



PRESTASHOP

Documentation Technique

v 1.0

Documentation

Documentation technique de la solution e-commerce open-source PrestaShop

DEFINITIONS	4
INTRODUCTION GENERALE ET PHILOSOPHIE DU PROJET.....	4
PRE-REQUIS TECHNIQUES.....	5
INSTALLER PRESTASHOP	5
INSTALLATION LOCALE OU DISTANTE ?	5
TELECHARGER PRESTASHOP	5
COPIER LES FICHIERS D'INSTALLATION	6
VERIFIER LES PERMISSIONS	6
CREER LA BASE DE DONNEES	6
SENSIBILISATION A LA SAUVEGARDE	6
LANCER L'INSTALLATION	7
<i>Etape 1 – Bienvenue</i>	7
Dans quelle langue souhaitez-vous effectuer l'installation ?	7
Méthode d'installation	7
<i>Etape 2 – Compatibilité système</i>	7
Merci de vous assurer que chacun des paramètres ci-dessous est validé	7
Paramètres optionnels	8
<i>Etape 3 – Configuration système</i>	8
Configuration de la base de données	8
Paramètres d'envoi des emails	10
<i>Etape 4 – Configuration boutique</i>	10
Nom de la boutique	10
Pays par défaut	10
Logo de la boutique.....	10
Prénom.....	10
Nom.....	10
Adresse e-mail.....	10
Mot de passe de la boutique.....	10
Confirmez le mot de passe.....	10
Recevoir mes informations par e-mail	10
<i>Etape 5 – Installation terminée !</i>	11
PERSONNALISEZ VOTRE BOUTIQUE	12
MISE A JOUR DE PRESTASHOP.....	13
PRECAUTIONS.....	13
SAUVEGARDE	13
<i>Sauvegarde de vos traductions</i>	13
<i>Sauvegarde de votre base de données</i>	13
<i>Sauvegarde de vos fichiers PrestaShop</i>	14
INSTALLATION DE LA NOUVELLE VERSION	14
Téléchargement	14
Extraction et copie.....	14
Restauration des données depuis l'ancienne version	14
Installer la mise à jour	14
ARCHITECTURE GENERALE	15
GRANDS PRINCIPES (MODULES, THEMES)	15
MODULES.....	15
Le concept	15
Fonctionnement technique d'un module.....	16
Liste des points d'accroche (hooks) de PrestaShop	16

THEMES	21
<i>Le concept</i>	21
<i>Fonctionnement technique d'un thème</i>	21
<i>La gestion des traductions</i>	22
<i>Transmettre des informations (données) à un thème PrestaShop</i>	22
L'ARBORESCENCE DE PRESTASHOP	23
BONNES PRATIQUES A RESPECTER (DEVELOPPEURS, INTEGRATEURS)	24
LA SECURISATION DE VOTRE INSTALLATION PRESTASHOP	24
OPTIONS DE CONFIGURATION ET FINE-TUNING/PERFORMANCES	25
FICHIER CONFIG.INC.PHP	25
FICHIER DEFINES.INC.PHP	25
FICHIER SMARTY.CONFIG.INC.PHP	25
AMELIORATION DES PERFORMANCES DE PRESTASHOP	25
TRAVAUX PRATIQUES	26
CREER UN THEME PRESTASHOP	26
<i>Thème par défaut PrestaShop</i>	26
<i>Créer un nouveau thème</i>	26
<i>Arborescence d'un thème</i>	26
<i>Thème des modules</i>	28
<i>Conseils pour l'édition d'un thème PrestaShop</i>	28
Firebug	28
JavaScript	28
Fichier preview.jpg	28
Normes W3C et compatibilité inter-navigateurs	29
CREER UN MODULE PRESTASHOP	29
<i>Principe des modules</i>	29
<i>Arborescence des modules</i>	29
<i>Structure de base d'un module</i>	29
<i>Configuration d'un module</i>	31
<i>Greffe d'un module</i>	32
<i>Affichage d'un module</i>	33
<i>Utilisation de Smarty</i>	36
<i>Traduction des modules</i>	37
CREER UN ONGLET DANS LE BACK-OFFICE, AINSI QUE SA CLASSE	39
EN CAS DE PROBLEME	40
FORUM OFFICIEL PRESTASHOP	40
OUTIL DE RAPPORTS DE BUGS : LE "BUG TRACKER"	40
POSTER SUR LE FORUM	41
SOUMETTRE UN BUG DANS LE BUG TRACKER	41
SITES OFFICIELS PRESTASHOP	41

Documentation

Documentation technique de la solution e-commerce open-source PrestaShop

Définitions

Back Office / BO / Panneau d'administration

Le **Back Office (BO)** est la partie d'un site permettant à l'administrateur de le gérer, par opposition au Front Office qui constitue l'interface avec l'utilisateur. Pour un site d'e-commerce, on peut également parler d'arrière-boutique.

Le commerçant peut y gérer ses produits, les clients, les commandes, le paramétrage, etc.

Front office / FO

Le **Front Office (FO)** est la partie d'un site destinée aux utilisateurs, c'est-à-dire aux clients pour un site d'e-commerce.

L'interface destinée à l'administrateur est quant à elle appelée Back Office.

Open-Source

La désignation Open-Source s'applique aux logiciels dont la licence respecte des critères précisément établis par l'Open Source Initiative, c'est-à-dire la possibilité de libre redistribution, d'accès au code source et de travaux dérivés.

Introduction générale et philosophie du projet

PrestaShop est un logiciel e-commerce Open-source qui est téléchargeable gratuitement depuis l'adresse Internet <http://www.prestashop.com>.

Le logiciel est édité par une société française basée à Paris, du nom de PrestaShop également. Une équipe de quinze personnes travaille sans relâche pour fournir un logiciel de qualité, le plus complet possible sur les plans fonctionnels et techniques.

Le projet est financé par des services spécifiques aux marchands (personnalisation graphique, développements techniques...) et également par deux projets :

Le projet est financé par :

- © des services spécifiques aux marchands (personnalisation graphique, développements techniques...),
- © des formations de 4 types :
 - © développement (PHP, MySQL) avec PrestaShop,
 - © intégration (XHTML, CSS, Smarty) avec PrestaShop,
 - © optimisation de son référencement avec PrestaShop,
 - © utilisation du Back-office PrestaShop,
- © et également par deux projets :
 - © www.prestastore.com
La boutique officielle pour PrestaShop (thèmes, modules, logos...).
 - © www.prestabox.com
Une solution d'hébergement clé en main pour le logiciel PrestaShop.

Les maîtres mots du projet PrestaShop sont :

- © Simplicité

- © Efficacité
- © Souplesse
- © Rapidité

Chaque utilisateur du logiciel PrestaShop peut contribuer selon ses envies et ses moyens :

- © En traduisant le logiciel dans de nouvelles langues,
- © En effectuant un don permettant de faire progresser le projet (<http://www.prestashop.com/fr/donate/>),
- © En participant au forum communautaire (<http://www.prestashop.com/forums/>),
- © ...

PrestaShop regroupe aujourd'hui plus de 40 000 marchands à travers le monde et progresse à grands pas, fort d'une communauté très active de plus de 100 000 membres.

Pré-requis techniques

PrestaShop est un logiciel conçu avec le langage de programmation PHP 5 et le système de base de données MySQL 5.

Pour l'installer vous aurez besoin d'un serveur web (Apache, IIS), vous pouvez également utiliser un système "tout-en-un" tel que WAMP, MAMP ou encore EasyPHP.

Ces éléments techniques sont supportés par 99% des hébergeurs Internet, et vous parviendrez donc à installer PrestaShop facilement quel que soit votre hébergeur.

Installer PrestaShop

Installation locale ou distante ?

Pour que votre boutique soit accessible depuis Internet, elle devra être installée sur un serveur distant, par exemple chez un hébergeur. Dans ce cas, vérifiez avant de souscrire un abonnement que celui-ci propose bien tous les pré-requis techniques cités plus haut. L'hébergeur mettra à votre disposition tous les outils nécessaires à la gestion de votre site (accès FTP, serveur HTTP, outil de gestion des bases de données MySQL tel que PHPMYAdmin, etc.).

Néanmoins, avant de se lancer et d'investir dans un tel abonnement, on peut vouloir tester PrestaShop sur sa propre machine (Mac, PC, etc.). Ceci est réalisable rapidement grâce à des logiciels qui permettent de disposer très simplement des pré-requis techniques.

Ces logiciels "tout-en-un" sont disponibles tant sous Mac, que sous Windows ou Linux.

Voici une liste non exhaustive de tels logiciels :

Système d'exploitation	Logiciel	URL
Linux	XAMPP	http://www.apachefriends.org/fr/xampp.html
Mac	MAMP	http://www.mamp.info/en/index.html
Windows	WAMP	http://www.wampserver.com/
	EasyPHP	http://www.easyphp.org/

Pour plus d'informations sur l'utilisation de ces programmes, vous pouvez consulter leur documentation respective.

Télécharger PrestaShop

La dernière archive stable de la solution PrestaShop peut être téléchargée à l'adresse :

<http://www.prestashop.com/fr/downloads/>

Les versions réputées stables à utiliser en production comportent le terme "final".

D'autres versions en cours de développement peuvent être mises à disposition sur cette page dans un but de tests. Elles sont alors identifiées par les mots "alpha", "beta" ou "RC". Il est **vivement déconseillé d'utiliser ces versions en production**.

Copier les fichiers d'installation

Une fois l'archive téléchargée, en extraire le contenu. Vous obtenez un dossier [prestashop](#) renfermant tous les dossiers et fichiers de la solution. Copiez l'ensemble de ces éléments sur votre site, soit à la racine, soit dans un sous-dossier.

Il est conseillé de garder précieusement l'archive téléchargée ayant servi à l'installation de votre boutique. En effet, en cas de problème majeur et en l'absence de sauvegardes, vous pourriez en avoir besoin ultérieurement pour réinstaller ou restaurer un fichier corrompu.

Vérifier les permissions

Assurez-vous que les dossiers suivants aient bien les permissions en lecture/écriture, également appelées CHMOD 666, de façon **non réursive**, c'est-à-dire appliquées au dossier lui-même mais pas aux sous-dossiers :

[/config](#)
[/download](#)
[/tools/smarty/compile](#)
[/upload](#)

Assurez-vous que les dossiers suivants aient bien les permissions en lecture/écriture, de façon **réursive** :

[/img](#)
[/mails](#)
[/modules](#)
[/themes/prestashop/lang](#)
[/translations](#)

Créer la base de données

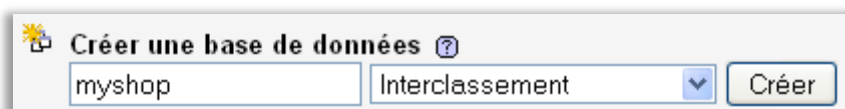
Lors de la validation de la troisième étape de l'installation de PrestaShop, les tables nécessaires au fonctionnement de la solution seront créées au sein de la base de données. Il est donc nécessaire de créer la base de données avant cette étape.

Le plus simple est de créer la base de données avec l'outil fourni par votre hébergeur (par exemple PHPMyAdmin) dès maintenant.

Exemple de création de la base de données, en local, avec WAMP

Pour créer une nouvelle base de données avec WAMP :

- © assurez-vous que les serveurs Apache et MySQL sont lancés,
- © connectez-vous à l'adresse <http://127.0.0.1/phpmyadmin/> depuis votre navigateur,
- © dans le champ **Créer une base de données**, tapez le nom de la base de données à créer (sans espace ni caractères spéciaux). Par exemple : `myshop`,



- © cliquez sur le bouton **Créer**.

La base de données est maintenant créée. Retenez le nom de la base de données, il devra être saisi à l'étape 3 de l'installation de PrestaShop.

Sensibilisation à la sauvegarde

Parce que l'on n'est jamais à l'abri d'un incident (mauvaise manipulation, virus, altération de données, etc.), il est judicieux de faire le plus souvent possible des sauvegardes de sa boutique.

Une boutique PrestaShop est constituée d'une part de **dossiers et fichiers** (ceux constituant la solution, mais aussi les images ajoutées par la suite, les fichiers mis à disposition pour le téléchargement ou uploadés par le client, etc.) et d'autre part, d'une **base de données** qui renferme les informations telles que descriptions des produits, comptes clients, commandes, etc.

Il faudra donc bien prendre soin de sauvegarder ces deux parties, pour ne pas risquer de perdre la moitié de sa boutique.

Lancer l'installation

Lancez l'installation en accédant, depuis votre navigateur, au dossier contenant vos fichiers PrestaShop. Par exemple saisir www.myshop.com s'ils se trouvent à la racine de votre site (ou localhost en local), et www.myshop.com/shop/ s'ils se trouvent dans un sous-dossier [shop](http://localhost/shop/) (ou localhost/shop/ en local).

Etape 1 – Bienvenue

Dans quelle langue souhaitez-vous effectuer l'installation ?

Choisissez la langue qui sera utilisée dans l'assistant d'installation. L'assistant d'installation PrestaShop est disponible dans trois langues : anglais, français, espagnol.



Ce choix détermine également quelle sera la langue par défaut du Front Office et du Back Office après l'installation.

Des packs de langues vous permettant de traduire à la fois le Front Office et le Back Office de votre boutique sont disponibles gratuitement à l'adresse :

<http://www.prestashop.com/fr/downloads/>

Méthode d'installation

Pour lancer une installation neuve de PrestaShop, cochez **Installation complète de PrestaShop**.

