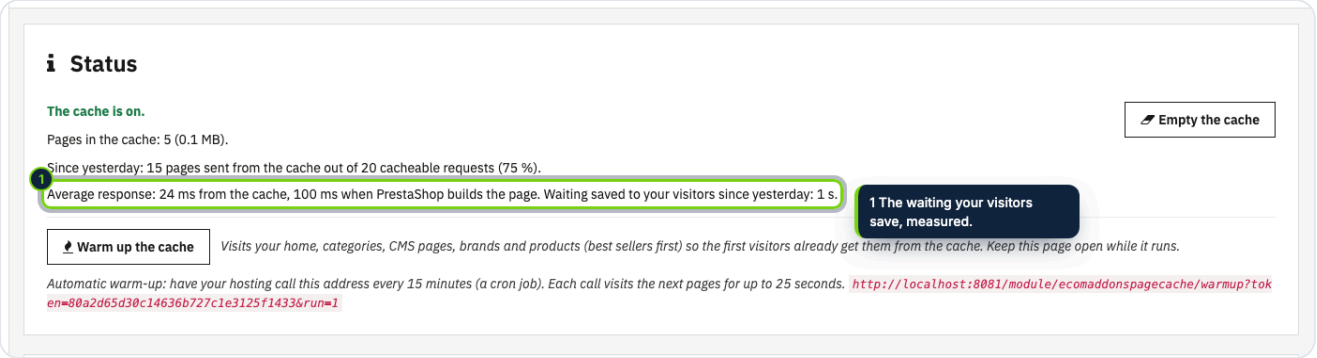
// prestashop · user guide

Page Cache for a Faster PrestaShop

Version 1.1.0
ecom-addons.com

Page Cache keeps the finished home, category, product and CMS pages of your PrestaShop shop and sends them again to the next visitor who is not signed in and has an empty cart, **before PrestaShop starts building the page**. Product, price and stock changes rebuild the pages that show them on their own, and nothing personal is ever shared.



The status of the cache: pages stored, their size and how many requests were answered from the cache.

Requirements

Component	Version
PrestaShop	8.2 · 9.0
PHP	8.1 or later
Theme	Any: the module stores the HTML your theme prints

Installation

1. Download the latest version from **My account › Downloads** on ecom-addons.com.
2. In the back office, go to **Modules › Module Manager** and click **Upload a module**.
3. Choose the zip, then click **Configure**. The cache is **off** after installing: run the check (below), then tick **Send cached pages to anonymous visitors** and save.

Who gets cached pages

- **Cached**: the home page, category, product, CMS, brand and supplier pages, the new products, best sellers and prices drop pages, and search results if you want (you choose which). The filters, sort and pagination of listings (the ajax answers of the faceted search) are cached too.
- **Sent from the cache** to visitors who are not signed in and have nothing in their cart, for a normal page view (GET), and, if you switch it on, to visitors with a cart or signed in (see below). Forms, ajax calls, currency and language switches and previews always get a fresh page.
- **Stored** only when the page was built for a visitor whose cookie holds nothing more than language, currency, country and visit counters. A page built for a visitor with a cart, an account or a history of viewed products is never stored, so nobody receives someone else's name, cart or recently viewed products.
- **Never cached**: cart, checkout, account, contact, module pages, error pages and redirects. Maintenance mode and geolocation (prices by IP address) switch the cache off.
- One page is kept per address, currency, country, device type (computer, tablet, phone; or one for all, see the settings) and shop: the currency and country PrestaShop will choose for the visitor, so a first visit and a returning visitor that get the same prices share the page. Tracking parameters such as `utm_source`, `gclid` or `fbclid` are ignored, so links from ads share the cached page.

When pages are rebuilt

Change	What happens
A product, combination, stock, image, specific price or review	The product page, its categories and the home, new, best-seller and price-drop pages are rebuilt on the next visit.
A specific price or catalogue price rule that starts or ends later	Cached pages expire at that moment.
A Promo Bar message, popup or button that starts or ends later (Promo Bar 1.1.0 or later)	Cached pages expire at that moment.
Any setting saved in the back office (PrestaShop's or a module's), a category, CMS page or carrier, a module installed, enabled, hooked or moved	Every page is rebuilt.
Clearing PrestaShop's cache (Advanced parameters › Performance)	The cache is emptied.
The lifetime is reached (60 minutes by default)	The page is rebuilt, as a safety net.

