Guide de l'administrateur système

Guide de l'administrateur système

Ce guide vous aidera à configurer un serveur Web plus efficace et sûr.

Une fois terminé, vous serez prêt à installer PrestaShop, à l'aide de notre Guide de démarrage.

Configuration PHP

Manipuler php.ini

Nombre des conseils de ce guide requièrent que vous modifiiez le fichier `php.ini`, qui se trouve dans le dossier d'installation de PHP sur votre serveur (et non le dossier de PrestaShop).

Votre hébergeur ne vous autorisera pas forcément à modifier ou même lire ce fichier. Dans ce cas, n'hésitez pas à le contacter pour réaliser les modifications demandées.

Par exemple, vous n'aurez sans doute pas accès au fichier `php.ini` sur un hébergeur mutualisé. Si votre hébergeur ne vous offre pas par défaut la configuration requise et que vous ne pouvez pas accéder au `php.ini`, vous devriez soit passer à un hébergement dédié, ou trouver un hébergeur plus permissif.

Notez cependant que modifier le fichier `php.ini` reste un acte technique et avancé. Si votre boutique fonctionne bien en l'état, inutile de forcément vouloir toucher à ce fichier, ni de changer d'hébergeur.

Modifier le fichier de configuration de PHP requiert que vous touchiez à certains réglages du fichier `php.ini`, ce qui la plupart du temps consiste à activer ou désactiver une option (passer de "On" à "Off" et vice-versa). Le fichier contient beaucoup de documentation pour chacune de ses lignes : lisez bien celles qui concernent votre modification afin de mieux comprendre celles-ci. Faites attention à ce que vous modifiez, car cela a un résultat direct sur la manière dont PHP fonctionne, et donc sur la stabilité ou même la sécurité de votre serveur.

Prérequis système essentiels

Pour que PrestaShop 1.6.x fonctionne correctement, il faut que votre configuration de PHP dispose des réglages et bibliothèques suivants :

- MySQL (ou Percona Server, voir la section "Améliorer les performances de PrestaShop" plus bas dans ce chapitre).
- Bibliothèque GD.
- Extension Dom.
- `allow_url_fopen`.

L'extension MySQL permet l'accès à vos données. PrestaShop ne peut tout simplement pas fonctionner sans.
Vous pouvez également utilise le remplacement direct Percona Server, qui offre de meilleures performances par rapport au serveur MySQL standard.

La bibliothèque GD permet à PHP de manipuler vos images de manière dynamique. PrestaShop l'utilise pour redimensionner et retravailler les fichiers image qui sont mis en ligne (ajout d'un filigrane, rognure, etc.). Sans images, votre boutique perd le plus gros de son attrait ; assurez-vous donc que GD est bien activé !

L'extension Dom permet d'analyser des documents XML. PrestaShop s'en sert pour diverses fonctionnalités, comme le localisateur de magasin. Elle est aussi utilisée par certains modules, ainsi que la bibliothèque `pear_xml_parser`.

La directive `allow_url_fopen` permet aux modules de PrestaShop d'accès à des fichiers distants, ce qui est un besoin fondamental du processus de paiement, entre autres choses. Il est donc essentiel que cette directive soit activée.

En résumé, il est **essentiel** que les directives suivantes aient les réglages indiqués :

```
extension = php_mysql.dll
extension = php_gd2.dll
allow_url_fopen = On
```

Réglages recommandés

Pour un usage optimal, votre installation PHP devrait disposer des réglages et bibliothèques suivants :

- Reconnaissance GZIP.
- Bibliothèque Mcrypt.
- `register_globals` désactivé.
- `magic_quotes` désactivé.
- `allow_url_include` désactivé.

La reconnaissance GZip permet au serveur de compresser les pages, images et script avant de les envoyer au navigateur. Cela accélère la navigation du site, et donc offre une meilleure expérience utilisateur.

Mcrypt offre une meilleure couche de sécurité à PHP, permettant l'usage d'algorithmes de hash et de chiffrement.

Une fois activée, la directive `register_globals` passe les variables d'environnement (GET, POST, COOKIE, SERVER...) en variables globales. **Il n'est pas recommandé d'utiliser de telles variables**, car un utilisateur pourrait facilement enregistrer une valeur à l'aide de la méthode GET, par exemple. Il est donc impératif de désactiver cette directive.