Pour effectuer une mise à jour d'une boutique PrestaShop existante vers une version plus récente, cochez **Mise à jour**.

Si l'assistant d'installation ne détecte pas de précédente installation, seule l'installation complète de PrestaShop est proposée.



Cliquez sur **Suivant** pour valider cette étape et passer à la suivante.

Etape 2 – Compatibilité système

Merci de vous assurer que chacun des paramètres ci-dessous est validé

Tous les éléments de cette liste doivent être précédés d'une coche verte.

Paramètres PHP :


- ✓ PHP 5.0 ou supérieur installé
- ✓ Envoi de fichier autorisé
- ✓ Création de nouveaux dossiers et fichiers autorisée
- ✓ Librairie GD installée
- ✓ Le support de MySQL est activé

Droits en écriture sur les fichiers et dossiers:

- ✓ /config
- ✓ /tools/smarty/compile
- ✓ /sitemap.xml

Droits en écriture sur les dossiers (et leurs sous-dossiers) :

- ✓ /img
- ✓ /mails
- ✓ /modules
- ✓ /themes/prestashop/lang
- ✓ /translations
- ✓ /upload
- ✓ /download


Si un panneau rouge  apparaît, c'est que la condition correspondante n'est pas respectée ; il faut alors corriger le problème. Les moyens d'y parvenir peuvent être différents d'un hébergeur à l'autre. Recherchez sur notre forum (<http://www.prestashop.com/forums>) les solutions les plus adaptées à votre cas.


Paramètres optionnels

Tous les éléments de cette liste sont optionnels, mais il est recommandé de faire en sorte qu'ils soient validés.

Paramètres PHP :

- ✓ Ouverture des URL externes autorisée
- ✓ L'option PHP "register global" est désactivée (recommandé)
- ✓ La compression GZIP est activée (recommandé)

Si après avoir résolu l'un des problèmes rencontrés vous voulez revérifier votre configuration, cliquez sur le bouton .

Cliquez sur  pour valider cette étape et passer à la suivante.

Etape 3 – Configuration système

Pour cette étape, la base de données doit avoir été créée pour que les tables puissent y être générées. Voir le paragraphe "Créer la base de données" ci-dessus pour plus de détails.

Configuration de la base de données

Vous devez au préalable créer une base de données (aide disponible dans le fichier readme.txt)

Serveur :	<input type="text" value="localhost"/>
Nom de la base :	<input type="text" value="myshop"/>
Identifiant :	<input type="text" value="root"/>
Mot de passe :	<input type="password"/>

Les quatre premiers champs **Serveur**, **Nom de la base**, **Identifiant** et **Mot de passe** permettent de paramétrer l'accès à la base de données précédemment créée.


Dans le cas d'une boutique installée en local et de l'utilisation d'un logiciel "tout-en-un" du type LAMP (EasyPHP, WAMP, MAMP, etc.) consultez la documentation officielle de celui-ci pour savoir quels paramètres spécifier.

Dans le cas d'une installation chez un hébergeur, celui-ci a dû vous communiquer tous les paramètres nécessaires (par e-mail ou dans le panneau d'administration de votre compte).

Tester la connexion SQL

Le bouton **Tester la connexion SQL** permet de vérifier la bonne connexion à la base de données. Il n'est pas possible de passer à l'étape suivante tant que ce test échoue.

Si l'accès au serveur n'est pas possible (serveur non lancé, nom de serveur spécifié incorrect, identifiant ou mot de passe incorrects), le message suivant s'affiche :

 **Le serveur de bases de données n'a pas été trouvé, merci de vérifier vos identifiants ou le nom du serveur.**

Si la base de données n'a pas été trouvée (base de données non créée, nom de base de données spécifié incorrect), le message suivant s'affiche :

 **Le serveur de bases de données est disponible mais la base de données n'a pas été trouvée**

Si les paramètres entrés sont corrects, le test sera validé par le message :

 **La base de données a été trouvée !**

Le champ **Préfixe des tables** permet de spécifier le préfixe qui sera affecté au nom de toutes les tables créées par l'outil d'installation PrestaShop dans la base de données. Ce préfixe permet de différencier facilement les tables de plusieurs applications qui utiliseraient cette même base de données.

Il est ensuite proposé de choisir entre 2 modes d'installation.

- ☒ **Mode complet** : installe les principaux modules ainsi que des produits de test
- ☐ **Mode simple** : aucun module installé

Mode complet

Les tables de la base sont créées, les modules essentiels installés et des données d'exemples (catégories, produits, clients, commandes, etc.) sont ajoutées à la base. Idéal pour la réalisation de tests !

Mode simple

Les tables de la base de données sont créées, mais aucun produit exemple n'est inclus et aucun module n'est installé. Après l'installation, le Front Office apparaîtra presque vide. **C'est bien entendu le mode à utiliser lorsque l'on installe sa boutique définitive**, afin qu'elle ne soit pas "polluée" par les données d'exemples de

l'autre mode !

Paramètres d'envoi des emails

Pour tester si le paramétrage d'envoi des e-mails est fonctionnel, il est possible de demander l'envoi d'un e-mail de test en spécifiant une adresse e-mail dans le champ situé en bas de page et en cliquant sur le bouton

M'envoyer un email de test

Par défaut, la fonction *mail()* de PHP est utilisée pour l'envoi d'e-mails.

Les utilisateurs plus avertis pourront spécifier leurs propres paramètres en cochant la case **Configurer l'envoi SMTP** et en remplissant les champs qui s'affichent alors.

Suivant

Cliquez sur **Suivant** pour valider cette étape. L'outil d'installation procède alors à la création des tables nécessaires au fonctionnement de la solution PrestaShop avant de passer à l'étape suivante. Ceci peut prendre quelques instants.

Etape 4 – Configuration boutique

Nom de la boutique

Saisissez le nom de votre nouvelle boutique PrestaShop. Ce nom sera utilisé dans la balise <title> et apparaîtra donc dans la barre système du navigateur des visiteurs.

Pays par défaut

Choisissez quel sera le pays utilisé par défaut dans votre boutique.

Logo de la boutique

Cliquez sur **Parcourir...** pour choisir sur votre disque dur l'image correspondant au logo de votre boutique. Ce champ est optionnel et vous pourrez ajouter votre logo plus tard, depuis le Back Office (**Back Office >> Préférences >> Apparence**).

Prénom

Prénom de l'administrateur de la boutique.

Nom

Nom de l'administrateur de la boutique.

Adresse e-mail

Adresse e-mail de l'administrateur de la boutique. Cette adresse sera utilisée comme identifiant de connexion au panneau d'administration de la boutique.

Mot de passe de la boutique

Mot de passe de l'administrateur de la boutique. Celui-ci doit comporter au minimum 8 caractères, uniquement des lettres, des chiffres ou des tirets : - _

Confirmez le mot de passe

Saisissez à nouveau le mot de passe de l'administrateur de la boutique pour vérification.

Recevoir mes informations par e-mail

Si vous désirez recevoir vos identifiants de connexion par e-mail, cochez cette case.

Etape 5 – Installation terminée !

Cette dernière étape vous récapitule quelques informations fournies précédemment (nom de la boutique, nom, prénom et e-mail de l'administrateur).

Vous pouvez d'ores et déjà accéder à votre boutique en cliquant sur le cadre **Front Office**. Si vous avez choisi l'installation simple, celle-ci peut avoir l'aspect d'une page presque vide.



Un clic sur le cadre **Back Office** vous conduit à votre panneau d'administration.



Mais pour l'instant celui-ci ne vous permet pas de vous connecter. Il vous invite, pour des raisons de sécurité, à supprimer le dossier [install](#) et à renommer le dossier [admin](#).



On évitera un nom trop simple à découvrir, tel que [admin2](#).

Il faut faire ces opérations manuellement dans le dossier d'installation de votre boutique, soit sur votre disque dur pour une installation en local, soit via votre client FTP si l'installation s'est faite sur un serveur distant.

Si vous tentez maintenant d'accéder à votre panneau d'administration en cliquant sur le cadre **Back Office**, vous obtiendrez une erreur 404 "Page non trouvée" car en renommant le dossier, nous avons modifié son chemin d'accès.

Pour y accéder, il suffit de remplacer dans la barre d'adresse de votre navigateur le dossier **admin** par le nom que vous lui avez donné.

Exemple en local	Exemple en ligne
Remplacer : http://127.0.0.1/shop/admin par : http://127.0.0.1/shop/admin567	Remplacer : http://myshop.com/shop/admin par : http://myshop.com/shop/admin567

Vous avez alors accès au formulaire de connexion au panneau d'administration de votre nouvelle boutique PrestaShop.



Adresse e-mail
admin@myshop.com

Mot de passe
.....

[Mot de passe oublié ?](#)

Saisissez alors l'adresse e-mail et le mot de passe spécifiés à l'étape 4 de l'installation.

Personnalisez votre boutique

Enfin, cette cinquième étape de l'assistant d'installation vous propose de découvrir les multiples façons de personnaliser votre boutique PrestaShop, tant au niveau de l'aspect que des fonctionnalités grâce aux nombreuses contributions disponibles sur PrestaStore.com, la place de marché de PrestaShop :

www.prestastore.com



Mise à jour de PrestaShop

Précautions

La mise à jour étant une opération délicate affectant à la fois les fichiers et la base de données de votre boutique PrestaShop, il est vivement conseillé de s'exercer auparavant à en effectuer une sur une boutique de tests.

Sauvegarde

Sauvegarde de vos traductions

Pour exporter vos traductions, utilisez l'outil du panneau d'administration de PrestaShop prévu à cet effet :



Sélectionnez la langue à exporter puis cliquez sur le bouton **Exporter**.



Répétez l'opération pour chaque langue à exporter.

Sauvegarde de votre base de données

Sauvegardez votre base de données en utilisant l'outil de sauvegarde mis à disposition par votre hébergeur (phpMyAdmin ou autre).

Sauvegarde de vos fichiers PrestaShop

Si votre boutique PrestaShop est installée dans un sous-dossier de votre site, la future mise à jour sera installée dans ce même dossier. Pour sauvegarder les fichiers de l'ancienne version de votre boutique, renommez simplement le dossier les contenant.

Par exemple, si votre boutique est située dans un dossier `myshop` :

`/myshop`

renommez-le en :

`/myshop_old`

Si les fichiers de votre boutique sont installés directement à la racine du site, créez un nouveau dossier :

`/prestashop_old`

et copiez y l'ensemble de votre boutique PrestaShop.

Installation de la nouvelle version

Téléchargement

Téléchargez la dernière version finale de PrestaShop à l'adresse :

<http://www.prestashop.com/fr/downloads/>

Extraction et copie

Extraire l'archive téléchargée sur votre disque dur puis transférez-les à l'aide de votre client FTP sur la racine de votre site ou dans le dossier contenant votre boutique.

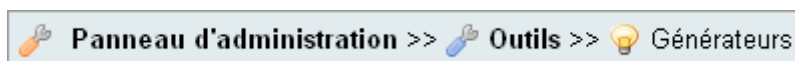
Restauration des données depuis l'ancienne version

Si vous aviez personnalisé certains éléments de PrestaShop (langues, templates de mails, etc.), il vous faudra réappliquer ces modifications sur cette nouvelle installation depuis la copie de sauvegarde précédemment réalisée.

Les images devront être restaurées depuis le dossier `/img` de l'ancienne copie vers le dossier `/img` de la nouvelle installation.

Il vous faudra également réinstaller les modules tiers utilisés dans votre ancienne boutique. Assurez-vous au préalable auprès du développeur que ceux-ci soient compatibles avec cette nouvelle version de PrestaShop.

Si vous avez été amené à modifier le fichier `.htaccess`, régénérez un nouveau fichier `.htaccess` avec l'outil :



puis ajoutez les informations contenues dans l'ancien fichier `.htaccess` dans le nouveau en supprimant les doublons éventuels.

Si vous utilisez un thème personnalisé, copiez le dossier correspondant à ce thème depuis l'ancienne version :

`/myshop_old/themes/montheme`

vers la nouvelle

`/myshop/themes/montheme`

Copiez le fichier :

`/myshop_old/config/settings.inc.php`

de votre ancienne boutique vers la nouvelle :

`/myshop/config/settings.inc.php`

Installer la mise à jour

Lancez l'assistant d'installation depuis votre navigateur and ajoutant `/install` à l'adresse de votre boutique :

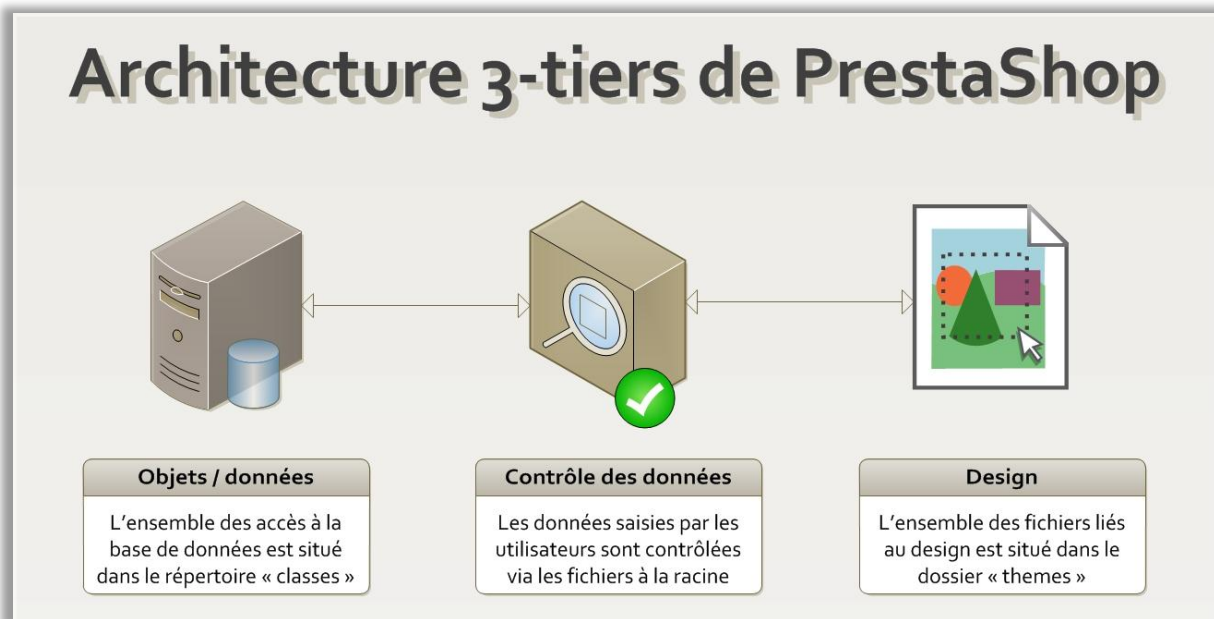
<http://www.myshop.com/myshop/install>

Reprenez le même processus que pour l'installation mais cette fois en cochant la case **Mise à jour** lors de la première étape de l'assistant d'installation.

