

Fondamentaux de la création de thème

Contenu

- [Fondamentaux de la création de thème](#)
 - [Concepts](#)
 - [Architecture technique de PrestaShop](#)
 - [Modèle](#)
 - [Vue](#)
 - [Contrôleur](#)
 - [Concepts et informations techniques](#)
 - [Bonnes pratiques](#)
 - [Comment fonctionne un thème PrestaShop](#)
 - [Dossiers propres aux thèmes](#)
 - [Aperçu du dossier d'un thème](#)

Fondamentaux de la création de thème

Concepts

Le système de thème de PrestaShop est basé sur un moteur de modèle/template, appelé Smarty (<http://www.smarty.net/>), qui permet aux web-designers et développeurs de créer facilement leur propre thème, avec peu de connaissances techniques.

✔ Tous les designers et développeurs web devraient utiliser les outils suivants :

- Firefox : installer Firebug (<http://getfirebug.com/>), une extension gratuite permettant de facilement comparer et déboguer du CSS et son résultat.
- Firefox/Chrome : installer Web Developer (<http://chrispederick.com/work/web-developer/>), une extension gratuite qui ajoute de nombreux outils pour développeurs, directement dans le navigateur.
- Safari : activer le Web Inspector (lire http://developer.apple.com/library/safari/#documentation/AppleApplications/Conceptual/Safari_Developer_Guide/2SafariDeveloperTools/SafariDeveloperTools.html).
- Chrome : activer les Developer Tools (lire <https://developers.google.com/chrome-developer-tools/docs/overview>).
- Opera : utiliser Dragonfly (<http://www.opera.com/dragonfly/>), un environnement de débogage complet.
- Internet Explorer 8+ : utiliser les Developer Tools (lire <http://msdn.microsoft.com/fr-fr/library/dd565628%28v=vs.85%29.aspx>).
- Internet Explorer 6 et 7 : installer la IE Developer Toolbar (lire <http://www.microsoft.com/en-us/download/details.aspx?id=18359>), ou utiliser Firebug Lite (<https://getfirebug.com/firebuglite>).

Ils offrent de nombreux outils, parmi lesquels un outil de parcours du DOM, un éditeur CSS en direct, un inspecteur du réseau, etc., et apportent une aide essentielle lorsqu'il s'agit de déboguer du code HTML, CSS, JavaScript, et même les requêtes Ajax.

Architecture technique de PrestaShop

PrestaShop est basé sur une architecture 3-tiers :

- **Object/données.** L'accès à la base de données se fait par le biais des fichiers du dossier `/classes`.
- **Contrôle des données.** Le contenu envoyé à l'utilisateur est vérifié par les fichiers du dossier racine.
- **Apparence.** Tous les fichiers du thème sont dans le dossier `/themes`.

Pour plus d'information, lire la page Wikipédia http://fr.wikipedia.org/wiki/Architecture_trois_tiers.

Architecture 3-tiers de PrestaShop



Il s'agit du même principe que pour l'architecture MVC (Modèle-Vue-Contrôleur), en plus simple et plus accessible.

Pour plus d'informations sur le MVC, lisez la page Wikipédia <http://fr.wikipedia.org/wiki/Mod%C3%A8le-Vue-Contr%C3%B4leur>.

Notre équipe de développeurs a choisi de ne pas utiliser de framework PHP, tel que Zend Framework, Symfony ou CakePHP, afin d'obtenir une meilleure lisibilité du code, et donc de permettre des modifications plus rapides.

Cela permet également d'obtenir de meilleures performances, étant donné que le logiciel n'est fait que de lignes de code qu'il utilise effectivement, et qu'il ne contient pas un lit de bibliothèques génériques ajoutées.

Une architecture 3-tiers a de nombreux avantages :

- Il est plus facile de lire le code du logiciel.
- Les développeurs peuvent ajouter et corriger le code plus rapidement.
- Les designers et intégrateurs HTML peuvent travailler en toute sécurité dans le dossier /themes sans avoir à comprendre ou même lire une seule ligne de code PHP.
- Les développeurs peuvent travailler sur des données et modules supplémentaires que les intégrateurs HTML peuvent exploiter.

Modèle

Un modèle représente le comportement de l'application : traitement des données, interaction avec la base de données, etc.

Il décrit ou contient les données qui ont été traitées par l'application. Il gère ces données, et garantit leur intégrité.

Vue

Une vue est l'interface avec laquelle l'utilisateur interagit.

Son premier rôle est d'afficher les données qui ont été fournies par le modèle. Son second rôle est de gérer toutes les actions en provenance de l'utilisateur (clic de la souris, sélection d'un élément, boutons, etc.) et d'envoyer ces données au contrôleur.

La vue ne fait aucun traitement : elle ne fait qu'afficher le résultat d'un traitement réalisé par le modèle, et interagit avec l'utilisateur.

Contrôleur

Le contrôleur gère la synchronisation des événements entre le modèle et la vue, et met les deux à jour en fonction des besoins. Il reçoit tous les événements utilisateur, et déclenche les actions à réaliser.

Si une action a besoin de modifier des données, le contrôleur "demandera" au modèle de modifier ces données, et en retour le modèle notifiera la vue que les données ont été changées, afin que la vue puisse se mettre à jour.

