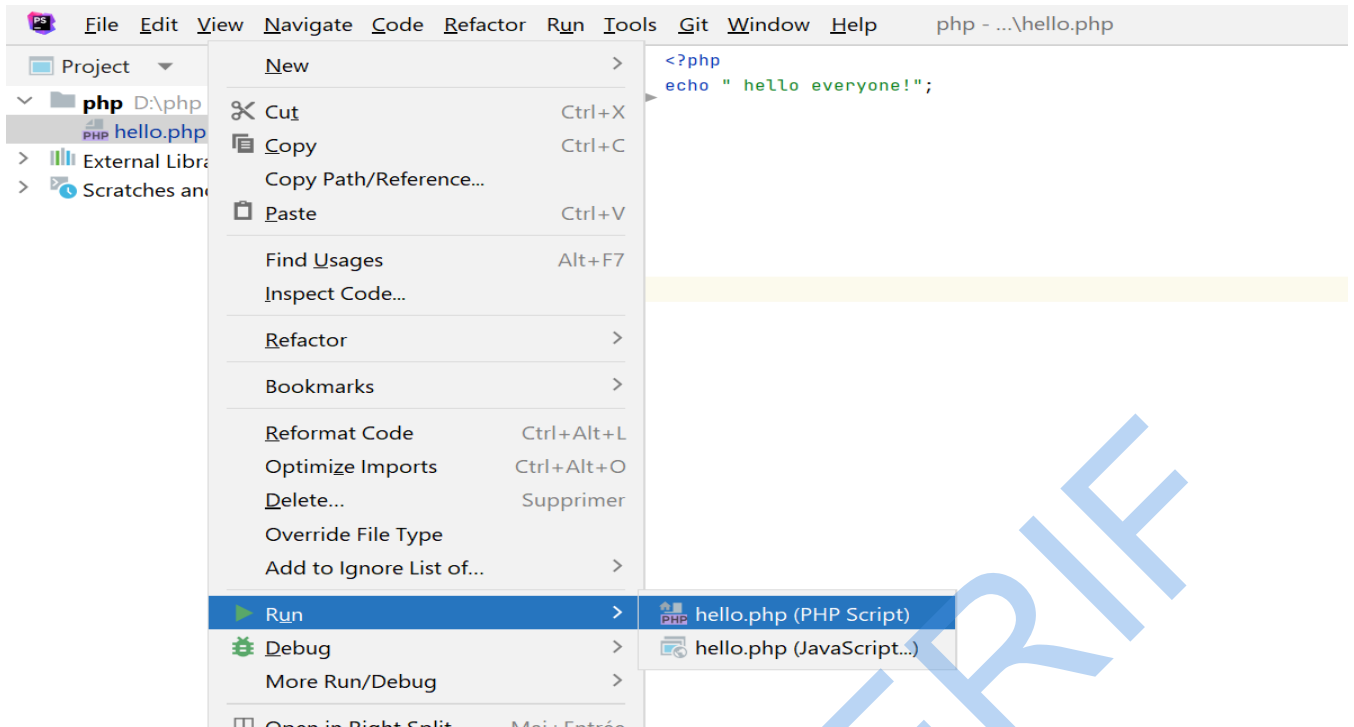


Figure 7 : Exécution du fichier PHP

Une erreur est affichée et qui exige une configuration de php pour la résoudre

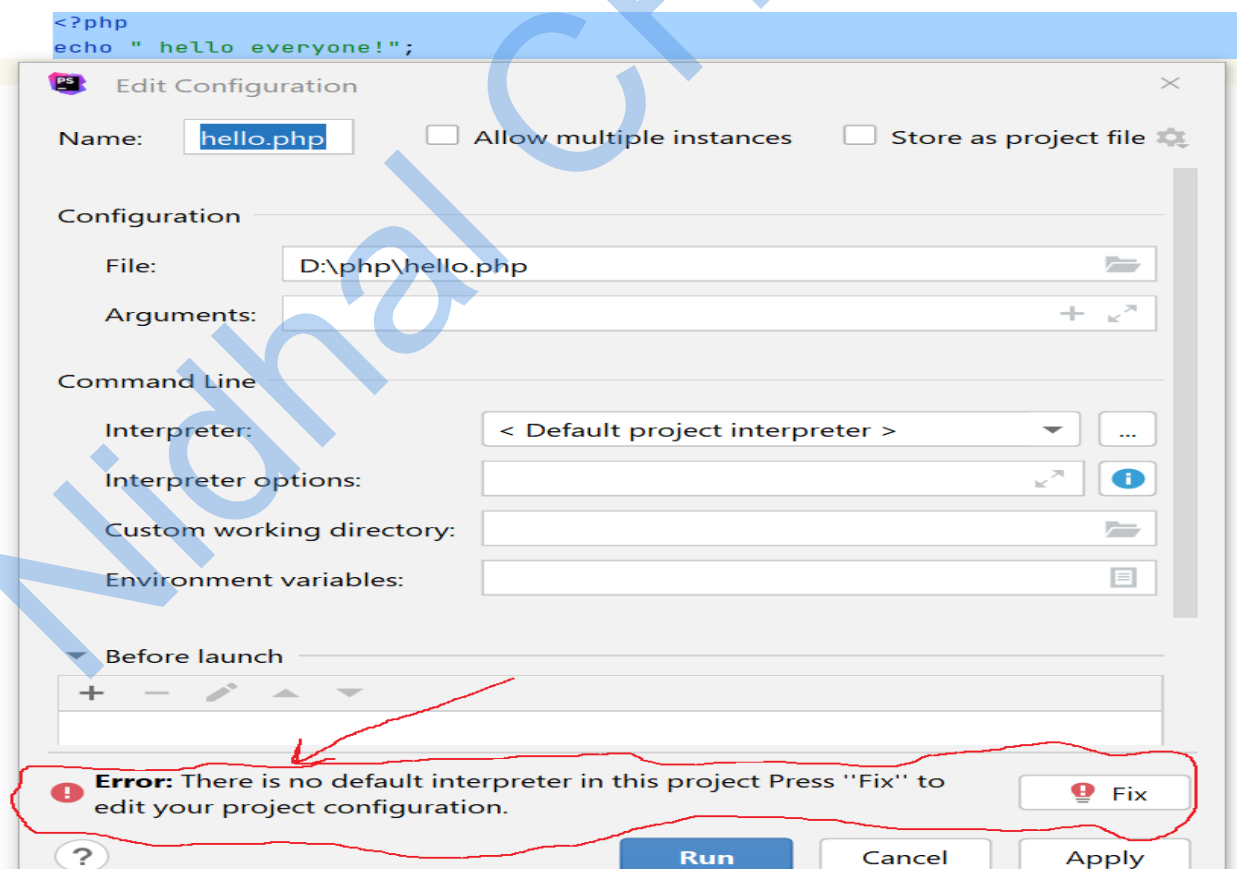