- ☐ Installation complète de PrestaShop
- ☒ Mise à jour : installez la dernière version de PrestaShop (version actuelle détectée : 1.2.5.0)

Architecture générale

PrestaShop est un logiciel basé sur une architecture 3-tiers, comme nous le montre le schéma ci-dessous :



Ce modèle reprend le principe du MVC ("Modèle-vue-contrôleur"), en le simplifiant et en le rendant accessible à tous.

Notre équipe technique a fait le choix de ne pas utiliser de Framework PHP (Zend Framework, Symfony, CakePHP...) afin de permettre une plus grande lisibilité du code et une modification plus rapide.

Ceci permet également d'obtenir des performances plus élevées car le logiciel contient uniquement le code PHP nécessaire à son bon fonctionnement (et non un ensemble de bibliothèques génériques).

Les avantages de l'architecture 3-tiers sont nombreux :

- Ⓢ La possibilité de comprendre facilement le code source du logiciel,
- Ⓢ Des modifications rapides et simples à mettre en œuvre pour les développeurs,
- Ⓢ La possibilité pour les graphistes/intégrateurs de travailler dans le dossier **themes** sans avoir à comprendre et à lire du code PHP,
- Ⓢ La possibilité pour les développeurs de préparer des données et modules complémentaires pour l'intégrateur.

Grands principes (Modules, Thèmes)

Modules

Le concept

Le logiciel e-commerce PrestaShop est articulé autour d'un principe de modules qui rend son utilisation, sa personnalisation et son extension très faciles.

Un module est un complément à PrestaShop qui permet de réaliser les actions suivantes :

- © Apporter une fonctionnalité supplémentaire à PrestaShop,
- © Afficher des éléments supplémentaires sur le site (sélection de produits, etc.),
- © Communiquer avec des acteurs de l'e-commerce (guides d'achat, plateformes de paiement, logisticiens...),
- © etc.

La société PrestaShop propose gratuitement avec le logiciel plus de 50 modules vous permettant de lancer rapidement et gratuitement votre activité.

Plus de 400 modules complémentaires, sont également disponibles sur le site www.prestastore.com.

Ces derniers ont été réalisés par la société PrestaShop et/ou la communauté PrestaShop et sont commercialisés à des coûts abordables.

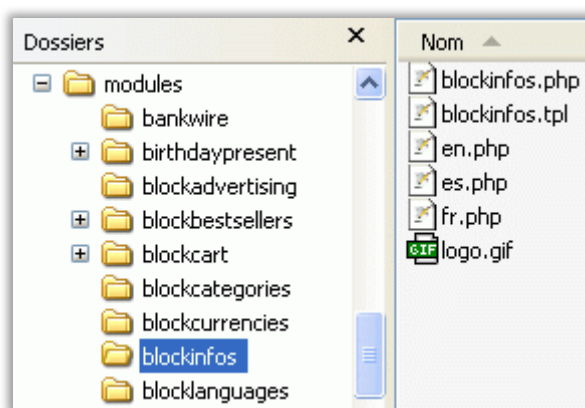
En tant que développeur, vous pouvez également proposer vos modules sur ce site et toucher 70% des montants liés à la vente de vos créations.

Fonctionnement technique d'un module

Un module est composé de :

- © Un dossier du nom du module, situé dans le dossier [modules](#) de PrestaShop,
- © Un fichier PHP du nom du module, situé dans ce même dossier,
- © Un fichier icône, [logo.gif](#), représentant ce module,
- © Éventuellement : des fichiers .tpl, représentant le thème du module,
- © Éventuellement : des fichiers de langue, si ce dernier comporte du texte à afficher,
- © Éventuellement : dans un dossier [/themes/modules/](#), un dossier du même nom que le module comprenant des fichiers TPL et fichiers de langue le cas échéant. Ce dernier dossier est essentiel lors de toute modification d'un module déjà existant, afin d'adapter celui-ci sans pour autant affecter les fichiers d'origine, et permet notamment de gérer différemment l'affichage d'un même module selon le thème utilisé.

Exemple avec le module **blockinfos** disponible gratuitement dans PrestaShop :



Chaque module PrestaShop, s'il a été installé par le marchand, peut intervenir sur un ou plusieurs "**points d'accroche**" également appelés "**hook**" en anglais. Un point d'accroche est la représentation dans la vue de l'instant auquel un traitement du module est effectué et/ou des informations affichées (exemple : dans le panier, la fiche produit, après la mise à jour des stocks). Plus précisément, le hook est un raccourci vers les différentes méthodes de l'objet Module assignées à celui-ci.

Liste des points d'accroche (hooks) de PrestaShop

Voici la liste des 53 points d'accroche disponibles dans PrestaShop :

Front-office

Homepage et contour du site

Nom du point d'accroche	Emplacement (fichier)	Visible par les internautes	Description
header	header.php	Non	Est appelé entre les balises <head></head> de la page. Idéal pour charger vos fichiers Javascript et CSS.
top	header.php	Oui	Est appelé dans l'en-tête de la page.
leftColumn	header.php	Oui	Est appelé lors du chargement de la colonne de gauche.
rightColumn	footer.php	Oui	Est appelé lors du chargement de la colonne de droite.
footer	footer.php	Oui	Est appelé dans le pied de page.
home	index.php	Oui	Est appelé au centre de la page d'accueil.

Fiche produit

Nom du point d'accroche	Emplacement (fichier)	Visible par les internautes	Description
extraLeft	product.php	Oui	Est appelé juste au-dessus du lien "Imprimer", sous la photo.
extraRight	product.php	Oui	Est appelé en dessous du bloc contenant le bouton "Ajouter au panier".
productActions	product.php	Oui	Est appelé à l'intérieur du bloc contenant le bouton "Ajouter au panier", sous ce bouton.
productOutOfStock	product.php	Oui	Est appelé à l'intérieur du bloc contenant le bouton "Ajouter au panier", en dessous de l'information "Disponibilité :".
productfooter	product.php	Oui	Est appelé au-dessus des onglets.
productTab	product.php	Oui	Est appelé dans la liste des onglets tels que "En savoir plus", "Caractéristiques", "Accessoires"... Idéal pour ajouter un onglet supplémentaire dont le contenu sera géré par le hook productTabContent .
productTabContent	product.php	Oui	Est appelé lorsque l'un des onglets est cliqué, idéal pour afficher du contenu correspondant à un onglet que vous auriez ajouté avec le hook productTab .

Panier

Nom du point d'accroche	Emplacement (fichier)	Visible par les internautes	Description
cart	Classe : Cart.php	Non	Est appelé juste <u>après</u> la création ou la mise à jour d'un panier.
shoppingCart	order.php	Oui	Est appelé en dessous du tableau listant les produits contenus dans le panier.
shoppingCartExtra	order.php	Oui	Est appelé en dessous du tableau listant les produits contenus dans le panier, en dessous des boutons de navigation.

createAccountTop	authentication.php	Oui	Est appelé dans le formulaire de création d'un compte client, au-dessus du bloc "Vos informations personnelles".
createAccountForm	authentication.php	Oui	Est appelé dans le formulaire de création d'un compte client, au-dessus du bouton "S'inscrire".
createAccount	authentication.php	Non	Est appelé juste <u>après</u> la création d'un compte client.
customerAccount	my-account.php	Oui	Est appelé sur l'accueil du compte client, en bas de la liste des liens disponibles. Idéal pour ajouter un lien au sein de cette liste.
myAccountBlock	Module : blockmyaccount.php	Oui	Est appelé dans le bloc "Mon compte" en colonne de gauche, en bas de la liste des liens disponibles. Idéal pour ajouter un lien au sein de cette liste.
authentication	authentication.php	Non	Est appelé juste <u>après</u> l'identification d'un client, <u>uniquement si l'identification est validée</u> (adresse e-mail et mot de passe OK).

Recherche

Nom du point d'accroche	Emplacement (fichier)	Visible par les internautes	Description
search	Classe : Search.php	Non	Est appelé après chaque recherche. Idéal pour analyser et/ou exploiter les recherches (et résultats de recherche) effectuées par vos clients.

Choix d'un transporteur

Nom du point d'accroche	Emplacement (fichier)	Visible par les internautes	Description
extraCarrier	order.php	Oui	Est appelé en dessous de la liste des transporteurs disponibles lors du processus de commande. Idéal pour ajouter un transporteur ayant été développé sous la forme d'un module (exemple : Transporteur relais colis).

Paiement

Nom du point d'accroche	Emplacement (fichier)	Visible par les internautes	Description
payment	order.php	Oui	Est appelé pour constituer la liste des moyens de paiements disponibles lors du processus de commande. Idéal pour permettre le choix d'un module de paiement que vous auriez développé.
paymentReturn	order-confirmation.php	Oui	Est appelé lors du retour sur la boutique après paiement. Idéal pour afficher un message de confirmation et/ou des précisions relatives au paiement.
orderConfirmation	order-confirmation.php	Oui	Identique à paymentReturn (doublon). Est appelé lors du retour sur la boutique après paiement. Idéal pour afficher un message de confirmation et/ou des précisions relatives au paiement.
backBeforePayment	order.php	Non	Est appelé lors de l'affichage de la liste des moyens de paiement disponibles. Idéal pour rediriger l'acheteur au lieu de

			lui afficher cette liste (Exemple : Checkout 1-click PayPal).
--	--	--	---------------------------------------------------------------

Retours marchandises

Nom du point d'accroche	Emplacement (fichier)	Visible par les internautes	Description
orderReturn	order-follow.php	Non	Est appelé lorsqu'une demande de retour de marchandises est effectuée par le client, <u>uniquement si aucune erreur n'est rencontrée</u> .
PDFInvoice	Classe : PDF.php	Oui	Est appelé lors de l'affichage d'une facture au format PDF. Idéal pour afficher du contenu dynamique ou statique au sein de cette facture.

Back-office et traitements annexes

Général

Nom du point d'accroche	Emplacement (fichier)	Visible par les internautes	Description
backOfficeTop	header.inc.php	Oui	Est appelé dans l'en-tête, au-dessus des onglets.
backOfficeHeader	header.inc.php	Non	Est appelé entre les balises <head></head> de la page. Idéal pour charger vos fichiers Javascript et CSS.
backOfficeFooter	footer.inc.php	Oui	Est appelé dans le pied de page, au-dessus de la mention "Powered By PrestaShop".
backOfficeHome	index.php	Oui	Est appelé au centre de la page d'accueil.

Commandes et détail de commandes

Nom du point d'accroche	Emplacement (fichier)	Visible par les internautes	Description
newOrder	Classe : PaymentModule.php	Non	Est appelé lors de la création d'une nouvelle commande, <u>après la création</u> .
paymentConfirm	Classe : Hook.php	Non	Est appelé lorsque le statut d'une commande passe à "Paiement accepté".
updateOrderStatus	Classe : OrderHistory.php	Non	Est appelé lors du changement de statut d'une commande, <u>avant le changement</u> .
postUpdateOrderStatus	Classe : OrderHistory.php	Non	Est appelé lors du changement de statut d'une commande, <u>après le changement</u> .
cancelProduct	AdminOrders.php	Non	Est appelé lorsqu'un produit est supprimé d'une commande, <u>après la suppression</u> .
invoice	AdminOrders.php	Oui	Est appelé sur le détail d'une commande, avant le bloc relatif aux informations du client.
adminOrder	AdminOrders.php	Oui	Est appelé sur le détail d'une commande, après le bloc relatif aux informations du client.
orderSlip	AdminOrders.php	Non	Est appelé lors de la création d'un avoir, <u>après la création</u> .

Produits

Nom du point d'accroche	Emplacement (fichier)	Visible par les internautes	Description
addproduct	AdminProducts.php	Non	Est appelé lorsqu'un produit est créé ou dupliqué, <u>après la création/duplication</u> .
updateproduct	AdminProducts.php	Non	Est appelé lorsqu'un produit est mis à jour avec l'ajout d'une nouvelle photo, <u>après la mise à jour</u> .
deleteproduct	Classe : Product.php	Non	Est appelé lorsqu'un produit est supprimé, <u>avant la suppression</u> .
updateQuantity	Classe : PaymentModule.php	Non	Est appelé lors de la validation d'une commande dont le statut n'est pas "Annulé" ou "Erreur de paiement", pour chaque produit de la commande.
updateProductAttribute	Classe : Product.php	Non	Est appelé lors de la mise à jour d'une déclinaison de produit, <u>après la mise à jour</u> .
watermark	AdminProducts.php	Non	Est appelé lors de l'ajout d'une image sur un produit, <u>après l'ajout</u> .

