PolarisPrestaConnector - Documentation

PolarisPrestaConnector

Ce document est la documentation unique du module `PolarisPrestaConnector`. Il regroupe la présentation du module, ses prérequis, l'exhaustivité de ce qu'il sait faire, les données qu'il manipule et l'endroit où il les place dans PrestaShop, son paramétrage, deux foires aux questions (« comment faire » et « résolution des problèmes »), et enfin une documentation technique destinée aux webmestres.

Vous pouvez télécharger la dernière version du module à l'adresse suivante : https://ftp.vega-net.net/dists/polarisprestaconnector/stable/latest/polarisprestaconnector.zip

1. Présentation

`PolarisPrestaConnector` est un module PrestaShop qui établit une passerelle de synchronisation bidirectionnelle entre votre boutique PrestaShop et votre logiciel de gestion de magasin **Polaris**.

La synchronisation se fait dans les deux sens :

- **importation** (Polaris → PrestaShop) : catalogue produits, stocks, prix et promotions, clients, bons et cartes prépayées ;
- **exportation** (PrestaShop → Polaris) : commandes (ventes), retours et remboursements, et clients.

Dans toute cette documentation, le terme *importation* désigne le transfert de données du backoffice Polaris vers PrestaShop ; par conséquent, *exportation* désigne le processus inverse. La suite du document précise quelles données sont optionnelles et lesquelles sont obligatoirement gérées, ainsi que le sens de chaque échange.

Support

Pour obtenir du support sur ce module, merci de joindre le service technique Polaris usuel.

2. Prérequis

Le module est conçu et développé pour fonctionner avec les versions minimales suivantes :

- **PrestaShop 8.1.7** ou supérieur (dont **PrestaShop 9**) ;
- **PHP 8.1** pour PrestaShop 8.x et **PHP 8.4** pour PrestaShop 9.x ;
- **MariaDB 13** ;
- un environnement d'hébergement capable de **communiquer vers l'extérieur sur le port de contact de Polaris** (souvent **3443**). Par exemple, les offres mutualisées d'os2switch ne sont pas compatibles en l'état (le port doit être déverrouillé dans le firewall applicatif), celles d'infomaniak demande également l'ouverture d'un port (https://www.infomaniak.com/fr/support/faq/2126/gerer-le-pare-feu-des-hebergements-web-serveur-cloud)…

! Si vous rencontrez une erreur persistente de Timeout au jumelage du connecteur vers le backoffice, vérifier que vous n'êtes pas dans un tel cas !

Aucun support ni correctif ne sera fourni pour supporter des versions moins élevées de ces dépendances.

3. Fonctionnalités

Sous réserve du choix du mode de fonctionnement et d'un paramétrage correct, le module prend en charge :

- l'importation des **quantités en stock** dédiées à la vente en ligne, des **prix** et des **prix promotionnels** (soldes comprises) des articles ;
- les **informations générales** sur les produits (nom, référence, photo, description, marque, classification, catégorisation) — *les marques et catégories ne sont pas créées automatiquement : elles sont assignées par correspondance* ;
- l'**éclatement des produits** par critères ou leur mise en déclinaison (couleur, taille…) — *les attributs utilisés pour la mise en déclinaison sont créés automatiquement, mais vous devez avoir préalablement créé les groupes d'attributs et les avoir affectés aux différents critères* ;
- la **classification automatique** des produits selon des règles de correspondance (mappage) ;
- la possibilité de **filtrer les références importées** depuis les règles de correspondance (si correspondance, alors ne pas importer) ;
- le support du **ramasse-miettes (catch-all)** dans les règles de correspondance (si aucune correspondance, alors…) ;

