fabrick

Fabrick Checkout

User manual

Version 1.0.1

PrestaShop 8 / 9 module

Audience: merchants / e-commerce administrators

Document date: August 2026

© Fabrick · PrestaShop module Fabrick Checkout 1.0.1

PrestaShop 8 / 9 module | Module version **1.0.1**
Audience: merchants / e-commerce administrators
Document date: September 2026

---

1. Introduction

Fabrick Checkout connects **Fabrick Payment Orchestra** payments to PrestaShop using the **REST API** and the **<fbk-checkout> Web Component**.

Main features:

- Standard payment (cards and APMs enabled on the shopLogin)
- Checkout on a **dedicated page (Redirect)**
- Optional **Pay by Link**
- Optional **card save** with buyer opt-in and My Account management
- Buyer callback + server-to-server notifications
- Reconciliation cron (e.g. Pay by Bank)
- Back-office refund tool
- **Multistore:** configuration and credentials **per shop**

---

2. Requirements

- PrestaShop **8.0+** or **9.x**
- Fabrick account with **Payment Orchestra**, Web Components and REST API enabled
- Credentials: **Shop Login** and **API Key** (sandbox or production)
- HTTPS on the shop domain (recommended / required in production)
- For Apple Pay / Google Pay: domain prerequisites and Fabrick-side activation (see official docs)

---

3. Installation

1. Copy the fabrickcheckout folder into the shop’s modules/ directory.
2. Back Office -> **Modules** -> find **Fabrick Checkout** -> **Install**
3. Open **Configure** and set the options (chapter 4).
4. Ensure the module is **enabled** and available for the cart currencies.

After an upgrade without reinstall, open the module configuration page at least once (self-heal for hooks/config if needed).