Statistiques

Nom du point d'accroche	Emplacement (fichier)	Visible par les internautes	Description
GraphEngine	Classe : ModuleGraph.php	Oui	Est appelé lors de l'affichage d'un graphique de statistiques.
GridEngine	ModuleGridEngine.php	Oui	Est appelé lors de l'affichage d'une liste de données statistiques.
AdminStatsModules	AdminStatsTab.php	Oui	Est appelé lors de l'affichage de la liste des modules de statistiques.

Clients

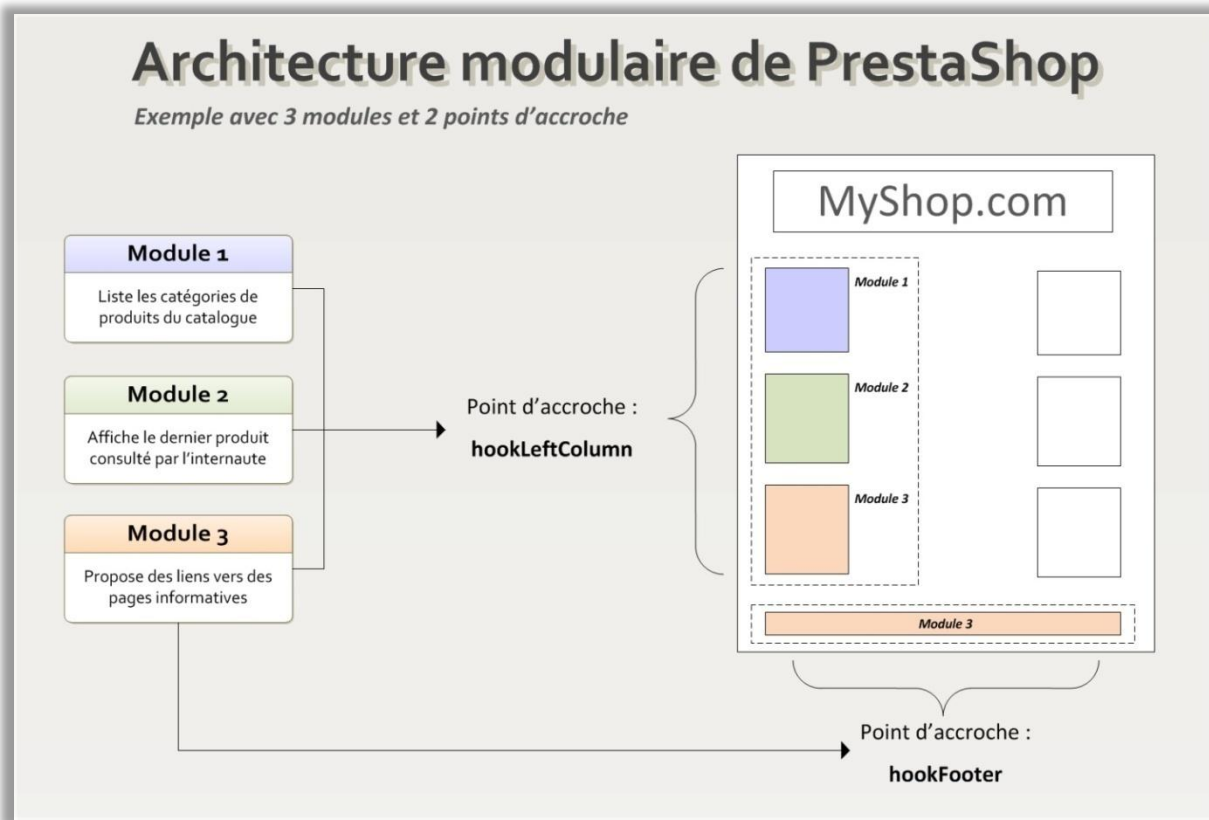
Nom du point d'accroche	Emplacement (fichier)	Visible par les internautes	Description
adminCustomers	AdminCustomers.php	Oui	Est appelé sur le détail d'un client, après la liste des groupes de clients auxquels il appartient.

Transporteurs

Nom du point d'accroche	Emplacement (fichier)	Visible par les internautes	Description
updateCarrier	AdminCarriers.php	Non	Est appelé lors de la mise à jour d'un transporteur, <u>après la mise à jour</u> .

Lorsqu'une page du site e-commerce est affichée, pour chacun des points d'accroche qui la composent, le moteur PrestaShop vérifie quels sont les modules à appeler.

Voici un récapitulatif de l'architecture modulaire de PrestaShop :



Thèmes

Le concept

Le logiciel e-commerce PrestaShop propose un système de thèmes afin de permettre aux marchands une personnalisation à leurs couleurs.

Ce système est articulé autour d'un moteur de templates appelé Smarty (<http://www.smarty.net>). Ce dernier permet aux intégrateurs de développer facilement, sans connaissances techniques particulières, leur propre thème.

PrestaShop propose en standard un thème neutre, aux tons gris, permettant aux marchands de démarrer facilement et gratuitement, quel que soit leur secteur d'activité.

Plus de 100 thèmes PrestaShop sont également disponibles sur le site www.prestastore.com.

Ces derniers ont été réalisés par la société PrestaShop et/ou la communauté PrestaShop et sont commercialisés à des coûts abordables.

En tant que graphiste/intégrateur, vous pouvez également proposer vos thèmes sur ce site et toucher 70% des montants liés à la vente de vos modules.

Fonctionnement technique d'un thème

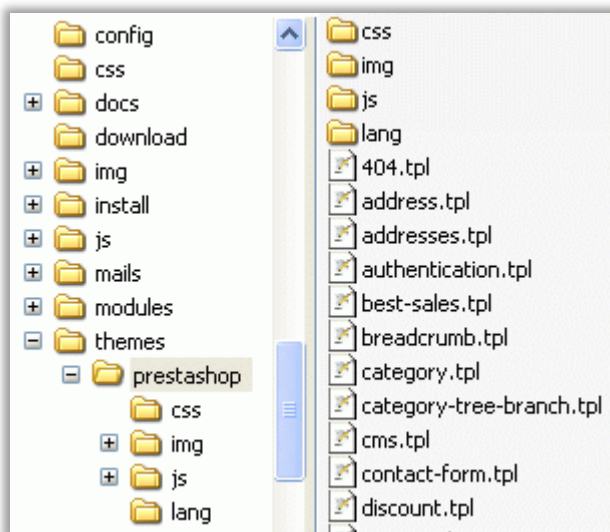
Un thème PrestaShop est un ensemble de fichiers vous permettant de modifier l'apparence de votre boutique.

Voici les informations importantes à noter :

- © L'ensemble des fichiers relatifs à un thème graphique est situé dans le dossier [/themes](#).
- © Chaque thème est représenté par un sous-dossier portant le nom du thème.
- © Chaque thème est constitué de fichiers template (.tpl), d'images (.gif, .jpg, .png), d'une ou plusieurs feuilles de style CSS et parfois de fichiers JS.

- © Chaque thème comporte un fichier [preview.jpg](#) dans son dossier, ce qui permet au marchand d'avoir un aperçu du thème dans le Back-office et également de sélectionner ce thème.

Voici un aperçu de l'arborescence d'un thème PrestaShop :



© Le dossier [css](#) contient les fichiers relatifs aux styles CSS du thème.

© Le dossier [img](#) contient les fichiers relatifs aux images du thème.

© Le dossier [js](#) contient les fichiers relatifs aux Javascript du thème.

© Le dossier [lang](#) contient les traductions du thème. Pensez à mettre les bons droits (ex : CHMOD 666) afin que l'outil de Back-office puisse enregistrer les traductions.

© La racine du thème contient uniquement les fichiers .tpl (templates Smarty) ainsi que le fichier [preview.jpg](#).

Note : les dossiers [css](#), [img](#) et [js](#) ne sont pas indispensables au fonctionnement d'un thème personnalisé.

La gestion des traductions

Pour l'ensemble des textes contenus dans vos fichiers .tpl, vous devez inclure la balise suivante :

```
{l s='Mon texte'}
```

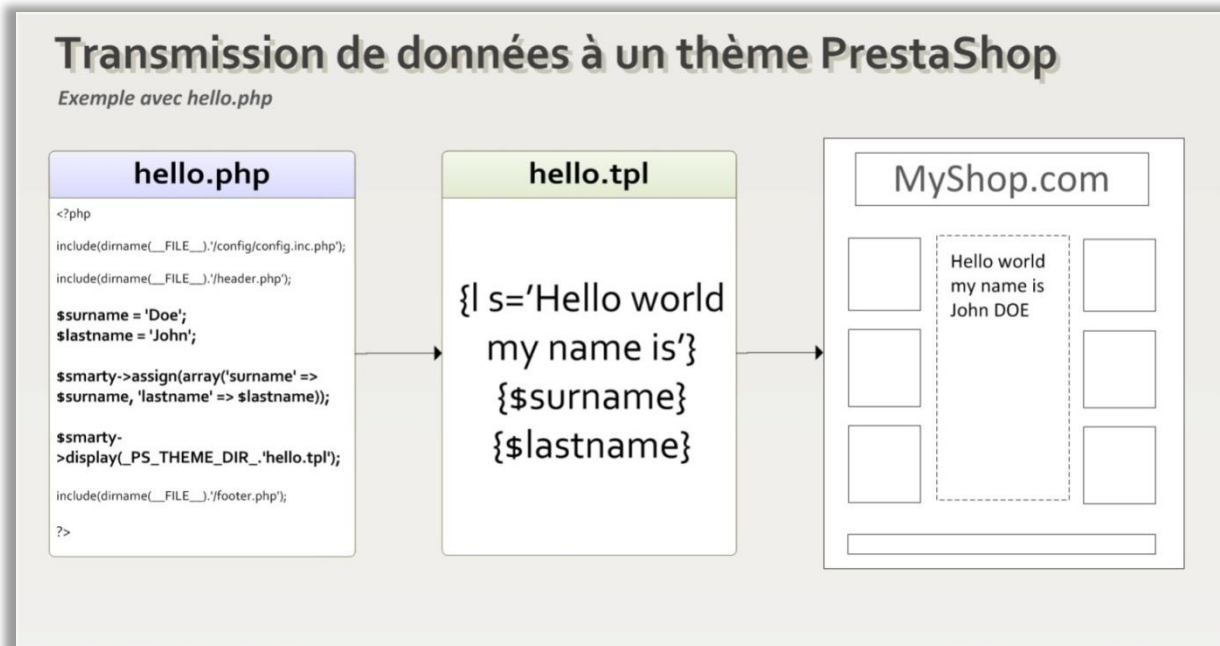
Ceci vous permettra de traduire facilement ces textes via l'onglet :



section **Modifier les traductions** du Back-office PrestaShop.

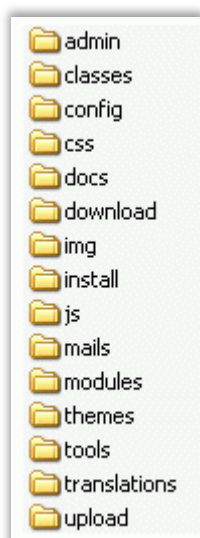
Transmettre des informations (données) à un thème PrestaShop

Voici un schéma de fonctionnement de la transmission de données entre le cœur de PrestaShop et les thèmes PrestaShop. L'appel à la fonction `l`, dans [hello.tpl](#), permet d'afficher le texte dans la langue choisie, si celui-ci a été traduit comme indiqué ci-dessus.



L'arborescence de PrestaShop

Notre équipe technique a souhaité séparer de façon claire et intuitive les différents éléments du logiciel. Voici une présentation de chaque dossier de l'arborescence de PrestaShop :



© **admin** : contient l'ensemble des fichiers relatifs au Back-office de PrestaShop. Il vous sera demandé de le renommer après l'installation de PrestaShop pour plus de sécurité.

Important : nous vous recommandons également de protéger ce dossier via un fichier `.htaccess/.htpasswd` !

© **classes** : contient l'ensemble des fichiers relatifs au modèle objet de PrestaShop. Chaque fichier représente une classe (un objet) et ses propriétés/méthodes.

© **config** : contient les fichiers de configuration de PrestaShop. Sauf cas exceptionnel, vous n'avez pas à les éditer manuellement, ces derniers sont gérés par l'outil d'installation et le Back-office.

© **css** : contient les fichiers CSS qui ne sont pas relatifs aux thèmes. Il s'agit donc par exemple des fichiers CSS du Back-office.

© **docs** : contient de la documentation et doit être supprimé si vous travaillez dans un environnement de production.

© **download** : contient vos produits dématérialisés (exemple : pdf, mp3...).

© **img** : contient l'ensemble des images, icônes et photos de PrestaShop, hormis ceux des thèmes. Vous y trouverez donc les photos des catégories de produit (sous-dossier **c**), celles des produits (sous-dossier **p**) ou encore celles du Back-office (sous-dossier **admin**).

© **install** : contient l'ensemble des fichiers relatifs à l'outil d'installation de PrestaShop. Il vous sera demandé de le supprimer après l'installation de PrestaShop pour plus de sécurité.