- le support des **méta-catégories** (mappage en fonction d'une combinaison logique de plusieurs catégories backoffice existantes et/ou de la marque) ;
- le **calcul automatique de la catégorie par défaut** ;
- l'acquisition et la présentation des **disponibilités des articles dans chaque magasin** ;
- la modification de la **visibilité des produits selon leur stock** ;
- le **classement automatique des produits en promotion** dans des catégories (par correspondance) ;
- la **multiplicité des grilles de tailles** ;
- l'**exportation des commandes et des retours** sous forme de ventes sur le backoffice ;
- l'**exportation des modes de règlement** (par correspondance) ;
- l'**exportation des clients** hors marketplace ayant commandé ;
- l'**importation des clients** disposant d'une adresse e-mail ;
- l'**importation des bons** d'avoir, d'acompte et cadeaux des clients importés, ainsi que des **cartes prépayées (cartes cadeaux)** ;
- la mise à disposition d'un **relevé de la collecte de TVA** facturée dans les commandes, par pays de livraison et par taux ;
- la **destruction automatique des photos** des produits désactivés et redirigés n'ayant plus d'activité depuis plus d'un mois ;
- la prise en charge d'une fonctionnalité de **redirection d'URL** pour les reprises de site.

Modes de fonctionnement

Le module possède deux modes :

1. le **mode basique** : le module synchronise uniquement les stocks, les prix et les promotions en se basant sur la référence fournisseur Polaris. *Dans ce mode, vous devez entretenir les références de ce fournisseur dans la fiche produit PrestaShop avec celles des produits correspondants sous Polaris, taille par taille.*
2. le **mode complet** : en plus de toutes les actions du mode basique, le module crée et tient à jour les fiches produits pour tous les produits déclarés comme publiés sur le site Internet. Quand une donnée existe sur le backoffice Polaris, elle est reportée dans PrestaShop à la création ; par la suite, si la donnée existe déjà sous PrestaShop lors d'une synchronisation, elle est écrasée par la valeur de Polaris en fonction de votre configuration. Les données vitales (♜) sont toujours écrasées dans PrestaShop.

Note : les données marquées d'une tour (♜) sont des données vitales de classification. Elles sont toujours écrasées par le module si elles existent sous Polaris.

Fonctionnement général

- À intervalle régulier, il est nécessaire d'appeler la **tâche cron** présente à l'adresse `https://votresite.com/modules/polarisprestaconnector/cron?guard=XXXX`, ou bien en ligne de commande (`php bin/console polarisprestaconnector:sync`). Consultez la page de configuration du module pour connaître la valeur exacte du jeton d'accès (*guard*) et l'URL complète.
- C'est cette tâche qui initie la communication entre le site PrestaShop et votre backoffice et réalise les opérations d'importation et d'exportation en se connectant aux API de Polaris.
- Il est nécessaire d'appeler cette tâche régulièrement.
- La tâche ne resynchronise pas systématiquement la totalité de vos données : dans la mesure du possible, et sauf ordre contraire ponctuel, elle travaille en **incrémentiel**, c'est-à-dire qu'elle ne s'occupe que des données qui ont changé.

Le module est conçu pour avoir un impact minimal sur le *front* du site.

Fonctionnalités avancées

Mise à jour automatique

Le module est doté d'un système de mise à jour automatique. Au moins une fois par jour, lors d'une tâche de synchronisation, il se connecte à nos serveurs pour vérifier si une nouvelle version est disponible. Le cas échéant, il télécharge l'archive ZIP de la mise à jour et la déploie dans le répertoire `/updates/latest`.

Il informe ensuite PrestaShop de la disponibilité de la mise à jour : les utilisateurs disposant des droits peuvent alors la déployer comme n'importe quel module depuis le gestionnaire de modules de PrestaShop. La mise à jour de la base de données est effectuée automatiquement lors de l'appel suivant au module.

Nous vous conseillons de mettre à jour le module régulièrement pour bénéficier d'une meilleure prise en charge de votre backoffice, de nos correctifs de sécurité et de fonctionnement, ainsi que des nouvelles fonctionnalités.

Par défaut, le module est branché sur la branche de mise à jour `stable`, qui contient une version de production. Pour passer sur la branche `testing` (pré-version destinée aux tests), créez un fichier `TESTING` à la racine du site PrestaShop. **Attention : nous le déconseillons formellement pour un site en production — vos données peuvent être endommagées.**

Disponibilité des pièces dans les boutiques

Le module permet, selon sa configuration, d'afficher la disponibilité d'un produit dans un magasin donné. Pour que cette option fonctionne, vous devez préalablement avoir mappé vos magasins Polaris avec vos magasins PrestaShop via le mapping des magasins, disponible dans les options.

