addons presta

Price Drop Alert

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

1. What this module does

Let logged-in customers follow a product and get an e-mail when its price drops. A signed-in customer clicks **Notify me when the price drops** on the product page, may type a target price, and finds the alert in the list **My price alerts** of their account. Your shop then checks the price this customer would pay and sends one e-mail when it has dropped by the percentage you set, or to the customer's target price. The configuration page shows which products customers are watching and which e-mails went out.

Works with **PrestaShop 1.7.8, 8.x and 9.x** (PHP 7.2.5 or newer), with the Classic and Hummingbird 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_pricealert_v1.0.0.zip` file and wait for the confirmation.
4. Click **Configure** on the module card.

If you have shell access, you can also unpack the zip into `modules/` and run `php bin/console prestashop:module install ap_pricealert` from the shop folder.

The module is on right after the installation. Signed-in customers see the button on product pages at once. Nothing is sent until a customer has set an alert, and then only to that customer. Before customers start using it:

1. **Check the settings** in the **Settings** panel, above all **Price drop that sends an e-mail (%)** (5 % by default).
2. **Check that your shop sends e-mails** (**Advanced Parameters** → **E-mail**) and send yourself a test with **Send a test e-mail** in the **Sending** panel.
3. **Set up the cron job** for each shop with the address shown in the **Sending** panel, once a day or more often (every hour for quicker e-mails). Without a cron job, leave **Check prices during shop visits** on.
4. **Mention the price alerts in your privacy policy** (see "What the module stores" in the next section).

3. How it works

Who can follow a price. Only customers signed in to their account. Visitors who are not signed in — and guests of the checkout without an account — see the link **Sign in to get a price drop alert** instead. It opens the login page and, after signing in (or creating an account), brings the customer back to the same product and combination. There is no alert for guests and no alert by e-mail address alone.

On the product page. The module adds a small block under the product information, for the combination that is selected (it changes when the customer picks another size or colour):

- not signed in: the link **Sign in to get a price drop alert**;
- signed in, no alert yet: the button **Notify me when the price drops**. It is a link to the page **My price alerts**, where the customer confirms the alert;
- signed in, alert already set: "You follow the price of this product.", with "You will get an e-mail at … or lower." when the alert has a target price, and the link **My price alerts**.

The block is not shown when the shop is in catalogue mode, when the product cannot be ordered or does not show its price, when the customer's group does not see prices, and while **Price alerts** is off. It is never shown on product tiles (category pages, search, home page) or in the quick view — only on the product page itself. It uses no JavaScript and takes the look of your theme's buttons.

Setting an alert. The button opens the page **My price alerts** with the card **New price alert**: the picture, the name and the combination of the product, "Price now:" with the price this customer pays today, the field **Target price (optional)** and the button **Set the alert**.

- **Field left empty:** the customer gets an e-mail when the price drops by at least the percentage of your setting **Price drop that sends an e-mail (%)**. The hint under the field says so ("Leave it empty to get an e-mail when the price drops by at least 5 %.").
- **Target price typed:** the customer gets an e-mail when the price reaches that amount or less. The target must be lower than the current price ("The target price must be lower than the current price (…)."). It replaces your percentage: such an alert waits for the target, even when the price drops by more than your percentage before.

After **Set the alert** the customer returns to the product page with "Your price alert is set. We will e-mail you when the price drops by at least 5 %." or "Your price alert is set. We will e-mail you when the price drops to … or lower." Other answers of the form:

- "A price alert cannot be set for this product." — the product cannot be followed by this customer (see below), or its price is zero. The same message for every reason, and nothing about the product is shown.
- "You already follow the price of this product." — one alert per product and combination.
- "You have reached the limit of 20 price alerts. Remove one to add another." — the setting **Alerts per customer**.
- "Enter the target price as a number, for example 17.99." — a target that is not a number above zero, or has more decimals than the currency.

- "The form has expired. Reload the page and try again." — the customer signed out or changed their password while the page was open.