© **js** : contient les fichiers Javascript qui ne sont pas relatifs aux thèmes. Il s'agit donc par exemple des fichiers Javascript du Back-office. Il contient également le framework Javascript jQuery.

© **mails** : contient l'ensemble des fichiers HTML et TXT relatifs aux e-mails envoyés depuis PrestaShop. Il y a un sous-dossier par langue disponible. C'est dans ce dossier que vous devez éditer manuellement ces e-mails si vous souhaitez en modifier le contenu.

© **modules** : contient un sous-dossier par module disponible. Si vous souhaitez désinstaller définitivement un module, commencez d'abord par le **désinstaller via le Back-office de PrestaShop** avant de supprimer son sous-dossier !

© **themes** : contient un sous-dossier par thème disponible.

© **tools** : contient des outils externes intégrés dans PrestaShop. On y retrouve notamment les outils suivants : Smarty (moteur de template/thème), FPDF (moteur de génération de fichiers PDF), Swift (moteur d'envoi d'emails), PEAR Xml parser (outil PHP).

© **translations** : contient un sous-dossier par langue disponible. Si vous souhaitez modifier les traductions, il faut toutefois le faire via le Back-office de PrestaShop et non directement dans ces dossiers !

© **upload** : contient les fichiers qui auraient été transmis par les clients lors de l'achat de produits personnalisables (exemple : la photo d'un client qui doit être imprimée sur un mug).

Bonnes pratiques à respecter (développeurs, intégrateurs)

Voici une liste non exhaustive de bonnes pratiques à respecter :

1. Lors de la création d'un thème, ne mélangez jamais du code XHTML et du code PHP.
2. Lors de la création d'un thème, ne mélangez jamais du code XHTML et du code CSS, placez son code dans un fichier CSS séparé.
3. n'effectuez jamais de requête SQL dans un contrôleur PHP (fichiers .php à la racine de PrestaShop), privilégiez l'utilisation des méthodes existantes dans les classes PHP et/ou créez de nouvelles méthodes dans ces classes.
4. Validez votre code XHTML et votre code CSS grâce au validateur W3C <http://validator.w3.org/>.
5. Vérifiez toujours si une méthode n'existe pas déjà dans les classes.
6. Ne modifiez pas les fichiers du cœur de PrestaShop, développez un maximum sous forme de modules afin de faciliter les mises à jour.
7. Respectez les normes de codes pour un code lisible et clair, facilement maintenable.
8. Commentez votre code, en anglais.
9. Mettre la boutique en maintenance via le Back-office PrestaShop lorsque vous effectuez des corrections.
10. Utilisez le navigateur Firefox ou Google Chrome et incitez les gens autour de vous à en faire autant.
11. Remerciez PrestaShop en effectuant un don.

La sécurisation de votre installation PrestaShop

Voici quelques recommandations à suivre si vous souhaitez sécuriser votre boutique PrestaShop, ces dernières sont numérotées et classées par ordre de priorité :

1. Sécurisez votre accès au Back-office PrestaShop :
 - a. renommez le dossier **admin** après l'installation de PrestaShop, choisissez un nom de dossier plus compliqué avec un mélange de chiffres et lettres,
 - b. protégez ce dossier avec un fichier **.htaccess/.htpasswd**, ou demandez à votre hébergeur de le faire pour vous,
 - c. n'enregistrez pas vos codes d'accès sur votre ordinateur,
 - d. choisissez un mot de passe complexe (mélange de chiffres et de lettres) et modifiez-le régulièrement.
2. Sécurisez votre installation de PHP :
 - a. l'option "register_globals" doit être configurée sur "Off",
 - b. l'option "magic_quotes" doit être configurée sur "Off",
 - c. l'option "allow_include_url" doit être configurée sur "Off".
3. Supprimez systématiquement le dossier **install** après installation ou mise à jour de PrestaShop.
4. Supprimez systématiquement les fichiers inutiles après installation ou mise à jour de PrestaShop :
 - a. **readme_fr.txt**, **readme_en.txt**, **readme_es.txt**,
 - b. **CHANGELOG**,
 - c. dossier **docs**.
5. Interdisez l'accès à vos fichiers de thème (templates), grâce à un fichier **.htaccess** comportant le code suivant :

```
<FilesMatch "\.tpl$" >
order deny,allow
deny from all
</FilesMatch>
```

Options de configuration et fine-tuning/performances

Ce paragraphe vous fait découvrir les variables de configuration de PrestaShop qui ne sont pas gérées dans le Back-office mais directement au sein des fichiers de configuration.

Quatre fichiers de configuration sont présents dans PrestaShop :

- © [config/config.inc.php](#) : contient la configuration du cœur de PrestaShop. Ce fichier est éditable manuellement.
- © [config/defines.inc.php](#) : contient la définition des constantes. Anciennement contenues dans le fichier `settings.inc.php`.
- © [config/settings.inc.php](#) : contient les accès à votre base de données ainsi que votre numéro de version. Ce fichier est géré directement par le Back-office de PrestaShop et l'outil d'installation.
- © [config/smarty.config.inc](#) : contient la configuration relative à l'outil Smarty (moteur de template/thème). Ce fichier est éditable manuellement.

Fichier config.inc.php

- © Si vous êtes en mode production, pensez à laisser `@ini_set('display_errors', 'Off');` sur `Off`. A l'inverse, si vous êtes en phase de développement, cela vous sera utile de la passer à `On` pour détecter d'éventuelles erreurs.
- © Si vous êtes en mode production, pensez à laisser `define('_PS_DEBUG_SQL_', false);` sur `false`. A l'inverse, si vous êtes en phase de développement, cela vous sera utile de la passer à `true` pour détecter d'éventuelles erreurs.

Fichier defines.inc.php

- © Ce fichier contient l'ensemble des emplacements des fichiers, via la méthode PHP `define()`. Si vous souhaitez modifier ces emplacements, n'oubliez pas de modifier les lignes correspondantes.

Fichier smarty.config.inc.php

- © Le système de cache de Smarty (`$smarty-> caching`) est désactivé car il n'est pas compatible avec le moteur dynamique de PrestaShop.
- © **IMPORTANT** : le paramètre `$smarty->force_compile`, doit être passé à `false` lorsque vous mettez un site en production. Ceci vous permettra de gagner jusqu'à 30% de performances sur le temps de chargement des pages. En contrepartie, lorsque vous modifierez un fichier `.tpl`, il vous faudra alors supprimer le contenu du dossier `/tools/smarty/compile/` (sauf `index.php`).
- © Le paramètre `$smarty->compile_check` doit être laissé sur `false`.
- © Le paramètre `$smarty->debugging` vous permet de bénéficier d'informations de DEBUG Smarty lors de l'affichage de vos pages.

Amélioration des performances de PrestaShop

Voici quelques conseils qui vous permettront d'optimiser les performances de votre installation PrestaShop.

- © Activez le cache MySQL (ou demandez à votre hébergeur de le faire) et configurez-le sur une valeur assez élevée (exemple : 256M).
- © N'oubliez pas de passer `$smarty->force_compile` à `false` (lire le paragraphe "Fichier `smarty.config.inc.php`").
- © Utilisez des techniques avancées d'intégration telles que le Spriting CSS :
<http://www.alistapart.com/articles/sprites>
- © Répartissez vos éléments statiques sur des sous-domaines différents afin de paralléliser les flux HTTP des visiteurs.

Pour cela, il faut vous rendre dans le fichier [config/defines.inc.php](#) et ajouter ceci :

```
if ($_SERVER['REMOTE_ADDR'] != '127.0.0.1')
```

```

{
    define('_THEME_IMG_DIR_', 'http://img2.xxx.com/');
    define('_THEME_CSS_DIR_', 'http://css.xxx.com/');
    define('_THEME_JS_DIR_', 'http://js.xxx.com/');
    define('_THEME_CAT_DIR_', 'http://img1.xxx.com/c/');
    define('_THEME_PROD_DIR_', 'http://img1.xxx.com/p/');
    define('_THEME_MANU_DIR_', 'http://img1.xxx.com/m/');
    define('_PS_IMG_', 'http://img1.xxx.com/');
    define('_PS_ADMIN_IMG_', 'http://img1.xxx.com/admin/');
}
else
{
    define('_THEME_IMG_DIR_', _THEMES_DIR_._THEME_NAME_.'/img/');
    define('_THEME_CSS_DIR_', _THEMES_DIR_._THEME_NAME_.'/css/');
    define('_THEME_JS_DIR_', _THEMES_DIR_._THEME_NAME_.'/js/');
    define('_THEME_CAT_DIR_', __PS_BASE_URI__.'img/c/');
    define('_THEME_PROD_DIR_', __PS_BASE_URI__.'img/p/');
    define('_THEME_MANU_DIR_', __PS_BASE_URI__.'img/m/');
    define('_PS_IMG_', __PS_BASE_URI__.'img/');
    define('_PS_ADMIN_IMG_', __PS_IMG_.'admin/');
}
}

```

© Installez un cache d'opcode (ou demandez à votre hébergeur de le faire), PrestaShop est compatible avec eAccelerator : <http://www.eaccelerator.net/>.

Ceci devrait vous permettre de gagner nettement en termes de charge processeur.

Travaux pratiques

Créer un thème PrestaShop

Thème par défaut PrestaShop

Le thème par défaut de PrestaShop a été conçu dans un style neutre pour s'adapter à tout secteur de l'e-commerce. De plus, ce thème respecte scrupuleusement les normes du W3C et a été optimisé pour s'afficher correctement dans tous les navigateurs les plus utilisés.

Malgré tout, vous pouvez vouloir modifier le thème de votre boutique afin de l'adapter à votre activité. Le plus simple est de partir d'un thème existant.

Créer un nouveau thème


Les thèmes sont situés dans le dossier [/themes/](#) ; c'est donc dans ce dossier que l'on devra placer les différents thèmes pouvant être appliqués à la boutique. On y trouve d'ailleurs un sous-dossier [prestashop](#) correspondant au thème PrestaShop par défaut.

C'est de ce thème que nous allons partir pour créer un nouveau thème.

Néanmoins, il est fortement conseillé de conserver intact le thème PrestaShop par défaut. L'une des raisons est que la modification du thème peut générer des bugs et le fait de remplacer temporairement le thème personnalisé par le thème PrestaShop par défaut permet ainsi de vérifier si le problème provient du thème personnalisé ou non. Si le problème ne survient plus avec le thème PrestaShop par défaut, c'est que le thème personnalisé en est la cause.

Nous allons donc dupliquer le dossier [/themes/prestashop/](#) et nommer ce dossier, par exemple, [/themes/montheme/](#). Le thème par défaut sera ainsi conservé intact.

Arborescence d'un thème

Fichier / Dossier	Description
preview.jpg	Image utilisée en tant qu'aperçu dans la section  Thèmes de la page Panneau d'administration >> Préférences >> Apparence . La présence de cette image est obligatoire, sinon le thème ne peut être sélectionné

	dans cette section.
404.tpl	Page affichée lorsque la page demandée n'est pas trouvée (erreur 404).
address.tpl	Page d'ajout ou de modification d'une adresse client.
addresses.tpl	Page listant les adresses d'un client.
authentication.tpl	Page d'identification et de création de compte.
best-sales.tpl	Page listant les meilleures ventes.
breadcrumb.tpl	Chemin de navigation (fil d'Ariane).
category-tree-branch.tpl	Utilisé uniquement pour le bloc Catégories.
category.tpl	Page listant les produits au sein d'une catégorie.
cms.tpl	Pages informatives (onglet Outil >> CMS du BO).
contact-form.tpl	Page du formulaire de contact.
discount.tpl	Page listant les bons de réduction d'un client.
errors.tpl	Est appelé par l'ensemble des pages. Affiche les erreurs.
footer.tpl	Pied de page.
header.tpl	En-tête de page.
history.tpl	Page listant les commandes d'un client.
identity.tpl	Page de modification des informations personnelles d'un client.
index.tpl	Page d'accueil.
maintenance.tpl	Page de mise en maintenance du site.
manufacturer-list.tpl	Page listant les fabricants.
manufacturer.tpl	Page listant les produits d'un fabricant.
my-account.tpl	Page d'accueil du compte d'un client.
new-products.tpl	Page listant les produits dernièrement ajoutés.
order-address.tpl	Page du processus de commande : Étape 1 – choix des adresses (livraison, facturation).
order-carrier.tpl	Page du processus de commande : Étape 2 – choix du mode de livraison.
order-confirmation.tpl	Page de confirmation de commande (après paiement).
order-detail.tpl	Page de détail d'une commande d'un client.
order-follow.tpl	Page de demande d'un retour produit pour un client.
order-payment.tpl	Page du processus de commande : Étape 3 – choix du mode de paiement.
order-return.tpl	Page de détail d'un retour d'un client.
order-slip.tpl	Page listant les avoirs d'un client.
order-steps.tpl	Barre d'évolution du processus de commande.
pagination.tpl	Est appelé par l'ensemble des pages listant des produits. Boutons permettant de changer de page dans une liste de produits.
password.tpl	Page de changement de mot de passe d'un client.
prices-drop.tpl	Page listant les promotions.
product-list.tpl	Est appelé par l'ensemble des pages listant des produits. Liste des produits.
product-sort.tpl	Est appelé par l'ensemble des pages listant des produits. Menu permettant de filtrer une liste de produits.
product.tpl	Page de détail d'un produit.
scenes.tpl	Page de détail d'une scène au sein d'une catégorie de produits.
search.tpl	Page listant les résultats d'une recherche.

shopping-cart-product-line.tpl	Page de détail d'une ligne du panier.
shopping-cart.tpl	Page listant les produits dans le panier d'un client.
sitemap.tpl	Page de plan du site.
supplier-list.tpl	Page listant les fournisseurs.
supplier.tpl	Page listant les produits d'un fournisseur.
thickbox.tpl	Page de zoom d'une photo d'un produit.
css	Contient les feuilles de style du thème. Le fichier global.css concerne la mise en forme de la majeure partie du site. <i>A moins que vous ne soyez un utilisateur averti, il est conseillé de laisser les fichiers d'origine.</i>
img	Contient les images du thème. Remplacez les images adéquates par vos créations personnelles. <i>A moins que vous ne soyez un utilisateur averti, il est conseillé de laisser les fichiers d'origine.</i>
js	Scripts JavaScript utilisés dans le thème. <i>A moins que vous ne soyez un utilisateur averti, il est conseillé de laisser les fichiers d'origine.</i>
lang	Fichiers de traduction. Ces fichiers sont générés par le Back Office et leur contenu doit être modifié depuis l'outil Panneau d'administration >> Outils >> Traductions , section Modifier les traductions .

