

WMODULE 1

# Introduction à Symfony

Découvrez le framework PHP professionnel qui structure, sécurise et accélère le développement d'applications web modernes.

# Objectifs pédagogiques

## Comprendre les frameworks

Découvrir le rôle essentiel d'un framework dans le développement web moderne et ses avantages

## Découvrir Symfony

Explorer l'écosystème Symfony et sa place dans le monde PHP professionnel

## Installer et créer

Mettre en place l'environnement de développement et créer votre premier projet Symfony

## Maîtriser MVC

Comprendre l'architecture Model-View-Controller et son application dans Symfony

## Développer un contrôleur

Créer votre premier contrôleur et comprendre son rôle dans l'application

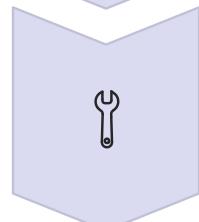
# Organisation du module

Ce module est structuré en trois parties progressives pour vous guider efficacement dans votre apprentissage de Symfony.



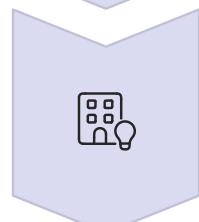
## Partie 1 : Introduction

Comprendre les frameworks web et découvrir Symfony dans l'écosystème PHP



## Partie 2 : Installation

Configurer l'environnement de développement et créer votre premier projet

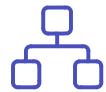


## Partie 3 : Structure & MVC

Explorer l'architecture Symfony et créer vos premiers contrôleur

# Qu'est-ce qu'un framework ?

Un framework est une infrastructure logicielle qui fournit une base solide pour construire des applications web de manière professionnelle et efficace.



## Structure le code

Impose une organisation claire et cohérente qui facilite la collaboration entre développeurs



## Accélère le développement

Fournit des composants réutilisables et des outils pour gagner du temps sur les tâches courantes



## Améliore la sécurité

Intègre des protections contre les vulnérabilités web les plus courantes (XSS, CSRF, injection SQL)



## Facilite la maintenance

Rend le code plus lisible et maintenable grâce à des conventions et une structure standardisée



## Respect des bonnes pratiques

Encourage l'adoption des design patterns et des standards de l'industrie

# Symfony dans l'écosystème PHP

Symfony est bien plus qu'un simple framework : c'est un écosystème complet qui propulse des milliers d'applications web à travers le monde.

## Framework open-source

Développé activement par une communauté mondiale et sponsorisé par SensioLabs

## Référence en entreprise

Choisi par de nombreuses grandes entreprises pour sa robustesse et sa maturité

## Stable et maintenu

Versions LTS (Long Term Support) avec 3 ans de maintenance garantie

## Utilisé par

- Drupal (CMS)
- PrestaShop (e-commerce)
- Magento (composants)
- Laravel (composants)

## Adapté pour

- Projets complexes
- Applications d'entreprise
- APIs REST
- Microservices

# Fonctionnement du Web

Comprendre le cycle requête-réponse est essentiel pour développer avec Symfony. Voici comment une application web communique.



## Client : Navigateur

L'utilisateur saisit une URL et envoie une requête HTTP

## Serveur : Symfony

Le framework traite la requête, exécute la logique métier



## Base de données

Les données sont récupérées ou modifiées selon les besoins

## Réponse HTTP

Le serveur renvoie une page HTML, JSON ou autre au navigateur

- ❑ **Le rôle de Symfony :** Le framework intercepte la requête HTTP, la route vers le bon contrôleur, gère la logique métier et génère la réponse appropriée.

# Outils nécessaires

Avant de commencer avec Symfony, assurez-vous d'avoir installé les outils essentiels suivants sur votre machine de développement.

1

## PHP 8.1 ou supérieur

Le langage de programmation côté serveur. Vérifiez votre version avec  
`php -v`

2

## Composer

Le gestionnaire de dépendances PHP indispensable pour installer Symfony et ses bibliothèques

3

## Symfony CLI

L'outil en ligne de commande officiel qui facilite la création et la gestion de projets Symfony

4

## Git

Le système de contrôle de version pour gérer l'historique de votre code

5

## Visual Studio Code

Éditeur de code recommandé avec support excellent pour PHP et Symfony

# Installation du projet

Créez votre première application Symfony en quelques commandes. Le paramètre `--webapp` installe tous les composants nécessaires pour une application web complète.

01

## Créer le projet

```
symfony new projet --webapp
```

Cette commande crée un nouveau projet avec tous les composants web (Twig, Doctrine, etc.)