Which products can be followed. A product in the current shop that is active, visible somewhere in the shop, available for order and shows its price, while the shop is not in catalogue mode. A product with combinations is followed per combination. With customer groups on, the customer's group must see prices and may see a category of the product. The same rules apply again before every e-mail.

The list in the account. My price alerts is a link in the customer's account. The page lists the customer's active alerts in the current shop: picture, product (a link to the product page with the combination selected), combination, "Price when set:", "Price now:", "Target: ..." or "Any drop of at least 5 %", "Set on:" and "Ends on:", and a **Remove** button. Under the list: "You get one e-mail per alert. An alert ends after the e-mail or after 90 days." When a product can no longer be followed (disabled, hidden from the customer's group), its name and combination stay in the list so that the customer can remove the alert, without picture, link and current price.

When the e-mail is sent. The module compares the price the customer would pay now with the price at the moment the alert was set:

- **Alert without a target:** the e-mail goes when the price has dropped by at least **Price drop that sends an e-mail (%)**. The lowest price that counts is rounded **down** to the cents of the currency (to the smallest unit the currency shows): with 10 %, a price of 33.33 has to fall to 29.99 or less; with 5 %, a price of 100.00 to 95.00 or less. A smaller drop is not announced — a 9.99 % drop is not a 10 % drop.
- **Alert with a target:** the e-mail goes when the price is at the target or lower.
- **Never** when the price is not below the price at the time the alert was set, whatever the target or the percentage. A price that goes up and comes back to where it was is not a drop.
- The percentage is your setting **at the time of each check**, not the value when the alert was set. When you change it, it changes for the existing alerts too — also for customers who were told "by at least 5 %".
- With stock management on, a product (or combination) that is out of stock and cannot be ordered when out of stock is not announced; the alert waits and is checked again later.
- **One e-mail per alert.** After the e-mail the alert is closed and leaves the customer's list. The customer can set a new alert for the same product at once.
- **Alerts expire.** An alert that sent no e-mail is deleted after **Alerts end after (days)** (90 days by default), counted from the day it was set.

Which price. The module computes the price exactly as this customer would see it for the smallest quantity they can order, wherever the check is started from:

- the customer's default group: group reduction, prices with or without tax as that group shows them;
- the currency the customer used when setting the alert (all amounts of the alert stay in that currency; a lower amount after a change of the exchange rate counts as a drop);

- the tax of the country (and state, postcode) of the customer's first address, or of your shop's default country when the customer has no address;
- specific prices and catalogue price rules, also those for that customer or group; the selected combination; the minimal quantity of the product.

When the customer's conditions change. When the customer's default group, the tax display of that group or the country of the tax address changes, the price changes without any change of yours (for example a customer moved to a group that sees prices without tax). The module then does **not** compare: it takes the new price as the new starting price, without an e-mail, and goes on from there. Two consequences:

- a real drop that happens at that same moment is missed;
- a target price that is no longer below the new price is removed without notice; the alert goes on with your percentage.

When the prices are checked. The checks run in short batches, one shop at a time; two runs of one shop never work at the same time. Each alert is checked **at most once an hour**, however often the runs come.

- **Cron job** (recommended): your server calls the shop's address from the **Sending** panel, once a day or more often (every hour for quicker e-mails). One run checks at most 2000 alerts, works for about 20 seconds and sends at most **E-mails per run** e-mails (50 by default); the next run goes on with the alerts that were not checked in the last hour. Two calls less than 60 seconds apart do not start a second run.
- **Shop visits** (**Check prices during shop visits**, on by default): for shops without a cron job, a page request of a visitor who is neither signed in nor a guest of the checkout starts a small run — at most 30 alerts, 5 e-mails and about 5 seconds — when the last run of that shop is at least 15 minutes old. Requests of signed-in customers and checkout guests never start a run.

A short sale can be missed: a price that is lower only between two checks of an alert is never seen. The time limit of a run is checked between two alerts, so with a slow mail server one e-mail can make a run last longer; in that case use a cron job and switch the visit checks off, so that no visitor waits.

