

Chapitre 4

Personnalisation

1. Traductions

1.1 Introduction

Par défaut, sans nécessités d'activer un module natif ou complémentaire, PrestaShop permet de gérer du contenu internationalisé.

Afin de pouvoir appréhender le système de traduction au sein de PrestaShop, il est important de prendre en considération quelques concepts clés.

Tout d'abord, nous avons la phrase à traduire – usuellement écrite en anglais – que l'on nomme le message (*wording*).

À celle-ci est associé un domaine de traductions (*Translation domain*) afin de pouvoir organiser les traductions en fonction du contexte souhaité.

Le catalogue (*Message catalogue*) comprend quant à lui l'ensemble des éléments traduits. On dispose d'un catalogue par langue.

À ce catalogue est associée une ressource (*Catalogue resource*) déterminant le procédé avec lequel les données sont obtenues. Ce procédé peut être un fichier ou encore la base de données, par exemple.

Enfin, l'utilisation d'un service permettant de mettre le tout en relation est de mise, il s'agit du service de traduction (*Translator*).

Celui-ci s'appuie sur le service fourni par défaut dans Symfony.

1.1.1 Les différentes ressources de catalogue

Lors de son initialisation, le service de traduction charge plusieurs ressources permettant de traduire la phrase souhaitée en fonction du contexte :

- Les traductions personnalisées encodées en base de données – dans la table `translations`.
- Les traductions du thème actif – les fichiers XLF présents dans `themes/yourtheme/translations`.
- Les traductions du cœur – les fichiers XLF présents dans `app/Resources/translations`.
- Les traductions issues des modules actifs – les fichiers XLF présents dans `modules/*/translations`.
- Les traductions issues des modules actifs selon la méthode legacy – les fichiers PHP présents dans `modules/*/translations`.

■ Remarque

Le chargement des ressources est effectué par ordre de priorité. La première chargée prévaut sur les suivantes.

1.1.2 Le format XLIFF

Les fichiers représentant les ressources de catalogue sont au format XLF. Il s'agit de fichiers XLIFF (*XML Localization Interchange File Format*).

S'appuyant sur le format XML, il s'agit d'un langage de balisage créé pour standardiser les échanges liés à la localisation.

Ce format est celui utilisé par Symfony pour son système de traduction par défaut.

1.2 Traduction des messages issus du cœur

Lorsque vous souhaitez traduire les messages qui sont par défaut fournis dans PrestaShop et pour lesquels le pack de localisation permet une traduction automatique (a contrario des modules tiers, par exemple), vous devez vous rendre dans la traduction des thèmes et choisir le thème par défaut (nommé « Classic »).

■ Remarque

Ceci est valable même si vous n'utilisez pas le thème par défaut.

1.3 Traduction d'un thème front-office

Ce point est vu plus amplement dans le chapitre Thèmes.

1.4 Domaines

Lors de l'utilisation du service de traduction, deux données sont passées à ce dernier : le message et le domaine de traductions.

■ Remarque

Il est également possible de lui transmettre des paramètres ainsi que la locale souhaitée.

Ainsi, l'appel de code suivant a pour but de demander au service de traduction de traduire notre message (*Hello World !*) en fonction de notre domaine (*Modules.Eniblog.Test*).

```
$translator->trans(  
    'Hello World !',  
    'Modules.Eniblog.Test'  
);
```

Pour ce faire, le service vérifie tout d'abord si la traduction ne se trouve pas déjà en cache.

Si celle-ci n'est pas trouvée, il poursuit sa recherche au sein de la table des traductions personnalisées.

Pour finir, ce sont les ressources de catalogue que sont les fichiers de traductions qui sont utilisés selon l'ordre de priorité défini dans la section précédente.

Lorsque le message ne peut être traduit via les précédents catalogues, le service de traduction peut se référer aux traductions effectuées en mode legacy – via les fichiers PHP – dès le moment où le domaine associé à la demande débute par Modules.

Il est impératif que le sous-domaine utilisé soit le nom du module (qui est remis en lettres minuscules automatiquement) lui-même.

La troisième partie du domaine permet de définir le fichier source dans lequel le message apparaît.

Le code illustré précédemment équivaldrait alors à celui-ci en matière de code legacy :

```
$this->l(
    'Hello World',
    'Test'
);
```

■ Remarque

À noter qu'il est donc possible d'utiliser une traduction issue d'un module dans un second module sans que celui-ci ne nécessite d'embarquer la traduction souhaitée.

1.4.1 Les domaines existants

Les domaines de traduction permet d'obtenir une organisation plus précise et d'associer un contexte particulier à un même message. PrestaShop s'appuie sur cinq principaux domaines pour l'organisation des traductions :

- **Install** : concerne les traductions utilisées lors du processus d'installation.
- **Shop** : pour les traductions de messages présentés en front-office (thème).
- **Admin** : pour les traductions de messages présentés en back-office.

- **Modules** : concerne les traductions des modules.
- **E-mails** : pour les traductions issues des e-mails.

Selon la nature du domaine principal, on peut y retrouver un à plusieurs sous-domaines selon la nécessité. Il se peut également qu'un domaine ne dispose pas de sous-domaine. C'est le cas concernant **Install**.

Remarque

Le nombre de parties composant un domaine n'excède jamais trois.

1.4.2 Les sous-domaines

Voici une liste des sous-domaines utilisés pour l'ensemble des domaines prévus par PrestaShop nativement.

Install

Ce domaine ne dispose pas de sous-domaine.

Shop

- Pdf
- Theme
 - Catalog
 - Checkout
 - Customeraccount
 - Actions
- Demo
 - Catalog
- Notification
 - Error
 - Success
 - Warning
 - Info

- Forms
- Labels
- Errors

Admin

- Global
- Actions
- Notifications
 - Error
 - Success
 - Warning
 - Info
- Navigation
 - Menu
 - Header
 - Footer
- Ordercustomers
 - Feature
 - Notification
 - Help
- Catalog
 - Feature
 - Notification
 - Help
- Modules
 - Feature
 - Notification
 - Help
- Design
 - Feature
 - Notification
 - Help

- Shipping
 - Feature
 - Notification
 - Help
- Payment
 - Feature
 - Notification
 - Help
- International
 - Feature
 - Notification
 - Help
- Shoppparameters
 - Feature
 - Notification
 - Help
- Advparameters
 - Feature
 - Notification
 - Help

Modules

La deuxième partie de ce domaine particulier équivaut au nom du module concerné par ce dernier. Pour l'exemple, il s'agit de `Pagesnotfound` pour le module *pagesnotfound*.

La troisième partie peut être utilisée pour déterminer la provenance du message et on utilise principalement **Admin** ou **Shop**.

Cette troisième partie peut également être renseignée comme étant le nom du fichier source où se situe le message à traduire.