Thème des modules

Certains modules possèdent leurs propres fichiers .tpl. Si vous créez votre propre thème, il vous faudra également adapter celui des modules. Vous pouvez le faire sans modifier les fichiers originaux des modules.

Pour cela, dans le dossier de votre thème, créez un dossier **modules** :

[/themes/montheme/modules/](#)

et copiez-y les dossiers des modules à retravailler. Dans ces dossiers, ne conservez que les fichiers .tpl que vous modifierez selon vos besoins. **Ces éléments seront utilisés prioritairement sur les originaux.**

Conseils pour l'édition d'un thème PrestaShop

Firebug

Pour vous faciliter grandement le travail d'intégration, nous vous conseillons de vous familiariser avec l'add-on **Firebug** pour Firefox qui vous permettra de délimiter rapidement chaque bloc et d'afficher les propriétés CSS qui s'y appliquent.

Lien utile : <http://www.prestashop.com/forums/viewthread/16736>

JavaScript

Les fonctions JavaScript doivent être stockées dans le dossier **js**.

Fichier preview.jpg

Une fois votre thème terminé, ne pas oublier d'en faire une capture d'écran, de la redimensionner, puis de copier le fichier à la place du fichier [preview.jpg](#) déjà présent dans le thème.

Pour faire une capture d'écran d'une page Web complète dont la taille est plus grande que l'écran, il peut être très pratique d'utiliser un add-on de Firefox tel que **FireShot** ou **ScreenGrab**.

Normes W3C et compatibilité inter-navigateurs

Il est judicieux de vérifier la compatibilité des pages de votre thème avec les normes du W3C à l'aide du validateur proposé par cet organisme à l'adresse <http://validator.w3.org/>.

Enfin, testez l'affichage des différentes pages de votre thème dans les navigateurs les plus utilisés (Firefox, Internet Explorer, Chrome, Safari, Opera).

Créer un module PrestaShop

Principe des modules

Les modules sont idéals pour laisser s'exprimer ses talents de développeur et son imagination, tant les possibilités de créations sont nombreuses.

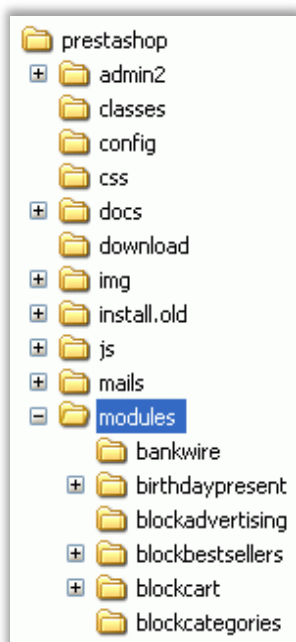
Les modules peuvent afficher du contenu (blocs, textes, etc.), réaliser des traitements (mise à jour en masse, import, export, etc.), s'interfacer avec d'autres logiciels, etc.

On peut rendre ses modules plus ou moins configurables. Plus le module que vous créez est paramétrable, plus il sera souple d'utilisation et donc répondra aux besoins d'un plus grand nombre de personnes.

L'un des intérêts essentiels d'un module est d'ajouter à la solution des fonctionnalités sans modifier le cœur de celle-ci, ceci afin de faciliter une éventuelle future mise à jour qui écrasera les modifications apportées au cœur.

Pour cette raison, et dans la mesure du possible, **le développement d'un module devra toujours éviter au maximum de modifier le cœur de la solution**. Ceci est malheureusement parfois inévitable pour le développement de certaines fonctionnalités.

Arborescence des modules



La solution PrestaShop contient un dossier [/modules](#) dans lequel sont regroupés tous les modules ; il s'agit aussi bien des modules fournis dans la solution de base que des modules tiers ajoutés ultérieurement.

Chaque module est stocké dans un sous-dossier distinct à l'intérieur du dossier [/modules](#) ([bankwire](#), [birthdaypresent](#), etc.).

Structure de base d'un module

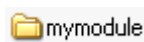
La structure de base est simple et similaire pour chaque module, facilitant ainsi l'implémentation de nouveaux modules.

Pour la décrire, nous allons créer pas à pas un module rudimentaire.

Cet exemple sera volontairement développé en anglais. En effet, d'une part développer en anglais est une bonne habitude à prendre, et d'autre part, le système de traduction de PrestaShop étant souple et puissant, il sera aisé plus tard de réaliser la traduction du module, chose que nous verrons plus bas.

Nous appellerons notre module "My module".

Pour débiter la création d'un module, commençons par créer le dossier qui le contiendra. Celui-ci porte le nom que l'on souhaite donner au module. Ce nom ne doit contenir que des lettres, chiffres ou tirets (-_) tout en minuscules.

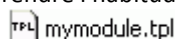


Ce dossier doit contenir un fichier PHP chargé du traitement des données. Ce fichier porte le même nom que le dossier parent.



Ce fichier est suffisant pour créer un module élémentaire. D'autres fichiers et sous-dossiers pourront venir en complément.

Un fichier TPL sera utile en cas d'affichage de données dans le Front Office. Les fichiers TPL doivent être situés à la racine du module. Les fichiers TPL peuvent porter n'importe quel nom. Lorsqu'il n'y en a qu'un, on le nomme souvent comme le module, mais il vaut mieux prendre l'habitude de leur donner des noms explicites.



Le fichier `mymodule.php` va contenir la classe de notre module.

Celle-ci porte le nom donné au module, et donc au répertoire parent.

De plus, cette classe hérite de la classe Module (`/classes/Module.php`) ou de n'importe quelle classe dérivée (PaymentModule, ModuleGridEngine, ModuleGraph) mettant ainsi à notre disposition quelques propriétés et méthodes.

Fichier `mymodule.php`

```
<?php

class MyModule extends Module
{
    public function __construct()
    {
        $this->name = 'mymodule';
        $this->tab = 'Test';
        $this->version = 1.0;

        parent::__construct();

        $this->displayName = $this->l('My module');
        $this->description = $this->l('Description of my module.');
```

```
    public function __construct()
```

Constructeur de la classe.

```
        $this->name = 'mymodule';
```

Attribution d'un nom à notre instance de classe.

```
        $this->tab = 'Test';
```

Titre du tableau qui contiendra le module dans la liste des modules du Back Office. Il peut s'agir d'un nom déjà existant (**Products**, **Blocks**, **Stats**) ou d'un nom personnalisé. Dans ce dernier cas, un nouveau tableau sera créé avec ce titre.

```
        $this->version = '1.0';
```

Numéro de version du module, affiché dans la liste des modules.

```
parent::__construct();
```

Appel du constructeur parent. Le constructeur doit impérativement être appelé avant l'utilisation des méthodes `$this->l()` et après l'enregistrement du nom.

Dans d'anciennes versions de PrestaShop (< 1.3) on trouve ici :

```
$this->page = basename(__FILE__, '.php');
```

Permet de récupérer le nom du fichier, sans l'extension '.php'. `__FILE__` est une constante magique contenant le chemin et le nom du fichier (depuis PHP 4.0.2, `__FILE__` contient toujours le chemin absolu). La fonction `basename()` sépare le nom du fichier du reste du chemin, l'argument '.php' servant à supprimer ce suffixe.

```
$this->displayName = $this->l('My module');
```

Nom du module affiché dans la liste des modules du Back Office. Nous verrons plus loin comment rendre possible la traduction du texte 'My module'.

L'appel à la méthode `l()` est expliqué dans le paragraphe "Traduction des modules" plus bas.

```
$this->description = $this->l('Description of my module.');
```

Description du module affichée dans la liste des modules du Back Office.

L'appel à la méthode `l()` est expliqué dans le paragraphe "Traduction des modules" plus bas.

```
public function install()
{
    if(parent::install() == false)
        return false;
    return true;
}
```

En l'état, cette méthode n'est pas indispensable, car elle ne fait que vérifier le retour de la méthode `install()` de la classe `Module.php`. D'ailleurs, si cette méthode était absente, la méthode de la classe mère serait automatiquement appelée et le résultat serait donc identique.

Néanmoins, nous mentionnons cette méthode ici car elle nous servira à faire des contrôles et des actions lors de l'installation du module (création de table, copie de fichier, création de variable de configuration, etc.).

De la même manière, le module pourra contenir une méthode `uninstall()` permettant de personnaliser la désinstallation du module.

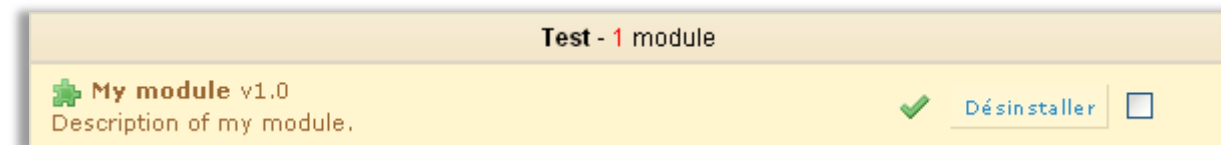
Un exemple d'utilisation de cette méthode :

```
public function uninstall()
{
    Db::getInstance()->Execute('DELETE FROM `'.DB_PREFIX_.'block_cms` WHERE `id_block` ='.intval($this->id);
    parent::uninstall();
}
```

Pour terminer notre module de base, nous allons ajouter une icône en regard du nom du module dans la liste des modules. Pour cela, il suffit de placer une image 16x16 dans le dossier du module et de l'appeler `logo.gif`.

Le site <http://www.famfamfam.com/lab/icons/silk/> propose une bibliothèque importante d'icônes gratuites.

Voici, dans l'onglet **Panneau d'administration >> Modules**, le résultat obtenu avec le code ci-dessus.



Un clic sur le bouton **Installer** fait un appel à la méthode `install()` du module qui appelle à son tour la méthode `install()` de la classe `Module.php` qui ajoute alors une occurrence à la table `ps_module`.

<input type="checkbox"/>			51	statssearch	1
<input type="checkbox"/>			52	statscheckup	1
<input type="checkbox"/>			53	mymodule	1

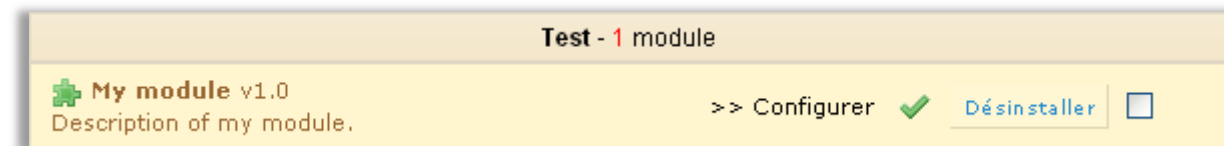
Configuration d'un module

Nous l'avons mentionné, il est possible de rendre un module configurable et donc adapté aux besoins d'un plus grand nombre d'utilisateurs.

Pour permettre l'accès à l'interface de configuration d'un module, il faut ajouter une fonction *getContent()* à la classe de notre module.

```
Fichier mymodule.php (extrait)
public function getContent ()
{
    // Instructions de la page de configuration...
}
```

Le simple ajout ci-dessus provoque dans l'encart de notre module l'apparition d'un lien **>> Configurer**.



Grefe d'un module

Pour qu'un module soit "accroché" à un endroit du Front Office ou du Back Office, par exemple pour l'affichage d'un bloc ou pour exécuter un traitement à un moment précis, il faut lui permettre d'utiliser l'un des hooks que nous avons listés plus haut.

Pour cela, nous allons modifier le code de notre module comme suit.

```
Fichier mymodule.php (extrait)
public function install()
{
    if(parent::install() == false OR !$this->registerHook('leftColumn'))
        return false;
    return true;
}
...
public function hookLeftColumn($params)
{
    global $smarty;
    return $this->display(__FILE__, 'mymodule.tpl');
}

public function hookRightColumn($params)
{
    return $this->hookLeftColumn($params);
}
```

```
if(parent::install() == false OR !$this->registerHook('leftColumn'))
```

Si l'installation ou la greffe rate, on communique l'échec au cœur.

```
public function hookLeftColumn($params)
{
    global $smarty;
    return $this->display(__FILE__, 'mymodule.tpl');
}
```

Cette méthode permet l'affichage du rendu final dans la colonne de gauche du Front Office.

```
public function hookRightColumn($params)
{
    return $this->hookLeftColumn($params);
}
```

Cette méthode permet l'affichage du rendu final dans la colonne de droite du Front Office. Elle consiste ici en un simple appel à la fonction *hookLeftColumn()*.