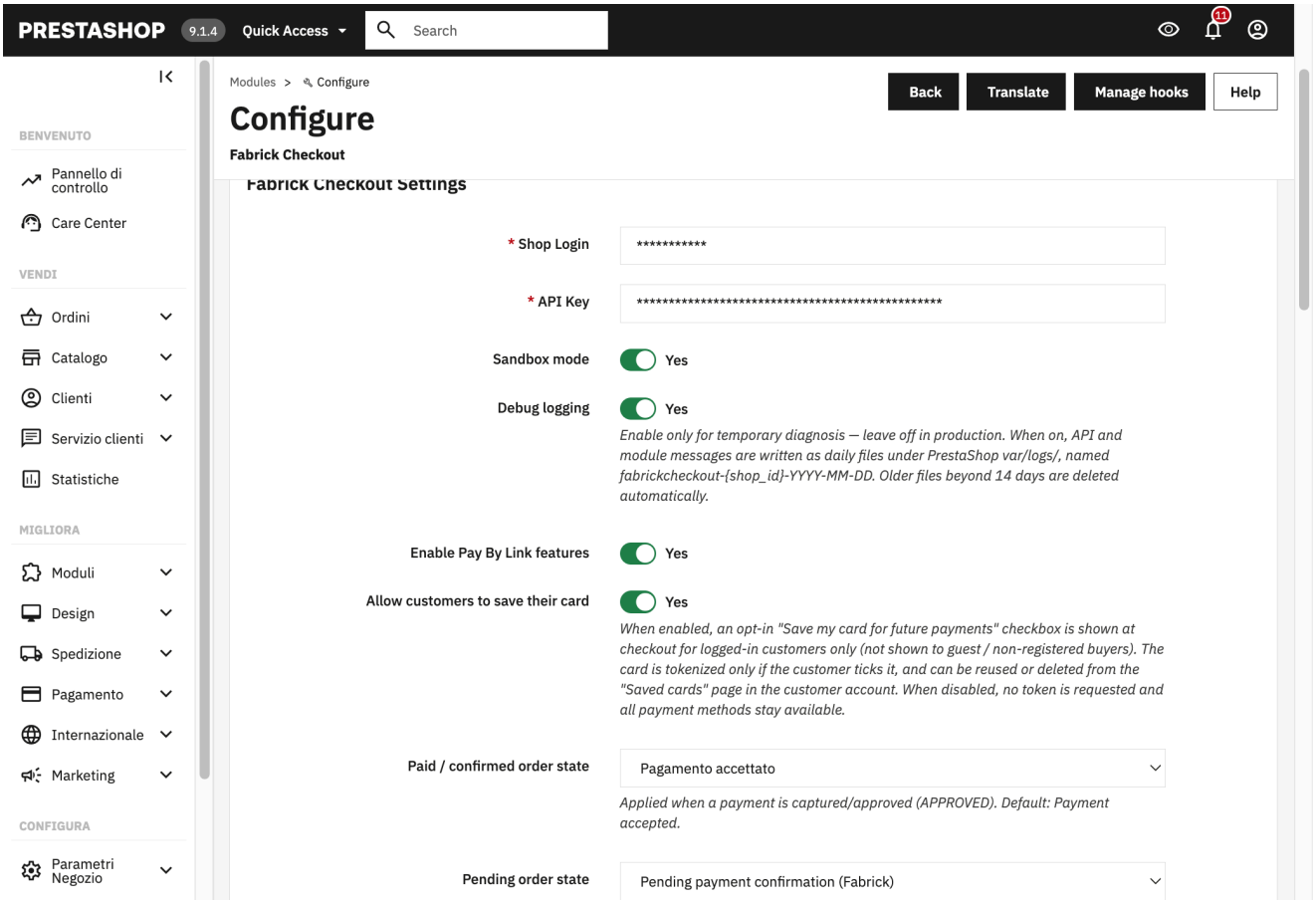
---

4. Back-office configuration

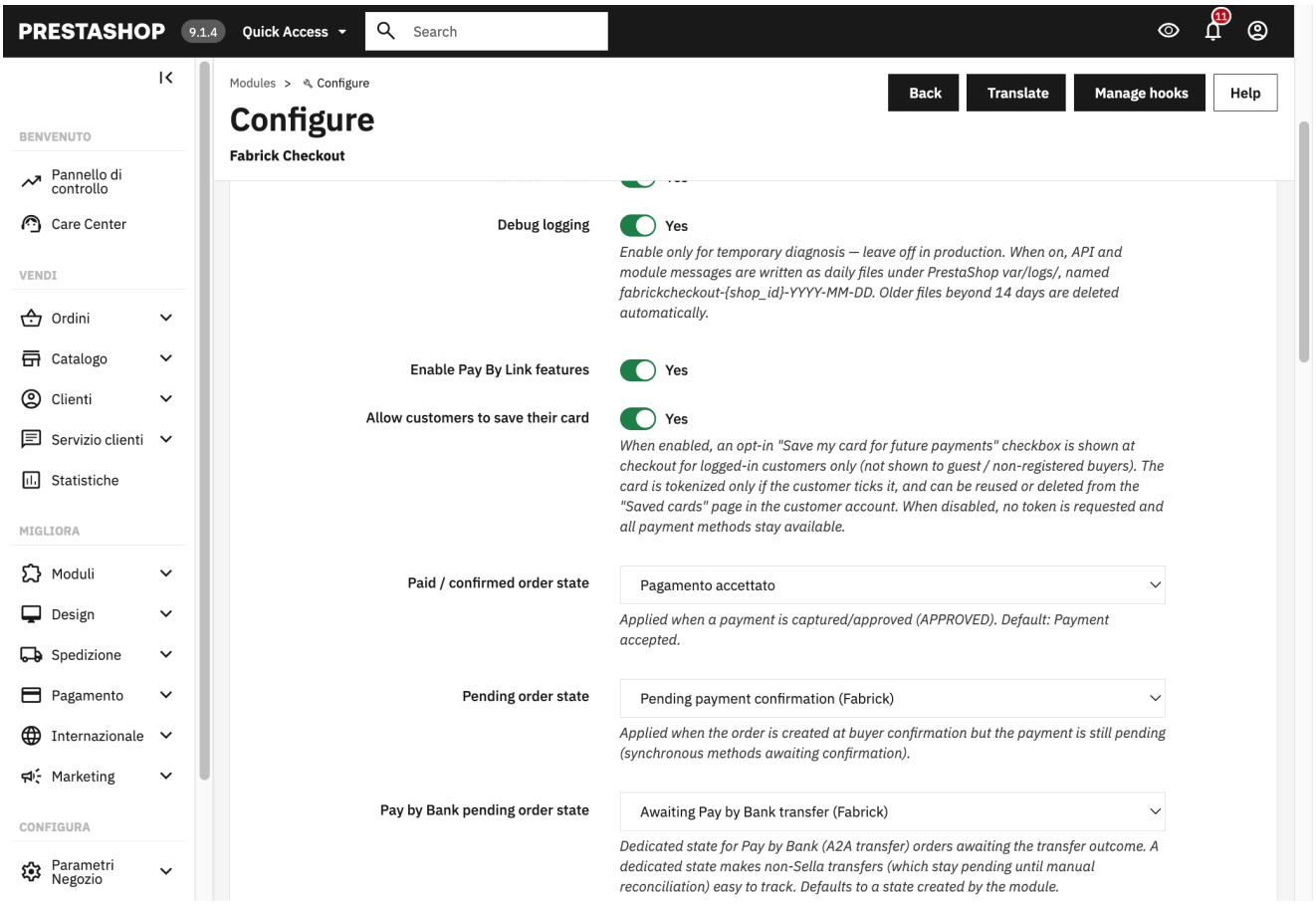
Path: **Modules -> Fabrick Checkout -> Configure**