Sélecteur de produits connexes

Le module permet, selon sa configuration, de scinder les produits Polaris en plusieurs produits PrestaShop, par exemple pour mieux gérer la couleur. Il garde la trace du lien de parenté des produits ainsi créés et incorpore un *hook* qui génère un sélecteur pour naviguer entre ces produits.

Pour utiliser ce sélecteur, appelez le *hook* `{hook h='displayMultiProductSelector' product=$product}` dans le *template* (`.tpl`) de la page produit, à l'endroit où vous souhaitez générer le code HTML. Vous pouvez en personnaliser le rendu par CSS (cacher la variante affichée actuellement, modifier les couleurs, etc.).

Le sélecteur ne s'affiche pas (rendu vide) lorsqu'il n'y a qu'une seule variante activée et affichable (avec du stock) pour le produit consulté.

4. Données manipulées et leur emplacement dans PrestaShop

Produits

Fonctionnement général :

- sur Polaris, certaines déclinaisons sont **discriminantes** (par exemple la couleur ou la matière), d'autres sont des **tailles** (longueurs de jambe, bonnets…) ;
- chaque couple `{produit – déclinaisons discriminantes}` de Polaris marqué pour être sur Internet provoque la création et l'entretien d'un **produit** sous PrestaShop ;
- dans ce produit PrestaShop, chaque couple `{article – déclinaisons de taille}` de Polaris provoque la création d'une **déclinaison** unique lorsqu'elle n'existe pas déjà ;
- **la clé de synchronisation (la donnée utilisée pour reconnaître l'entité) est le code-barre Polaris de l'article, injecté dans la référence fournisseur du produit, pour le fournisseur nommé `Polaris`.**

Données synchronisées sur le produit

Quand une donnée n'apparaît pas dans la colonne PrestaShop ci-dessous, c'est qu'elle n'est pas synchronisée.

Donnée PrestaShop	Notes
Référence produit ♜	Référence produit sous Polaris concaténée aux libellés des déclinaisons discriminantes, séparées par le caractère `/`.
Référence fournisseur du fournisseur Polaris ♜	L'ID interne du modèle Polaris, agrégé avec ceux des déclinaisons discriminantes, est utilisé ici.
Nom	Constructible selon un modèle à partir des informations.
URL simplifiée	Déduite automatiquement du nom du produit lorsqu'elle est vide sous PrestaShop.
Marque	—
Taux de taxes (TVA)	Autodétecté (1).
Écotaxe	—
Description	—
Code douane (code HS) & pays d'origine	Supporté quand le module Colissimo officiel est installé.
Description Méta	—
Largeur, hauteur, profondeur, poids	—
Prix de vente de base ♜	Le prix le plus bas, toutes déclinaisons et tailles confondues.
Caractéristiques	Définies par mappage des champs dynamiques de Polaris ; voir le paramétrage.
Catégories	Définies par mappage des `RFS`, des `Collections`, `Saisons` et `Catégories dynamiques`.
Photos	—

1. : le backoffice envoie le taux de taxe du produit ; le module sélectionne la première règle de taxe incluant ce taux pour le magasin par défaut du site.

Données synchronisées sur la déclinaison

Donnée PrestaShop	Notes
Référence de la déclinaison ♜	C'est le code-barre Polaris qui est utilisé ici.

Référence fournisseur du fournisseur Polaris ♜	Le code-barre Polaris est utilisé ici comme clé de synchronisation.
Attributs constituant la déclinaison ♜	Les déclinaisons non discriminantes et les tailles Polaris sont utilisées pour créer les déclinaisons PrestaShop.
Prix de vente ♜	Prix de vente en boutique.
Code-barre EAN13	Code-barre fournisseur sous Polaris.
Emplacement du stock	Zonage de la déclinaison concernée ; à défaut, celui du modèle.
Association de photo	Le module rétablit automatiquement les bonnes photos sur les bonnes déclinaisons.

