

Prestashop - Développement d'un module

David Annebicque

Janvier 2017

1 Objectif des dernières séances sur Prestashop

Développement d'un module pour la gestion de commentaire (inutile d'en récupérer un déjà fait, le module demandé est très simpliste). Vous pouvez récupérer un exemple de code du précédent module directement sur <https://github.com/Dannebicque/prestashop-lpmim>.

L'évaluation portera sur le module de commentaires

2 Objectifs du module de commentaires

Créer un module sur la page d'un produit qui permet de lister et d'ajouter un commentaire. On fera une gestion basique, dans un premier temps, où on ne demandera que le pseudo et le commentaire au visiteur.

Pour ce faire, il va falloir créer un nouveau module, et modifier le template de *product.tpl* afin d'ajouter le *hook* correspondant à notre module.

Normalement, il faudrait créer un thème et ne pas modifier le thème par défaut.

```
1 <?php
2 {hook h='nomDeVotreHook' product=$product }
```

Ajoutez la ligne ci-dessus à l'endroit où vous souhaitez afficher la partie commentaire. A noter que l'on passe la variable du produit dans cet appel. Ainsi, notre module sera "exécuté" à cet endroit, et on pourra récupérer les commentaires, afficher le formulaire et traiter les messages postés, en rapport avec le produit spécifié.

3 Création d'une table pour notre module

Pour ce module nous allons avoir besoin d'une table permettant de sauvegarder les messages. Pour cela, nous devons définir lors de l'installation du module la table à créer.

Dans le code des séances précédentes nous utilisons une table déjà existante (la table configuration) pour sauvegarder des clés en rapport avec un module. Ici nous souhaitons une vraie table, afin de sauvegarder les commentaires.

Cela se fait assez simplement avec Prestashop, puisqu'il suffit d'écrire le code SQL.

```
1 <?php
2 public function installDB()
3 {
4     $return = true;
5     $return &= Db::getInstance()->execute('
6         ICI votre requête SQL de création
7     ');
8
9     return $return;
10 }
```

A noter qu'il faut manipuler le préfixe choisit par l'utilisateur lors de l'installation, pour cela, on peut écrire :

```
1 <?php
2 $table = _DB_PREFIX_.'nomdelatable';
```

Cette méthode *installDB* doit être appelée dans la méthode *install* du module en même temps que l'association du module aux différents *hook*. D'ailleurs n'oubliez pas d'enregistrer le hook (*registerHook*) que vous avez défini précédemment dans le template *product.tpl*.

Il faudra ajouter le code sur la méthode de désinstallation qui permettra de supprimer la table précédemment créée, en suivant le même principe.

4 Récupération des commentaires liés à un produit

Dans le template *product.tpl*, le produit est passé en paramètre, il est donc possible de le récupérer dans la méthode décrivant le fonctionnement

de votre *hook*. Le paramètre *param* est un tableau associatif qui contient, entre autre, un clé "produit".

```
1 <?php
2 public function hookNomdeVot reHook ($param)
3 {
4
5 }
```

5 Affichage du formulaire

L'affichage du formulaire se fait de manière très classique, en l'écrivant, en HTML, directement dans le template du hook. Pensez aux traductions.

Il faut également veillez à construire l'URL du produit, afin de pouvoir définir la propriété ACTION de votre formulaire. Pour cela vous pouvez utiliser :

```
1 <?php
2 $link = new Link();
3 $url = $link->getProductLink(iddevotreproduit);
```

6 Traitement du formulaire

C'est la méthode qui affiche le hook, qui va aussi traiter les informations soumises du formulaire. Pour cela, il faut détecter que le bouton submit est cliqué. Le helper "Tools" permet la manipulation de formulaire, notamment avec les méthodes `isSubmit('nomduchamp')` et `getValue('nomduchamp')`;

L'ajout des informations se fait en écrivant la requête SQL appropriée et en l'exécutant.

- `Db::getInstance()->execute(requete)` : exécute une requête sans retourner de résultat
- `Db::getInstance()->executeS(requete)` : exécute une requête et retourne un tableau contenant les éléments.

7 Critères d'évaluations du module

Les fonctionnalités seront évaluées, ainsi que l'aspect esthétique de votre affichage dans la page produit.

- Possibilité d'ajouter et de lister les commentaires pour un produit
- Aspect "esthétique" de la partie commentaire
- Module en plusieurs langues
- Installation / désinstallation fonctionnelle
- Back office pour gérer les messages (modération) - Vue lors de la dernière séance

Un respect strict des consignes entrainera une note sur 14 points. Les 6 derniers points seront attribués pour les améliorations que vous apporterez.