DAWAN

Plan de la formation

www.dawan.fr

Formation Symfony Avancé

Durée :	5 jours (35 heures)
Tarifs inter-entreprise :	2 975,00 € HT (standard) 2 380,00 € HT (remisé)
Public :	Développeur PHP confirmé
Pré-requis :	Avoir suivi la formation Symfony initiation et approfondissement ou connaissance équivalente
Objectifs :	Découvrir les fonctions avancés de Symfony - Gérer les événements et l'automatisation de tâches - Tester et valider son application
Modalités pédagogiques, techniques et d'encadrement :	• Formation synchrone en présentiel et distanciel. • Méthodologie basée sur l'Active Learning : 75 % de pratique minimum. • Un PC par participant en présentiel, possibilité de mettre à disposition en bureau à distance un PC et l'environnement adéquat. • Un formateur expert.
Modalités d'évaluation :	• Définition des besoins et attentes des apprenants en amont de la formation. • Auto-positionnement à l'entrée et la sortie de la formation. • Suivi continu par les formateurs durant les ateliers pratiques. • Évaluation à chaud de l'adéquation au besoin professionnel des apprenants le dernier jour de formation.
Sanction :	Attestation de fin de formation mentionnant le résultat des acquis
Référence :	PHP724-F
Note de satisfaction des participants:	5,00 / 5
Contacts :	commercial@dawan.fr - 09 72 37 73 73

Modalités d'accès :	Possibilité de faire un devis en ligne (www.dawan.fr, moncompteformation.gouv.fr, maformation.fr, etc.) ou en appelant au standard.
Délais d'accès :	Variable selon le type de financement.
Accessibilité :	Si vous êtes en situation de handicap, nous sommes en mesure de vous accueillir, n'hésitez pas à nous contacter à referenthandicap@dawan.fr, nous étudierons ensemble vos besoins

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
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 composé
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

Passage de la certification (si prévue dans le financement)