When nothing is sent.

- **Price alerts** is off in that shop, or PrestaShop's e-mails are switched off (**Advanced Parameters** → **E-mail**, never send e-mails): nothing is checked and nothing is sent.
- **The e-mail failed.** When the mail server refuses an e-mail, the alert is closed with the status **Not sent** and the e-mail is **never sent again** — a server that reports an error may still have delivered it, and the promise is one e-mail per alert. After **three failures in a row** the run stops and writes an entry in **Advanced Parameters** → **Logs** ("Price Drop Alert: the run stopped because 3 e-mails in a row could not be sent..."); the other alerts that are due stay open and are sent by a later run.
- The customer's account is disabled or deleted, the currency of the alert was disabled or deleted, the product cannot be followed any more or has no name in the language of the alert: the alert is skipped and checked again an hour later, until it expires.
- Another module held the e-mail back with the module's hook (see `README.md`).

Cleaning up only happens during runs. Expired alerts, closed alerts older than 365 days and alerts of deleted customers, products or shops are deleted by the runs. While **Price alerts** or PrestaShop's e-mails are off, or while neither a cron job nor a visit starts a run, nothing is cleaned up either: the alerts stay in the database until runs happen again, or until you uninstall the module.

The e-mail. The subject is "Price drop: " followed by the product name. The e-mail greets the customer by first name and shows the product with its picture, name and combination, the price when the alert was set and the price now (in the currency of the alert), the target price when there is one, a button to the product page with the combination selected, a link to **My price alerts**, and why the customer gets it: they set this alert, this is its only e-mail, and prices can change at any time. It goes out through your shop's own e-mail settings, from the shop where the alert was set, **in the language the customer used when setting the alert**. When neither the module nor your theme has the e-mail template in that language, the e-mail goes in English; when that language has been deleted from the shop, in the shop's default language. There is no tracking pixel, and nothing is loaded from outside your shop.

Addresses that nobody confirmed. PrestaShop does not confirm the e-mail address of a new account. Someone could therefore register with another person's address, set price alerts and make your shop send that person e-mails about real price drops of your products. The module limits what this can cause: an account can have at most **Alerts per customer** active alerts, each alert sends one e-mail at most and only after a real drop of your price, and the names in the e-mail are printed as plain text, never as a link. The module has no double opt-in.

What the module stores. One database table, `<prefix>ap_pricealert_alert`, with one row per alert: the shop, the customer number, the product and the combination, the currency and the language, the price when the alert was set, the target price, the price in the e-mail, a fingerprint of the customer's price conditions (group, tax display, tax address; it cannot be read back), the status and the dates (set, last checked, closed). It records **who follows which product**, so it is personal data. It stores no e-mail address and no name: they are read from the customer account when an e-mail is sent or the list **Notifications sent** is shown.

- **Active alerts** are deleted when they expire, when the customer uses **Remove**, and when the customer is deleted.
- **Closed alerts** (sent, not sent, interrupted) are kept for **365 days** for the list **Notifications sent**, then deleted.
- Deleting a customer from the database deletes all their alerts (see "A customer asks to be erased" in the FAQ for the option to choose). Deleting a product or a combination deletes its alerts **in the shops where it no longer exists**; the other shops of a multistore keep theirs. Alerts whose customer, product, combination or shop disappeared in another way (for example while the module was disabled) are deleted by the next runs, a few hundred alerts per run.
- **Uninstalling the module, and Reset, delete every alert** — active and closed — together with the settings.

The module does not connect to addonspresta.com or any other server, uses no external mailing service, sets no cookies of its own, writes no files and does not track whether e-mails are opened or links clicked. It sends no SMS and no push messages, and it does not tell customers that a product is

back in stock. The product picture in the e-mail is loaded from your shop by the recipient's mail program.