La directive `magic_quotes` échappe (autrement dit, elle "ajoute un antislash" (http://php.about.com/od/phpfunctions/g/addslashes_php.htm]) automatiquement les caractères spéciaux (`'`, `"`, `\`, `NULL`) pour toutes les variables d'environnement (GET, POST, COOKIE, SERVER...). Cette option doit être désactivée car elle ajoute un antislash à toutes les variables, même celles n'en ayant pas besoin. De plus, certaines applications web sont mal conçues, et pourraient se retrouver avec deux antislash au lieu d'un, aboutissant à des donnés inexploitables.

La directive `allow_url_include` permet d'inclure un fichier à l'aide des déclarations `require` et `include`, même s'il ne vient pas de votre serveur. Cette option doit être désactivée, car dès qu'une application sur votre serveur la prend mal en compte, votre serveur devient vulnérable : un utilisateur pourrait ajouter n'importe quel fichier en provenance de n'importe quel serveur, et celui-ci serait exécuté sur votre propre serveur.

En somme, il est **fortement recommandé** que les directives suivantes aient les réglages indiqués :

```
register_globals = Off
magic_quotes_gpc = Off
allow_url_include = Off
```

Configuration MySQL

Le compte d'administrateur MySQL par défaut est souvent "root" ou "admin", et donne accès à l'ensemble du contenu de la base de données, quel que soit le propriétaire de la base de données. L'administrateur a tous les droits, et peut lancer n'importe quelle action. Il vous faut donc protéger vos bases de données, afin d'empêcher que vos applications ne tombent sous le coup d'injections SQL (lire http://fr.wikipedia.org/wiki/Injection_SQL), ce qui peut arriver quand un utilisateur a accès au mot de passe de l'administrateur.

> Si vous venez d'installer MySQL, donnez un mot de passe au compte "root", qui par défaut n'en a pas.

Un utilisateur MySQL par application web

Chaque fois que vous installez une nouvelle application web sur votre serveur, vous devez créer un nouvel utilisateur MySQL qui ne dispose que des droits strictement nécessaires à la gestion des données de cette application. N'utilisez PAS le même nom d'utilisateur pour gérer les bases de données de toutes les applications web installées.

Ainsi, si vous disposez d'un compte MySQL capable de créer d'autres utilisateurs, voici comme s'y prendre en ligne de commande :

```
mysql -u USERNAME -p PASSWORD
```

Vous pourriez également utiliser la requête SQL suivante:

```
mysql> USE mysql;
mysql> CREATE USER 'username'@'servername' IDENTIFIED BY 'new_password';
```

Notez que votre hébergeur vous donne sans doute accès à un outil en ligne pour administrer MySQL plus facilement, par exemple cPanel. Utilise-le, car il y a moins de chance que l'hébergeur vous donne accès à la ligne de commande.

Maintenant, nous disposons d'un utilisateur avec uniquement les droits nécessaires pour se connecter à la base de données locale.

Nous devons autoriser cet utilisateur à utiliser la base de données de PrestaShop, et configurer ses droits par la même occasion. Voici un modèle de requête SQL :

