

Characteristics of a front-office theme

Table of contents

- [Characteristics of a front-office theme](#)
 - [The theme's folders](#)
 - [Preview image](#)
 - [Translation files](#)
 - [Handling translations](#)
 - [Transmitting data to a PrestaShop theme](#)
 - [Cached files](#)
 - [Image files](#)
 - [Style-sheets \(CSS\) files](#)
 - [JavaScript \(JS\) files](#)
 - [Mobile version of the theme](#)
 - [Overriding the e-mails and PDF template files](#)
 - [Overriding a module's template files](#)

Characteristics of a front-office theme

The theme's folders

The main folder for the theme should have a unique name. It must be put in PrestaShop's root `/themes` folder in order to work.

A PrestaShop front-office theme is built using the following standard folders, within the main theme folder:

- `/cache`: contains temporary files that are generated and reused in order to alleviate the server's load.
- `/css`: contains the style sheets.
- `/img`: contains the theme's images.
- `/js`: contains the theme's JavaScript files.
- `/lang`: contains a sub-folder per translation, one for each language.
- `/mobile`: contains the mobile version of the theme.
- `/mails`: you can override PrestaShop's default mail templates with you own.
- `/modules`: you can provide modules with the theme, and override the default modules with your own.
- `/pdf`: you can override PrestaShop's default PDF templates with your own.

The last three sub-folders in this list (`/mails`, `/modules` and `/pdf`) are greatly useful in order to fully customize your installation of PrestaShop, and make the templates better match the theme's style. This way you can provide a full experience with your theme.

All of these folders are optional. Having them is simply a best practice: it makes your theme cleaner and more complete.



If you do not have time to build a specific mobile version of your theme, you can at first just rely on the default theme's mobile version, by copying its `/mobile` folder into your theme's folder.

Preview image

When displaying the theme in the theme selection page of the back-office, located in the "Themes" page under the "Preferences" menu, PrestaShop displays a specific image file. This file's characteristics are thus:

- Filename: `preview.jpg`
- Size: 180*200 pixels
- Format: JPEG (do not use GIF or PNG).

As a theme creator, you should make that preview image very recognizable: making a screen capture of the theme's home page is the best way to catch the PrestaShop administrator's eyes.

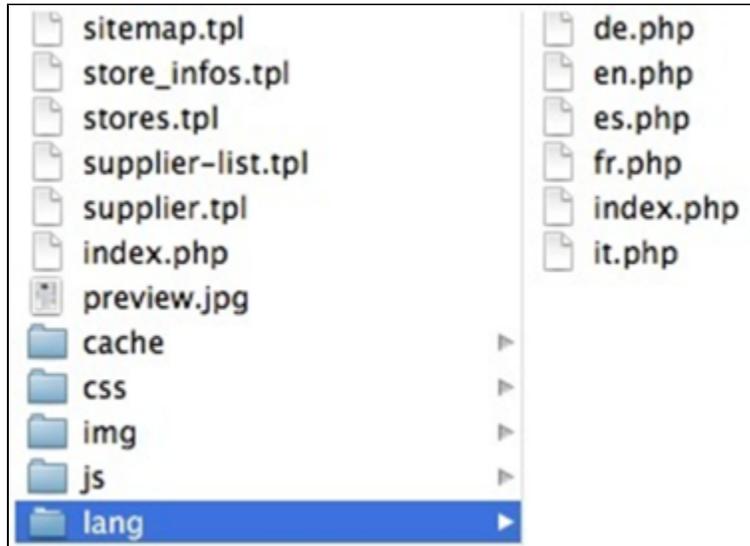
Even the default PrestaShop theme has its own:



Translation files

The translation files are stored in the `/lang` folder.

Each translation file must be named with the two-letters [ISO 639-1](#) code for its language: for instance, for French, use `fr.php`.



Translation files are automatically created by PrestaShop translation tool, which you can find in the "Translations" page of the "Localization" menu in the back-office.