Note : les déclinaisons PrestaShop correspondant à des produits supprimés ou marqués comme non publiés sous Polaris sont automatiquement supprimées de PrestaShop. Si un produit se retrouve ainsi sans déclinaison, il est désactivé s'il ne l'est pas déjà.

Marques

Les marques ne sont pas créées automatiquement par le module : elles sont **mappées**. Chaque fois que le module rencontre une nouvelle marque, il crée une ligne de mappage qu'il faudra configurer dans les options (ou paramétrer un catch-all). Une resynchronisation complète est nécessaire après modification du mappage.

Caractéristiques, tailles & attributs

Les caractéristiques, tailles et autres attributs ne sont pas synchronisés de la même façon que le reste des données. Lorsque le module synchronise un produit référençant l'une de ces données et que la caractéristique ou le groupe d'attributs a été mappé, il vérifie si une valeur équivalente existe : si c'est le cas, il l'utilise ; sinon, il crée la valeur de caractéristique ou l'attribut et l'utilise.

Pour ces données, **seul** la caractéristique ou le groupe d'attributs est mappé — pas les valeurs qui le composent.

Propriétés personnalisables

À partir de **Polaris 15**, votre backoffice peut exposer des **propriétés personnalisables** (composition, conseils d'entretien, pays d'origine, indicateurs oui/non, etc.). Le module les liste dans les *Paramètres du catalogue* et vous permet d'associer chacune à une **caractéristique** PrestaShop.

- Pour chaque propriété mappée, la valeur du produit est reportée **en texte libre** dans la caractéristique choisie ; la valeur de caractéristique est créée automatiquement si elle n'existe pas encore.
- Les propriétés de type **case à cocher** (booléen) sont reportées en **« Oui »** ou **« Non »** (une valeur vide, absente ou égale à `0` vaut « Non »).
- Lorsqu'une propriété **texte** est vide pour un produit, la caractéristique correspondante est **retirée** du produit.
- Une propriété laissée sur *Ne pas utiliser* (valeur par défaut) n'est jamais synchronisée.

Toutes ces opérations sont tracées dans l'audit (création d'une valeur de caractéristique, changement, suppression). Sur les versions de Polaris antérieures à la 15, aucune propriété n'est remontée : la section reste vide et rien n'est modifié. La liste des propriétés disponibles est actualisée au téléchargement de la configuration du backoffice.

Catégories

Les catégories sont systématiquement mappées. À chaque nouvelle catégorie de produit sur le backoffice Polaris, une ligne de mappage est créée. Quand elle est renseignée, tous les produits appartenant à cette catégorie Polaris sont automatiquement assignés aux catégories PrestaShop indiquées pour cette ligne.

Le module essaie au mieux de ne pas toucher aux associations que vous auriez faites manuellement, mais il peut modifier les catégories d'un produit en pensant défaire une association caduque si vous avez associé manuellement votre produit à l'une des catégories gérées par le mappage. Une resynchronisation complète est donc souvent nécessaire après modification du mappage.

La **catégorie par défaut** assignée aux produits est déterminée :

- lors de la création originale du produit ;
- lorsque le produit est resynchronisé et que la catégorie par défaut actuelle est `Accueil` ou n'est plus associée au produit.

Le module choisit systématiquement la catégorie la plus profonde (celle qui a le plus de parents). À défaut, la plus ancienne ; à défaut, la catégorie `Accueil`.

Méta-catégories

Les méta-catégories sont des **catégories virtuelles** côté backoffice, définies entièrement dans la configuration du module. Elles n'existent pas en tant que telles dans Polaris : elles sont construites par le

module comme la **combinaison logique de plusieurs catégories dynamiques existantes**. L'objectif est de créer des catégories PrestaShop dont le peuplement dépend de l'appartenance simultanée d'un produit à plusieurs catégories backoffice.