Concepts et informations techniques

Bonnes pratiques

Voici une liste non exhaustive de bonnes pratiques à respecter :

1. Ne mélangez jamais du code XHTML et du code PHP ; utiliser les balises Smarty pour gérer votre contenu dynamique.
2. Ne mélangez jamais du code XHTML et du code CSS, placez le code dans un fichier `.css` à part.
3. Validez toujours votre code XHTML et votre code CSS grâce aux validateurs du W3C : <http://validator.w3.org/> pour le code HTML et XHTML, <http://jigsaw.w3.org/css-validator/> pour le code CSS.
4. 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.
5. Vérifiez toujours si une méthode n'existe pas déjà dans les classes disponibles.
6. Ne modifiez pas les fichiers du coeur de PrestaShop ; développez au 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. Lorsque vous effectuez des corrections, mettez la boutique en maintenance via le back-office de PrestaShop.
10. Utilisez les navigateurs Firefox, Google Chrome, IE9 ou Opera, et incitez les gens autour de vous à en faire autant.
11. Autant que possible, utilisez des CSS sprites (lire <http://www.alsacreations.com/tuto/lire/1068-sprites-css-background-position.html>).

Comment fonctionne un thème PrestaShop

Un thème PrestaShop est un ensemble de fichiers que vous pouvez modifier afin de personnaliser la mise en page de votre boutique.

Voici quelques points importants :

- Tous les thèmes ont leurs fichiers dans le dossier racine `/themes`.
- Chaque thème dispose de son propre sous-dossier, dans le thème racine.
- Chaque thème est fait de fichiers modèles (`.tpl`), d'images (`.gif`, `.jpg`, `.png`), d'un ou plusieurs fichiers CSS `.css`, et parfois même de fichiers JavaScript (`.js`).
- Chaque thème a un fichier `preview.jpg` dans son dossier racine, permettant au propriétaire du site d'avoir un aperçu du thème directement depuis le back-office, et donc de choisir son thème en connaissance de cause.

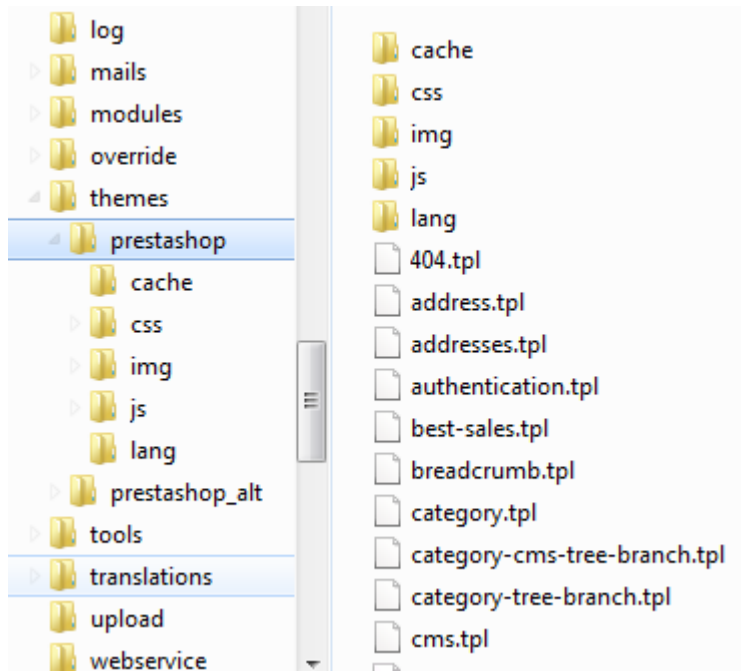
Dossiers propres aux thèmes

En tant que développeur de thèmes, il y a des dossiers auxquels vous devez faire attention :

- `/modules` : c'est ici que se trouvent tous les modules. Un module peut interagir avec des fichiers template, qui redéfinissent des parties du thème.
- `/themes` : c'est ici que se trouvent tous les thèmes. Le thème par défaut de la version 1.5 se trouve dans le dossier `/default` (celui de la version 1.4 était dans `/prestashop`).
- `/mails` : c'est ici que se trouvent tous les modèles d'e-mails. Ces modèles devraient correspondre autant que possible au style et au design du thème principal. Chaque sous-dossier contient des modèles par langue.
- `/img` : c'est ici que se trouvent toutes les images de la boutique. **Les images propres à un thème se trouvent dans le dossier du thème.**
- `/pdf` : c'est ici que se trouvent les modèles de documents.

Aperçu du dossier d'un thème

Voici un aperçu de l'arborescence de fichiers d'un thème PrestaShop (ici, le thème par défaut) :



- Le dossier `/css` contient tous les fichiers CSS.
- Le dossier `/img` contient toutes les images.
- Le dossier `/js` contient tous les fichiers JavaScript.
- Le dossier `/lang` contient toutes les traductions du thème. Ses droits d'accès devraient être à CHMOD 666 (par exemple), afin que l'outil de traduction puisse y avoir accès.
- La racine du dossier contient seulement les fichiers TPL (Smarty), ainsi que le fichier `preview.jpg`.

✔ Les dossiers `/css`, `/img` et `/js` sont facultatifs, un thème peut parfaitement fonctionner sans, il vous revient de les créer selon vos besoins.