Settings that modules write by themselves while visitors browse (the time of a scheduled task, a refreshed token) do not empty the cache: the module notices they are written by the shop itself, not by you.

Settings

Open **Modules › Module Manager › Page Cache › Configure**.

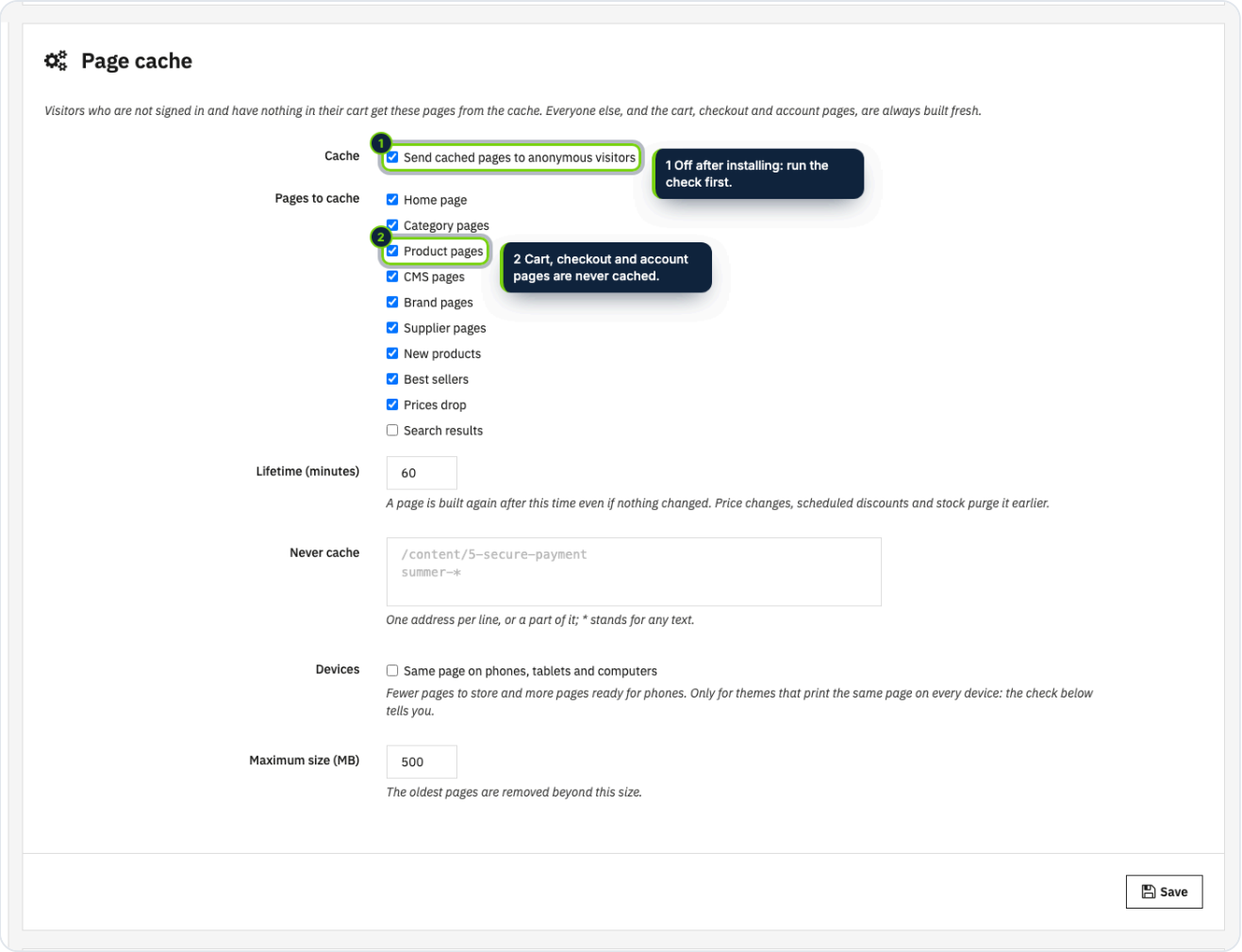
Setting	What it does
Cache	Sends cached pages to anonymous visitors. Off after installing.
Pages to cache	The page types that are cached.
Lifetime (minutes)	From 5 to 1440, 60 by default. A page is rebuilt after this time even if nothing changed.
Never cache	Addresses that are never cached, one per line: the whole address, a part of it, `*` for any text (`/content/5-*`).
Devices	One page for phones, tablets and computers. Only for themes that print the same page on every device: the check tells you.
Maximum size (MB)	From 50 to 10000, 500 by default. Beyond it, the oldest pages are removed.

