addons presta

Auto Lazy Load

USER MANUAL · Version 1.0.0 · PrestaShop 1.7.8 – 9.x

1. What this module does

A page of a shop often holds dozens of images, and most of them are below the part of the page the visitor sees first. With lazy loading the browser downloads an image only when the visitor scrolls near it, so the page shows up faster and uses less data. Default themes do this for most of their own images, but third-party themes, module templates and everything typed into the editor — product descriptions, CMS pages — usually do not. Some themes also lazy-load the one image that must not wait: the large image at the top of the page.

The module adds lazy loading to the images of your shop without editing a single file of the theme or of a module. It also makes sure the first images of every page load at once, which is what Google measures as Largest Contentful Paint (LCP).

Works with **PrestaShop 1.7.8, 8.x and 9.x** (PHP 7.2.5 or newer), with Classic, Hummingbird and third-party themes and with multistore.

2. Installation

1. Download the module from **My account → Order history** at addonspresta.com.
2. In your Back Office, open **Modules → Module Manager** and click **Upload a module**.
3. Drop the `ap_autolazyload_v1.0.0.zip` file and wait for the confirmation.
4. Click **Configure** on the module card.

The module starts working right after installation with the native technique and needs no further setting. You do not have to clear any cache: saving the settings does it for you.

3. How it works

The module works in two places.

In each template. Every page is printed from templates: the page itself, the blocks of the modules, the product list loaded after filtering or sorting, the quick view, the cart window. When a template has printed its HTML, the module reads it, finds the `<img>` and `<iframe>` tags and adds lazy loading to them. Images you typed into product descriptions and CMS pages are part of that HTML, so they are covered too.

In the finished page. Just before a page is sent to the visitor, the module makes one more fast pass over it: the first images of the page are set to load at once, and the first large one gets a high priority.

There are two techniques to choose from:

- **Native** (default) — the module adds the attribute `loading="lazy"`. The browser decides when to load the image. No JavaScript is added to the shop, and search engines see ordinary images.
- **Script** — the module puts a small transparent placeholder with the proportions of the image into `src` and keeps the real address aside. A small script (about 4 KB, no library) gives the address back when the image comes near the screen. This technique lets you choose how early images start to load, fade them in and lazy-load inline CSS backgrounds.

What the module leaves alone:

- images another lazy-loading script of your theme already handles;
- images and templates you switch off (see *Excluded templates* and *CSS classes* below);
- tracking pixels and images embedded in the page as `data:`;
- in the script technique, SVG icons and images marked as structured data (`itemprop`), which must keep their address;
- e-mails, PDF documents such as invoices, and the Back Office;
- images added by scripts, backgrounds from stylesheets and videos.

The module edits no file of your theme or modules and writes nothing to products or pages. It changes only the HTML sent to the visitor; disabling or uninstalling it brings the original pages back at once. It does not resize, compress or convert images, stores only its settings (and, while you record, the list of templates) and does not connect to addonspresta.com or any other server.

4. Features and settings

Go to **Modules** → **Module Manager** → **Auto Lazy Load** → **Configure**. The Help box at the top links to this manual and to the module page at addonspresta.com, in the language of your Back Office. Three panels with settings follow, each with a **Save** button that saves all of them, and the table **Templates with images**.

Lazy loading

- **Lazy-load images** — switches the whole module on or off for the shop.
- **Technique** — *Native: loading="lazy", no script* or *Script: placeholder and loader*, as described above.
- **Iframes too** — videos and maps embedded with an `<iframe>` get `loading="lazy"`, in both techniques.
- **Images loaded at once** — the number of images at the top of every page that are never lazy, from 0 to 20 (default 4). The rule also removes the lazy loading a theme put on these images, for example on the main image of a product. Count what a visitor sees without scrolling on your home page and on a product page: the logo, the slide, the product image. 0 switches the rule off.

- **High priority for the first large image** — the first of those images that is at least 300 pixels wide gets `fetchpriority="high"`, which asks the browser to download it before other files. This is usually the slide on the home page and the main image on a product page.

Templates

- **Covered templates** — *All storefront templates except the excluded ones* (default) or *Only the templates listed below*.
- **Listed templates** — for the second choice: one template per line.
- **Excluded templates** — templates that are never lazy-loaded, whatever is chosen above. By default the two templates of the product gallery in Classic and Hummingbird, which the shop prints again when a visitor chooses another combination.
- **CSS classes that switch lazy loading off** — comma-separated, `no-lazy, skip-lazy` by default. An image, iframe or element with one of these classes, or with the attribute `data-no-lazy`, is left as it is. Handy for a single image in a description: add `class="no-lazy"` to it in the source view of the editor.