Multistore: if PrestaShop has multiple shops, select the desired shop in the Back Office header **before** opening or saving configuration. Settings are not applied in bulk to “All shops”.

Example configuration page (Shop Login / API Key values redacted):



Fabrick Checkout settings - Shop Login and API Key (back office)



Fabrick Checkout settings - feature switches (back office)

4.1 Credentials and environment

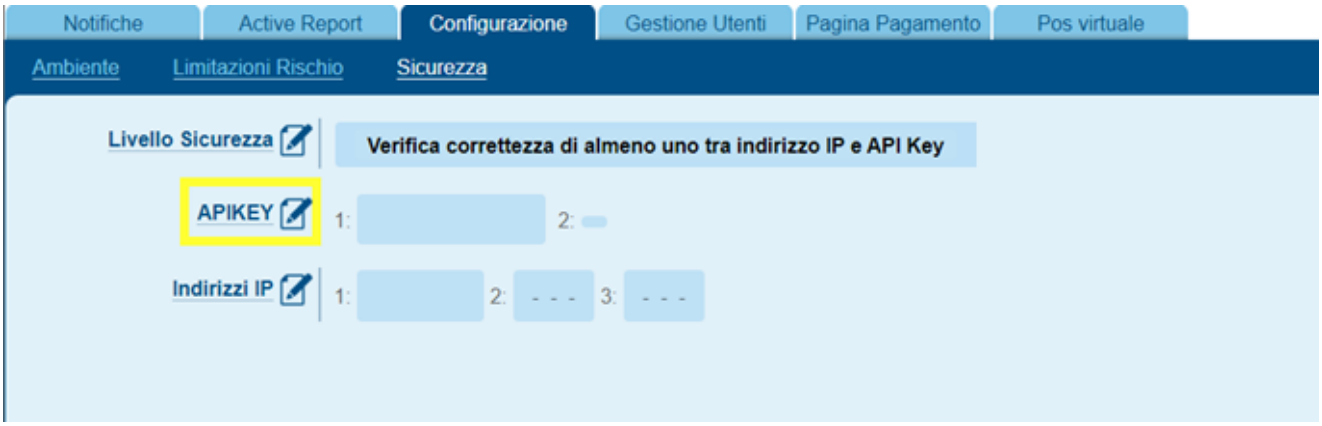
Setting	Description
Shop Login	Fabrick merchant code
API Key	API key (keep secret)
Sandbox mode	Yes = test environment; No = production
Debug logging	Writes daily files under var/logs/fabrickcheckout-{shop_id}-YYYY-MM-DD (diagnostics only; disable in production if unused)

4.1.1 Fabrick merchant back office

In the Fabrick **merchant back office** you can retrieve the credentials (**Shop Login** and **API Key**) to use later in the PrestaShop module configuration.