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER
     > ON 'prestashop'.* TO 'new_user'@'localhost';
mysql> FLUSH PRIVILEGES;
```

Nous disposons maintenant d'un utilisateur pour la base de données "prestashop". Pensez à faire ceci pour chacune des applications web que vous ajoutez à votre serveur.

Sécurité

Lisez notre page dédiée à la sécurisation de PrestaShop, pleine de conseils faciles à appliquer.

Peaufinage des performances

Cette section vous aidera à mieux comprendre les variables de configuration qui ne sont pas gérées par le back-office, mais directement dans les fichiers de configuration.

Il y a quatre fichiers de configuration dans PrestaShop, tous se trouvant dans le dossier `/config` :

- `config.inc.php` : principal fichier de configuration de PrestaShop.
- `defines.inc.php` : contient toutes les valeurs constantes de PrestaShop. *Elles étaient auparavant définies dans `settings.inc.php`.*
- `settings.inc.php` : contient les informations d'accès à la base de données, ainsi que le numéro de version de PrestaShop.
- `smarty.config.inc.php` : contient toutes les configurations liées à Smarty, le moteur de template/modèle de PrestaShop.

Fichier config.inc.php

La plupart des données de ce fichier sont mises en place lors de l'installation de PrestaShop, et ne devraient donc pas être éditées à la main. Modifiez ce fichier à vos risques et périls.

Fichier defines.inc.php

En mode production, assurez-vous que `define('_PS_MODE_DEV_', false);` est bien réglé sur `false`.
Au contraire, pour passer en mode développement/test et faire remonter les erreurs potentielles, passez `define('_PS_MODE_DEV_', false);` de `false` à `true`.

Il est également possible de mettre en place l'outil de profilage du code (*code profiling*), qui affiche de nombreuses informations en bas de chaque page : passez la ligne `define ('_PS_DEBUG_PROFILING_', false);` à `true`, puis ouvrez n'importe quelle page du front-end ou du back-end. En bas de la page se trouvera un résumé complet des performances du chargement. Il est impératif que la boutique soit inaccessible aux visiteurs

En plus des valeurs constantes, ce fichier contient l'emplacement de tous les fichiers et dossiers. Si vous avez besoin de les changer, n'oubliez pas de conserver l'original à portée de main, dans le cas où vous souhaiteriez revenir à l'ancien emplacement.

Fichier smarty.config.inc.php

- `$smarty->caching = false;` : le système de cache de Smarty doit être désactivé, parce qu'il n'est pas compatible avec PrestaShop.
- **IMPORTANT** : en production, `$smarty->force_compile` doit être mis à "false", car cela améliorera de 30 % le temps de chargement de la page. En revanche, lors de la modification d'un fichier `.tpl`, vous devrez effacer le dossier `/tools/smarty/compile` (sauf son fichier `index.php`) afin de voir les modifications en place dans votre navigateur. Notez que ce réglage peut également être fait dans le back-office, dans la page "Performance" du menu "Paramètres avancés", section "Smarty".
- `$smarty->compile_check` devrait être laissé à "false".
- `$smarty->debugging` vous donne accès aux informations de débogage de Smarty quand les pages sont affichées.

Améliorer les performances de PrestaShop

Voici quelques conseils qui devraient vous permettre d'optimiser PrestaShop.

Performances de PHP

Si vous le pouvez, utiliser un cache d'opcode (ou demandez à votre hébergeur d'en installer un pour vous), afin d'alléger la charge de traitement du serveur. "Opcode" signifie "code opérationnel", et défini l'état de compilation des fichiers dynamiques, qui peuvent être traités plus rapidement.
PrestaShop est compatible avec eAccelerator (http://eaccelerator.net/) ainsi que la nouvelle fonctionnalité OPcache de PHP 5.5.0 : http://www.php.net/manual/fr/intro.opcache.php.

Performance de MySQL

- Activer le cache de MySQL (ou demandez à votre hébergeur de le faire pour vous), et donnez-lui une taille élevée (par exemple, 256M).
- Si vous le pouvez, utilisez Percona Server (http://www.percona.com/software/percona-server), un remplacement direct de MySQL qui offre de nettes améliorations des performances par rapport au serveur MySQL standard grâce à son moteur de base de données XtraDB.
  Vous trouverez une comparaison des performances sur cette page : http://www.percona.com/doc/percona-server/5.5/feature_comparison.html

Autres améliorations

Si possible, placez vos éléments statiques sur plusieurs domaines ou sous-domaines, afin de profiter de connexions HTTP parallèles. Pour ce faire, ouvrez le fichier `/config/defines.inc.php` et ajoutez-y ces lignes (adaptées à vos propres besoins) :

