

## PROGRAMME DE FORMATION

# "DRUPAL 8 DEVELOPPEUR BACK-END"

### Public concerné et pré-requis :

**Développeurs Web** souhaitant exploiter Drupal comme un framework de développement et pas uniquement comme un CMS.

**Développeurs Drupal 7** souhaitant se mettre à niveau sur la dernière version 8.

#### Pré requis :

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

### Objectifs de la formation :

- ✓ Créer son propre module qui sera capable d'interagir avec Drupal 8.
- ✓ Utiliser Drupal comme un framework pour accélérer ses développements maison, en s'appuyant sur ses APIs.
- ✓ Comprendre le fonctionnement global de Drupal 8 et le container (Symfony 2).
- ✓ Adapter le fonctionnement du cœur de Drupal 8 ou de modules communautaires sans toucher à leur code source.

### Programme de la formation :

**IMPORTANT :** La version installée et utilisée tout au long de la formation est **Drupal 8**, mais les différences entre Drupal 7 et 8 seront signalées -- oralement, et dans le support de cours -- à chaque fois que ça sera opportun.

#### Introduction

- Présentation rapide de Drupal et de son jargon (node, taxonomie...).
- Présentation et installation des outils de développement (WAMP, Editeur PHP).
- Téléchargement et installation de Drupal 8.
- Installation des outils du développeur : module « devel » , Firebug.
- PHP objet : les concepts de base (classes, interfaces, injection de dépendance, PSR4, annotations...).

#### Création d'un module

- ✓ Les fichiers nécessaires.
- ✓ Structures des fichiers et autoloading.
- ✓ PSR-0 et PSR-4.
- ✓ Concepts hérités de Symfony 2 : container, services et injection de dépendance.

## Plugins

- ✓ Qu'est-ce qu'un plugin ?
- ✓ Mécanisme de découverte (fichier YAML, annotation PHP).
- ✓ Exemple de plugin et implémentation.

## Le système de routing

- ✓ Comment Drupal gère le routing.
- ✓ Comment faire correspondre une URL à une page.
- ✓ Utilisation des méthodes `create()` et `__construct()`.
- ✓ Créer des URLs dynamique.
- ✓ Ajout de liens de menu, de liens contextuels, de liens d'action et de tâche.

## Cache API

- ✓ Comment Drupal gère le cache.
- ✓ Les différents types de cache.
- ✓ Cache API : keys, contexts, tags et max-age.
- ✓ Les différents modules de cache : Internal Page Cache, Internal Dynamic Page Cache et Big Pipe.
- ✓ Auto-placeholder et lazy builders.

## Le système de bloc

- ✓ Gestion des blocs avec Drupal 8.
- ✓ Introduction aux types de bloc.
- ✓ Comment créer un bloc avec Bloc Plugin.

## La base de données

- ✓ Comprendre la couche d'abstraction BDD de Drupal.
- ✓ Entity API : gestion des entités de contenu et de configuration (EntityTypeManager et EntityQuery).
- ✓ Se connecter à la base et effectuer des requêtes simples avec la Database API.
- ✓ Enregistrer ses propres données en base avec la Schema API : fichier d'installation.

## Les noeuds (exemple d'entité de contenu)

- ✓ Qu'est-ce qu'un noeud ?
- ✓ Créer un "node module".
- ✓ Manipuler les noeuds programmatiquement.
- ✓ Intercepter les actions effectuées sur les noeuds avec les `hook_entity_XXX()`.
- ✓ Ajout d'onglet sur chaque page de noeud.

## Système de templating

- ✓ Principes de la couche de thème : séparation fonctionnel/présentation.
- ✓ TWIG
- ✓ Render Arrays.
- ✓ Déclarer son propre formattage sous forme de template.

## Formulaire

- ✓ Créer ses propres formulaires programmatiquement avec la Form API.
- ✓ Gérer la validation et la soumission d'un formulaire.
- ✓ Formulaire AJAX.
- ✓ Changer la présentation d'un formulaire.
- ✓ Modifier un formulaire généré par Drupal ou un autre module sans toucher au code source de ce formulaire (hook\_form\_alter).

## State API

- ✓ Principe du Configuration Management : fichiers YAML, import/export.
- ✓ Créer un formulaire dans le back-office de Drupal.
- ✓ Ajouter sa propre configuration : manipulation de l'objet Config.
- ✓ Traduction de configuration.

## Sécurité

- ✓ Comprendre le contrôle d'accès de Drupal.
- ✓ Protéger l'accès à un bloc.
- ✓ Protéger l'accès à une page.
- ✓ Créer son propre système de contrôle d'accès sous forme de service.