Figure 8 : fenêtre de configuration

-Cliquer sur le bouton fix , pour entrer le chemin de votre interpréteur php :

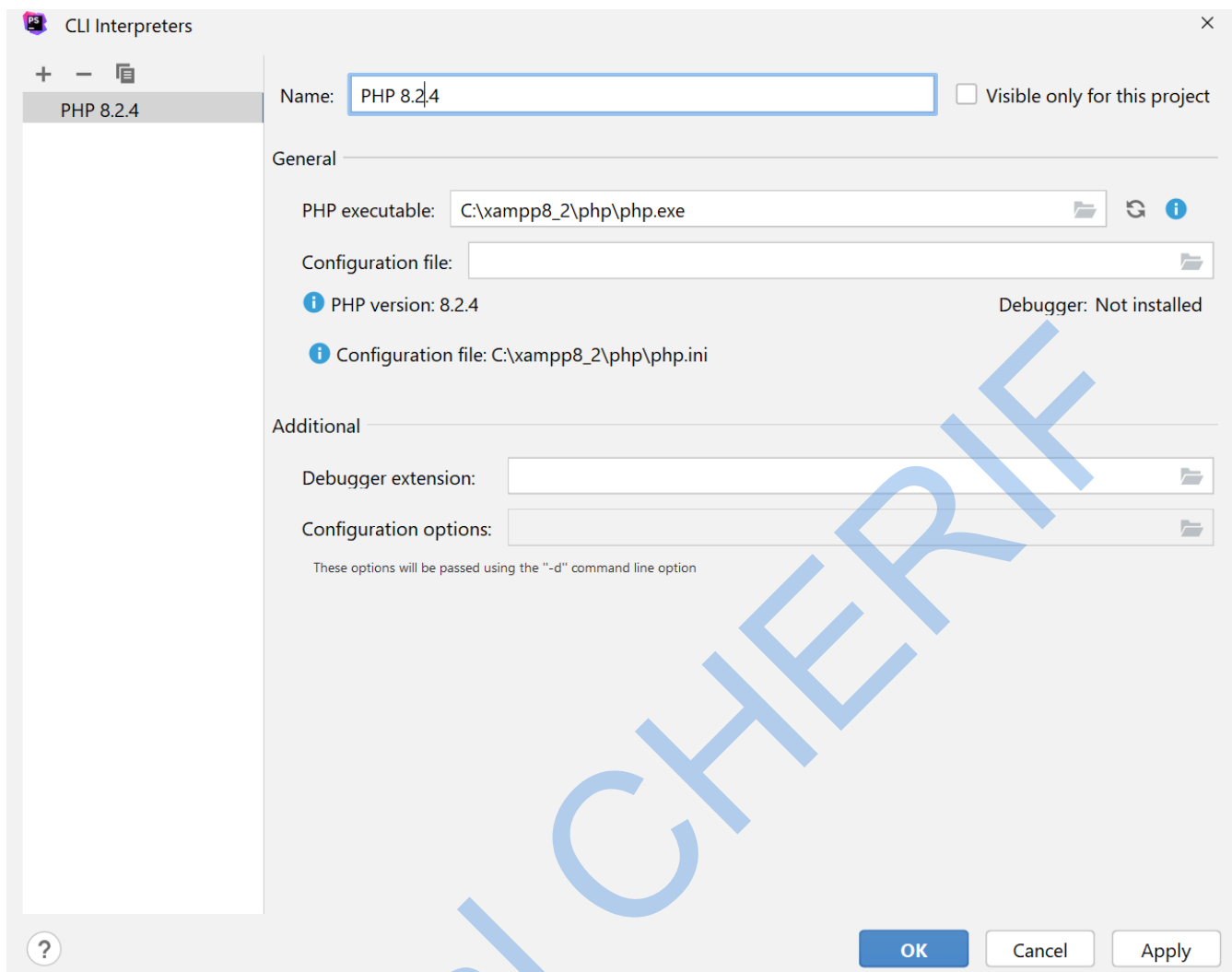


Figure 9 : indication du chemin de l'interpréteur