

Prestashop - Développement d'un module

David Annebicque

Novembre 2017

1 Objectif de cette troisième partie sur Prestashop

Dans cette partie nous allons améliorer notre module en intégrant du CSS, des librairies annexes et des traductions.

2 Pré-requis

2.1 Séance 2

Il est nécessaire que la séance 2 soit fonctionnelle pour poursuivre sur ce module.

3 CSS et Librairies annexes

Pour utiliser des librairies JS ou CSS dans prestashop, la démarche consiste à déposer les fichiers dans notre modules.

Le CSS et le JS ainsi associé à notre module ne doit concerner que ce dernier. Si le CSS (ou le js) doit avoir un impact sur d'autres éléments de la boutique, alors il faut passer par le thème.

On est libre de choisir l'organisation de son répertoire. La pratique veut malgré tout que l'on ajoute un répertoire js et un répertoire css à la racine de notre module.

Dans l'exemple ci-dessous nous installons une librairie JS/CSS pour gérer des graphiques <http://chartjs.org/>. Nous ajouterons également un second fichier JS pour gérer notre graphique.

Dans notre controleur nous devons donc utiliser la méthode setMedia qui permet d'ajouter des fichiers CSS et JS de manière temporaire.

Le code pourrait ressembler au code ci-dessous.

```
1 <?php
2 public function setMedia()
3 {
4     parent::setMedia();
5
6     $this->context->controller->registerStylesheet (
7         'module-da_commentaires-style',
8         'modules/'. $this->module->name. '/css/fichier',
9         [
10            'media' => 'all',
11            'priority' => 200,
12        ]
13    );
14
15    $this->context->controller->registerJavascript (
16        'module-da_commentaires-chartjs',
17        'modules/'. $this->module->name. '/js/fichier.js',
18        [
19            'priority' => 200,
20            'position' => 'bottom'
21        ]
22    );
23
24 }
```

Comme vous le constatez, il existe deux méthodes, associés au context, pour définir le CSS ou le JS. On peut l'appeler autant de fois que nécessaire. Il convient de donner un nom à votre ressource (ligne 7 et 16). Ensuite il faut définir, selon les cas

- le "media" (en css) : c'est à dire à quoi s'applique la feuille de style
- la "priority" (en css et js) : pour définir l'ordre.
- la "position" (en js) : pour définir si le JS doit être dans le head ou en bas du fichier

Il ne vous reste plus qu'à exploiter la librairie pour définir vos graphique. Il est nécessaire de définir un autre fichier js contenant votre code.

4 Localisation de votre module

Prestashop permet de gérer les traductions dans vos templates. Théoriquement il est maintenant possible de gérer les traductions soit selon la

méthode de Symfony, soit selon celle d'origine de Prestashop. Nous verrons ici la méthode traditionnelle.

4.1 Dans vos templates

Le principe de traduction est d'écrire tout le site en Anglais, et de définir que chaque "texte" doit être traduit. En procédant comme cela, on s'assure que par défaut, il y aura au minimum le texte en anglais.

Pour cela, il faut utiliser dans les templates l'instruction "l", qui prend deux paramètres obligatoires "s" et "mod".

- "s" permet de définir la phrase à traduire (en anglais par défaut)
- "mod" permet de définir le nom du module. Cela facilite, entre-autre, la traduction, car nos phrases sont regroupées sous notre module dans le back-office. Cette possibilité peut aussi permettre de ré-exploiter une traduction déjà existante dans un autre module.

Un exemple ci-dessous :

```
1 {l s='Statistiques !' mod='da_commentaires'}
```

Si vous souhaitez passer des paramètres dans vos textes, il faut utiliser une notation particulière, indiquant quelles sont les données à passer, et les valeurs associées.

Exemple ci-dessous, d'une phrase affichant le nombre de produits

```
1 <p>{l s='There is %d products on the website'  
↪ sprintf=$nbProducts mod='da_commentaires' } </p>
```

- %d : indique qu'il faut mettre une valeur numérique (on peut aussi utiliser %s pour une valeur de type texte)
- sprintf : permet de fixer la valeur de l'argument. La variable doit être un tableau, et les éléments sont pris dans l'ordre.

4.2 Dans le back office

Une fois toutes les traductions mise en place, vous devez aller dans le back office de Prestashop pour ajouter les traductions en fonction des langues activées sur le site.

Pour cela, vous devez aller dans "international" > "Traductions".

Choisissez ce que vous souhaitez traduire dans le premier bloc de la page :

Vous devriez vous apparaître un formulaire avec les champs de traduction présent dans votre module. Il ne vous reste plus qu'à saisir la correspondance dans la langue sélectionnée.

MODIFIER LES TRADUCTIONS

Vous pouvez modifier les traductions de chacune des portions de texte de PrestaShop.
 Choisissez une section (comme "Back-office" ou "Modules installés"), puis choisissez la langue que vous souhaitez modifier.

Type de traduction: Traductions des modules installés

Choisissez votre module: DA Commentaires

Choisissez votre langue: Français (French)

FIGURE 1 – Back-Office Traduction

DISPLAY 1 EXPRESSIONS

Statistiques 1

Annuler Enregistrer et rester Enregistrer

FIGURE 2 – Formulaire de traduction

5 Travail à faire

- Continuer d’afficher des statistiques sur la boutique, en exploitant la librairie de graphique (vous pouvez choisir une autre librairie que GraphJs)
- Gérer le module pour qu’il soit en plusieurs langues (définissez le français, l’anglais et une troisième langue de votre choix)