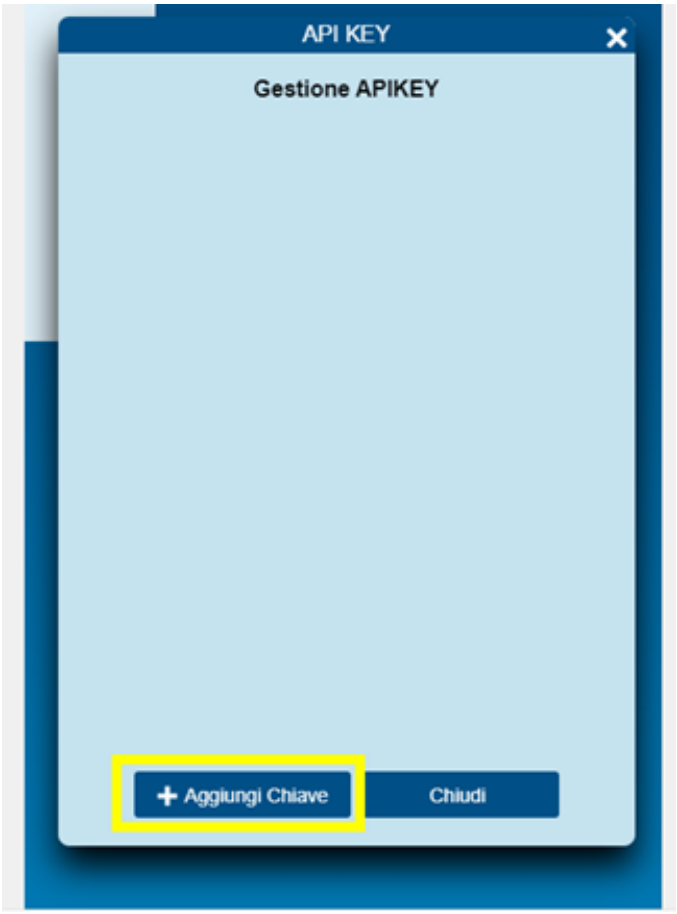
Create and copy the API Key

1. Go to **Configuration -> Security -> APIKEY** (Configurazione -> Sicurezza -> APIKEY) and create the required key.



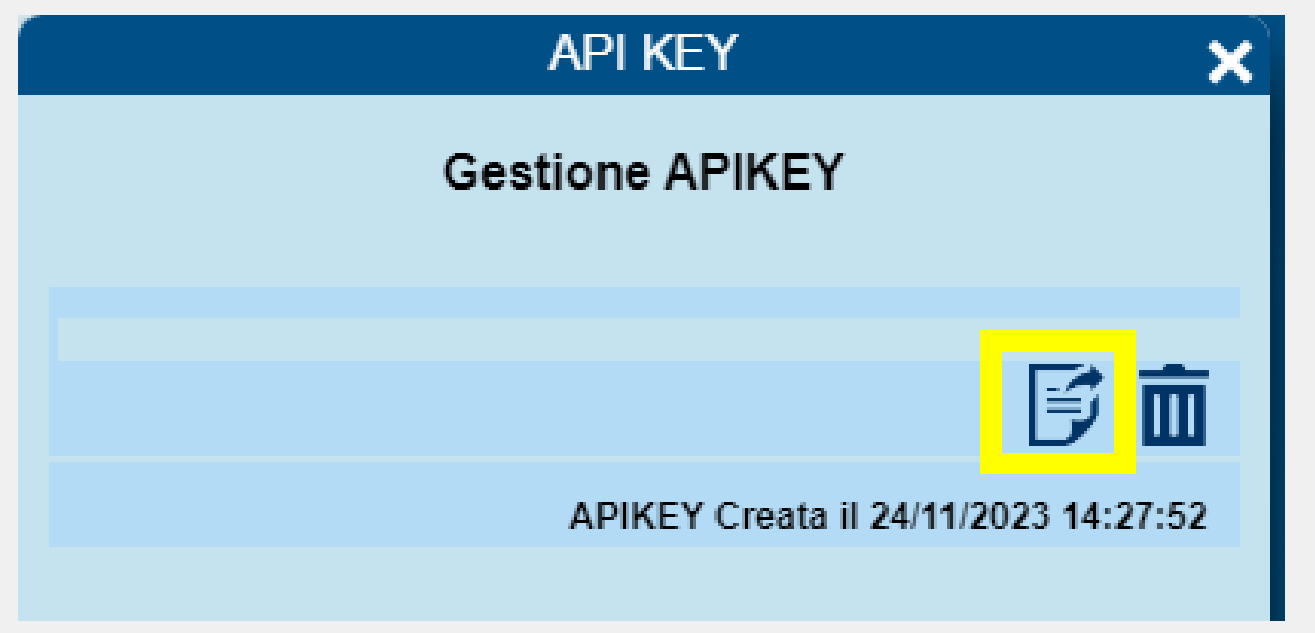
Fabrick merchant back office - Configuration > Security, APIKEY section

2. Click **Add Key** (Aggiungi Chiave).



Fabrick merchant back office - APIKEY management, Add Key

3. Use the **copy** icon (highlighted in the screenshot) to put the key on the clipboard, then paste it into the PrestaShop module **API Key** field.