4. Features and settings

Open **Modules → Module Manager → Price Drop Alert → Configure**. The **Help** box at the top links this manual and the module page at addonspresta.com in the language of your Back Office, gives our support address and shows the installed version. Below it: the **Settings** panel with the **Save** button, then the panels **Sending**, **Watched products** and **Notifications sent**.

Settings

Price alerts — the main switch, on after the installation. Off: nothing of the module is shown in the shop (button, account link, the page **My price alerts**, which then leads to the customer account), no price is checked and no e-mail is sent. The alerts stay in the database and work again when you switch it back on; their expiry time keeps running.

Price drop that sends an e-mail (%) — for alerts without a target price, the drop that sends the e-mail: a whole number from 1 to 90 (default 5). The price must be lower by at least this percentage than when the alert was set (see "When the e-mail is sent" in the previous section). Alerts with a target price wait for that price instead.

Alerts end after (days) — an alert that sent no e-mail is deleted after this many days, from 1 to 365 (default 90). The customer sees the end date in the list.

Alerts per customer — the most active alerts one customer can have in a shop, from 1 to 50 (default 20). Lowering it later does not remove existing alerts: the customer sees and can remove all of them, but cannot add a new one while over the limit.

E-mails per run — the most e-mails one run of the cron job sends, from 1 to 500 (default 50). A run started by a visit sends at most 5 (or fewer when this setting is lower). Failed e-mails count too.

Check prices during shop visits — on by default. For shops without a cron job: a visit starts a small check (up to 30 alerts and 5 e-mails) at most every 15 minutes. With a cron job you can switch it off.

A value outside its range is refused with a message, for example "The price drop must be a whole number from 1 to 90."; the other values are kept in the form and nothing is saved until all of them are valid.

Sending

The table shows, for every active shop of the current context that you manage:

- **Shop**;
- **Price alerts** — **On**, **Off**, or **PrestaShop e-mails are off** when the shop is set to never send e-mails;
- **Address for the cron job** — the address of that shop;
- **Last run** — when the last run of that shop started or ended, or **not yet**;

- **Active alerts** — the number of alerts waiting for a drop.

The cron job. Give the address of each shop to a cron job and call it once a day or more often (every hour for quicker e-mails) — in your hosting panel, or on a server with crontab, for example every hour:

```
0 * * * * curl -fsS "https://your-shop.com/module/ap_pricealert/cron?token=…"
>/dev/null
```

Copy the address from the panel: each shop has its own, on its own domain, and its token is valid only for that shop. Keep the addresses secret. Anyone who knows one can start the run of that shop — nothing else — and a run sends only the e-mails that are due anyway. With many alerts, call it more often (for example every 15 minutes): one run checks up to 2000 alerts, and each alert is checked at most once an hour.

Replace the cron addresses — creates new addresses for every shop at once; the old ones stop working ("The cron addresses have been replaced. Update your cron jobs: the old addresses no longer work."). Only an employee who manages every shop of the installation sees and can use this button.

Send a test e-mail — type an **E-mail address** and click **Send the test e-mail**. The test goes out for the first shop of the table, in your Back Office language, with sample data only: your first name, the first active product of that shop's catalogue with its price in the shop's default currency as the price now, and a made-up earlier price (the price before a 10 % drop). No alert and no customer is used, and the links lead to the home page. The test also works while **Price alerts** is off, but not while the shop's e-mails are switched off. At most 5 test e-mails can be sent in 10 minutes for the whole installation; after that: "Too many test e-mails in a short time. Wait a few minutes and try again."

Watched products

The 50 products and combinations with the most active alerts in the shops of the current context that you manage: **Shop**, **Product**, **Combination**, **Active alerts** and **Lowest target price**. Target prices are amounts in the currency of their alert, so the lowest target is given per currency (for example `79,00 zł · 18,00 €`), only for the alerts that have a target. The panel shows no customer data: use it to see which prices hold your customers back.

