



## FORMATION SYMFONY 1.4 & DOCTRINE 1.2

**Durée** : 3 jours – 21 heures

**Tarifs** : 1 490 € HT

**REF.** : SF1C1

### OBJECTIFS

En trois jours, cette formation vous présentera les aspects techniques de Symfony et vous montrera comment les mettre en œuvre pour créer des applications réelles.

### PRE-REQUIS

Maîtrise de la programmation orientée objet en PHP

### PUBLIC

Développeurs PHP

### PROGRAMME DETAILLE

#### DU PHP « A PLAT » A LA STRUCTURE MVC

- Comprendre la séparation en couches
- Comprendre la théorie du design pattern MVC
- Mettre la logique métier dans des classes
- Utiliser la syntaxe PHP alternative pour les templates
- Découper un projet en niveaux application, module, action
- Naviguer dans l'arborescence des fichiers d'un projet Symfony
- Créer des liens grâce aux aides de vue
- Gérer un menu de navigation

#### LE TEMPLATING AVEC SYMFONY

- Lire les paramètres d'une requête et les passer à la vue
- Exploiter les aides de vue pour générer des templates réutilisables
- Concevoir des gabarits modulaires avec les partiels, slots
- Gérer l'apparence du site grâce à un layout
- Exploiter les CSS
- Construire un composant réutilisable dans tous les modules
- Explorer les aides de vue disponibles (texte, nombre, date)
- Bonnes pratiques de templating

#### L'INSTALLATION DU FRAMEWORK

- Installer symfony depuis une sandbox, PEAR, SVN
- Comprendre l'organisation du dépôt SVN de symfony
- Initialiser un projet, un module
- Initialiser un suivi de version
- Mettre à jour symfony
- Installer symfony sur un serveur de production

#### L'ENVIRONNEMENT DE TRAVAIL SYMFONY

- Utiliser la ligne de commande symfony
- Utiliser plusieurs environnements de développement
- Comprendre le déroulement d'une requête avec la web debug toolbar
- Comprendre la configuration en cascade
- Apprendre la syntaxe YAML
- Utiliser le cache de la configuration la commande cache:clear
- Explorer les fichiers de configuration
- Débugger une application symfony
- Installer des plugins : intérêts et contraintes

## LE ROUTAGE ET LE PROTOCOLE HTTP

- Comprendre ce qu'est une URL
- Distinguer un GET et un POST
- Utiliser Apache et la réécriture d'URL pour les requêtes entrantes
- Découvrir le contrôleur frontal
- Comprendre le routage bidirectionnel
- Savoir écrire un fichier routing.yml
- Distinguer une redirection HTTP d'une redirection interne
- Ajouter une extension à une page
- Utiliser les routes nommées
- Utiliser le routage pour gérer une arborescence de navigation

## BONNES PRATIQUES

- Sécurité : se protéger des attaques XSS et CSRF
- Refactoriser son code et bénéficier de la puissance de l'ORM
- Exploiter la ligne de commande et les tâches automatiques
- Respecter les conventions de codage

## INTERAGIR AVEC DES BASES DE DONNEES RELATIONNELLES

- Décrire son modèle en YAML pour Doctrine
- Configurer Symfony pour utiliser Doctrine
- Connecter une application à une base de données
- Manipuler des objets métiers
- Générer un modèle objet grâce à Doctrine
- Abstraire le SQL pour pouvoir changer de base de données
- Ecrire ses requêtes en Doctrine Query Language
- Utiliser les classes générées par Doctrine
- Utiliser le DQL pour les relations entre objets
- Refactoriser le code d'une action vers le modèle

## LE GENERATEUR D'ADMINISTRATION

- Générer ou initier un module
- Comprendre la différence entre un scaffolding et une administration
- Initier une administration
- Explorer le code généré
- Utiliser le fichier generator.yml pour modifier le code généré

## LA GESTION DES DROITS ET LA SESSION

- Gérer des messages d'alerte et d'erreur
- Stocker des informations en session et y accéder
- Gérer l'authentification des utilisateurs
- Déconnecter un utilisateur
- Tester les permissions d'accès
- Paramétrer les droits d'accès grâce au système de permissions

## ENVOYER DES E-MAILS

- Découvrir Swift Mailer
- Envoyer des e-mails depuis votre application
- Comprendre le fonctionnement de la queue

## LES TESTS AUTOMATISES

- Valider un élément de code avec un test unitaire
- Utiliser Lime et la ligne de commande de symfony pour exécuter les tests
- Alimenter une base de tests avec des données de test
- Valider une fonctionnalité avec un test fonctionnel
- Simuler une navigation entre des pages
- Vérifier un élément de page grâce à des sélecteurs CSS3
- Automatiser l'exécution de la suite de tests unitaires