

Adobe Commerce / Magento 2 Open Source : Back End

Concepts et bonnes pratiques de développement Back Office avec Adobe Commerce / Magento 2 Open Source

Filière : **Commerce & Marketplaces** Sous-filière : **e-commerce / Marketplaces**

| RÉFÉRENCE | DURÉE | PRIX UNITAIRE HT |
|---------------|----------------------|------------------|
| MAGB01 | 5 JOURS (35H) | 2 300 € |

Description

La formation Adobe Commerce / Magento 2 Open Source Back-end est une formation technique qui permet d'acquérir les compétences nécessaires pour pouvoir comprendre et étendre les possibilités fonctionnelles de Adobe Commerce / Magento 2 Open Source. Orienté travaux pratiques, vous aborderez l'ensemble des aspects du développement sous Adobe Commerce / Magento 2 Open Source.

Objectifs pédagogiques

- Installer Adobe Commerce / Magento 2 Open Source
- Créer un module Front et BackOffice Adobe Commerce / Magento 2 Open Source en respectant les bonnes pratiques de conception, codage et de test
- Mettre en place ou enrichir les APIs Adobe Commerce / Magento 2 Open Source

Public cible

Profil à dominante technique : Lead Dev, Ingénieur de développement, Développeur junior et senior souhaitant découvrir le framework Adobe Commerce / Magento 2 Open Source

Pré-requis

Expérience en développement objet PHP et de préférence connaissant Magento 1

Modalités d'évaluation

L'évaluation des acquis se fait tout au long de la session au travers d'ateliers de mise en pratique des notions et concepts abordés pendant la formation.

OXiane Institut



Méthodes pédagogiques

50 % Théorie / 50 % Pratique

Programme de la formation

Environnement

- Environnement Magento
- Environnement de développement
- Installation de Adobe Commerce / Magento 2 Open Source
- Configuration de l'environnement de développement
- Les modes : Default, Developer, Production, Maintenance
- Arborescence des fichiers

Modules

- Présentation générale
- Qu'est-ce qui constitue un module
- Contenu minimum d'un module
- Classes d'un module : PSR-2, autoloader, dependency injection, organisation...
- Fichiers XML d'un module : validation obligatoire, cloisonnement par area
- Traductions

Instanciation des objets

- Dependency Injection
- Object Manager
- Compilateur

Routage et Controllers

- Présentation générale
- Les différents Routers disponibles
- Principe de fonctionnement d'un Router
- Controllers (alias Action Classes)
- Interprétation d'une URL module/action-path/action
- Configuration du routage pour chaque module
- Résultats possibles: page, JSON, redirection...
- Instanciation à l'intérieur d'un controller

OXiane Institut



Évènements et Observers

- Déclencher un évènement
- Déclarer un Observer

Plug-ins

- Concept et limitations
- Déclaration
- Convention de nommage

Zoom sur les tests unitaires

- Commandes: installation, exécution
- Création d'un TU

TP1

- Création d'un controller : Retour d'un JSON, Redirection, Gestion des paramètres
- Observer
- Plug-ins : Remplacement d'une méthode via plugin

Layouts

- Définition
- Les trois types de layout (page layout, configuration de page, layout générique)
- Utilisation des fichiers de layout : Conventions de nommage
- Contenus d'un layout: Containers, Blocks : Les différents types de Blocks prédéfinis
- Templates et leur utilisation
- UiComponents :
 1. Cas d'utilisation
 2. Définition
 3. UiComponents basiques et secondaires
 4. Templates des UiComponents
 5. Utilisation des UiComponents
 6. Exemples
- Références vers les Containers et Blocks existants
- Actions sur les Blocks
- Déplacement et suppression des Blocks et Containers

TP2

OXiane Institut



- Surcharge du controller créé dans le TP1
- Création d'une page avec une structure particulière : Création des Blocks avec templates
- Modification du layout créé dans un module différent

Base de données et ORM

- Models :

1. Définitions: ORM, Models, Resource Models, Collections, Resource Adapters
2. Généralités pour le CRUD
3. Liaison Model / Resource Model
4. Liaison Model / BDD
5. Liaison Collection / Model
6. Model Type Interface
7. Vue détaillée : Model, Resource Model, Collection
8. Scripts d'install et d'upgrade
9. Types de script d'install (Install et Upgrade, Schema et Data)
10. Installation d'un module
11. Exemples de scripts

- EAV (Entity-Attribute-Value) :

1. Concept
2. Méta-tables
3. Models utilisés
4. Chargement et sauvegarde (Différences dans les implémentations des classes ORM, Process de chargement, Source Models : utilisation des valeurs ou labels pour les attributs de type "liste déroulante")
5. Gestion des attributs (Deux aspects de l'EAV : Méta-information et Contenu, Types des valeurs, Classe Setup spécifique, Création d'un nouvel attribut, Modèles des attributs: Backend, Source et Frontend)
6. Entity Increment Model

TP3

- Création de deux tables via installer : "catalog_product_series" et "vendor" : Implémentation des classes ORM associées
- Création d'un attribut EAV "series" sur un produit via installer : Création d'un Source Model et Frontend Model pour cet attribut qui utilisent les données de la table "catalog_product_series"
- Création d'une table d'associations entre "vendor" et "catalog_product_entity", utilisation des Resource Models et Collections pour afficher les vendors pour un produit en front

Service Contracts, API et Services Web

OXiane Institut



- Service Contracts
- Définition, avantages, inconvénients
- Composition d'un Service Contract : Data API, Service API
- Tags dans les commentaires nécessaires pour la génération des Web Services
- Implémentation des DTO (Data Transfer Objects) : "Custom Attributes" et "Extension Attributes": déclaration et injection
- Service API : Business Logic API et Repositories
- Services Web (Publication d'un WS REST ou SOAP, Déclaration d'une méthode de webservice et mapping avec une interface Magento, REST : paramètres et requêtes via POST et PUT, SOAP: WSDL auto-généré et son URL, Gestion des ACL et authentification, Paramétrage en BO)

TP4

- Utilisation d'un Repository existant pour effectuer des recherches
- Création d'un Service Contract pour l'entité "vendor" créée précédemment
- Création d'un DTO et d'un Repository
- Publication des méthodes du Repository en tant que Webservice
- Test de l'API SOAP

Back-office

- Présentation
 - Implémentation d'un Controller BO en tenant compte des ACL
 - Éléments récurrents: grilles et formulaires: UiComponents :
1. Présentation des UiComponents dédiés: "listing" (grille) et "form" (formulaire) : Définition des composants enfants, Création des sources de données (composant dataSource et classe dataProvider), Exemples
 2. Pour les grilles : Définitions des filtres et des colonnes, Autres composants moins complexes: actions en masse, paginations etc
 3. Pour les formulaires : Déclaration des champs et leur organisation en fieldsets., Déclaration des boutons, Validation
- System Configuration :
1. Présentation de la configuration BO
 2. Déclaration d'un point de config
 3. Inclusion d'un Source Model et d'un Frontend Model
- Menu : Configuration du menu de navigation en BO
ACL :
1. Définition
 2. Organisation des ACL par rôles

OXiane Institut



-
3. Exemples de contrôle des ACL: `_isAllowed()`
 4. Configuration dans la BO
 5. Création d'ACL

TP5

- Gestion des entités précédemment créées "series" et "vendor"
- Création des grilles et formulaires
- Ajout des liens dans le menu
- Création et utilisation des ACL adaptées
- Adaptation du WS précédemment créé pour utiliser ces ACL



Programme mis à jour le **23 février 2024**