The status at the top shows whether the cache is on, how many pages it holds, how many of the requests since yesterday were answered from it, and the average response time from the cache and when PrestaShop builds the page, with the waiting your visitors were spared. **Empty the cache** starts again from scratch.

Warm-up

Warm up the cache visits your home, categories, CMS pages, brands and products, best sellers first and in every language, as a new visitor, two at a time: the first visitors already get the pages from the cache. Keep the page open while it runs (up to 5,000 addresses per run).

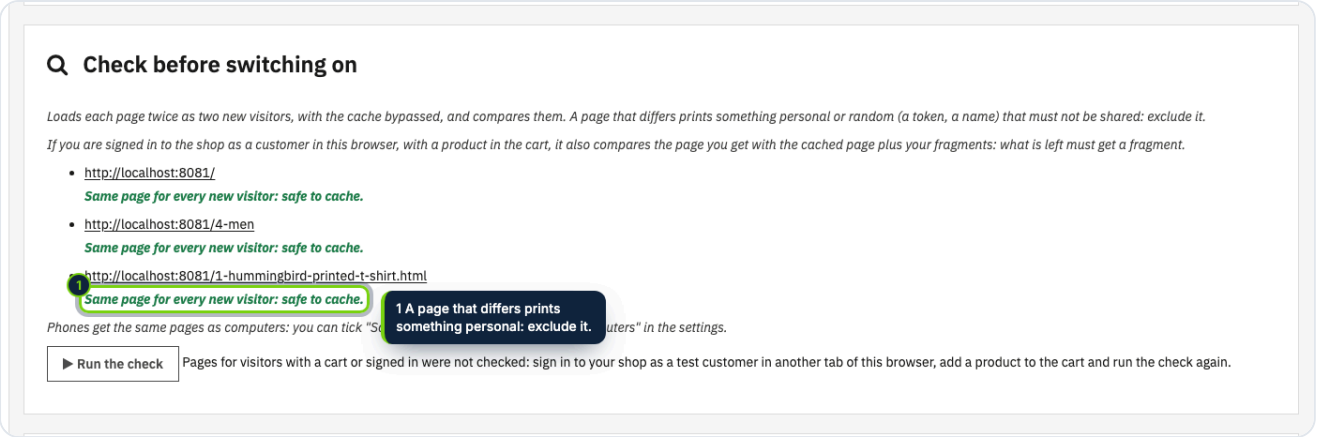
For an automatic warm-up, have your hosting call the address shown under the button every 15 minutes (a cron job, for example `curl -s "<address>"`). Each call visits the next pages for up to 25 seconds and the next call goes on from there; pages already in the cache answer in a few milliseconds, so once the cache is warm a round is quick.



Settings: which pages are cached, for how long at most, addresses never cached and the maximum size.

The check

Run the check loads your home page, a category and a product twice, as two new visitors without cookies and with the cache bypassed, and compares them. Numbers are ignored (themes print random ids in menus and filters). If a page differs, some module prints something personal or random on it: the differing lines are shown, so you can add the page to **Never cache** or ask that module's author. It also compares the category page a phone gets with the one a computer gets, and tells you whether **Same page on phones, tablets and computers** can be ticked. Run it again after installing new modules.

