

PrestaShop's developer tools

PrestaShop's developer tools

PrestaShop tries its best to make it easy for developers to build clean and working themes, modules and overriding classes. This is done through a set of developer tools that you can enable whenever you needs them.

By default, PrestaShop turns error reporting off, which makes that if your code fails, you will see a blank page with no information. This is because customers should not see anything from your server. You can get a lot of information using the following tools.

Most of these tools should only be enabled when you are in the development process, on your local installation. If you have to enable them on a production store, you should first disable the store so that only you can see the results of your tests.

Developer mode

PrestaShop's developer mode enables a series of tools to help developers:

- Sets PHP's `display_errors` to "on": errors are displayed in the browser instead of just blank pages.
- Sets PrestaShop's `_PS_DEBUG_SQL_` constant to "true": PrestaShop displays potential problems within your SQL queries.
- Sets PrestaShop's `_PS_DISPLAY_COMPATIBILITY_WARNING_` constant to "true": PrestaShop displays a warning message if a piece of code you are using is deprecated in the current version.

All in all, this mode displays a lot more error messages than the regular mode.

Enabling it is quite easy:

- On your server, open the `/config/defines.inc.php` file.
- Find this line: `define('_PS_MODE_DEV_', false);`
- In this line, change "false" to "true".
- Save your changes.

Now reload any blank or suspicious page: you should get a lot more information.

p() and d()

PHP developers are used to regularly call `print_r($var)` in order to know what lies inside `$var`. In order to properly display the content of `$var`, they often encapsulates `print_r()` within HTML or XML tags.

PrestaShop does that for you! `p()` and `d()` are both wrappers around `print_r()`. `p()` returns the content of the variable, while `d()` finishes with `die('END')`.

On top of that, PrestaShop defines the `ppp()` and `ddd()` method, which are respectively the aliases of `p()` and `d()`. They work exactly the same, but are often easier to search and find in a huge block of code.

These debug methods are not available by default. To activate them, you must enable the Developer mode (see above).

Code profiler

PrestaShop's profiler is a great way to discover the bottlenecks in your code: once enabled, each page on either the front-end or the back-end of your store will display a lot of information about the page and its files:

- Load time.
- Processed hooks.
- Memory usage.
- Number of database queries and their overall duration.
- Database stopwatches per query.
- Duplicate queries.
- Database table stress.
- ObjectModel instances.
- Included files.

This is a huge step in knowing which files or SQL query you should work on when you need to improve the performance of your store.

Enabling it is quite easy:

- On your server, open the `/config/defines.inc.php` file.
- Find this line (around line 43): `define('_PS_DEBUG_PROFILING_', false);`
- In this line, change "false" to "true".
- Save your changes.

Now load any page of your store, either front-end or back-end, and it will display a lot of statistics at the bottom of the page, below the regular content of that page.

In order to disable the Profiling Mode, simply open the `defines.inc.php` file and change the value back to "false" in the line mentioned above.

Here is the result from the home page:

© 2014 Ecommerce software by PrestaShop™

Load time: 263 ms Good boy! That's what I call a webservice!	Hook processing: 153 ms / 12.14 Mb 65 methods called in 39 modules	Memory peak usage: 25.7 Mb	Total cache size (in Cache class): 0.44 Mb
------------------------------------------------------------------------	------------------------------------------------------------------------------	-----------------------------------	---------------------------------------------------