Fabrick merchant back office - copy API Key

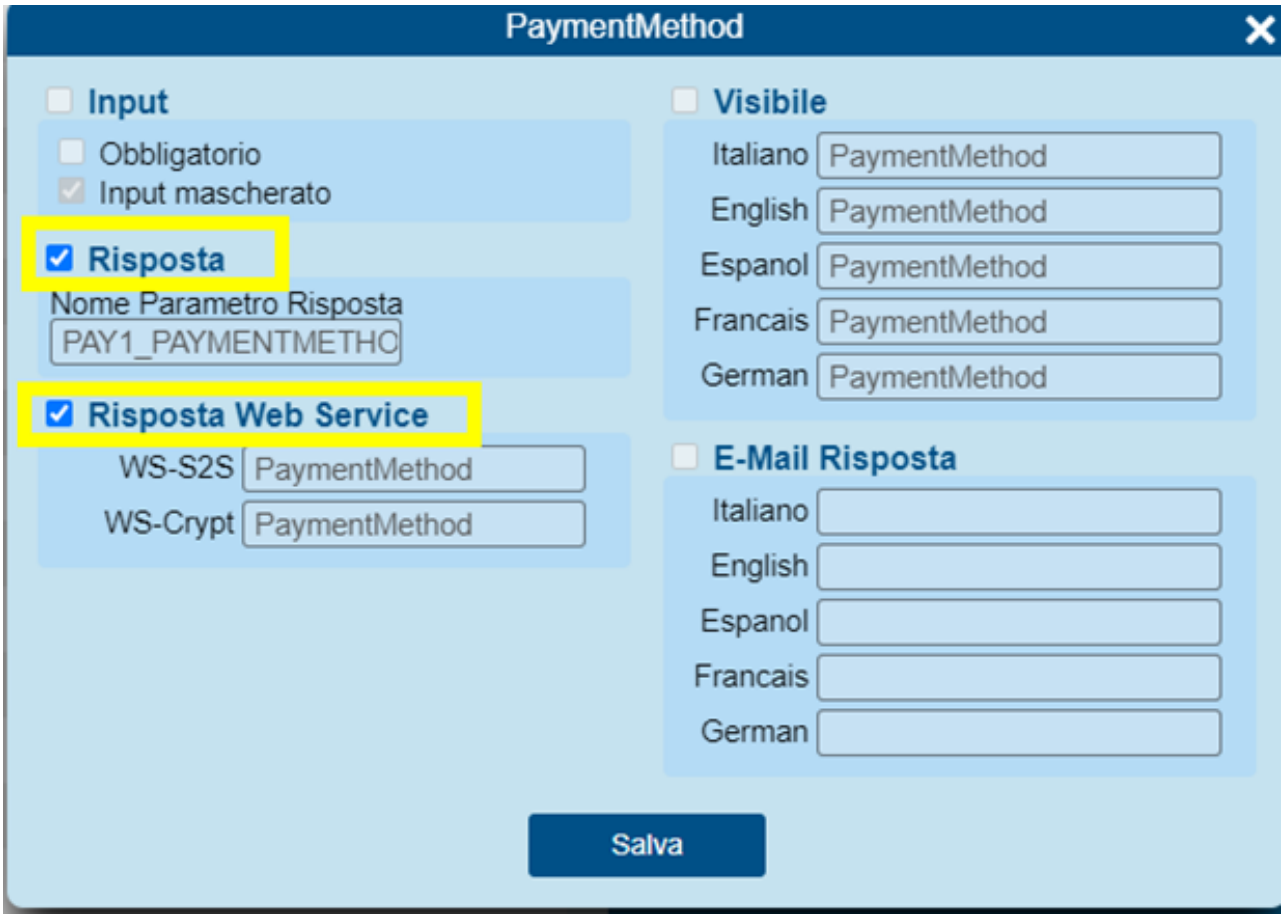
Response parameters (Payment page)

Under **Payment page -> Fields & Parameters** (Pagina di Pagamento -> Campi & Parametri), enable in the response the parameters **PaymentMethod**, **Token**, **Token ExpiryMonth** and **Token ExpiryYear** on both pages. The screenshots below show **PaymentMethod**; the other parameters follow the same logic.



Fabrick merchant back office - parameter list, PaymentMethod

In the parameter dialog, tick **Response** (Risposta) and **Web Service response** (Risposta Web Service), then save.