03

## Démarrer le serveur

```
symfony serve
```

Lance un serveur de développement local avec support HTTPS automatique

02

## Accéder au répertoire

```
cd projet
```

Déplacez-vous dans le dossier du projet nouvellement créé

04

## Accéder à l'application

```
http://localhost:8000
```

Ouvrez votre navigateur et accédez à votre nouvelle application Symfony

# Structure du projet

Un projet Symfony est organisé selon une structure claire et standardisée. Chaque dossier a un rôle précis dans l'architecture de l'application.



## **src/**

Contient tout votre code métier : contrôleurs, entités, services, repositories



## **templates/**

Fichiers Twig pour générer les vues HTML de votre application



## **public/**

Point d'entrée web (index.php) et ressources publiques (CSS, JS, images)



## **config/**

Fichiers de configuration de l'application et des bundles



## **var/**

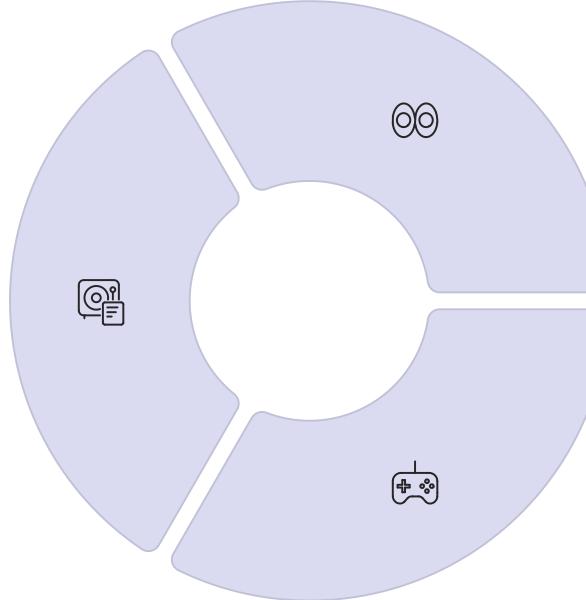
Données temporaires : cache pour les performances et logs pour le débogage

# Architecture MVC

Symfony utilise le pattern MVC (Model-View-Controller) pour séparer clairement les responsabilités et faciliter la maintenance du code.

## Model (Modèle)

Représente les données et la logique métier. Gère les interactions avec la base de données via Doctrine.



## View (Vue)

Interface utilisateur générée avec les templates Twig. Affiche les données sans logique métier.

## Controller (Contrôleur)

Reçoit les requêtes, orchestre la logique, interagit avec le modèle et retourne une réponse.

## Avantages de MVC

- Séparation claire des rôles et responsabilités
- Maintenance et évolution facilitées
- Testabilité accrue du code
- Travail en équipe plus efficace

 **En pratique :** Un contrôleur reçoit une requête, consulte le modèle pour récupérer des données, puis passe ces données à la vue qui génère le HTML final.

# Premier contrôleur

Symfony facilite la création de contrôleurs grâce à la commande `make:controller` qui génère automatiquement la structure nécessaire.



## Commande `make:controller`

Utilisez le Maker Bundle pour générer un nouveau contrôleur



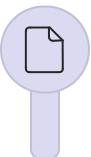
## Création automatique

Le fichier PHP et le template Twig sont créés simultanément



## Route intégrée

L'annotation de route est ajoutée automatiquement dans le contrôleur



## Template associé

Un fichier Twig de base est créé et lié au contrôleur

## Exemple de commande

```
php bin/console make:controller HomeController
```

## Fichiers générés

- `src/Controller/HomeController.php`
- `templates/home/index.html.twig`

Vous pouvez immédiatement tester votre contrôleur en accédant à la route définie !

# Outils développeur

Symfony intègre des outils puissants pour faciliter le développement et le débogage de vos applications.

## Symfony Profiler

Barre de débogage contextuelle qui apparaît en bas de chaque page en mode développement. Accédez à des informations détaillées sur chaque requête.



## Analyse des performances

Identifiez les goulots d'étranglement et optimisez le temps de chargement



## Requêtes SQL

Visualisez toutes les requêtes exécutées et leur temps d'exécution



## Variables et paramètres

Inspectez les variables de requête, session, et configuration



## Temps d'exécution

Mesurez précisément le temps de génération de chaque page

MODULE 1 TERMINÉ

# Conclusion : Les bases sont posées

Félicitations ! Vous avez acquis les fondamentaux de Symfony et êtes maintenant prêt à approfondir vos connaissances.

## ✓ Bases acquises

Vous comprenez le rôle des frameworks et l'architecture MVC

## ✓ Projet fonctionnel

Votre première application Symfony est opérationnelle

## ✓ Environnement configuré

Tous les outils sont installés et prêts à l'emploi

## Prochaine étape

Le **Module 2** vous permettra d'approfondir vos connaissances sur le routage et les contrôleurs avancés.

### Module 2

#### Routage & Contrôleurs

Routes dynamiques, paramètres, redirections et gestion des réponses