Execution	Load time (ms)	Hook	Processing	Execution	Memory (Mb)	Total (Mb)
config	69	displayHeader	7.62 Mb in 71 ms	config	8.36	8.6
constructor	0	displayTop	2.89 Mb in 32 ms	constructor	0.00	8.6
init	24	displayFooter	0.69 Mb in 22 ms	init	3.89	12.4
checkAccess	0	displayHome	0.33 Mb in 8 ms	checkAccess	0.00	12.4
setMedia	5	displayHomeTabContent	0.28 Mb in 6 ms	setMedia	0.25	12.6
postProcess	0	displayHomeTab	0.10 Mb in 6 ms	postProcess	0.00	12.6
initHeader	0	displayNav	0.12 Mb in 4 ms	initHeader	0.01	12.6
initContent	123	displayTopColumn	0.08 Mb in 2 ms	initContent	11.24	24.1
initFooter	23	displayBanner	0.03 Mb in 2 ms	initFooter	0.70	24.7
display	20	moduleRoutes	0.00 Mb in 0 ms	display	0.73	25.7
		actionFrontControllerSetMedia	0.00 Mb in 0 ms			
		DisplayOverrideTemplate	0.00 Mb in 0 ms			
		actionObjectAddBefore	0.00 Mb in 0 ms			
		actionObjectAddAfter	0.00 Mb in 0 ms			
		actionDispatcher	0.00 Mb in 0 ms			
		actionObjectConnectionAddBefore	0.00 Mb in 0 ms			
		actionObjectConnectionAddAfter	0.00 Mb in 0 ms			

Stopwatch

Disables

Tables stress

ObjectModel instances

Files included

Stopwatch (with SQL_NO_CACHE) (total = 40)

Query	Time (ms)	Rows	Filesort	Group By	Location
<pre>INSERT INTO `ps_connections` (`id_guest`, `id_page`, `ip_address`, `http_referer`, `id_shop`, `id_shop_group`, `date_add`) VALUES ('3', '2', '2130786433', '', '1', '1', '2014-05-07 05:12:00')</pre>	6.266	1			in C:\wamp\www\ps16en\classes\ObjectModel.php:455 from /classes/Connection.php:154 from /classes/Connection.php:85 from /modules/statistics/statsdata.php:104 from /classes/Hook.php:509 from /tools/profiling/Hook.php:60 from /classes/controller/FrontController.php:808 from /tools/profiling/Controller.php:233 from /classes/Dispatcher.php:373 from /index.php:28
<pre>SELECT SQL_NO_CACHE cp.`id_product_attribute`, cp.`id_product`, cp.`quantity` AS cart_quantity, cp.`id_shop`, pl.`name`, p.`is_virtual`, pl.`description_short`, pl.`available_now`, pl.`available_later`, product_shop.`id_category_default`, p.`id_supplier`, p.`id_manufacturer`, product_shop.`on_sale`, product_shop.`ecotax`, product_shop.`additional_shipping_cost`, product_shop.`available_for_order`, product_shop.`price`, product_shop.`active`, product_shop.`unit_weight`, product_shop.`unit_price_ratio`...</pre>	3.974	1	Yes	Yes	in C:\wamp\www\ps16en\classes\Cart.php:508 from /modules/blockcart/blockcart.php:62 from /modules/blockcart/blockcart.php:185 from /modules/blockcart/blockcart.php:235 from /classes/Hook.php:509 from /tools/profiling/Hook.php:60 from /classes/controller/FrontController.php:448 from /classes/controller/FrontController.php:37 from /tools/profiling/Controller.php:227 from /classes/Dispatcher.php:373 from /index.php:28
<pre>SELECT SQL_NO_CACHE h.`name` as hook, m.`id_module`, h.`id_hook`, m.`name` as module, h.`live_edit` FROM `ps_module` m INNER JOIN `ps_module_shop` module_shop ON `module_shop`.`id_module` = m.`id_module` AND `module_shop`.`id_shop` = 1 AND `module_sh`</pre>	1.598	67	Yes	Yes	in C:\wamp\www\ps16en\classes/Hook.php:346 from /tools/profiling/Hook.php:50 from /classes/Dispatcher.php:413 from /index.php:28

Here is the result from the back office:

Customer Name	Products	Total	Date	Action
John DOE	2	\$61.37	05/05/2014	Q Details
John DOE	3	\$75.90	05/05/2014	Q Details
John DOE	3	\$76.01	05/05/2014	Q Details
John DOE	4	\$87.41	05/05/2014	Q Details
John DOE	3	\$74.37	05/05/2014	Q Details

DEMO MODE

YES NO

This mode generates fake data so you can try your Dashboard without real numbers.