Handling translations

All of your theme's text string should be enclosed in a Smarty function, like so:

```
{l s='My Text'}
```

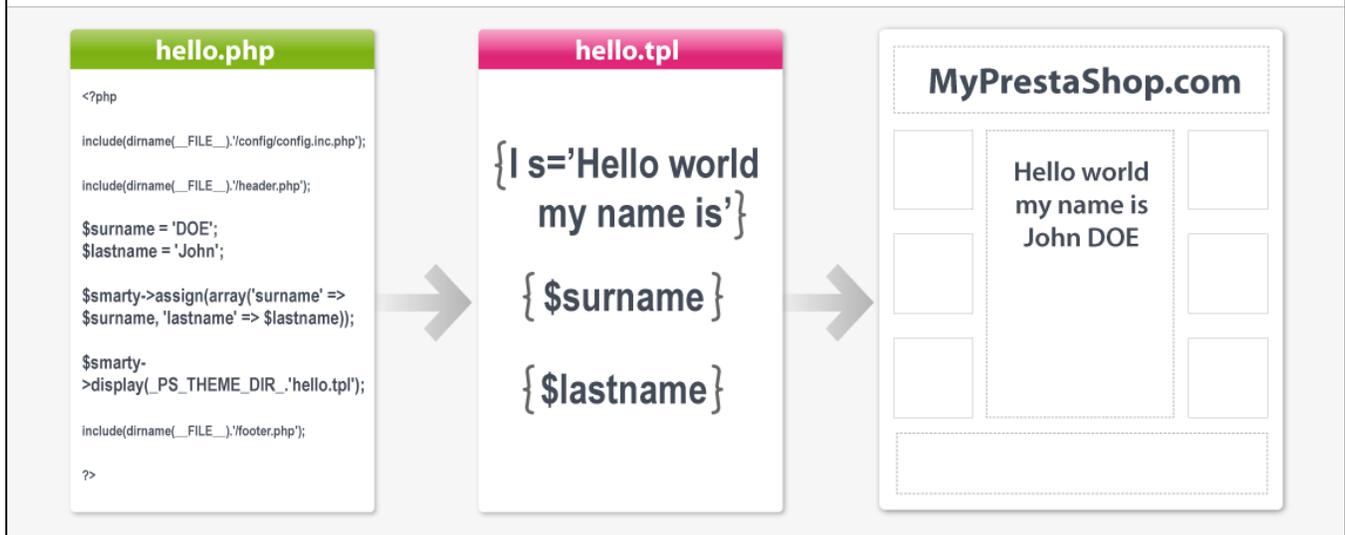
This will enable anyone to translate the theme into his own language, using the internal translation tool. In the "Modify translations" section, use the drop-down menu, choose "Front Office translation", then click on the flag of the language you wish to translate strings into. All the front-office strings will then be displayed: translate them all then save your work, and PrestaShop will create the translation file at the correct location.

Transmitting data to a PrestaShop theme

The following graphic explains how data is transmitted from PrestaShop's core to its theme. It uses the `l()` method to present how the theme displays its text in the chosen language, if it has been translated beforehand.

Transmitting data to a PrestaShop theme

Example with hello.php



Cached files

PrestaShop's CCC tool caches the theme's CSS and JavaScript files in a specific folder: `/cache`, right in the theme's main folder.

The cached CSS files are grouped into one file per page and per media.