A template is what the shop prints in one go: a page (`index.tpl`, `catalog/product.tpl`, `cms/page.tpl`), the block of a module (`module:ps_imageslider/views/templates/hook/slider.tpl`), a product list loaded after filtering. Write it the way the table **Templates with images** shows it. `*` stands for any characters: `catalog/listing/*` covers every product list page, `module:ps_imageslider/*` every template of the slider. A small template that another one includes belongs to that other template; to keep a single image as it is, give it the class `no-lazy`.

Script technique

These settings apply only when the technique is *Script*.

- **Start loading before the image is visible** — the distance from the screen, in pixels, at which an image starts to load (0 to 2000, default 300). A larger value loads earlier, so a fast scroll shows fewer empty places.
- **Fade images in** — a loaded image appears with a short fade.
- **Copy of each image in `<noscript>`** — adds the original image for visitors and robots without JavaScript. It makes the pages a little larger; Google runs JavaScript and does not need it.
- **Inline CSS backgrounds too** — elements with a background image in their `style` attribute, a common way to build banners. Backgrounds from stylesheets are not covered.

Templates with images

The table shows which templates your shop printed and what the module did in each of them. Click **Record for 10 minutes**, open the pages you want to check in your shop — the home page, a category, a product, a CMS page — and reload the configuration page. The recording stops by itself; outside it the shop writes nothing down. Each new recording, and each save of the settings, starts a new list.

Column	Meaning
Template	the name to copy into the lists above
Images and iframes	how many the template contains
Lazy-loaded by the module	how many of them the module changes
Left as they are	how many it does not, and why: already lazy-loaded, switched off, cannot be read safely, or with nothing to load later
Status	Covered, Excluded by you, Not listed, or Nothing to do

"Cannot be read safely" means the tag is written in a way the module will not risk changing; the image then simply loads as before. "Switched off" covers images with `data-no-lazy`, a class from the list above, a high priority or `loading="eager"`.

5. Multistore

The settings are saved for the shop selected in the Back Office header; the table **Templates with images** is shown when one shop is selected, because it depends on the theme of that shop. Each shop can have its own technique and lists, also when several shops share one theme. In the All shops or group context only an employee who manages every shop of that context can save. One licence covers all shops of the installation.

6. Uninstall

Module Manager → Auto Lazy Load → Uninstall deletes the module's settings and the recorded list; the pages are again as your theme made them. To keep the settings, choose **Disable**.

7. FAQ and troubleshooting

How do I check that it works? Load a page of the shop, right-click an image far down the page and choose **Inspect**. With the native technique the tag has `loading="lazy"`; with the script technique it has `data-ap-lazy` until you scroll to it. In the **Network** tab of the browser tools, filtered by images, new files appear as you scroll.

PageSpeed says "Largest Contentful Paint image was lazily loaded". Raise **Images loaded at once** so that it covers the large image at the top of that page, and keep **High priority for the first large image** on.

A slider, a zoom or a gallery of my theme shows an empty or wrong image. Its script works with the addresses of the images itself. Record, find its template in the table **Templates with images** and add it to **Excluded templates**, or give the image the class `no-lazy`. The native technique rarely has this problem; consider it if your theme has many such scripts.

An image in a product description should not be lazy. Add `class="no-lazy"` to it in the source view of the editor.

I changed a setting and the page looks the same. Saving empties PrestaShop's template cache, but a full-page cache module, a CDN or the cache of your browser may still serve the old page. Refresh that cache, or clear the cache in **Advanced Parameters → Performance**.

Do search engines see my images? Yes. With the native technique the page contains ordinary images. With the script technique Google runs the loader; for robots that do not run JavaScript switch on **Copy of each image in** `<noscript>`.

My theme already lazy-loads images with its own script. The module leaves those images to the theme and covers the rest: module templates, editor content and iframes.

The table shows "cannot be read safely" for a template. Nothing is broken: the image in that template loads as before. If you want it lazy, add `loading="lazy"` to the tag in the template yourself.

Invoices and e-mails are never changed, whatever the settings.

8. Updates and support

Updates are free for life: download new versions from **My account → Order history** at addonspresta.com and upload them like a new module. Your settings are kept.

Support is included for 12 months from your order: write to **mail@addonspresta.com** or go through the contact form at addonspresta.com (Monday to Friday, 10:00–16:00 CET; we usually answer within 2 business days). Please include your PrestaShop version, the module version, the name of your theme and the address of a page that shows the problem.