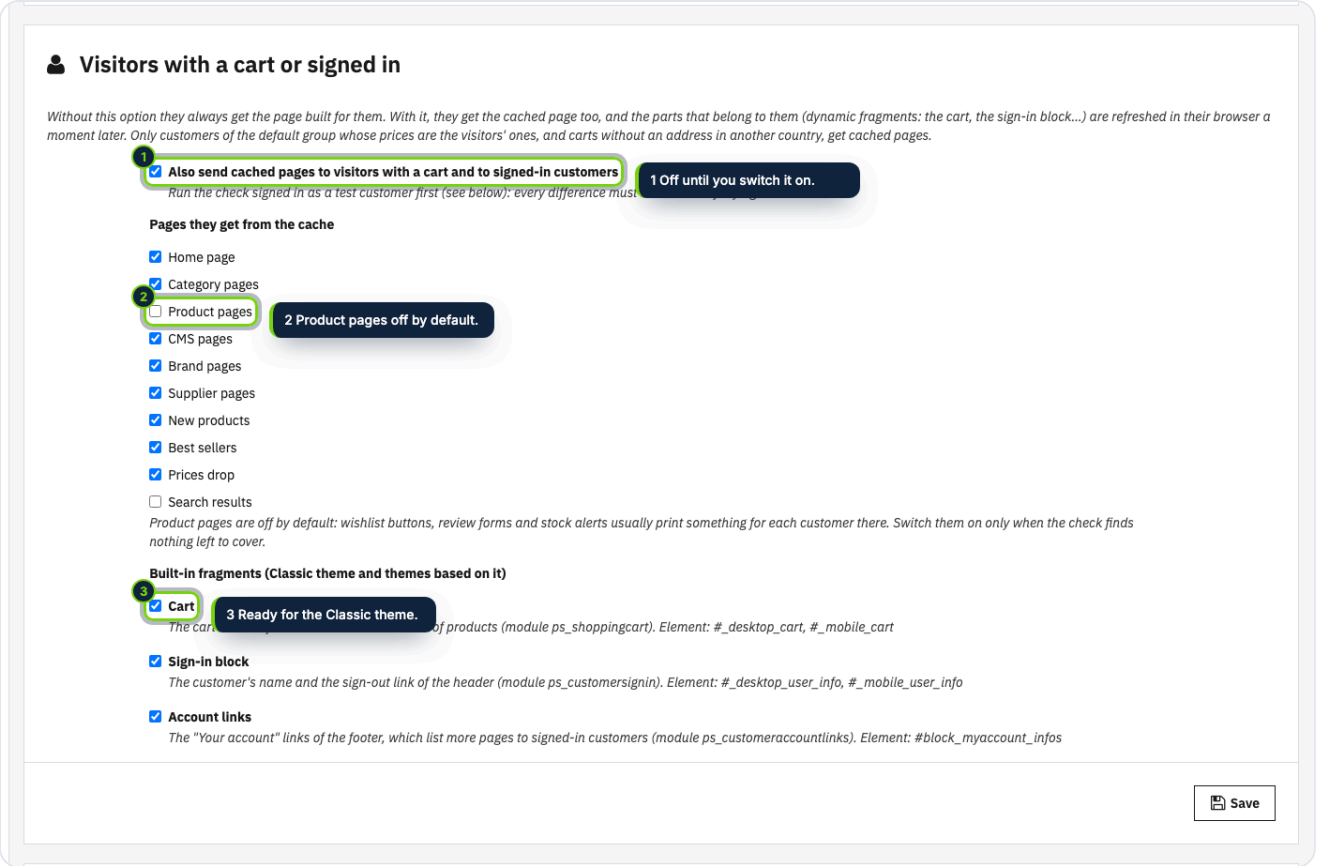


The check loads your home, a category and a product as two new visitors and compares them before you switch the cache on.

Visitors with a cart or signed in (dynamic fragments)

By default, visitors who are signed in or have something in their cart always get the page built for them. In the **Visitors with a cart or signed in** panel you can send them cached pages too: the page arrives as fast as for anyone else, and the parts that belong to them, the **dynamic fragments**, are refreshed in their browser a moment later.

