Mettre en place votre environnement de développement

Table des matières

Mettre en place votre environnement de développement

Maintenant que vous souhaitez développer pour PrestaShop, il serait préférable que vous disposiez d'un environnement local de travail. L'avantage principal de cet environnement local sera de vous permettre de ne pas devoir mettre un fichier en ligne afin de le tester. Autre avantage, l'environnement local vous permet de tester du code sans risque de mettre à mal votre boutique de production. Avoir à disposition un environnement local, c'est la première étape essentielle du développement web.

Installer PrestaShop 1.5 en local

Prérequis

PrestaShop nécessite la configuration serveur suivante pour fonctionner :

- Système: Unix, Linux ou Windows.
- Serveur web: Apache Web Server 1.3 ou n'importe quelle version suivante.
- PHP : 5.2 ou plus.
- MySQL : 5.0 ou plus.
- Au moins 32 Mo de RAM. 64 Mo pour être confortable. Plus il y a de mémoire, mieux c'est...

> ⚠ PrestaShop peut fonctionner avec PHP 5.1.3 et plus, mais les versions inférieures à la 5.2 ont des bugs qui peuvent empêcher certaines fonctionnalités de fonctionner normalement, comme la gestion de fuseaux horaires.
>
> PrestaShop peut également fonctionner avec Microsoft IIS 6.0 et plus, et nginx 1.0 et plus.

Installer un environnement local

Installer n'importe quelle application web localement requiert que vous installiez d'abord un environnement adéquat, à savoir le serveur web Apache, l'interpréteur de langage PHP, le serveur de base de données MySQL, et idéalement l'outil phpMyAdmin. L'ensemble est connu sous l'acronyme AMP : Apache+MySQL+PHP, les systèmes d'exploitation, ce qui donne WAMP (Windows+Apache+MySQL+PHP), MAMP (Mac OS X+...) et LAMP (Linux+...). Étant donné que tous les logiciels de ce pack sont open-source, les installeurs de ces environnements sont la plupart du temps gratuits.

Voici une sélection d'installeur AMP gratuits :

- XAMPP : http://www.apachefriends.org/en/xampp.html (Windows, Mac OS X, Linux, Solaris)
- WampServer : http://www.wampserver.com/en/ (Windows)
- EasyPHP : http://www.easyphp.org/ (Windows)
- MAMP : http://www.mamp.info/ (Mac OS X)

Configurer PHP

PrestaShop nécessite quelques additions à PHP et MySQL afin de fonctionner au mieux. Faites en sorte que votre configuration PHP dispose des réglages et outils suivants :

- Bibliothèque GD.
- Extension Dom.
- `allow_url_fopen` activé.

Voici une section du fichier `php.ini` (le fichier de configuration de PHP) :

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

# also recommended
register_globals = Off
magic_quotes_gpc = Off
allow_url_include = Off
```

⚠ La bibliothèque GD (http://www.boutell.com/gd/) permet à PrestaShop de retravailler les images que vous mettez en ligne, notamment de les redimensionner.

L'extension Dom permet d'analyser des documents XML. PrestaShop l'utilise pour diverses fonctionnalités, comme le Localisateur de Boutique. Elle est également utilisée pour certains modules, ainsi que la bibliothèque `pear_xml_parse`.

La directive `allow_url_fopen` permet aux modules d'accéder à des fichiers distants, ce qui est un élément essentiel du processus de paiement, entre autre choses. Il est donc impératif qu'elle soit sur **ON**.

Télécharger et extraire les fichiers de PrestaShop

Vous pouvez télécharger la dernière version de PrestaShop à l'adresse http://www.prestashop.com/en/downloads.

Vous pouvez télécharger la version de développement (non stable) sur Github : https://github.com/PrestaShop/PrestaShop/archive/master.zip

Décompressez l'archive, et mettez les fichiers à la racine web du serveur AMP que vous avez choisi d'utiliser :

- XAMPP : `C:\xampp\htdocs` or `/Applications/xampp/htdocs`
- WampServer : `C:\wamp\www`
- EasyPHP : `C:\easyphp\www`
- MAMP : `/Applications/MAMP/htdocs/`

Créer une base de données pour votre boutique locale

Ouvrez l'outil phpMyAdmin dans votre navigateur. Son emplacement dépend du pack AMP que vous avez choisi :

- `http://127.0.0.1/phpmyadmin` (XAMPP, WampServer, MAMP),
- `http://127.0.0.1/mysql` (EasyPHP)

Dans l'onglet "Base de données", donnez un nom à votre base et cliquez sur le bouton "Créer une base de données".

Installer PrestaShop

Lancer l'installateur de PrestaShop, qui doit se trouver à l'adresse `http://127.0.0.1/prestashop`, et suivez-en les instructions.

Vous pouvez lire le Guide de Démarrage pour plus d'information.

Configurer PrestaShop

L'installation par défaut de PrestaShop est configurée afin de fournir un environnement stable et sécurisé tant pour l'administrateur de la boutique que pour ses clients.

En tant que développeur, vous pouvez et devez appliquer certaines modifications à cette installation par défaut pour vous aider à mieux développer, à repérer les problèmes plus rapidement, et en général pour vous aider à faire de PrestaShop un meilleur outil.

Désactiver le cache et forcer la compilation de Smarty

Lorsque votre développement a un impact sur le front-office, que vous soyez en train de créer un thème ou simplement un module qui envoie des informations au client, vous devriez forcer la compilation des fichiers de template et désactiver le cache, afin de toujours voir le résultat direct de vos modifications.

Allez dans le menu "Paramètres avancés" de PrestaShop, ouvrez la page "Performances", et modifiez les réglages suivants :

- Cache des templates : passez-le à "Forcer la compilation à chaque appel ".
- Cache : désactivez-le.

> Le fait de forcer la compilation Smarty ralentira toujours le temps de chargement d'une page. Assurez-vous donc que votre boutique de production est configurée pour ne recompiler que les fichiers mis à jour, et que son cache est activé.

Afficher les messages d'erreur

Les réglages par défaut de PrestaShop font en sorte que le client ne puisse jamais voir de message d'erreur serveur ni code de débogage.

En ce qui vous concerne, il est dans votre intérêt de voir les messages d'erreur afin de repérer tout problème potentiel avec votre code. Pour ce faire, ouvrez le fichier `/config/defines.inc.php`, et modifiez les lignes suivantes (en passant le mode de développement de false à true) :