Fabrick merchant back office - PaymentMethod: Response and Web Service response

4.2 Feature switches

Setting	Description
Enable Pay By Link features	Shows Pay by Link at checkout
Allow customers to save their card	Shows the opt-in “Save my card” checkbox and the Saved cards account page

4.3 Order statuses

Map PrestaShop statuses for:

- Paid / confirmed
- Generic pending payment
- Pay by Bank pending
- Pay by Link pending / paid
- Failed payment

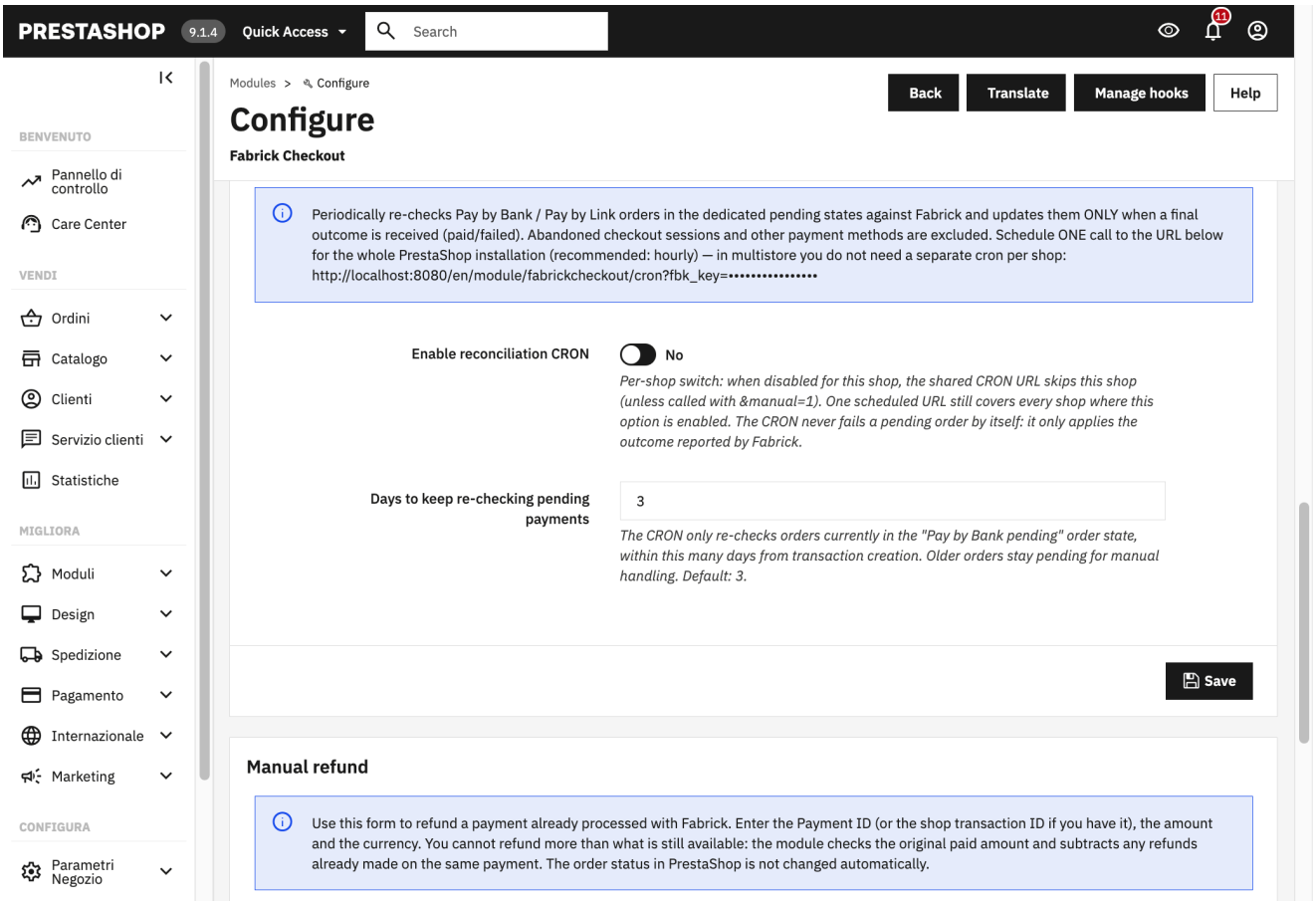
Dedicated statuses are created on install; you may remap them to existing ones.

4.4 Pay by Link validity

Pay by Link validity (hours): link lifetime (typically up to 48-49 hours per Fabrick rules).

4.5 Pending payments reconciliation (CRON)

Back-office section: **Pending payments reconciliation (CRON)**.