- **Who**: customers of the default customer group, when that group sees the same prices, categories and modules as visitors and the customer has no prices of their own; and carts without an address, or with one in the shop's country where taxes do not depend on the state or the postcode. Everyone else keeps getting the page built for them.
- **Which pages**: you choose them. Product pages are off by default, because wishlist buttons, review forms and stock alerts usually print something for each customer there.
- **Built-in fragments** for the Classic theme and the themes based on it: the header cart, the sign-in block and the "Your account" links of the footer. The customer's form tokens are put in the page before it leaves, so "Add to cart" works from the first click.
- **Check first**: sign in to your shop as a test customer in another tab of the same browser, add a product to the cart and run the check. For each page it compares what you get with the cached page plus your fragments, and lists whatever is left: each of those lines needs a fragment.



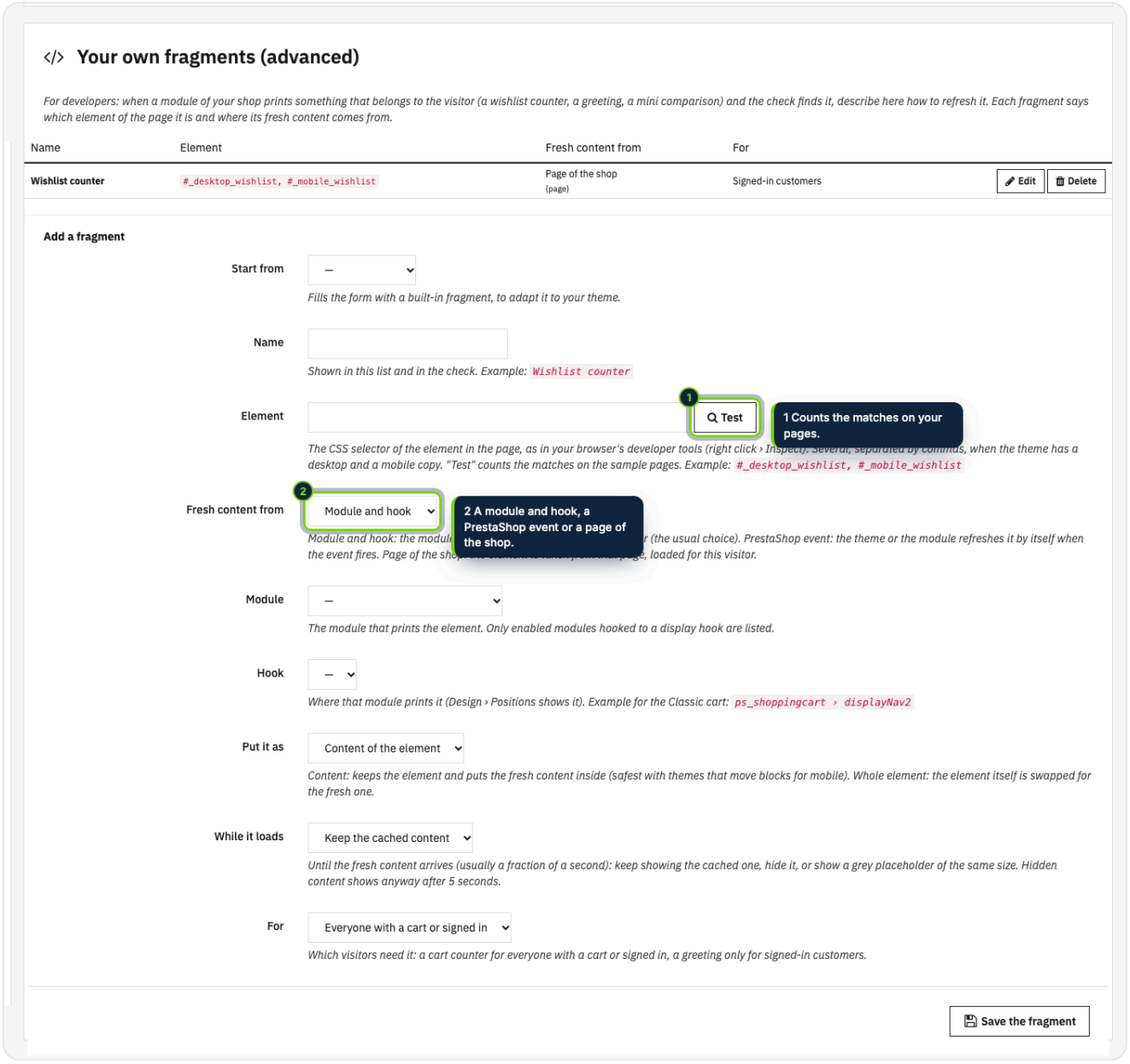
Visitors with a cart or signed in: cached pages for them too, with their cart and name refreshed by built-in fragments.