Par exemple, une méta-catégorie `Jeans longues tailles` qui regroupe les produits correspondant à une combinaison RFS donnée **ET** une longueur de jambe donnée.

Du point de vue du moteur de synchronisation, une méta-catégorie se comporte exactement comme une catégorie dynamique ordinaire : elle peut être mappée vers une ou plusieurs catégories PrestaShop, elle supporte le catch-all (*), et son mappage est géré dans l'onglet de mappage habituel. Les entrées de mappage sont créées à la volée, au fur et à mesure que le moteur rencontre de nouvelles clés composites. Tant qu'une clé composite n'est pas mappée (et qu'aucun catch-all n'est défini), le produit n'est pas assigné à une catégorie PrestaShop au titre de cette méta-catégorie.

Stock (♜)

Le module télécharge les stocks ventilés par point de vente et dépôts dans la table `ps_pc_pol_stocks`, puis les **agrège** dans la table PrestaShop `ps_stock_available` prévue à cet effet, en passant par les objets internes pour déclencher tous les *hooks* du système (afin que les modules réagissant aux changements de stock fonctionnent correctement).

Vous pouvez définir quels magasins participent à l'agrégation pour obtenir les stocks mis à disposition pour la vente en ligne (voir le paramétrage). Vous pouvez aussi vous appuyer sur la table `ps_pc_pol_stocks` pour calculer et afficher une disponibilité d'article en magasin dans votre propre module.

Sur option, une quantité minimale de **stock de sécurité** est déduite par taille de ces stocks agrégés ; cette quantité n'est jamais déduite des informations présentes dans la table `ps_pc_pol_stocks`.

Promotions

Les promotions sont synchronisées sous la forme de **prix spécifiques** PrestaShop. Le module fait au mieux pour en créer le moins possible et les regroupe lorsque la promotion concerne toutes les déclinaisons d'un produit. Il ne crée que des promotions en pourcentage et ne touche pas aux promotions manuelles que le marchand pourrait créer lui-même. Pour cela, il conserve une copie de ses promotions dans les tables `ps_pc_pol_promotion` et `ps_pc_pol_promotion_detail`.

Le module retire de lui-même les promotions caduques. Il est possible de **mapper une promotion avec une catégorie**, de sorte qu'un produit concerné par cette promotion soit automatiquement associé à cette catégorie PrestaShop le temps de la promotion — utile pour nommer ou présenter une opération commerciale sur le site.

Clients

Les clients sont exportés vers le backoffice Polaris lors de la tâche automatisée de synchronisation si la configuration le permet, puis importés vers PrestaShop selon les mêmes conditions, afin qu'une modification sur l'une des plateformes se répercute sur l'autre, tout en donnant la priorité au site Internet.

- Un client est importé sous PrestaShop depuis Polaris s'il possède une **adresse e-mail valide**.
- Les clients primo-importés sous PrestaShop depuis Polaris sont marqués comme faisant partie du groupe de clients **« Clients Polaris »**.
- Les modifications clients sont exportées vers Polaris si le client a passé au moins une commande sur le site PrestaShop **ou** s'il fait partie du groupe « Clients Polaris », **et** s'il ne s'agit pas d'un compte invité ni d'un compte écran créé automatiquement par une plateforme e-commerce (*compte écran* `…@amazon.marketplace.com`) ; le module tente de les reconnaître dans la mesure du possible.

Donnée	Notes
Email	Mail du chef de famille.
Raison sociale	En majuscules, si applicable.
Nom	En majuscules.
Prénom	Capitalisé.
Civilité	Importée/exportée seulement si un titre de civilité identique existe sous PrestaShop, sinon ignorée.
Inscription à la newsletter	Selon l'adhésion du client à la liste de diffusion système « Mailing par mail ».
Mot de passe	Non synchronisé ; créé aléatoirement à la création du client sous PrestaShop.
Adresse	Le système remonte l'adresse de facturation de la dernière commande comme adresse principale. Si le client n'a jamais commandé, aucune information ne remonte. *Attention :* c'est l'adresse utilisée lors de la commande qui remonte, et non les modifications faites par l'internaute *après* la commande. Les numéros de téléphone ne remontent pas. Les

	adresses créées sous Polaris ne redescendent pas sous PrestaShop.

