

# Akéneo POC Bundles

## CoreBundle

Le coreBundle contient toutes les entités, tout le routing et les templates pour le POC.

### Command

#### *ExportCommand.php*

Étends la classe Commande de symfony et permet de lancer l'export de produit avec la commande :

```
php app/console poc:export --env=prod --channel=<nom du channel>
```

#### *ImportCommand.php*

Lance l'import du catalogue cDiscount avec la commande :

```
Php app/console poc:import --env=prod
```

### Controller

#### *IndexAction*

Renvoie la page index du POC avec les deux boutons Import Settings et Export Settings.

#### *ImportMappingSettingsAction*

Récupère le catalog cDiscount, le masterCatalog, les CatalogMaps et traite les requêtes post pour le formulaire de marge, passe ces variables aux templates twig importMappingSettings.twig.

#### *ExportMappingSettingsAction*

Retourne les channels, la masterCategory et les channelMaps au template exportMappingSetting.twig.

#### *changeImportMapAction*

Récupère la requête AJAX envoyée par le template et met à jour la base de données.

#### *changeExportMapAction*

Récupère la requête AJAX envoyée par le template et met à jour la base de données.

### Entity

#### *CatalogMap*

Associe un label de catégorie du catalogue cDiscount à un label de catégorie du catalogue Master.

#### *ChannelCategoryMap*

Gère un lien ManyToOne entre un channel et des catégories.

#### *Margin*

Gère la valeur de la marge en base de données.

### Form

#### *MarginType*

formulaire pour changer la valeur de la marge.

## Resources

### *Config*

#### [masterCatalogs.yml](#)

Contient le nom des libellés des catalogues cDiscount et Master.

#### [Routing.yml](#)

Routing pour le bundle.

### *Views*

#### [Index.html.twig](#)

Extend le `bp.layout` d'akéneo et réutilise ces classes pour présenter deux boutons, `export settings` et `import settings`.

#### [importMappingSettings.html.twig](#)

Utilise des macros twig pour afficher toutes les catégories du catalogue cDiscount avec à côté un dropdown de toutes les catégories du catalogue Master.

Affiche également le formulaire Margin.

Utilise du jQuery pour remplir les bonnes cases au chargement et des requêtes AJAX quand un dropdown est modifié.

#### [exportMappingSettings.html.twig](#)

Utilise des macros twig pour afficher une liste de checkbox de toutes les catégories du catalogue Master pour chaque channel.

Pré rempli les bonnes checkboxes avec du jQuery et envoie les données en AJAX dès que le bouton save est actionné.

## CSVCleanerConnectorBundle

### ArrayConverter

#### *[ProductStandardConverter.php](#)*

Override le `productStandardConverter` d'akéneo pour associer les headers du csv de cdiscount avec des catégories reconnu par akéneo et télécharge les images.

### DependencyInjection

#### *[POCCsvCleanerConnectorExtension.php](#)*

Utilise le `YamlFileLoader` pour charger les fichiers YML

### Ressources

#### *Config*

#### [Array\\_converters.yml](#)

Définit un service `poc_csvcleanerconnector.array_converter.flat.product` et lui passe les arguments nécessaires.

#### [Batch\\_jobs.yml](#)

Ce fichier est utilisé par akéneo pour créer de nouvelles tâches d'import ou d'export, il définit le `reader` (celui par défaut), le `processeur` (notre processeur), et le `writer` par défaut.

### Csv.yml

Contient le mapping de tous les headers du csv cDiscount avec les attributs akéneo.

### MasterCatalogs.yml

Contient les libellés du catalogue cDiscount et master.

### Processors.yml

Définit un service qui utilise notre class ProductStandarConnector.php

### Translations

Contient les fichiers yml pour la traduction des noms de services qui s'affichent dans le dropdown du backoffice akeneo au moment de la création d'un nouveau profil d'import

### EnhancedExportConnectorBundle

Ce bundle permet de proposer des profils d'export beaucoup plus configurables.

Il utilise notamment les mappings entre les channels et les catégories pour exporter que les produits issues de catégories associés au channel sélectionné.

Il est construit comme le CsvCleanerConnectorBundle mais utilise un Reader pour trier les produits à exporter au lieu d'un Processor pour gérer l'import.

Le tri se fait dans le ProductReader.php dans la méthode applyCategorizationFilter()

Ici on récupère les ids des catégories associées avec le channel sélectionné et on exporte seulement les produits qui ont une catégorie dans cette liste.

### EnrichBundle

Ce bundle spécifique à akéneo permet de rajouter des onglets dans les différentes interfaces du BO, Ici nous utilisons le fichier vendor/pim-community-dev/src/pim/Bundle/EnrichBundle/resources/navigation.yml

pour ajouter un onglet POC qui renvoie sur l'index du CoreBundle.

Il faut pour cela rajouter les lignes :

```
poc_menu:  
  label: 'POC'  
  route: poc_home
```

À la fin des items et

```
poc_menu:  
  children:  
    poc_menu: ~
```

À la fin du tree.

- ▼ CoreBundle
  - ▼ Command
    - ExportCommand.php
    - ImportCommand.php
  - ▼ Controller
    - POCController.php
  - ▼ DependencyInjection
    - Configuration.php
    - POCCoreExtension.php
  - ▼ Entity
    - CatalogMap.php
    - CatalogMapRepository.php
    - ChannelCategoryMap.php
    - Margin.php
    - MarginRepository.php
  - ▼ Form
    - MarginType.php
  - ▼ Resources
    - ▼ config
      - masterCatalogs.yml
      - routing.yml
    - ▼ views
      - ▼ POC
        - exportMappingSettings.html.twig
        - importMappingSettings.html.twig
        - index.html.twig
  - ▶ Tests
    - POCCoreBundle.php

- ▼ CsvCleanerConnectorBundle
  - ▼ ArrayConverter
    - ProductStandardConverter.php
  - ▼ DependencyInjection
    - POCCsvCleanerConnectorExtension.php
  - ▼ Resources
    - ▼ config
      - array\_converters.yml
      - batch\_jobs.yml
      - csv.yml
      - masterCatalogs.yml
      - processors.yml
    - ▶ doc
    - ▶ public
    - ▼ translations
      - messages.en.yml
      - messages.fr.yml
  - POCCsvCleanerConnectorBundle.php

- ▼ EnhancedExportConnectorBundle
  - ▼ ArrayConverter
    - ProductStandardConverter.php
  - ▼ DependencyInjection
    - EnhancedExportConnectorExtension.php
  - ▶ Doctrine
  - ▼ Reader
    - ProductReader.php
  - ▼ Resources
    - ▼ config
      - ▶ storage\_driver
        - array\_converters.yml
        - batch\_jobs.yml
        - query\_builder.yml
        - readers.yml
    - ▶ translations