```
if ( $_SERVER['REMOTE_ADDR'] != '127.0.0.1' )
{
  define( '_THEME_IMG_DIR_',   'http://img2.votredomaine.com/'       );
  define( '_THEME_CSS_DIR_',   'http://css.votredomaine.com/'        );
  define( '_THEME_JS_DIR_',    'http://js.votredomaine.com/'         );
  define( '_THEME_CAT_DIR_',   'http://img1.votredomaine.com/c/'     );
  define( '_THEME_PROD_DIR_',  'http://img1.votredomaine.com/p/'     );
  define( '_THEME_MANU_DIR_',  'http://img1.votredomaine.com/m/'     );
  define( '_PS_IMG_',          'http://img1.votredomaine.com/'       );
  define( '_PS_ADMIN_IMG_',    'http://img1.votredomaine.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/'                     );
}
```

Vous trouverez également de bons conseils sur notre site:

- Nos astuces d'optimisation : http://www.prestashop.com/fr/conseils-optimisation
- 10 Best Tips to speed up your PrestaShop store: http://www.prestashop.com/blog/en/10-best-tips-to-speed-up-your-prestashop-store-3/ (en anglais)

Autres recommandations

Safe Mode

Le Safe Mode de PHP est obsolète dans la dernière version de PHP, et ne devrait plus être utilisé. En particulier, sous PrestaShop, il peut rendre vos modules de paiement inutilisable.

Mises à jour

Le code PHP de vos applications est la seule vulnérabilité de votre serveur. Il vous faut donc vous assurez de toujours utiliser les dernières versions de vos applications : PHP, MySQL, Apache et toute les autres applications en place sur votre serveur.

URL simplifiées dans Nginx

La plupart des instructions serveur de cette page se rattachant au serveur web Apache. Mais certains d'entre vous préfèrent sans doute utilise le serveur Nginx. PrestaShop fonctionne bien avec Nginx, mais n'est pas capable de générer les bonnes règles de redirection pour profiter des URL simplifiées.

Voici quelques règles à mettre dans votre fichier `nginx.conf` afin de faire fonctionner les URL simplifiées.

```
location /PRESTASHOP_FOLDER/ {
  index /PRESTASHOP_FOLDER/index.php;

  rewrite ^/PRESTASHOP_FOLDER/api/?(.*)$ /PRESTASHOP_FOLDER/webservice/dispatcher.php?url=$1 last;
  rewrite ^/PRESTASHOP_FOLDER/([0-9])(-[_a-zA-Z0-9-]*)?/[_a-zA-Z0-9-]*.jpg$ /PRESTASHOP_FOLDER/img/p/$1/$1$2.
jpg last;
  rewrite ^/PRESTASHOP_FOLDER/([0-9])([0-9])(-[_a-zA-Z0-9-]*)?/[_a-zA-Z0-9-]*.jpg$ /PRESTASHOP_FOLDER/img/p/$1
/$2/$1$2$3.jpg last;
  rewrite ^/PRESTASHOP_FOLDER/([0-9])([0-9])([0-9])(-[_a-zA-Z0-9-]*)?/[_a-zA-Z0-9-]*.jpg$ /PRESTASHOP_FOLDER/img
/p/$1/$2/$3/$1$2$3$4.jpg last;
  rewrite ^/PRESTASHOP_FOLDER/([0-9])([0-9])([0-9])([0-9])(-[_a-zA-Z0-9-]*)?/[_a-zA-Z0-9-]*.jpg$
/PRESTASHOP_FOLDER/img/p/$1/$2/$3/$4/$1$2$3$4$5.jpg last;
  rewrite ^/PRESTASHOP_FOLDER/([0-9])([0-9])([0-9])([0-9])([0-9])(-[_a-zA-Z0-9-]*)?/[_a-zA-Z0-9-]*.jpg$
/PRESTASHOP_FOLDER/img/p/$1/$2/$3/$4/$5/$1$2$3$4$5$6.jpg last;
  rewrite ^/PRESTASHOP_FOLDER/([0-9])([0-9])([0-9])([0-9])([0-9])([0-9])(-[_a-zA-Z0-9-]*)?/[_a-zA-Z0-9-]*.jpg$
/PRESTASHOP_FOLDER/img/p/$1/$2/$3/$4/$5/$6/$1$2$3$4$5$6$7.jpg last;
  rewrite ^/PRESTASHOP_FOLDER/([0-9])([0-9])([0-9])([0-9])([0-9])([0-9])([0-9])(-[_a-zA-Z0-9-]*)?/[_a-zA-Z0-9-]
*.jpg$ /PRESTASHOP_FOLDER/img/p/$1/$2/$3/$4/$5/$6  /$7/$1$2$3$4$5$6$7$8.jpg last;
  rewrite ^/PRESTASHOP_FOLDER/([0-9])([0-9])([0-9])([0-9])([0-9])([0-9])([0-9])([0-9])(-[_a-zA-Z0-9-]*)?/[_a-zA-
Z0-9-]*.jpg$ /PRESTASHOP_FOLDER/img/p/$1/$2/$3/$4/$5/$6/$7/$8/$1$2$3$4$5$6$7$8$9.jpg last;
  rewrite ^/PRESTASHOP_FOLDER/c/([0-9]+)(-[_a-zA-Z0-9-]*)/[_a-zA-Z0-9-]*.jpg$ /PRESTASHOP_FOLDER/img/c/$1$2.jpg
last;
  rewrite ^/PRESTASHOP_FOLDER/c/([a-zA-Z-]+)/[a-zA-Z0-9-]+.jpg$ /PRESTASHOP_FOLDER/img/c/$1.jpg last;
  rewrite ^/PRESTASHOP_FOLDER/([0-9]+)(-[_a-zA-Z0-9-]*)/[_a-zA-Z0-9-]*.jpg$ /PRESTASHOP_FOLDER/img/c/$1$2.jpg
last;
  try_files $uri $uri/ /PRESTASHOP_FOLDER/index.php?$args;
}
```

Notez bien que cet exemple utilise `/PRESTASHOP_FOLDER/` comme marqueur du dossier PrestaShop. Vous devez remplacer toutes les occurrences de `/PRESTASHOP_FOLDER/` par le bon chemin de votre installation de PrestaShop.

Par exemple, si PrestaShop se trouve à la racine de votre serveur, remplacez `/PRESTASHOP_FOLDER/` par `/` , tout simplement :