Notifications sent

The last 50 closed alerts of the shops of the current context that you manage: **Date**, **Shop**, **Product**, **Combination**, **Price when set → price in the e-mail** (in the currency of the alert), **Status**, **Customer** and **E-mail address**. Status:

- **Sent** — the e-mail was handed over to the mail server;
- **Not sent** — the mail server refused it; it is not sent again;
- **Interrupted** — the run stopped before the result was known (for example a PHP timeout); it is not sent again.

The **Customer** and **E-mail address** columns are shown only to employees whose profile may view **Customers**. Other employees see the remaining columns and the note "Customer names and e-mail addresses are not shown: your profile has no access to Customers."

Product and combination names in both lists are in the language of your Back Office.

5. Multistore

Settings are saved for the shop or shop group selected in the Back Office header. Select **All shops** to use the same settings everywhere, or one shop to give it its own percentage, expiry and limits. In the All shops or group context only an employee who manages every shop of that context can save; others are asked to select one of their shops.

Each shop works on its own:

- **Alerts:** an alert belongs to the shop where it was set and uses that shop's prices, settings and e-mail addresses. **My price alerts** lists the alerts of the current shop only.
- **Cron:** every shop has its own cron address on its own domain; set up one cron job per shop. The visit checks work per shop as well.
- **Deleted products:** deleting a product in one shop deletes its alerts in that shop only.

The **Sending**, **Watched products** and **Notifications sent** panels show the active shops of the current context that you manage, each row with its shop name. **Replace the cron addresses** is only for an employee who manages every shop. One licence covers every shop of the installation.

6. Uninstall

Module Manager → Price Drop Alert → Uninstall deletes **every price alert** — active and closed, in every shop — together with the settings and the secret key of the cron addresses. **Reset** in the Module Manager does the same: it uninstalls and installs again, so all alerts are gone and the customers' lists are empty. There is no way to keep the alerts through an uninstall or a reset.

After a reinstallation or a reset the cron addresses are new: the old ones answer HTTP 403. Copy the new addresses from the **Sending** panel into your cron jobs, or remove the cron jobs from your server if you do not reinstall.

What is kept: the declaration of the module's hook for developers, so that other modules attached to it keep their place.

Disable keeps everything, but while the module is disabled: nothing is shown in the shop, no price is checked, no e-mail is sent, nothing is cleaned up, deleted customers and products do not delete their alerts (the next runs after you enable it again do), and the cron address and the link to My price alerts in e-mails already sent show the shop's error page 404 (the link to the product still works).

7. FAQ and troubleshooting

No e-mail goes out. Check in this order:

1. The **Sending** panel says **On** for the shop. **Off** means **Price alerts** is off in that shop's context; **PrestaShop e-mails are off** means **Advanced Parameters → E-mail** is set to never send e-mails.
2. **Last run** changes: your cron job calls the shop's address, or **Check prices during shop visits** is on and the shop has visitors who are neither signed in nor checkout guests.
3. **Active alerts** is above zero, and **Watched products** lists the product.
4. The price has really dropped for that customer by at least **Price drop that sends an e-mail (%)**, or to the target (next question).
5. Send yourself a test e-mail from the **Sending** panel.

Notifications sent shows what was sent; **Advanced Parameters → Logs** shows errors of runs, with entries starting with "Price Drop Alert".

A customer says the price dropped, but no e-mail came.

- **Below your percentage.** A drop of 4 % does not send anything with 5 %, and the limit is rounded down to the cents: from 33.33 with 10 %, 30.00 is not enough, 29.99 is.
- **The alert has a target price.** It waits for the target only: a drop of 20 % that stays above the target sends nothing.
- **Another price.** The module compares the customer's own price (their group, the currency of the alert, the tax of their first address, the smallest quantity). A drop for another group, for a larger quantity or in another currency is not a drop for them.
- **Their conditions changed.** After a change of group, tax display or tax country the alert starts again from the new price, without an e-mail (see "When the customer's conditions change").
- **Already notified.** One e-mail per alert: the alert was closed by an earlier e-mail (see **Notifications sent**), possibly with the status **Not sent**. The customer can set a new alert.
- **Too short.** The sale ended before the next check of that alert (at most once an hour, and only when runs happen).
- **Out of stock**, or the alert expired, or it was removed.