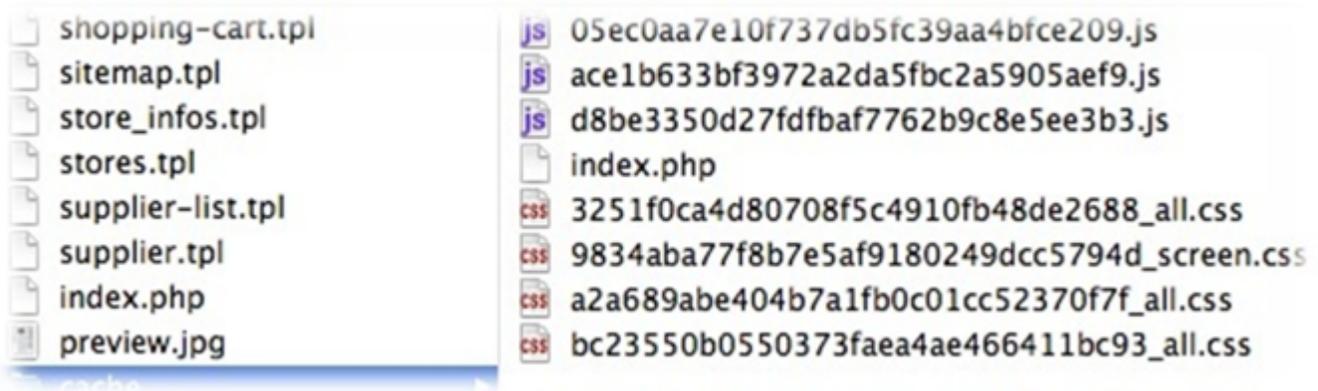
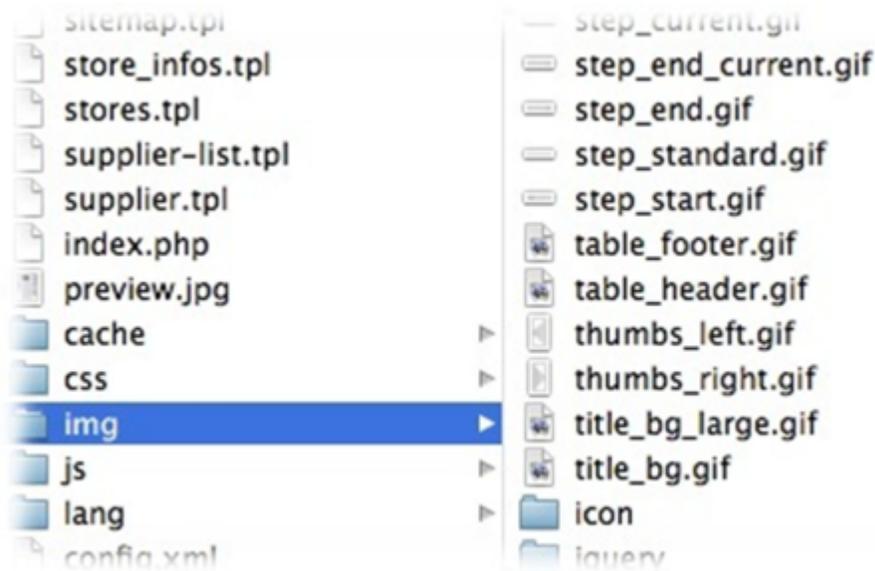


Image files

The theme's images are to be stored in the `/img` folder, within the theme's main folder.

In order to improve readability, the `/img` may contain sub-folders for specific groups of images, such as `/icons` or `/jquery`.

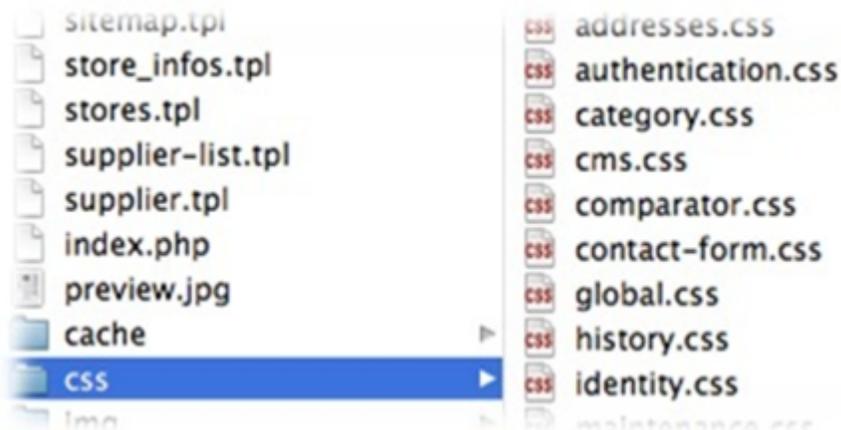


Style-sheets (CSS) files

All of the theme's CSS files should be stored in the `/css` folder (except when cached).

A theme usually features one CSS file for all the common rules within every pages of the theme. That file should be named `global.css`.