Nous pouvons donc d'ores et déjà greffer ce module sur le hook désiré et le déplacer à l'emplacement souhaité au sein de ce hook.

Le choix du ou des hooks auxquels accrocher le module se fait dans l'onglet [Panneau d'administration >> Modules >> Positions](#) du Back Office.



Cliquer sur **Greffer un module**, dans le champ **Module** sélectionner le module à greffer, ici "My module" et dans le champ **Greffer le module sur** choisir "Left column blocks" puis valider en cliquant sur **Enregistrer**.

The screenshot shows the 'Greffer un module' form. It has two dropdown menus. The first is labeled 'Module :' and has 'My module' selected. The second is labeled 'Greffer le module sur :' and has 'Left column blocks' selected. Both dropdowns have a red asterisk to their right, indicating they are required fields.

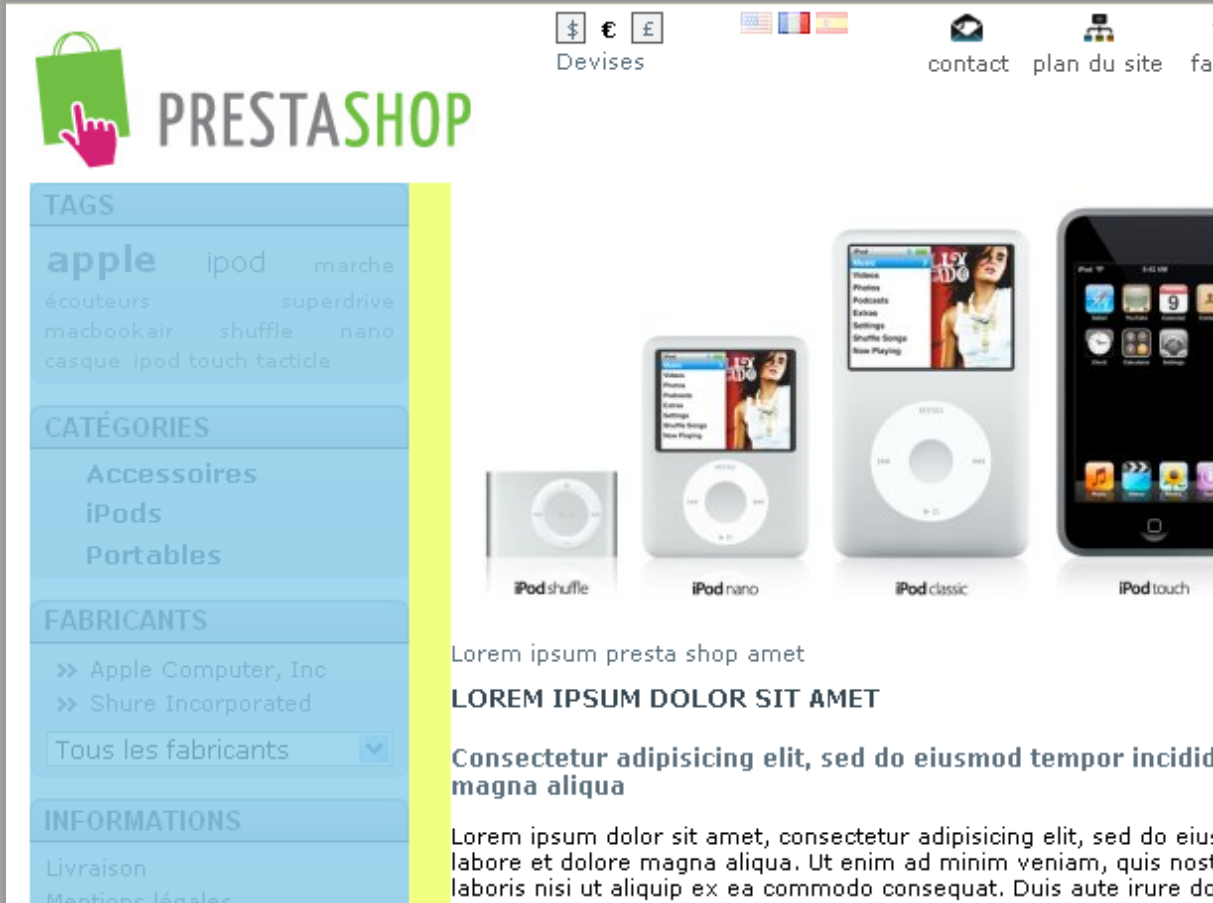
Attention : greffer un module sur un hook pour lequel il n'a aucune méthode implémentée est inutile.

De retour sur la page [Panneau d'administration >> Modules >> Positions](#), dans le tableau "Left column blocks" déplacer l'encart du module "My module" à l'emplacement souhaité, à l'aide des flèches σ et τ , ou avec un glisser-déplacer.

Left column blocks - 9 modules			
1	▼	My module v1.0 Description of my module.	
2	↕	Bloc Mon compte v1.2 Ajoute un bloc proposant des liens vers le compte du client	
3	↕	Bloc tags v1.0 Ajoute un bloc contenant les tags et un lien vers la recherche correspondante	
4	↕	Bloc catégories v1.0 Ajoute un bloc proposant une navigation au sein de vos catégories de produits	
5	↕	Bloc produits déjà vus v0.9 Ajoute un bloc proposant les derniers produits vus par le client	
6	↕	Bloc fabricant v1.0 Ajoute un bloc avec les fabricants	
7	↕	Bloc informations v1.1 Ajoute un bloc avec des liens vers vos pages d'informations	
8	↕	Bloc publicité v0.1 Ajoute un bloc affichant une publicité	
9	▲	Bloc logo de paiement v0.1 Ajoute un bloc affichant tous les logos des modes de paiement	

Affichage d'un module

Notre module va maintenant devoir afficher quelque chose dans la colonne gauche du Front Office.



The screenshot shows the PrestaShop v1.0 interface. On the left is a sidebar with navigation menus: TAGS (apple, ipod, marche, écouteurs, superdrive, macbookair, shuffle, nano, casque, ipod touch, tactile), CATÉGORIES (Accessoires, iPods, Portables), FABRICANTS (Apple Computer, Inc, Shure Incorporated, Tous les fabricants), and INFORMATIONS (Livraison, Mentions légales). The main content area displays four iPod models (iPod shuffle, iPod nano, iPod classic, iPod touch) and placeholder text: "Lorem ipsum presta shop amet", "LOREM IPSUM DOLOR SIT AMET", "Consectetur adipisicing elit, sed do eiusmod tempor incididunt magna aliqua", and "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure do".

Below the screenshot is a Firebug console window showing the HTML structure of the page. The breadcrumb trail is: Éditer | h4 < div#cate...ft.block < **div#left...n.column** < div#columns < div#page < body#index < htr. The HTML code is as follows:

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
<head>
<body id="index">
  <noscript> <ul> <li> Cette boutique n&#039;cessite JavaScript afin de fonctionner correctement. Merci de l&#039;activer dans votre navigateur. </li> </ul> </noscript>
  <div id="page">
    <div id="header">
      <div id="columns">
        <div id="left_column" class="column">
          <div id="tags_block_left" class="block tags_block">
            <script src="http://127.0.0.1/psdoc/themes/prestashop/js/tools/treeManagement.js" type="text/javascript">
          <div id="categories_block_left" class="block">
            <script type="text/javascript">
```

Comme nous le montre Firebug, la colonne de gauche de PrestaShop est matérialisée par une balise `<div>` portant le nom `"left_column"`. La fonction `hookLeftColumn()` va donc afficher le rendu de notre module dans cette div.

Nous l'avons vu plus haut, le fichier contenant les éléments à afficher est dans notre cas le fichier `mymodule.tpl` que nous avons passé en paramètres de la méthode `display()` dans la méthode `hookLeftColumn()`.

```
return $this->display(_FILE_, 'mymodule.tpl');
```

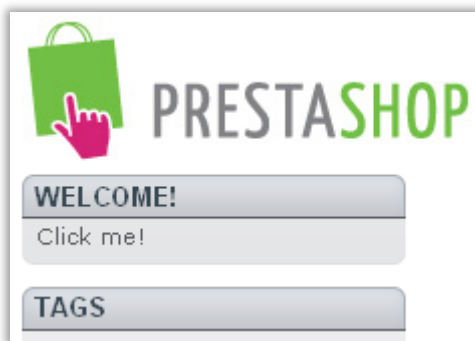
Éditons le fichier `mymodule.tpl` pour y ajouter le contenu suivant.

Fichier `mymodule.tpl` (extrait)

```
<!-- Block mymodule -->
<div id="mymodule_block_left" class="block">
  <h4>Welcome!</h4>
```

```
<div class="block_content">
  <ul>
    <li><a href="{ $base_dir }modules/mymodule/page.php" title="Click this link">
      Click me!
    </a></li>
  </ul>
</div>
<!-- /Block mymodule -->
```

En rechargeant la page du Front Office, nous voyons le module apparaître.



Le lien hypertexte ne mène pour le moment nulle part. Pour le tester, il suffit de créer le fichier mentionné dans `mymodule.tpl`, c'est-à-dire un fichier `mymodule_page.php` dans notre dossier `mymodule`, puis d'y ajouter du texte.

Fichier **mymodule_page.php**

Welcome to my shop!

Un clic sur le lien nous mène bien à notre page, mais l'affichage n'est pas très "PrestaShop".

http://127.0.0.1/...mymodule/page.php

Welcome to my shop!

Nous allons donc créer un fichier `mymodule_page.tpl`, dans lequel nous allons copier le contenu de notre fichier `mymodule_page.php`, qui dans notre cas est :

Fichier **mymodule_page.tpl**

Welcome to my shop!

Note : il est conseillé de nommer vos fichiers TPL avec des noms plus explicites pour mieux les retrouver dans le Back Office de PrestaShop plus tard (en particulier au niveau des traductions).

Et dans le fichier `mymodule_page.php`, nous allons bien sûr appeler le fichier TPL nouvellement créé pour que Smarty se charge de l'afficher, mais nous allons également provoquer l'appel des fichiers TPL chargés d'afficher le reste de l'interface du Front Office, à savoir les fichiers `header.tpl` et `footer.tpl`.

Le code de notre fichier PHP devient donc :

Fichier **mymodule_page.php**

```
<?php
global $smarty;
include('../..../config/config.inc.php');
include('../..../header.php');

$smarty->display(dirname(__FILE__).'/mymodule_page.tpl');

include('../..../footer.php');
?>
```

```
global $smarty;
```

Récupère l'instance courante de Smarty. Doit être fait avant le premier `display()`.

```
include('../..../config/config.inc.php');
```

Charge la configuration de PrestaShop.

```
include('../..../header.php');
```

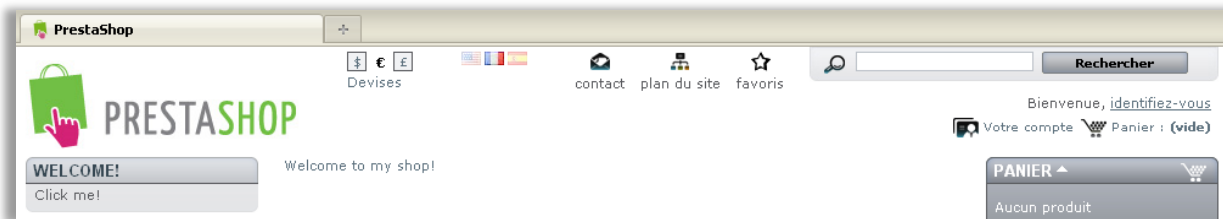
```
include('.././footer.php');
```

Appel des fichiers PHP chargés de l'affichage du reste de la mise en page, notamment en affectant à chaque variable Smarty correspondant aux hooks leur contenu. Ces fichiers appellent respectivement [header.tpl](#) et [footer.tpl](#) qui effectuent le rendu final.