Notes : pour des raisons de sécurité des données, les clients ne sont exportés depuis PrestaShop vers le backoffice que lorsqu'ils subissent une modification sur PrestaShop. Ils ne sont pas réexportés en totalité lorsqu'une resynchronisation totale est demandée. *Pour obtenir le renvoi de tous les clients vers Polaris, prévoyez un* `UPDATE` *du champ* `date_upd` *de la table* `ps_customer`*.* La demande de resynchronisation provoque en revanche la redescente de tous les clients depuis le backoffice.

Bons

Les bons sont importés depuis Polaris lors de la tâche automatisée de synchronisation, si la configuration le permet, sous forme de **réduction de panier** avec un code. Seuls les bons cadeaux, les chèques cadeaux et les avoirs peuvent (sous options) être exportés depuis Polaris vers PrestaShop ; l'inverse n'est pas pris en charge.

Donnée	Notes
Code du bon	Code EAN généré par Polaris.
Montant du bon TTC	Exprimé TTC, ou 0 si le bon est en pourcentage.
Remise en %	0 quand le bon est exprimé en valeur monétaire.
Quantité disponible	Toujours 1 quand le bon est créé, 0 si utilisé.
Activé	Vrai si le bon est valide ; faux s'il est désactivé ou plus valable sous Polaris.
Sécable	Non : les bons Polaris ne sont pas sécables.
Client	Lorsque l'importation client est activée, les bons sont reliés au client. Sinon, les bons reliés à un client inexistant sous PrestaShop sont ignorés et non importés.
Nom	Libellé du bon sous Polaris.

Note : les bons avec l'option « Plus cher » activée ne sont pas exportés vers PrestaShop.

Cartes prépayées (cartes cadeaux)

Les cartes prépayées sont importées depuis Polaris lors de la tâche automatisée de synchronisation, si la configuration le permet. Elles sont importées sous forme de **réduction panier non sécable** avec un code ; le montant de la carte est réapprovisionné par le module.

Donnée	Notes
Code de la carte	Code EAN généré par Polaris.
Montant disponible	Exprimé TTC.
Quantité disponible	1 tant que le solde n’est pas entièrement consommé.
Activé	Vrai si la carte est valide ; faux si désactivée ou plus valable sous Polaris.
Sécable	Non.
Client	Aucun.
Nom	« Carte prépayée », suivi du code de la carte.

Commandes

Les commandes sont exportées vers Polaris lors de la tâche automatisée de synchronisation dès qu’elles remplissent **toutes** les conditions suivantes :

- le statut actuel de la commande est considéré comme payé (une facture a été émise par PrestaShop) ;
- le statut actuel de la commande n’est pas « Erreur d’exportation vers Polaris » ;
- la date de création de la commande remonte à moins de 90 jours ;
- la dernière mise à jour de la commande remonte à moins de 30 jours ;
- tous les règlements utilisés pour la payer sont mappés dans la configuration ;
- le montant initial de la commande est strictement égal au montant encaissé ;
- la commande contient au moins un article à livrer.

Lorsqu’un paiement n’est pas mappé, ou qu’un nouveau type de paiement est rencontré lors d’une exportation, ce dernier est automatiquement ajouté aux modes de règlement devant être mappés.

À l’issue de l’exportation, la commande reçoit l’un des deux statuts suivants, créés automatiquement par le module lorsqu’ils n’existent pas :

- **Exportée vers Polaris** : la commande a bien été exportée, sans erreur ;
- **Erreur d’exportation vers Polaris** : la commande n’a pas pu être exportée. L’erreur figure dans le journal de PrestaShop et dans l’audit de synchronisation.

Le système est prévu pour retenter l'envoi de la commande dès qu'il détecte une modification respectant les conditions ci-dessus.