Pending payments reconciliation (CRON) - back office

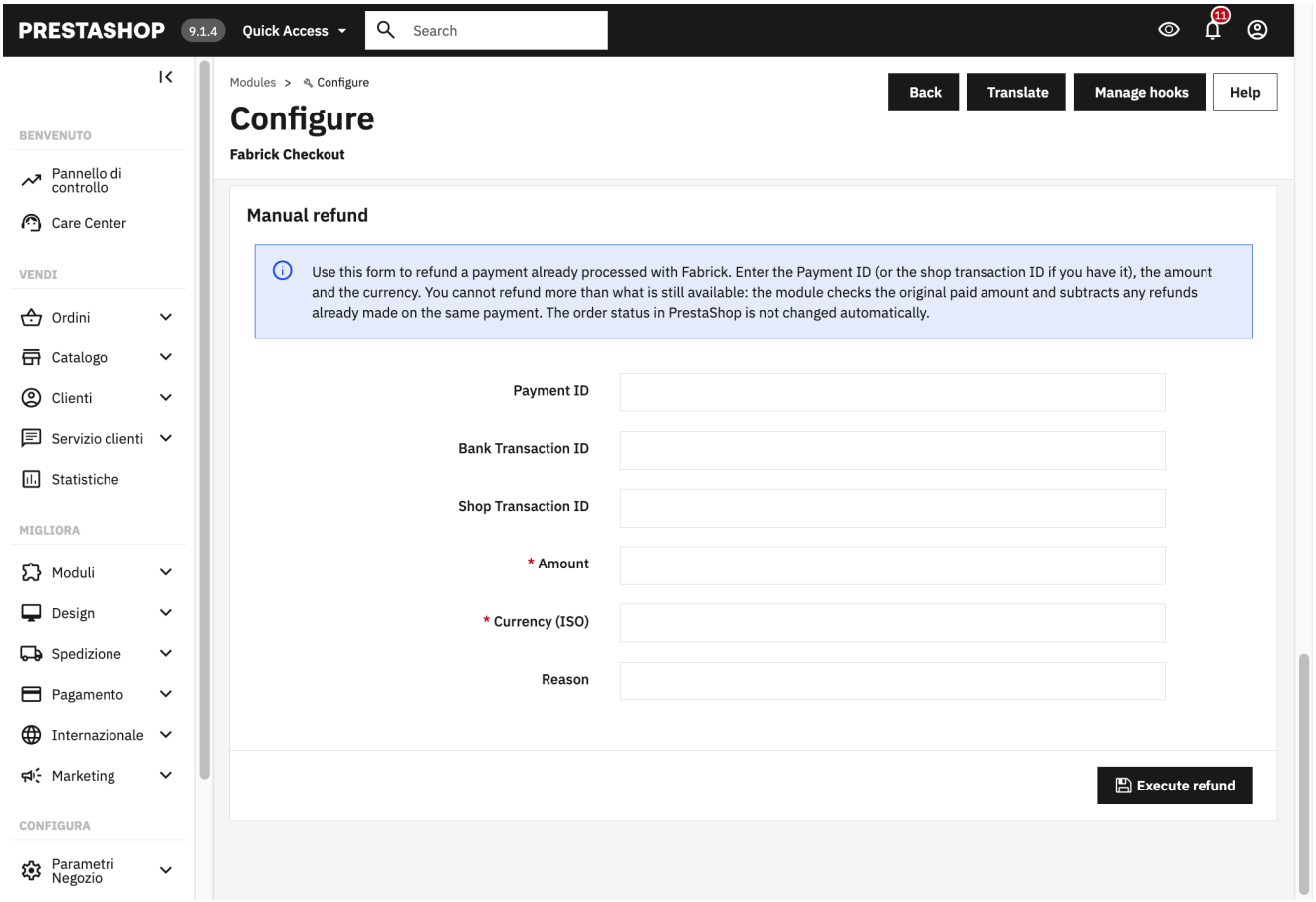
Setting	Description
Enable reconciliation CRON	Enables reconciliation logic
Days to keep re-checking pending payments	Look-back window in days

Schedule **one** periodic call (e.g. hourly) to the cron URL shown in configuration. In **multistore** you do not need a cron per shop: the same URL (main domain, without /shop2/) reconciles every shop where “Enable reconciliation CRON” is on.

Example: https://yourdomain/module/fabrickcheckout/cron?fbk_key=...
(The fbk_key in the URL is the default shop’s secret; do not share it publicly.)

