

SensioLabs Symfony2 Certification

Objectifs

- Comprendre la philosophie et les apports des frameworks MVC et plus particulièrement Symfony.
- Mettre en place un environnement de développement
- Créer un projet Symfony.
- Configurer une application Symfony.
- Découper une application en couche en suivant le modèle MVC et identifier le rôle de chaque élément.
- Développer une couche de persistance métier avec Doctrine 2 : le framework de persistance de Symfony.
- Concevoir le routage des requêtes HTTP avec les contrôleurs.
- Créer des vues d'affichage et de saisie de données et construire des formulaires interactifs.
- Maîtriser les interactions entre les vues, les contrôleurs et le modèle.
- Utiliser l'injection de dépendance entre les composants.
- Appliquer une stratégie de sécurité efficace dans une applications Symfony.

Prérequis

- Développer une application Web en PHP
- Avoir suivi la formation T44B-010 – Développer des applications Web ou posséder des connaissances équivalentes

Programme

Introduction aux frameworks MVC PHP

- Qu'est-ce que MVC ?
 - Apports et avantages
- La répartition des responsabilités dans une application PHP
 - Implémentation du modèle avec un ensemble de classes
 - Réalisation des vues avec les mécanismes de Templates PHP
 - Codage des contrôleurs pour l'aiguillage des flux

Présentation et introduction à Symfony

- Historique de Symfony
 - Différentes versions du framework : Les évolutions
 - Comprendre le versioning de Symfony : Le versionning sémantique
 - Choisir sa version pour un projet
- Architecture du framework
 - Les composants de base

Mettre en place un projet Symfony

- Création d'un projet

- Utilisation de Composer
 - Utilisation de l'installateur Symfony
 - Découverte de l'application générée
- Paramétrage du serveur Web pour l'hébergement de l'application
 - Utilisation du serveur Web embarqué
 - Création et configuration d'un hôte virtuel
- Configuration du framework pour l'utiliser dans ses applications
 - Liaison à la base de données
 - Les différents environnements : dev, test, prod
- Travaux pratiques:
 - Installer une plateforme et un IDE de développement pour Symfony
 - Créer un projet d'application Symfony et explorer sa structure et sa configuration

Configuration de Symfony

- Les fichiers de configuration
 - Répartition hiérarchique des fichiers
- Les syntaxes de configuration
 - Syntaxes YAML, Annotations, XML et PHP
 - Faire un choix de syntaxe
- Présentation de la console pour la génération des squelettes code de l'application
- Travaux pratiques:
 - Comparaison des formats et choix pour le projet
 - Compléter la configuration du projet d'application

La manipulation des bases de données pour le modèle

- Présentation des techniques de mapping objet/relationnel
 - Avantages et principes
- Doctrine 2 : Le framework de manipulation de données de Symfony
 - Installer et configurer Doctrine 2 dans l'application
 - Les entités et le mapping sur les tables
- Génération avec la console
 - Les différentes approches : Top-down, Bottom-Up
- Utiliser l'EntityManager
- Les repositories et le DQL
 - Personnalisation du repository pour les requêtes de sélection
 - Manipulation et syntaxe DQL
- Le QueryBuilder
- Travaux pratiques:
 - Création d'un schéma de données
 - Génération des entités et du mapping à partir du schéma de base de données
 - Codage de classes de manipulation des entités en implémentant les opérations CRUD
 - Exposition des fonctionnalités de manipulation de données sous forme de services

L'injection de dépendances pour les services applicatifs

- Le principe de l'injection de dépendances
- Le Service Container
 - Principe de fonctionnement et responsabilités
- La déclaration de services dans Symfony
 - Les services automatiques
 - Déclaration explicite de services
- Liaison entre les services
 - Injection par constructeur
 - Injection par méthode
 - Typage explicite des paramètres
 - Injection dans les contrôleurs
- Travaux Pratiques
 - Création de services applicatifs pour les fonctionnalités d'accès aux données
 - Utilisation de services Symfony

Les contrôleurs et le routage des requêtes

- Mécanisme de contrôleur frontal et de routage
- Format et configuration des routes
 - Routage global dans l'application
 - La définition du routage selon les différents formats de configuration
- Implémentation des contrôleurs
 - Récupération des informations de la requête
 - Interaction avec le modèle
 - Redirections internes et externes
 - Transmissions d'information et délégation aux vues
 - Injection de service dans les contrôleurs
- Travaux pratiques :
 - Définition de la stratégie de routage dans l'application ainsi que son format de configuration
 - Génération des contrôleurs et des actions.
 - Ajustement du routage dans la configuration.
 - Implémentation des actions et lien avec les services d'accès aux données métiers

Les vues avec les Templates Twig

- Présentation de Twig et de sa syntaxe
 - Instructions et expressions
- Les notions de layout/views et de blocs
 - Organisation des fichiers
 - Principes de l'héritage de blocs
 - Composition de vues
- Fonctionnalités avancées du langage
 - Structures conditionnelles et itératives
 - Les filtres et fonctions
- Échange de données et interactions avec les contrôleurs
- Travaux pratiques :

- Création de vues Twig
- Lien avec les contrôleurs pour l'échange de données

Les formulaires

- Une librairie MVC pour créer les formulaires
- Initialisation et affichage d'un formulaire
 - Les objets Form et FormBuilder
 - Les différents types de contrôle
- Traitement et validation des données d'un formulaire
 - Le lien avec la couche modèle
 - Mise en place des contraintes de validation
 - Gestion des erreurs de validation et de leurs messages
- Utilisation des thèmes de formulaire pour la personnalisation du rendu
- Travaux pratiques:
 - Création de formulaire dans les vues nécessitant de la saisie de données
 - Implémentation de formulaires réutilisables
 - Exploration des différents types de champs
 - Ajouts de contraintes de validation sur les données saisies

La sécurité dans les applications Symfony

- L'authentification
 - Les différentes techniques d'authentification et leur configuration dans l'application
- Gestion des permissions utilisateurs
 - Utilisateur et rôles
 - Exploiter les fournisseurs d'utilisateurs
 - Le cryptage des mots de passe
- Gestion des autorisations
 - Sécuriser les accès en fonction des rôles
- Travaux pratiques:
 - Implémentation d'une stratégie de sécurité sur une partie de l'application
 - Mise en place de rôles applicatifs pour la répartition des permissions d'accès

Perspectives : La version STS en cours

- Explication du principe de dépréciation
 - Le cycle dans Symfony
 - Savoir anticiper les changements d'API
- Contraintes de portage entre les versions
- Exploration des nouvelles fonctionnalités
- Mettre en place la veille technologique
 - Surveiller les évolutions de version
 - Identifier la prochaine version LTS
 - Surveiller les évolutions fonctionnelles