```
location / {
  index /index.php;

  rewrite ^/api/?(.*)$ /webservice/dispatcher.php?url=$1 last;
  rewrite ^/([0-9])(-[_a-zA-Z0-9-]*)?/[_a-zA-Z0-9-]*.jpg$ /img/p/$1/$1$2.jpg last;
  rewrite ^/([0-9])([0-9])(-[_a-zA-Z0-9-]*)?/[_a-zA-Z0-9-]*.jpg$ /img/p/$1/$2/$1$2$3.jpg last;
  rewrite ^/([0-9])([0-9])([0-9])(-[_a-zA-Z0-9-]*)?/[_a-zA-Z0-9-]*.jpg$ /img/p/$1/$2/$3/$1$2$3$4.jpg last;
  rewrite ^/([0-9])([0-9])([0-9])([0-9])(-[_a-zA-Z0-9-]*)?/[_a-zA-Z0-9-]*.jpg$ /img/p/$1/$2/$3/$4/$1$2$3$4$5.
jpg last;
  rewrite ^/([0-9])([0-9])([0-9])([0-9])([0-9])(-[_a-zA-Z0-9-]*)?/[_a-zA-Z0-9-]*.jpg$ /img/p/$1/$2/$3/$4/$5
/$1$2$3$4$5$6.jpg last;
  rewrite ^/([0-9])([0-9])([0-9])([0-9])([0-9])([0-9])(-[_a-zA-Z0-9-]*)?/[_a-zA-Z0-9-]*.jpg$ /img/p/$1/$2/$3/$4
/$5/$6/$1$2$3$4$5$6$7.jpg last;
  rewrite ^/([0-9])([0-9])([0-9])([0-9])([0-9])([0-9])([0-9])(-[_a-zA-Z0-9-]*)?/[_a-zA-Z0-9-]*.jpg$ /img/p/$1/$2
/$3/$4/$5/$6/$7/$1$2$3$4$5$6$7$8.jpg last;
  rewrite ^/([0-9])([0-9])([0-9])([0-9])([0-9])([0-9])([0-9])([0-9])(-[_a-zA-Z0-9-]*)?/[_a-zA-Z0-9-]*.jpg$ /img
/p/$1/$2/$3/$4/$5/$6/$7/$8/$1$2$3$4$5$6$7$8$9.jpg last;
  rewrite ^/c/([0-9]+)(-[_a-zA-Z0-9-]*)/[_a-zA-Z0-9-]*.jpg$ /img/c/$1$2.jpg last;
  rewrite ^/c/([a-zA-Z-]+)/[a-zA-Z0-9-]+.jpg$ /img/c/$1.jpg last;
  rewrite ^/([0-9]+)(-[_a-zA-Z0-9-]*)/[_a-zA-Z0-9-]*.jpg$ /img/c/$1$2.jpg last;
  try_files $uri $uri/ /index.php?$args;
}
```

Si votre installation de PrestaShop utilise le mode multiboutique, vous devez ajouter quelques lignes pour chaque boutique. Par exemple, si l'une de vos boutiques utilise l'identifiant "high-tech" :