4.6 Manual refund

Back-office section: **Manual refund**.



Manual refund - back office

Use this form to refund a payment already processed with Fabrick. Enter the **Payment ID** (or the shop transaction ID if you know it), the amount, the currency and an optional reason, then click **Execute refund**.

- **You cannot refund more than what is still available**: if you already issued one or more partial refunds on the same payment, only the remaining difference can be refunded.
- The **order status in PrestaShop is not updated** automatically. Change it manually from the order page if needed.

---

5. Checkout display mode

The module uses **Redirect** mode (dedicated page with the Web Component).

- The buyer selects **Pay with Fabrick**.
- They are sent to a **dedicated module page** hosting the Web Component.
- With tokenization on, the “Save my card” checkbox lives in the PrestaShop payment form and is posted with the session create.

---

6. Front-office flows

6.1 Standard payment

1. Buyer completes cart and checkout to the payment step.
2. Selects **Pay with Fabrick** (and save-card opt-in if applicable).
3. Completes payment in the Fabrick Web Component (card or available APMs).
4. Lands on PrestaShop order confirmation (or a pending state depending on the method).

6.2 Pay by Link

When enabled, a dedicated option appears at checkout. The buyer uses the link per Fabrick setup; the order may stay pending until a final outcome.

6.3 Saved cards

When tokenization is enabled and the buyer has at least one saved card:

- Checkout may show **Pay with saved card** (brand / last digits / expiry).
- **Customer account** -> **Saved cards**: list and delete (remote revoke on Fabrick when possible).

6.4 Callbacks and notifications

- **Browser callback:** buyer return after payment.
- **S2S notification:** reliable server-side status updates (keep module public URLs reachable).

---

7. Sandbox and testing

1. Enable **Sandbox mode**.
2. Use official Fabrick **test cards** (e.g. frictionless / challenge / failed 3DS authentication).
3. Verify order creation, statuses and card save if used.
4. With **Debug logging** on, inspect var/logs/fabrickcheckout-{shop_id}-YYYY-MM-DD on the server.
5. On a **declined** payment (DECLINED, e.g. test card with failed 3DS) the buyer returns to the payment step and sees a **header** notice that payment was not completed, including Fabrick's reason when available (e.g. "Authorization denied").

Do not use real cards in sandbox.

---

8. Production checklist

[ ] Sandbox mode = **No**
[ ] Production credentials
[ ] Debug logging = **No** (or limited)
[ ] HTTPS and callback/notify URLs reachable by Fabrick
[ ] Cron scheduled for async methods (e.g. Pay by Bank)
[ ] E2E tests for enabled card and APMs

[ ] Apple Pay / Google Pay domain verified if required

9. Troubleshooting

If something does not work as expected, start with the simplest checks (configuration and cache) before digging into logs and network traffic.

The payment method does not appear at checkout.

Confirm the module is installed and enabled, that the cart currency is allowed, and that Shop Login and API Key are set correctly. Country or currency restrictions on the Fabrick profile can also hide the method.

Blank page after “Pay with Fabrick”.

This is often theme or Smarty cache: clear it and retry. Also check the browser console for Fabrick script load errors (fbk-elements.js) and for CSP or network blocks.

The card is not saved.

Tokenization must be enabled in the back office, and the buyer must tick “Save my card” **before** completing payment. With debug on, the paid session log should show requestToken=MASKEDPAN and save_card_requested=1.

The order stays pending.

This can be normal for asynchronous methods (for example Pay by Bank). Check that server-to-server notifications arrive, that cron is scheduled when required, and that order statuses in configuration match expectations.

No message after a failed payment.

With the updated module, returning from buyerKO / DECLINED should show a checkout header notice that payment was not completed (optionally with Fabrick’s reason). If it is missing, clear PrestaShop cache, confirm the module is up to date, and that the actionFrontControllerInitAfter hook is registered (opening the module configuration in the back office triggers self-heal).

A refund does not change the order status.

The **Manual refund** tool refunds via Fabrick and will not let you exceed what is still available (including after multiple partial refunds). Update the PrestaShop order status yourself under **Orders** if needed.

Where to read logs.

With **Debug logging** enabled, API calls are written under the shop’s PrestaShop var/logs/ as a daily file fabrickcheckout-{shop_id}-YYYY-MM-DD (e.g. fabrickcheckout-1-2026-09-04). At most 14 days are kept. In production, leave debug on only for the time needed to diagnose.

© Fabrick | PrestaShop module Fabrick Checkout 1.0.1