Données exhaustives prises en charge par l'export (manuel ou automatique) :

Donnée	Notes
Référence de la commande	L'ID de la commande n'est pas significatif. Pour gérer les commandes scindées, seule la référence de la commande, préfixée par l'année (ex. `2024/UXHBSDSF`), sert de dénominateur.
Date de la vente	C'est la date du paiement qui est prise en compte (et non la date de la commande), pour des questions fiscales.
Type de la vente	Les commandes exportées sont typées « Vente Internet » sous Polaris.
Magasin	C'est le magasin de référence Internet qui est indiqué comme ayant réalisé la vente.
Produits	Pour chaque produit est exporté : le code-barre Polaris (à défaut le Gencode s'il est présent), la quantité, le prix de vente TTC consenti, le montant TTC de la remise éventuelle (typée en Promotion), et le taux de TVA pratiqué sur le site.
Les règlements	La liste exhaustive des règlements, mappés selon la configuration.
Remise panier	Considérée comme un mode de règlement et, à ce titre, soumise au mappage (par défaut `RE`).
Client	Si le client n'est pas un compte invité et si l'option d'exportation des clients vers Polaris est activée, le client est lié à la vente.

Les commandes sont toujours exportées dans la devise par défaut du site : si la commande est passée dans une autre devise, tous les montants sont convertis dans la monnaie principale du site.

Modification, annulation, retours & remboursement

Le module ne traite les modifications, annulations, retours & remboursements que si l'option éponyme a été activée dans sa configuration ; sinon, seule la version primitive de la commande est exportée vers Polaris.

Note : en raison de la grande liberté d'édition des commandes sous PrestaShop, il est possible d'aboutir à un résultat non importable dans Polaris, notamment en cas de remboursement partiel ou de

modifications successives. Dans ce cas, une erreur d'exportation se produit et il faut corriger la situation manuellement, directement sur le backoffice.

Quand le système détecte une modification d'une commande payée, une annulation, un retour ou un remboursement, il suit toujours la procédure suivante :

- si la commande est exportable et ne l'a jamais été, le module exporte d'abord une version originale de la commande en l'état ;
- si la commande a été dans un état exportable, n'a pas été exportée à l'époque et ne se trouve plus dans un état exportable, le module n'exportera pas cette commande ;
- s'il y a déjà eu une ou plusieurs modifications exportées pour cette commande, le module annule ces modifications en les passant sur le backoffice en sens inverse ;
- enfin, si la commande se trouve dans un état exportable, ou si elle n'y est plus mais l'a été un jour et a déjà été synchronisée, le module exporte une vente qui reflète la **différence** entre la commande d'origine et la commande à l'heure de l'exportation.

Redirections URL

Le module intègre un mécanisme de redirection d'URL permettant de mapper d'anciennes URL (issues d'un ancien site ou d'un autre CMS) vers les URL PrestaShop cibles. Il repose sur un fichier texte de règles (`redirects.list`) et s'articule en deux volets : la gestion manuelle via l'onglet **Redirections** du module (voir le paramétrage), et la génération automatique lors des synchronisations produits.

Fichier de règles `redirects.list`

Le fichier est stocké à la racine du site, dans `var/redirects.list`. S'il n'existe pas, le module le crée automatiquement avec un en-tête commenté décrivant le format.

Chaque ligne active représente une règle :

```
<masque_regex> <séparateur> <destination>
```

- le **masque** est une expression régulière PHP appliquée à l'URL entrante (sans domaine ni *query string*) ;
- le **séparateur** peut être `|`, `;` ou une tabulation — il est auto-détecté ligne par ligne ;
- la **destination** est soit une URL cible, soit une directive spéciale (voir ci-dessous) ;
- les lignes vides et celles commençant par `#` sont ignorées (commentaires).