```
location /PRESTASHOP_FOLDER/high-tech/ {
  rewrite ^/PRESTASHOP_FOLDER/high-tech/(.*)$ /PRESTASHOP_FOLDER/$1 last;
  try_files $uri $uri/ /PRESTASHOP_FOLDER/index.php?$args;
}
```

Divers

L'arborescence des fichiers de PrestaShop

Les développeurs de PHP ont fait de leur mieux pour séparer les différentes parties du logiciel de manière claire et intuitive.

Voici comment les fichiers sont organisés :

- `/admin` : contient tous les fichiers de PrestaShop relatifs au back-office. Si vous cherchez à accéder à ce dossier à l'aide de votre navigateur, il vous sera demandé de vous authentifier, pour des raisons de sécurité. **Important** : faites en sorte que ce dossier reste protéger par des fichiers `.htaccess` et `.htpasswd`.
- `/cache`: contient des dossiers temporaires qui sont générés et réutilisés afin d'alléger la charge du serveur.
- `/classes` : contient tous les fichiers relatifs au modèle Objet de PrestaShop. Chaque fichier représente (et contient) une classe PHP, et ses méthodes et propriétés.
- `/config` : contient tous les fichiers de configuration de PrestaShop. Ne modifiez **jamais** ces fichiers, sauf si on vous le demande expressément, car ils sont gérés directement par l'installateur et le back-office de PrestaShop.
- `/controllers` : contient tous les fichiers relatifs aux contrôleurs de PrestaShop (dans le cadre Modèle-Vue-Contrôleur, ou "MVC"), l'architecture logicielle sur laquelle repose PrestaShop. Chaque fichier contrôle une partie précise de PrestaShop.
- `/css` : contient tous les fichiers CSS qui ne sont pas liés aux thèmes ; de fait, ils sont la plupart du temps utilisés par le back-office de PrestaShop.
- `/docs` : contient un peu de documentation. **Note** : vous devriez les effacer dans un environnement de production.
- `/download` : contient tous vos produits numériques, pouvant être téléchargés : fichiers PDFs, MP3s, etc.
- `/img` : contient toutes les images par défaut de PrestaShop, c'est à dire celles qui n'appartiennent pas au thème. C'est ici que vous trouverez les sous-dossiers des images pour les catégories de produits (`/c`, celles des produits (sous-dossier `/p`) et celle pour le back-office lui-même (sous-dossier `/admin`}}.
- `/install` : contient tous les fichiers relatifs à l'installateur de PrestaShop. Vous devriez l'effacer après installation, afin d'améliorer la sécurité.
- `/js` : contient tous les fichiers JavaScript qui ne sont pas liés aux thèmes. La plupart appartiennent au back-office. C'est également ici que vous trouverez le framework jQuery.
- `/localization` : contient tous les fichiers de localisation de PrestaShop – c'est à dire, les fichiers qui contiennent des informations locales, comme la monnaie, la langue, les règles de taxes et les groupes de règles de taxes, les états et autres unités utilisées dans le pays choisi (par exemple, le volume en litre, le poids en kilogrammes, etc.).
- `/log` : contient les fichiers de log générés par PrestaShop lors de diverses étapes, par exemple pendant le processus d'installation.
- `/mails` : contient tous les fichiers HTML et textes relatifs aux e-mails envoyés par PrestaShop. Chaque langue dispose de son propre dossier, où vous pouvez modifier manuellement ce que vous souhaitez.
- `/modules` : contient tous les modules de PrestaShop, chacun dans son propre dossier. Si vous souhaitez enlever définitivement un module, commencez par le désinstaller depuis le back-office, puis effacez son dossier à la main.
- `/override` : il s'agit d'un dossier particulier, apparu à partir de la version 1.4 de PrestaShop. En utilisant la convention de nommage de dossier et fichiers de PrestaShop, il devient possible de créer des fichiers qui supplantent les classes et contrôleurs par défaut de PrestaShop. Cela vous permet de modifier le comportement fondamental de PrestaShop sans toucher aux fichiers originaux, ce qui les garde intact en prévision de la prochaine mise à jour.