Each controller has its own CSS file, which contains all the rules that are specific to its page. For instance, for the product page, handled by the Product controller, the default theme has the `product.css` file.

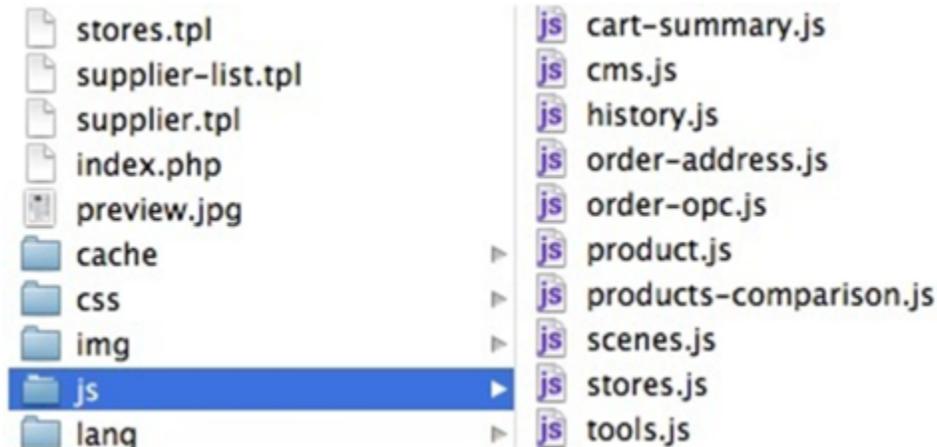


JavaScript (JS) files

All of the theme's JS files should be stored in the `/js` folder (except when cached).

Unlike CSS, there is no "common" file for JavaScript code.

Not all controllers have their own JS file. Each has its own needs, and therefore some have their own JavaScript files, some have none.



Mobile version of the theme

You can (and should) provide a mobile version of your theme. PrestaShop 1.5 is built to automatically use the theme available in the `/mobile` sub-folder in situations when the store is being viewed from a mobile device.

All of its files should be in the `/mobile` folder, within the theme's main folder.

The `/mobile` sub-folder has its own set of CSS, JS and image files, respectively stored in the `/mobile/css`, `/mobile/js` and `/mobile/img` folders.

 If you do not have time to build a specific mobile version of your theme, you can at first just rely on the default theme's mobile version, by copying its `/mobile` folder into your theme's folder.

Overriding the e-mails and PDF template files

You should provide your own mail and PDF templates, and make them match the theme's style.

The easiest way to get started is to copy PrestaShop's default mail templates folder (from the root `/mails` folder) into the theme's `/mails` sub-folder, and edit the files to fit your theme's style.

Likewise, copy PrestaShop's default PDF templates folder (from the root `/pdf` folder) into the theme's `/pdf` sub-folder, and edit the files to fit your theme's style.

Note that the mail files are duplicated as many folders as there are available languages. The translation is to be made using PrestaShop's integrated translation tool.

You can selectively override the default templates: if one of the file is missing in your version of the `/mails` or `/pdf` folder, PrestaShop will use the one from the equivalent root folder.

Overriding a module's template files

PrestaShop's modules can have templates of their own, in their `/views/templates/front`, `/views/templates/admin` and `/views/templates/hooks` folders.

For instance, the Bankwire module has the following template files:

- `/modules/bankwire/views/templates/front/payment_execution.tpl`
- `/modules/bankwire/views/templates/hook/payment.tpl`
- `/modules/bankwire/views/templates/hook/payment_return.tpl`

In order to override a module template file, your theme must have a file of the exact same name in the same exact file path, but within its own folder. So, for instance, for the three Bankwire templates above, the resulting file paths should be:

- /themes/my_theme/modules/bankwire/views/templates/front/payment_execution.tpl
- /themes/my_theme/modules/bankwire/views/templates/hook/payment.tpl
- /themes/my_theme/modules/bankwire/views/templates/hook/payment_return.tpl

You can override the template files, CSS files or JS files of installed modules using the same principle: same exact file name, same exact file path within the theme's folder.