[+ Add more graph and data](#)

Load time: 390 ms
Good boy! That's what I call a webservice!

Hook processing: 151 ms / 4.34 Mb
10 methods called in 7 modules

Memory peak usage: 17.9 Mb

Total cache size (in Cache class): 0.42 Mb

Execution	Load time (ms)
config	97
constructor	0
init	41
checkAccess	0
setMedia	16
postProcess	0
initHeader	8
initContent	210
initFooter	1
display	18

Hook	Processing
dashboardZoneTwo	0.62 Mb in 91 ms
dashboardZoneOne	0.98 Mb in 31 ms
displayBackOfficeHeader	1.99 Mb in 20 ms
actionAdminControllerSetMedia	0.80 Mb in 8 ms
displayForm	0.00 Mb in 0 ms
displayBackOfficeFooter	0.00 Mb in 0 ms
moduleRoutes	0.00 Mb in 0 ms
displayAdminForm	0.00 Mb in 0 ms
displayBackOfficeTop	0.00 Mb in 0 ms
actionDispatcher	0.00 Mb in 0 ms

Execution	Memory (Mb)	Total (Mb)
config	9.59	9.7
constructor	0.00	9.7
init	3.87	13.7
checkAccess	0.00	13.7
setMedia	0.81	14.5
postProcess	0.00	14.5
initHeader	0.56	15.1
initContent	1.98	17.1
initFooter	0.04	17.1
display	0.67	17.9

DB type: D6PDO

SQL Queries: 39 queries

Time spent querying: 24 ms

Included Files: 104

Size of included files: 2.00 Mb

Globals (> 1 Ko only): 1294 Ko
 _LANGADM ≈ 1120.4 Ko
 corect ≈ 84.4 Ko
 _MODULES ≈ 62.1 Ko
 _SERVER ≈ 10 Ko
 _MODULE ≈ 9.5 Ko
 _COOKIE ≈ 4.2 Ko

- Stopwatch
- Doubles
- Tables stress
- ObjectModel instances
- Files included

Stopwatch (with SQL_NO_CACHE) (total = 39)

Query	Time (ms)	Rows	Filesort	Group By	Location
SELECT SQL_NO_CACHE c.`name`, c.`id_lang`, IF(c.`id_lang` IS NULL, c.`value`, cl.`value`) AS `ps_configuration_kpi` FROM `ps_configuration` c LEFT JOIN `ps_configuration_kpi` cl ON (c.`id_configuration_kpi` = cl.`id_configuration_kpi`)	7.246	14			in C:\wamp\www\ps16en\classes\YConfiguration.php:118 from /classes/Configuration.php:154 from /classes/ConfigurationKPI.php:62 from /modules/dashgoals/dashgoals.php:131 from /modules/dashgoals/dashgoals.php:140 from /classes/Hook.php:507 from /tools/profiling/Hook.php:60 from /controllers/admin/AdminDashboardController.php:218 from /classes/controller/AdminController.php:1760 from /tools/profiling/Controller.php:227 from /classes/Dispatcher.php:373 from /admin-dev/index.php:54
SELECT SQL_NO_CACHE h.`name` AS hook, m.`id_module`, h.`id_hook`, m.`name` AS module FROM `ps_module` m INNER JOIN `ps_module_shop` module_shop ON (module_shop.`id_module` = m.`id_module` AND module_shop.`id_shop` = 1 AND module_shop.`id_shop` = 0) INNER JOIN `ps_module_shop` ms ON ms.`id_module` = m.`id_module`	1.509	67	Yes	Yes	in C:\wamp\www\ps16en\classes\Hook.php:346 from /tools/profiling/Hook.php:50 from /classes/Dispatcher.php:413 from /classes/Dispatcher.php:230