- `/pdf` : contient tous les fichiers template (`.tpl`) relatifs à la génération de fichier PDF (factures, bons, etc.). Modifiez ces fichiers afin de modifier la présentation des fichiers PDF de PrestaShop.
- `/themes` : contient tous les thèmes actuellement installés, chacun dans son propre dossier.
- `/tools` : contient les outils externes qui ont été intégré à PrestaShop. Par exemple, c'est ici que vous trouverez Smarty (moteur de thème), FPDF (générateur de fichiers PDF), Swift (expéditeur d'e-mails), PEAR XML Parser (outil PHP).
- `/translations` : contient un sous-dossier pour chaque langue. Cependant, si vous souhaitez modifier les traductions, vous devez utiliser l'outil interne de PrestaShop, et **surtout pas** les modifier directement dans ce dossier.
- `/upload` : contient les fichiers qui ont été mis en ligne par les clients pour personnalisés vos produits (par exemple, une image à imprimer sur un mug).
- `/webservice` : contient les fichiers qui permettent aux applications tierces de se connecter à PrestaShop, au travers de son API.

Déplacer PrestaShop

Une installation de PrestaShop ne reste pas à vie au même emplacement physique. Il existe de nombreuses raisons pour lesquelles vous pourriez avoir besoin de déplacer les fichiers et données de PrestaShop :

- Déplacer votre boutique depuis votre ordinateur vers votre serveur en ligne.
- Déplacer votre boutique depuis un sous-domaine de test vers le domaine principal.
- Déplacer votre boutique depuis un serveur vers un autre.
- Déplacer votre boutique depuis un sous-domaine vers un autre.

Pour chacune de ces circonstances, vous devez faire attention à déplacer correctement tous vos fichiers (dont vos images personnalisées, vos thèmes, les modules que vous avez acheté...) et toutes vos données (qui sont contenues dans votre base de données MySQL).

Déplacer PrestaShop vers un nouveau serveur

Voici les principales étapes à valider lors d'un changement de serveur, ou lors du transfert depuis votre disque dur jusqu'au serveur en ligne :

1. Mettez votre boutique en mode de maintenance, afin de ne pas perdre de nouveaux clients ou des commandes pendant le déplacement des données.
   Rendez-vous sur votre back-office, et dans la page "Maintenance du menu "Paramètres avancés", réglez l'option "Activer la boutique" à "Non".
2. Déplacez vos fichiers :
   a. **Faites une sauvegarde de tous vos fichier** : connectez-vous à votre serveur FTP, et copiez tous les fichiers et dossiers vers votre disque dur.
   b. **Transférez les fichiers vers le nouvel hébergeur** : connectez-vous au serveur FTP de votre nouvel hébergeur, et copiez-y tous les fichiers et dossier que vous avez téléchargé précédemment sur votre disque dur, tels quels.
3. Déplacez vos données :
   a. **Faites une sauvegarde de votre base de données (un "dump")** : connectez-vous à phpMyAdmin, cliquez sur l'onglet "Exporter", sélectionnez la base de données de votre installation de PrestaShop, et cliquez sur le bouton "Exécuter". Enregistrez le fichier sur votre disque dur. Si phpMyAdmin arrive à expiration avant l'export de toutes vos données, contactez votre hébergeur.
   b. **Transférez votre dump SQL vers votre nouvelle base de données** : connectez-vous au phpMyAdmin de votre nouveau serveur, cliquez sur le bouton "Importer", cliquez sur le bouton "Parcourir...", trouvez le fichier SQL que vous avez téléchargé, et cliquez sur le bouton "Exécuter" pour le mettre en ligne. Si phpMyAdmin arrive a expiration avant l'import de toutes vos données, contactez votre nouvel hébergeur.
4. Configurer votre boutique :