What does the cron address answer? A short JSON text, for example `{"status":"done","checked":120,"sent":3,"failed":0}`, where `checked` counts the alerts looked at, and `sent` and `failed` the e-mails of that run. The status:

- `done` — the run finished;
- `off` — **Price alerts** is off in that shop;
- `mail_off` — the shop's e-mails are switched off in **Advanced Parameters → E-mail**;
- `busy` — another run of that shop is working at this moment;
- `wait` — the last run of the shop (by the cron job or by a visit) was less than 60 seconds ago;

- `cart` — the call carried the cookie of a shop visitor whose cart has an address. Call the address without cookies, as `curl` or `wget` do by default;
- HTTP 403 with `{"error":"invalid token"}` — the address is wrong, was replaced, or the module was reinstalled or reset; copy the current one from the **Sending** panel. Nothing was run;
- HTTP 500 with `{"error":"run failed"}` — the run stopped on an error; the beginning of the error message is in **Advanced Parameters → Logs** ("Price Drop Alert: the cron run failed: ...");
- the shop's error page 404 — the module is disabled or not installed in that shop.

The test e-mail does not arrive. Check the mail settings in **Advanced Parameters → E-mail** with PrestaShop's own test there, and your spam folder. "The test e-mail could not be sent. Check the e-mail settings of the shop." means that the mail server refused it. After 5 tests in 10 minutes, wait a few minutes.

The status of a notification is "Not sent" or "Interrupted". The mail server refused the e-mail, or the run stopped before the result was known. That alert is closed and its e-mail is not sent again — deliberately, to never send it twice. Fix the mail settings; the customer can set a new alert.

A customer cannot find the button. They must be signed in; the product must be available for order and show its price; the shop must not be in catalogue mode; their group must see prices. The button is only on the product page, never on product lists or in the quick view.

A customer got the e-mail in English. The e-mail template does not exist in the language in which they set the alert. Copy the English template into your theme for that language (next question).

Can I change the look of the e-mail? Copy `mails/<language>/ap_pricealert.html` and `.txt` from the module to `themes/<your-theme>/modules/ap_pricealert/mails/<language>/` and edit the copies; they survive updates of the module. Keep the variables in braces, such as `{product_url}` and `{alerts_url}`.

Can someone receive e-mails they never asked for? PrestaShop does not confirm the address of a new account, so someone could register with another person's address and set alerts. That address then gets at most one e-mail per alert, only after real drops of your prices, and no more alerts than **Alerts per customer** at a time (see "Addresses that nobody confirmed").

A customer asks which data you hold about their price alerts. Their active alerts are in **My price alerts** in their account. The module adds nothing to the export file of PrestaShop's GDPR module. For a complete list, including closed alerts (kept up to 365 days), run this SQL query on the shop's database (with the prefix of your tables, usually `ps_`, and the customer's ID from **Customers → Customers**):

```
SELECT * FROM <prefix>ap_pricealert_alert WHERE id_customer = <customer ID>;
```

A customer asks to be erased. Delete the customer in **Customers → Customers** with the option that lets them register again with the same e-mail address — PrestaShop then deletes the account from the database, and the module deletes all their alerts at once. The other option keeps the account in the database, marked as deleted: no e-mail is sent to it any more, but its alerts stay until they expire, its closed alerts for up to 365 days, and **Notifications sent** can still show its name and

address. When only the price alerts must go, the customer can remove each of them with **Remove** in **My price alerts**.

Does the module tell customers that a product is back in stock? No. It only announces price drops. PrestaShop's own e-mail alerts module (ps_emailalerts) handles back-in-stock notices.

Does the module send a copy to me, SMS or push messages, or track opens? No. It sends only the e-mail to the customer who set the alert, through your shop's mail 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 and alerts are kept.

Support is included for 12 months from your order: write to **mail@addonspresta.com** or use 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 and the steps that lead to the problem.