

# Symfony Avancé

## INFORMATIONS GÉNÉRALES

**Type de formation :** Formation continue

**Éligible au CPF :** Non

**Domaine :** Développement

**Action collective :** Non

**Filière :** Open Source - LAMP : Linux Apache PHP

**Rubrique :** Frameworks ZEND et Symfony

**Code de formation :** LEDN224

## PRÉSENTATION

### Objectifs & compétences

A l'issue de la formation, le stagiaire sera capable :

- Découvrir les fonctions avancées de Symfony
- Gérer les événements et l'automatisation de tâches - Tester et valider son application

### Public visé

Développeur PHP confirmé

### Pré-requis

Avoir suivi la formation Symfony initiation et approfondissement ou connaissance équivalente

## € Tarifs

**Prix public :** 2975 €

### Tarif & financement :

Nous vous accompagnons pour trouver la meilleure solution de financement parmi les suivantes :

**Le plan de développement des compétences de votre entreprise :** rapprochez-vous de votre service RH.

**Le dispositif FNE-Formation.**

**L'OPCO** (opérateurs de compétences) de votre entreprise.

**France Travail:** sous réserve de l'acceptation de votre dossier par votre conseiller Pôle Emploi.

**CPF -MonCompteFormation**

Contactez nous pour plus d'information : [contact@aston-institut.com](mailto:contact@aston-institut.com)

## 📍 Lieux & Horaires

**Durée :** 35 heures

**Délai d'accès :**  
**8 jours avant le début de la formation**

## PROGRAMME

### Introduction

Revue de l'architecture du framework  
Évolution suivant les versions

### Les événements et écouteurs

Découpler d'avantage de code métier via le gestionnaire d'événement  
Créer un écouteur d'événement : EventListener  
Créer un souscripteur d'événement : EventSubscriber  
Événement natifs symfony et événements personnalisés

### Altérer un comportement sans héritage via souscripteur d'événement

#### Le composant Cache

Présentation du composant Cache  
Cache contract vs PSR-6  
Accéder et sauvegarder des données en cache  
Supprimer, invalider ou programmer l'expiration des données  
Liste des Adapters disponibles  
**Atelier :** Mise en cache sous Doctrine

### Le composant Messenger

Comprendre les principes de communications inter-application  
Créer le message et le handler  
Diffuser le message  
Les transports disponibles : AMQP, Redis, Doctrine, In Memory, ...  
Configurer les transports et le superviseur

### Gérer des traitements en parallèle via Messenger

#### Le composant Mail

Les composant Mail et Swift\_mailer  
Installation et configuration du transport  
Créer un mail, gérer les adresses  
Gérer le format de contenu : text/html, utiliser twig

## 📅 Prochaines sessions

Consultez-nous pour les prochaines sessions.

Attacher un fichier, embarquer un image

### **Mettre en place les services d'envoi de mail**

#### **Le composant Console Commands**

Définir une commande

Gestion des entrées / sorties

Accès aux services

Tester les commandes

Sortie avancée : style et couleur

Sortie avancée : barre de progression, section, tableau

Entrées : distinguer arguments et options

Créer des questions : confirmation, information, choix

### **Définir des actions automatisables via des commandes**

#### **Formulaire avancé**

Créer un type de champs personnalisé avec son thème

Gérer les données associées modèle-normalisée-vue

Définir les DataTransformer associés et y injecter des services

Définir un DataMapper pour les données composées

Associer son type de champs personnalisé via un FormGuesser

Modifier dynamiquement le formulaire via les événements de formulaire

### **Mise en place de champs de formulaires avancés**

#### **Intégration de WebPack Encore**

Installation et configuration Yarn / Webpack

Définir les entrées webpack et leur ressources associées

Inclure les entrées dans Twig

Gérer les dépendances dynamiquement avec les modules ES6

Utiliser Sass, Less ou Stylus

Utiliser Typescript

Passer des données de Twig au javascript

Gestion du cache et versionning

Mettre en place une interface riche

### **Les tests unitaires et fonctionnels**

Utilisation de PHPUnit Bridge

Tests unitaires des services métier

Tests fonctionnels et gestion des formulaires

Gestion des dates : ClockMock

### **Mise en place de procédures de tests**

#### **Mise en Application : Mettre en place une API REST avec authentification**

## **MODALITÉS**

### **Modalités**

Jusqu'à 8 jours avant le début de la formation, sous condition d'un dossier d'inscription complet

### **Méthode**

**Fin de formation :** entretien individuel.

**Satisfaction des participants :** questionnaire de satisfaction réalisé en fin de formation.

**Assiduité :** certificat de réalisation.

**Validations des acquis :** grille d'évaluation des acquis établie par le formateur en fin de formation.