Your own fragments (advanced)

When another module prints something personal (a wishlist counter, a greeting), a developer describes in **Your own fragments** how to refresh it. Every field of the form is explained on the page:

Field	What it means
Name	Shown in the list and in the check.
Element	The CSS selector of the element in the page (`#_desktop_wishlist, #_mobile_wishlist`). **Test** counts the matches on the sample pages.
Fresh content from	**Module and hook**: the module prints the element again for the visitor (pick both from lists). **PrestaShop event**: a name passed to `prestashop.emit()` that the theme or the module listens to (`updateCart`). **Page of the shop**: a path of the shop, or `{page}` for the page itself, from which the element is taken.
Put it as	The content of the element (safest with themes that move blocks for mobile) or the whole element.
While it loads	Keep the cached content, hide it or show a grey placeholder; hidden content shows after 5 seconds anyway.
For	Everyone with a cart or signed in, visitors with a cart, or signed-in customers only.

Start from fills the form with a built-in fragment to adapt it to your theme.



Your own fragments, for developers: the element, where its fresh content comes from and how to put it, every field explained.

Checking it in your browser

Every cacheable page answers with an `X-Ecom-Cache` header (browser developer tools › Network): `HIT` (sent from the cache), `HIT dynamic` (from the cache, with fragments), `MISS` (built and, if allowed, stored) or `BYPASS` followed by the reason (`customer` , `cart` , `action` , `excluded` , `controller` , `maintenance` , `geolocation`). Open the shop in a private window to see it as a new visitor.

Good to know

- Keep PrestaShop's own performance settings (Smarty cache, CCC): they make each page faster to build, this module avoids building it again.
- PrestaShop's built-in statistics record a visit while they build the page, so visits answered from the cache are not counted there. Analytics tags in the page (Google Analytics, Matomo…) still count them.
- The "viewed products" block shows empty for visitors who get a cached page.
- If you use another full-page cache (a hosting cache, Varnish), empty it after changing products as well.

For developers

A module whose output changes at a time it knows (a promotion that starts) can make cached pages expire then. Register the hook `actionEcomPageCacheExpiry` and add Unix times to `$params['changes']`:

```
public function hookActionEcomPageCacheExpiry(array $params): void
{
    $params['changes'][] = $this→nextStartOrEnd(); // Unix time
}
```

Send the header `X-Ecom-Cache: bypass` to get a page built fresh and never stored.

A module can also declare its own dynamic fragments, with the same fields as the form (they show in the list as "Declared by a module" and cannot replace yours):

```
public function hookActionEcomPageCacheFragments(array $params): void
{
    $params['fragments'][] = [
        'id' ⇒ 'wishlist-counter', 'name' ⇒ 'Wishlist counter',
        'selector' ⇒ '#_desktop_wishlist', 'source' ⇒ 'hook',
        'module' ⇒ $this→name, 'hook' ⇒ 'displayNav2',
        'apply' ⇒ 'inner', 'loading' ⇒ 'keep', 'when' ⇒ 'customer',
    ];
}
```

Data and uninstallation

Pages are stored compressed in `var/cache/<environment>/ecompagecache/`, with a small index in the database (address, expiry, products shown), daily counters for 30 days and the names of settings written by the shop itself. No personal data is stored. Uninstalling removes the files, the tables and the settings.

Languages

The configuration page is translated into English, Spanish and French. To change a text, use **International › Translations › Module translations** (module "Page Cache").

Updates and support

Your licence includes every new version and 6 months of support. New versions are in **My account › Downloads**; if you need help, open a ticket in **My account › Support**.