```
/* Debug only */
define('_PS_MODE_DEV_', false);
```

Activer les méthodes de débogage

PrestaShop dispose de méthode spécifiques à destination des développeurs : `p($variable)` et `d($variable)`. Elles sont à utiliser pour afficher le contenu d'une variable. Dans les faits, il s'agit d'une mise en forme de la méthode `print_r()` (http://php.net/manual/fr/function.print-r.php) :

La method p()

```
echo '<xmp style="text-align: left;">';
print_r($variable);
echo '</xmp><br />';
return $variable;
```

Ce type de fonction est typique des sessions de développement PHP, et PrestaShop vous permet de ne pas devoir réinventer la roue en l'intégrant directement dans son code.

`p()` est la méthode principale, et `d()` fonctionne de la même manière, sauf qu'elle fait appel à `die()` au lieu de renvoyer la variable :

La méthode d()

```
echo '<xmp style="text-align: left;">';
print_r($variable);
echo '</xmp><br />';
die('END');
```

Ces deux méthodes vous permettent de vérifier l'état d'une variable à un emplacement spécifique de votre code.

En plus de cela, PrestaShop met en place les méthodes `ppp()` et `ddd()`, qui sont les alias respectifs de `p()` et `d()`. Ils fonctionnent strictement de la même manière, mais sont souvent plus faciles à trouver dans un long bloc de code.

Ces méthodes de débogage ne sont cependant pas disponibles par défaut. Afin de les rendre disponibles, vous devez activer le mode Debug, en modifiant cette ligne du fichier `/config/defines.inc.php` :

```
define('_PS_MODE_DEV_', true);
```

Activer le mode multiboutique

PrestaShop 1.5 est capable d'héberger plus d'une boutique au sein d'une même installation du logiciel. De nombreux administrateurs de boutique choisissent d'activer cette fonctionnalité, et cela peut avoir un certain impact sur la manière dont PrestaShop fonctionne. Vous devriez donc faire en sorte que votre création fonctionne aussi bien en mode monoboutique qu'en multiboutique.

Il est facile d'activer le mode multiboutique : rendez-vous dans la page des préférences générales, et passez l'option "Activer le multiboutique" à "Oui".

Vous pouvez activer et désactiver le mode multiboutique en fonction de vos besoins.

Vous en apprendrez plus sur le mode multiboutique en lisant le Guide de l'Utilisateur PrestaShop 1.5.

À propos des fichiers de configuration

Il y a trois principaux fichiers de configuration, tous situés dans le dossier `/config` :

- `config.inc.php`
- `defines.inc.php`
- `smarty.inc.php`

config.inc.php

Il s'agit du fichier de configuration principal de PrestaShop. Vous n'avez rien à modifier ici pour le moment.

defines.inc.php

Ce fichier contient les valeurs constantes de PrestaShop.

Il contient également l'emplacement des fichiers et dossiers de PrestaShop. Si vous avez besoin de modifier leur emplacement, n'oubliez pas de conserver à portée de main le chemin original, par exemple dans un commentaire PHP, au cas où vous deviez revenir en arrière plus tard.

Lorsque vous êtes en mode de développement/débogage, vous devez faire en sorte de voir tous les messages d'erreur serveur :

- Mettez `define('_PS_MODE_DEV_', false);` à "`true`".

Au contraire, en mode production, vous devez impérativement cacher ces messages ! Assurez-vous donc que :

- `define('_PS_MODE_DEV_', false);` est bien à "`false`".

smarty.inc.php

Ce fichier contient les réglages de configuration Smarty.

Le système de cache de Smarty devrait toujours être désactivé, car il n'est pas compatible avec celui de PrestaShop : laissez `$smarty->caching = false;` tel quel.

`$smarty->compile_check` devrait toujours être à "`false`".

`$smarty->debugging` vous donne accès aux informations de débogage de Smarty lors de l'affichage de la page. Cette option peut être plus lisiblement modifiée dans la page "Performances" des paramètres avancés : l'option Smarty "Console de débogage" vous propose de ne jamais afficher ces informations, de toujours les afficher, ou de ne l'afficher que dans le cas où vous ajoutez le paramètre d'URL `?SMARTY_DEBUG` dans l'adresse de la page à tester, ce qui peut se révéler très pratique.

En mode production, `$smarty->force_compile` doit est configuré à "false", car il accélérera de 30% l'affichage de votre page.

En revanche, lorsque vous modifiez un fichier `.tpl`, vous devez effacer le dossier `/tools/smarty/compile` (sauf son fichier `index.php`) afin de voir vos modifications appliquées).

Notez que tout ceci peut être fait directement depuis la page "Performances" du menu "Paramètres avancés".