

# SensioLabs Twig for Designers

## Objectifs

- Identifier les apports et décrire le fonctionnement des frameworks MVC
- Mettre en place un environnement de développement
- Créer un projet Symfony et mettre en place l'environnement de développement
- Développer une couche de persistance métier avec Doctrine 2
- Concevoir le routage des requêtes HTTP avec les contrôleurs.
- Créer des vues d'affichage et de saisie de données avec le moteur de template Twig
- Utiliser l'injection de dépendance entre les composants
- Appliquer une stratégie de sécurité efficace dans une applications Symfony

## Prérequis

- Maîtriser les fondamentaux de Symfony 3

## Programme

### Module 1 - Présentation de Symfony

- Historique
- Dépendances logicielles (Serveur web, PHP, Base de données)
- Communauté et écosystème (knpbundles, Sensiolab insight, etc...)
- Framework existants (en PHP et tous langage confondus)

### Module 2 - Environnement de développement

- Installation de Sublime Text 3
- Prise en main de Sublime Text 3 (interface, raccourcis clavier)
- Installation de Package Control
- Installation des plugins Sublime Text 3 essentiels
- Installation de WAMP
- Prise en main de WAMP (répertoire www, phpmyadmin)
- Configuration de WAMP (activation des modules nécessaires, paramétrage du php.ini)

### Module 3 - Mise en place d'un projet Symfony

- Présentation de composer (Gestionnaire de dépendence, Packagist)
- Installation de composer
- Installation de Symfony via Composer
- Présentation du dossier app & var
- Présentation du dossier src
- Présentation du dossier web
- Présentation du dossier bin et de la ligne de commande
- Configurer Symfony (Présentation des fichiers parameters.yml et config.yml)

## Module 4 - Concepts fondamentaux

- Le design pattern MVC
- PHP procédural vs Symfony (Comparaison du traitement d'une requête HTTP en PHP et classique et en Symfony)
- L'autoloading (absence de include ou require, liens entre les namespace des fichiers PHP et le filesystem)
- Les environnements d'exécution (dev, prod, test)
- Le cache (Intérêt du cache, comment le vider manuellement grâce à la ligne de commande)

## Module 5 - Les "bundles"

- Le concept de bundle et génération d'un bundle en ligne de commande
- Architecture d'un bundle

## Module 6 - Le debuggage

- La debug toolbar (Quand apparaît-elle ? Comment la faire disparaître)
- Le profiler

## Module 7 - La couche "contrôleur" et le routing

- Création d'un contrôleur (création d'une classe avec une action retournant un objet "Response", mapping avec le routeur)
- Gestion de paramètres variables dans l'URL
- Format d'URL avancé
- Effectuer une redirection
- Effectuer une redirection interne
- Retourner une erreur
- L'objet "Request"
- Génération d'un contrôleur en ligne de commande

## Module 8 - La couche "vue"

- Présentation de Twig
- Utiliser un template dans un contrôleur
- Afficher une variable
- La syntaxe Twig
- Les conditions
- Les boucles
- Créer un lien vers une page interne
- Créer des liens vers des ressources (images, js, css)
- Les helpers utiles
- Définir un layout grâce à l'héritage
- Inclure un template
- Inclure un contrôleur
- Debuggage template Twig

## Module 9 - La couche "modèle"

- Présentation de Doctrine
- Créer une entité
- Générer le schéma de la BDD
- Enregistrer une entité en base
- Afficher une entité
- Utiliser le ParamConverter
- Lister les entités
- Supprimer une entité
- Relation OneToMany / ManyToOne
- Relation ManyToMany
- Relation OneToOne
- Debuggage de Doctrine

## Module 10 - Les formulaires

- Créer un formulaire dans le contrôleur
- Afficher un formulaire dans un template Twig
- Définir les règles de validation du formulaire
- Gérer le traitement du formulaire
- La sécurité CSRF
- Debuggage des formulaires
- Installer des bundles tiers avec Composer