

**PROGRAMME DE FORMATION**

# "**DRUPAL 8 / SYMFONY 2 DEVELOPPEMENT AVANCE**"

**Public concerné et pré-requis :**

**Développeurs Web** souhaitant exploiter Drupal 8 en comprenant l'intégration de Symfony2.

**Pré requis :**

- ✓ Connaître PHP Objet (indispensable)
- ✓ Connaître HTML et CSS (fortement recommandé)
- ✓ Connaître Drupal 8 en tant que Développeur

**Objectifs de la formation :**

- ✓ Comprendre les fondements de Drupal8 au travers de Symfony2.
- ✓ Comprendre l'injection de dépendance dans Drupal8.
- ✓ Savoir utiliser l'EventDispatcher.
- ✓ Créer son propre type de plugin.
- ✓ Créer une entité de configuration "maison".
- ✓ Importer ou exporter des données dans/hors de Drupal.

**Programme de la formation :**

**IMPORTANT :** La version installée et utilisée tout au long de la formation est **Drupal 8**.

**1) Injection de dépendances**

- ✓ Comprendre les principes de l'injection de dépendance.
- ✓ Quels sont les différents types d'injection ?
- ✓ Comment l'injection de dépendance est implémentée dans Symfony.

**2) Le container de Symfony**

- ✓ Comprendre à quoi sert le Container.
- ✓ Quels sont les principaux services de Drupal 8 ?
- ✓ Comment déclarer un nouveau service ?
- ✓ Comment remplacer un service de Drupal 8 ? Exemple de service simple à trouver.
- ✓ Concepts hérités de Symfony 2 : le container, les services...

### 3) EventDispatcher

- ✓ Découvrir les patrons de conception « Observateur » et « Médiateur »
- ✓ Savoir découpler son code avec le gestionnaire d'événements
- ✓ Connaître les événements principaux du framework Symfony
- ✓ Connecter des écouteurs à des événements du noyau de Symfony et Drupal
- ✓ Propager des événements personnalisés dans du code métier
- ✓ Créer de nouveaux types d'événements
- ✓ Enregistrer des « écouteurs » et « abonnés » d'événements

### 4) Créer son propre type de plugin

- ✓ Déclaration du plugin manager (service).
- ✓ Class DefaultPluginManager.
- ✓ Méthode de découverte (annotation par défaut).

### 5) Implémenter un plugin

- ✓ Déclaration du plugin Condition Plugin API : ajout d'une condition de visibilité sur les blocs.
- ✓ Utilisation de la Mail API.

### 6) Créer sa propre entité de contenu

- ✓ Ajouter le routing de base pour la gestion de l'entité.
- ✓ Déclarer une interface (ContentEntityInterface).
- ✓ Déclarer une classe (ContentEntityBase).
- ✓ Déclarer le schema correspondant.
- ✓ Ajouter les contrôleurs et formulaires nécessaires.
- ✓ Soumettre ses données à Views.