Shopper Signals for PrestaShop

Installation, connection, operation, privacy and troubleshooting guide

Module version	2.0.4
Platform	PrestaShop
Purpose	Collect product questions and shopping intent signals, and provide the Shopper Signals administrative experience inside the PrestaShop Back Office.
External service	A Shopper Signals account and application credentials are required. The module communicates with the Shopper Signals service over HTTPS.

Important: Keep this document with your module package. Support for Marketplace purchases must be requested through the PrestaShop Marketplace support channel.

1. Overview

Shopper Signals connects a PrestaShop store to the Shopper Signals service. Once connected, the module can display the storefront widget on product pages, securely proxy storefront interactions, expose the store catalog to the connected service through signed requests, and embed the Shopper Signals administrative interface inside the PrestaShop Back Office.

The module is designed for multistore installations. Each **id_shop** receives its own local installation identifier, shared secret and connection status.

2. Requirements

- A supported PrestaShop installation within the compatibility range declared by the module.
- PHP cURL extension enabled for HTTPS API requests.
- PHP OpenSSL extension enabled to encrypt local connector secrets.
- Outbound HTTPS access to the Shopper Signals service.
- A Shopper Signals application with a Client ID and Client Secret.
- For multistore installations, select a specific shop before configuring the connector.

3. Installation

Install the ZIP from the PrestaShop Back Office using the standard module upload workflow. The archive contains one top-level folder named **shoppersignals**.

- Open the Module Manager and upload the Shopper Signals ZIP.
- Install the module.
- Open the Shopper Signals connection page from the Back Office menu.
- If multistore is enabled, select the shop that you want to connect.
- Enter the Client ID and Client Secret generated for your Shopper Signals application.
- Click **Connect and test**. The Client Secret is used for pairing and is not stored locally.

During installation, the module creates its own database table for connection metadata and registers the required hooks. Existing PrestaShop core tables are not modified.

4. Connection and security model

The connector uses a unique installation identifier and a cryptographically random shared secret for each shop. The shared secret is encrypted locally with key material derived from the PrestaShop installation before being stored in the module table.

Control	Behavior
Pairing	Uses the merchant-provided Client ID and Client Secret over HTTPS. The Client Secret is not stored locally after pairing.
Service-to-store requests	Requests are authenticated with an installation ID, timestamp, nonce and HMAC signature.

Control	Behavior
Back Office session	The embedded administrative interface receives a short-lived signed session token associated with the authenticated employee and active shop.
Stored secrets	Connector shared secrets are encrypted at rest in the module table.
Transport	Remote API communication verifies TLS certificates and hostnames.

5. Storefront behavior

On product pages, the module renders the Shopper Signals widget container only when the current shop is connected. The storefront runtime is intentionally delivered from the Shopper Signals service and is registered as a remote JavaScript asset. Storefront API operations are sent through the module proxy rather than exposing connector secrets to the browser.

The widget receives product, category, locale and customer context needed by the configured Shopper Signals features. For authenticated customers, customer identifier, name and email may be supplied to the widget so that supported interactions can be associated with the customer.

Privacy note: The merchant remains responsible for presenting any privacy notices and obtaining any consent required by applicable law for the store configuration and enabled Shopper Signals features.

6. Embedded Back Office dashboard

When the shop is connected, the module embeds the Shopper Signals administrative interface in the Back Office. The host page validates the origin of cross-window messages before issuing a short-lived session. Connection credentials are managed on a separate Back Office connection page.

Billing return parameters are validated before the merchant is redirected back to the embedded dashboard. The return state is encrypted locally and has a limited lifetime.

7. Catalog access

The connected service can query products and categories through a signed module front controller. Supported catalog operations include single product lookup, paginated product listing, product search, category search and category lookup.

Requests are bound to the installation identifier and signed query values. Product and category queries are scoped to the connected **id_shop**.

8. Personal data and GDPR hooks

Depending on the enabled features and customer state, Shopper Signals may process personal data such as customer identifiers, names, email addresses, product interaction context and submitted questions/signals.

- The module integrates with the PrestaShop GDPR export hook when that hook is available.
- The module integrates with the PrestaShop GDPR deletion hook when that hook is available.
- GDPR requests are sent to the connected Shopper Signals service over authenticated HTTPS requests.

- Employee name and email are included in the short-lived Back Office session token to identify the authenticated administrator in the embedded experience.

9. Multistore behavior

Each store is configured independently. A merchant must select a specific shop in the PrestaShop multistore selector before connecting or testing Shopper Signals. Connection state is never intentionally shared across different **id_shop** values.

10. Updates

Install updates using the standard PrestaShop module update workflow. Version 2.0.4 includes a non-destructive upgrade step and keeps the existing Shopper Signals installation table and connector state.

Do not delete the module folder manually when updating. Use PrestaShop module management so upgrade logic can run correctly.

11. Uninstallation

Uninstall the module from the PrestaShop Module Manager. During uninstallation, the module attempts to notify the connected Shopper Signals service for each registered shop, removes its Back Office tabs, removes its own installation table, and then completes the standard PrestaShop module uninstall process.

Data impact: Uninstalling removes the local Shopper Signals connector table. Data already stored in the external Shopper Signals service is governed by that service and the merchant account configuration.

12. Troubleshooting

Symptom	Checks
Connection cannot be established	Confirm the active shop, Client ID, Client Secret, outbound HTTPS access, cURL and TLS certificate validation.
Integration test fails	Verify the shop is connected, the service is reachable and the server clock is accurate because signed requests are time-bound.
Widget does not appear	Confirm that the current shop is connected, the page is a product page, and the remote storefront runtime is reachable from the customer browser.
Dashboard does not load	Confirm the connection is active, the browser permits the embedded service, and content security or network policies allow the Shopper Signals host.
Multistore configuration is unavailable	Select one specific shop instead of an all-shops or shop-group context.

13. Support

For a module obtained through the PrestaShop Marketplace, use the support channel available from the Marketplace purchase or product page. Include the PrestaShop version, module version, active shop identifier, the exact error message and the steps needed to reproduce the problem. Never send a Client Secret in a support message.

14. Security recommendations for merchants

- Keep PrestaShop, PHP and the module updated to supported security releases.
- Protect Back Office administrator accounts with strong authentication and least-privilege access.
- Do not expose Client Secrets in screenshots, logs or support tickets.
- Keep server time synchronized because signed connector requests have a short validity window.
- Restrict outbound and browser content security policies to the external host declared by the module when custom hosting policies are used.

End of merchant documentation.