```
$smarty->display(dirname(__FILE__).'/mymodule_page.tpl');
```

Appel de notre fichier [mymodule_page.tpl](#) pour affichage.

Si nous rechargeons la page du Front Office, le résultat est beaucoup plus sexy ! ;-)



Utilisation de Smarty

Smarty est le moteur de templates utilisé dans PrestaShop.

Il parcourt les fichiers TPL à la recherche d'éléments dynamiques pour les remplacer par les données adéquates avant d'afficher le résultat obtenu. Ces éléments sont indiqués entre accolades { ... }. L'utilisateur peut créer ses propres variables et les utiliser dans les fichiers TPL.

Par exemple, dans le fichier [mymodule_page.php](#), on crée une telle variable.

Fichier **mymodule_page.php**

```
<?php
global $smarty;
include('.././config/config.inc.php');
include('.././header.php');
include('MyModule.php');

$mymodule = new MyModule();
$message = $mymodule->I('Welcome to my shop!');
$smarty->assign('messageSmarty', $message);

$mymodule->display(dirname(__FILE__).'/mymodule_page.tpl');

include('.././footer.php');
```

On peut désormais demander à Smarty l'affichage du contenu de cette variable dans le fichier [page.tpl](#).

Fichier **mymodule_page.tpl**

```
{ $messageSmarty }
```

De la même manière, l'élément { \$HOOK_LEFT_COLUMN } sera remplacé par le contenu de la colonne de gauche, c'est-à-dire par le contenu de tous les modules greffés au hook correspondant à cette colonne.

Toutes les variables Smarty sont globales ; il faut donc veiller à ne pas utiliser un nom similaire à une variable existante afin de ne pas l'écraser. On évitera donc les noms de variables trop simples tels que "products".

Voici une liste des variables Smarty communes à toutes les pages :

Fichier / Dossier	Description
img_ps_dir	URL du dossier image.
img_cat_dir	URL du dossier image des catégories.
img_lang_dir	URL du dossier image des langues.
img_prod_dir	URL du dossier image des produits.
img_manu_dir	URL du dossier image des fabricants.
img_sup_dir	URL du dossier image des fournisseurs.
img_ship_dir	URL du dossier image des transporteurs.
img_dir	URL du dossier image du thème.
css_dir	URL du dossier css du thème.

js_dir	URL du dossier javascript du thème.
tpl_dir	URL du dossier du thème courant.
modules_dir	URL du dossier des modules.
mail_dir	URL du dossier des templates de mail.
pic_dir	URL du dossier d'upload.
lang_iso	ISO code de la langue utilisée.
come_from	URL de provenance de l'utilisateur.
shop_name	Nom de la boutique.
cart_qties	Nombre de produits dans le panier.
cart	Le panier.
currencies	Les différentes devises disponibles.
id_currency_cookie	ID de la devise utilisée.
currency	Objet Currency (devise actuellement utilisées).
cookie	Cookie de l'utilisateur.
languages	Liste des différentes langues disponibles.
logged	Indique si l'utilisateur est connecté à un compte client.
page_name	Nom de la page.
customerName	Nom du client (lorsqu'il est connecté).
priceDisplay	Méthode d'affichage du prix (avec ou sans taxes...).
roundMode	Méthode d'arrondi utilisée.
use_taxes	Indique si les taxes sont activées.

Pour afficher toutes les variables Smarty disponibles dans une page donnée, on peut ajouter l'instruction suivante dans le fichier TPL :

```
{debug}
```

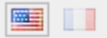





Pour mettre une portion d'un fichier TPL en commentaire, plutôt que d'utiliser les commentaires HTML qui se retrouveront dans le code source de la page, on peut utiliser les commentaires Smarty `{* ... *}`.

```
{* Cette chaîne est un commentaire et ne sera pas affichée. *}
```

Traduction des modules

Nous avons rédigé les textes de notre module en anglais. Nous voulons maintenant traduire celui-ci en français. Cette tâche va nous être grandement facilitée par Smarty ainsi que l'outil de traduction de PrestaShop.

Actuellement, les textes ont été spécifiés de façon statique. Ainsi, quelle que soit la langue choisie dans le Front Office ou le Back Office, les textes du module restent en anglais.

Langue	Back Office	Front Office
	<p>Test - 1 module</p> <p> My module v1.0 Description of my module.</p> <p>>> Configure  Uninstall <input type="checkbox"/></p>	<p>WELCOME!</p> <p>Click me!</p>
	<p>Test - 1 module</p> <p> My module v1.0 Description of my module.</p> <p>>> Configurer  Désinstaller <input type="checkbox"/></p>	<p>WELCOME!</p> <p>Click me!</p>

Nous allons donc rendre possible la traduction de nos textes.

Pour cela, nous allons utiliser dans les fichiers PHP la méthode `l()` de la classe abstraite `Module.php`. Dans les fichiers TPL en revanche ces textes statiques deviendront des éléments dynamiques que Smarty remplacera par la traduction correspondant à la langue choisie.

Ainsi, dans `mymodule.php` nous avons déjà aperçu plus haut l'utilisation de la méthode `l()` :

```
$this->displayName = $this->l('My module');
$this->description = $this->l('Description of my module.');
```

Dans `mymodule.tpl` :

```
<h4>Welcome!</h4>
...
Click me!
```

L'utilisation des éléments dynamiques Smarty donne :

```
<h4>{! s=' Welcome!' mod='mymodule'}</h4>
...
{! s='Click me!' mod='mymodule'}
```

Et dans `page.tpl` :

```
Welcome to my shop!
donne :
{! s='Welcome to my shop!' mod='mymodule'}
```

Le paramètre 'mod' est obligatoire dans les templates du module pour que l'outil de traduction puisse faire la correspondance entre la chaîne de caractères à traduire et la traduction afin de récupérer celle-ci lors de l'affichage.

Seules les apostrophes ' doivent être utilisées pour délimiter les chaînes à traduire. Échapper les éventuelles apostrophes de la chaîne avec un antislash '\'.

Ceci va nous permettre de traduire nos chaînes de caractères dans les langues installées dans notre boutique. Pour cela, rendez-vous sur l'onglet [Panneau d'administration >> Outils >> Traductions](#) et dans la section **Modifier les traductions** choisir "Traduction des modules" dans la liste, puis cliquer sur le drapeau français pour traduire notre module en français.

La section correspondant à notre module est déployée car celle-ci contient des chaînes non encore traduites. Il suffit donc de remplir les champs de la partie droite avec les équivalents en français des chaînes situées à gauche puis de valider en cliquant sur le bouton [Enregistrer les modifications](#) situé en haut ou en bas de la page.

```
prestashop - mymodule - 4 expressions (4)
```

My module	=	Mon module
Description of my module.	=	Description de mon module.
Welcome!	=	Bienvenue !
Click me!	=	Cliquez-moi !

```
prestashop - page - 1 expressions (1)
```

Welcome to my shop!	=	Bienvenue dans ma boutique !
---------------------	---	------------------------------

```
prestashop - mymodule - 4 expressions (4)
```

My module	=	Mon module
Description of my module.	=	Description de mon module.
Welcome!	=	Bienvenue !
Click me!	=	Cliquez-moi !

Dans le dossier `mymodule` correspondant à notre module, un fichier `fr.php` contenant les traductions est alors créé.









```
Fichier fr.php
<?php

global $_MODULE;
$_MODULE = array();
$_MODULE['<{mymodule}>prestashop>mymodule_2ddddc2a736e4128ce1cdfd22b041e7f'] = 'Mon module';
$_MODULE['<{mymodule}>prestashop>mymodule_d6968577f69f08c93c209bd8b6b3d4d5'] = 'Description de mon module.';
$_MODULE['<{mymodule}>prestashop>mymodule_9a843f20677a52ca79af903123147af0'] = 'Bienvenue !';
$_MODULE['<{mymodule}>prestashop>mymodule_f42c5e677c97b2167e7e6b1e0028ec6d'] = 'Cliquez-moi !';
$_MODULE['<{mymodule}>prestashop>page_c0d7cfa0105851272f83d5c1fe63a1c'] = 'Bienvenue dans ma boutique !';
```

Ces fichiers de langues ne doivent pas être édités directement. En effet l'utilisation, pour les modifier, d'un éditeur peu approprié pourrait poser problème. Les traductions doivent être réalisées depuis l'outil de traduction du Back Office.

Si nous choisissons le français comme langue dans le Front Office ou le Back Office, nous pouvons maintenant voir notre module traduit.

Langue	Back Office	Front Office
	<div>Test - 1 module</div> <div>  My module v1.0 Description of my module. </div> <div> >> Configurer  Uninstall <input type="checkbox"/> </div>	<div>WELCOME!</div> <div>Click me!</div>
	<div>Test - 1 module</div> <div>  Mon module v1.0 Description de mon module. </div> <div> >> Configurer  Désinstaller <input type="checkbox"/> </div>	<div>BIENVENUE !</div> <div>Cliquez-moi !</div>

Pour que les traductions soient prises en compte par l'outil de PrestaShop, il faut que les fichiers PHP et TPL contenant les chaînes à traduire soient situés à la racine du module.

Créer un onglet dans le Back-office, ainsi que sa classe

Cette section vous permet de créer en quelques minutes un nouvel onglet ou sous-onglet dans le Back-office de PrestaShop.

Suivez ces étapes dans l'ordre :

- Ajoutez une nouvelle table dans votre base de données PrestaShop, intitulée **ps_test**, comportant deux champs : **id_test (INT 11)**, **test (VARCHAR 32)**
- Créez un fichier vide intitulé **Test.php** dans le dossier **classes**.
- Dans votre fichier ajoutez les lignes suivantes :

```
<?php
class Test extends ObjectModel
{
    /** @var string Name */
    public $test;

    protected $fieldsRequired = array('test');
    protected $fieldsSize = array('test' => 64);
    protected $fieldsValidate = array('test' => 'isGenericName');
    protected $table = 'test';
    protected $identifier = 'id_test';

    public function getFields()
    {
        parent::validateFields();
        $fields['test'] = pSQL($this->test);
        return $fields;
    }
}
?>
```

- Créez un fichier vide intitulé **AdminTest.php** dans le dossier **admin/tabs**.
- Dans votre fichier ajoutez les lignes suivantes :

```
<?php
include_once(PS_ADMIN_DIR.'../classes/AdminTab.php');

class AdminTest extends AdminTab
{
    public function __construct()
    {
        $this->table = 'test';
        $this->className = 'Test';
        $this->lang = false;
        $this->edit = true;
        $this->delete = true;

        $this->fieldsDisplay = array(
            'id_test' => array('title' => $this->l('ID'), 'align' => 'center', 'width' => 25),
            'test' => array('title' => $this->l('Name'), 'width' => 200));
    }
}
```



```
$this->identifiant = 'id_test';

parent::__construct();
}

public function displayForm()
{
    global $currentIndex;

    $defaultLanguage = intval(Configuration::get('PS_LANG_DEFAULT'));
    $languages = Language::getLanguages();
    $obj = $this->loadObject(true);

    echo '
    <script type="text/javascript">
        id_language = Number('.$defaultLanguage.');
    </script>';

    echo '
    <form action="'. $currentIndex . '&submitAdd'.$this->table.'=1&token='.$this->token.'" method="post"
    class="width3">
        ' . ($obj->id ? '<input type="hidden" name="id_'.$this->table.'" value="'.$obj->id.'" />' : '') . '
        <fieldset><legend>'.$this->l('Profiles').</legend>
            <label>'.$this->l('Name:').</label>
            <div class="margin-form">';
        foreach ($languages as $language)
            echo '
                <div id="name_'.$language['id_lang'].'" style="display: ' . ($language['id_lang'] ==
                $defaultLanguage ? 'block' : 'none').'; float: left;">
                    <input size="33" type="text" name="name_'.$language['id_lang'].'" value="'.htmlentities($this-
                >getFieldValue($obj, 'name', intval($language['id_lang']), ENT_COMPAT, 'UTF-8').'" /><sup> *</sup>
                </div>';
            $this->displayFlags($languages, $defaultLanguage, 'name', 'name');
            echo '
                <div class="clear"></div>
            </div>
            <div class="margin-form">
                <input type="submit" value="'. $this->l(' Save ') .'" name="submitAdd'.$this->table.'"
            class="button" />
            </div>
            <div class="small"><sup>*</sup> '.$this->l('Required field').</div>
            </fieldset>
        </form>';
    }
}
?>
```

En cas de problème

En cas de problème, nous vous conseillons de suivre les étapes suivantes afin de trouver de l'aide.

Forum officiel PrestaShop

Rendez-vous sur notre forum à l'adresse <http://www.prestashop.com/forums> et effectuez une recherche en utilisant des mots-clés pertinents.

Pour affiner votre recherche, n'hésitez pas à utiliser le formulaire de recherche avancée disponible.

Certains forums comportent également des sujets "accrochés" de façon persistante en haut de la liste des sujets et contenant des informations pertinentes.

Outil de rapports de bugs : le "Bug Tracker"

La plupart des problèmes rencontrés par les membres de notre communauté active sont postés dans cet outil consultable par tous à l'adresse http://www.prestashop.com/bug_tracker/. Le cas échéant, les personnes ayant trouvé solution à un problème donné peuvent la poster en attendant qu'un développeur le corrige.

Cette correction sera disponible dans la prochaine version de PrestaShop, ou consultable en temps réel sur notre serveur SVN dont le "trunk" est accessible publiquement en lecture à l'adresse :

<http://svn.prestashop.com/trunk/>

Pour utiliser cette version SVN dans un but de test ou de développement, vous pouvez utiliser par exemple l'un des logiciels suivants (liste non exhaustive) :

Système d'exploitation	Logiciel	URL
Linux	KDESVN	http://kdesvn.alwings-world.de/
Mac	svnX	http://www.lachoseinteractive.net/fr/communaute/subversion/svnX/telecharger/
Windows	TortoiseSVN	http://tortoisesvn.net/

Poster sur le forum

Si avec ceci aucune solution n'a été trouvée, posez votre question sur notre forum, de la façon la plus détaillée possible, dans la section adéquat (langue, thème).

Il est nécessaire de créer un compte pour pouvoir poster sur notre forum. Pour cela, cliquez sur le lien **S'inscrire** situé tout en haut à droite du site Prestashop.com.

Soumettre un bug dans le Bug Tracker

Si votre problème provient d'un bug avéré, merci de soumettre celui-ci, **en anglais**, dans notre Bug Tracker en utilisant le formulaire suivant :

http://www.prestashop.com/bug_tracker/report/bug/

Sites officiels PrestaShop

Voici la liste des sites officiels de PrestaShop qui vous permettront d'aller plus loin dans votre expérience de l'e-commerce.

Adresse	Description
www.prestashop.fr	Site de la société PrestaShop (éditeur).
www.prestashop.com	Site de la communauté PrestaShop (Forums, Bug Tracker, démo de la solution, blog).
www.prestastore.com www.prestastore.fr www.prestastore.es	Place de marche de la solution PrestaShop proposant des modules et des thèmes pour votre boutique (versions anglophone, francophone et hispanophone).
www.prestabox.com www.prestabox.fr www.prestabox.es	Hébergement de boutiques PrestaShop sans les contraintes techniques.