   a. Sur le nouveau serveur, ouvrez le fichier `/config/settings.inc.php`et mettez à jour les réglages de la base de données (avec vos propres informations plutôt que les exemples donnés ci-dessous) :
      - `define('_DB_SERVER_', 'sql.domainname.com');`
      - `define('_DB_NAME_', 'prestashop');`
      - `define('_DB_USER_', 'PS-user');`
      - `define('_DB_PASSWD_', 'djsf15');`
      - `define('_DB_PREFIX_', 'ps_');`
   b. Connectez-vous à votre back-office, rendez-vous dans la page "SEO & URLs" du menu "Préférences", allez dans la section "Configuration des URL" et modifiez le nom du domaine pour y mettre votre nouveau domaine. Faites de même pour votre domaine SSL. Dans les faits, cela mettra à jour la table ps_shop_url (ainsi que les lignes "PS_SHOP_DOMAIN" et "PS_SHOP_DOMAIN_SSL" de la table "ps_configuration" pour des raisons de rétrocompatibilité).
5. Connectez-vous à votre serveur FTP et effacez le contenu des dossiers suivants, sauf le fichier `index.php`:
   - `/cache/smarty/cache`
   - `/cache/smarty/compile`
6. Dans votre back-office, dans la page "Maintenance", remettez l'option "Activer la boutique" à "Oui".

C'est fait ! Vérifiez bien que tous vos liens fonctionnent, que tous vos produits, images, modules et thèmes sont toujours en place, et essayez de créer un nouveau compte ainsi que de passer une commande afin de vous assurer que votre boutique fonctionne comme attendu.

Déplacer PrestaShop vers un nouveau domaine

Voici les principales étapes à valider lors d'un changement de domaine. Ce sont en fait une version simplifiée des étapes ci-dessus, sauf que nous ne touchons pas aux données, qui restent sur le même serveur SQL.

1. Mettez votre boutique en mode de maintenance, afin de ne pas perdre de nouveaux clients ou des commandes pendant le déplacement des données.
   Rendez-vous sur votre back-office, et dans la page "Maintenance du menu "Paramètres avancés", réglez l'option "Activer la boutique" à "Non".
2. Déplacez vos fichiers :
   a. **Faites une sauvegarde de tous vos fichier** : connectez-vous à votre serveur FTP, et copiez tous les fichiers et dossiers vers votre disque dur.
   b. **Transférez les fichiers vers le nouvel hébergeur** : connectez-vous au serveur FTP de votre nouvel hébergeur, et copiez-y tous les fichiers et dossier que vous avez téléchargé précédemment sur votre disque dur, tels quels.
3. Configuration
   a. Sur le nouveau serveur, ouvrez le fichier `/config/settings.inc.php`et mettez à jour les réglages de la base de données (avec vos propres informations plutôt que les exemples donnés ci-dessous) :
      - `define('_DB_SERVER_', 'sql.domainname.com');`
      - `define('_DB_NAME_', 'prestashop');`
      - `define('_DB_USER_', 'PS-user');`
      - `define('_DB_PASSWD_', 'djsf15');`
      - `define('_DB_PREFIX_', 'ps_');`

    b. Connectez-vous à votre back-office, rendez-vous dans la page "SEO & URLs" du menu "Préférences", allez dans la section "Configuration des URL" et modifiez le nom du domaine pour y mettre votre nouveau domaine. Faites de même pour votre domaine SSL. Dans les faits, cela mettra à jour les lignes "PS_SHOP_DOMAIN" et "PS_SHOP_DOMAIN_SSL" de la table SQL "ps_configuration".
4. Connectez-vous à votre serveur FTP et effacez le contenu des dossiers suivants, sauf le fichier `index.php`:
    - `/tools/smarty/cache`
    - `/tools/smarty/compile`
    - `/tools/smarty_v2/cache`
    - `/tools/smarty_v2/compile`
5. Dans votre back-office, dans la page "Maintenance", remettez l'option "Activer la boutique" à "Oui".

C'est fait ! Vérifiez bien que tous vos liens fonctionnent, que tous vos produits, images, modules et thèmes sont toujours en place, et essayez de créer un nouveau compte ainsi que de passer une commande afin de vous assurer que votre boutique fonctionne comme attendu.