Exemple de fichier :

```
# Fichier de règles de redirections URL - PrestaConnector
#
# Format : <masque_regex> <séparateur> <url_destination>

/ancienne-page/(?P<slug>[^/]+)\.html       | /nouvelle-page/{{slug}}
/fr/catalogue/new/femme/chaussures\.awp    | category(51)
/fr/catalogue/new/homme/mocassins\.awp     | product(102)
^/fr/article/.*?/(?P<id>\d+)\.awp$         | /produit/{{id}}
```

Groupes capturés et directives spéciales

- **Groupes nommés** : les groupes nommés du masque (`(?P<nom>...)`) peuvent être réutilisés dans la destination via `{{nom}}`. Les groupes positionnels classiques (`$1`, `$2`…) sont également acceptés.
- **Directive `category(<id>)`** : lorsque la destination est de la forme `category(<id>)`, le module résout automatiquement le lien PrestaShop de la catégorie correspondante (via `Category::getLink()`).
- **Directive `product(<id>)`** : lorsque la destination est de la forme `product(<id>)`, le module résout automatiquement le lien PrestaShop du produit correspondant (via `Link::getProductLink()`).

Comportement et application

> **Si une règle avec le même masque existe déjà dans le fichier, elle n'est pas modifiée.** Pour changer une destination existante, supprimez manuellement la ligne dans `redirects.list`, puis ressaisissez la règle. Ce comportement protège les règles ajoutées manuellement ou automatiquement contre tout écrasement involontaire.

À chaque synchronisation d'un **nouveau produit**, le module peut générer automatiquement une règle de redirection (selon le pattern configuré dans l'onglet **Redirections**) : si elle n'existe pas, elle est ajoutée ; si elle existe déjà, elle est conservée telle quelle. La génération automatique ne s'applique qu'aux **nouveaux produits** : pour reconstruire les redirections sur des produits déjà synchronisés, lancez une **resynchronisation totale** en ajoutant `&rebuild-redirects=1` à l'URL de la tâche, par exemple :

```
https://www.monsite.fr/modules/polarisprestaconnector/cron?guard=XXXX&full=1&rebuild-redirects=1
```

Lorsqu'une page n'est pas trouvée (erreur 404), le module confronte séquentiellement l'URL entrante à chaque règle du fichier : la **première** règle dont le masque correspond déclenche une redirection

HTTP 302 vers la destination calculée. Si aucune règle ne correspond, la requête suit son cheminement normal dans PrestaShop.

5. Paramétrage

Installation et association

Pour fonctionner, le module a besoin de se connecter aux API de votre backoffice. L'**association** est le processus qui établit cette connexion. Rendez-vous dans la page de configuration du module, accessible depuis l'écran de gestion des modules.

Si le module n'a jamais été associé ou détecte des identifiants périmés, il vous proposera automatiquement de procéder à une association avant de poursuivre la configuration ; sinon, pour une association manuelle, cliquez sur le bouton `Déconnecter`.

Pour associer le module à votre installation Polaris :

1. **Dans la page de configuration du module PrestaShop**, saisissez votre **numéro de licence Polaris**. Il s'agit d'une chaîne alphanumérique de la forme `pl-p8-111-2222`. Vous pouvez également saisir directement l'**adresse de connexion** de votre Polaris si vous la connaissez (elle doit être en `https://`) : dans ce cas, le module n'interroge pas le gestionnaire de licence et utilise l'adresse telle quelle.
2. **Sur Polaris**, dans `Système > Gestion des appareils` :
   - *si votre site n'a jamais été paramétré* : cliquez sur « `nouveau prestataire` » en choisissant les API « `WebConnect` » ; sélectionnez les paramètres les plus adaptés à votre site (droits, données…) ; à l'issue, un **code PIN** vous est remis ;
   - *si votre site a déjà été paramétré* : choisissez son accès dans la liste, modifiez-le et cliquez sur « `ré-associer` » pour générer un nouveau code PIN.
3. **Reportez ensuite le code PIN sur PrestaShop.**

Mise en place de la tâche automatisée

Le module se connecte au backoffice Polaris via une tâche automatisée appelée à intervalles réguliers et relativement courts (en général toutes les 5 minutes), soit par *Webcron*, soit par *Cron*.

Méthode Webcron

À privilégier lorsque vous n'avez pas accès à la ligne de commande de votre serveur (cas fréquent). Elle consiste à appeler une URL dédiée, protégée par un jeton d'accès (*guard*). Consultez la page de

configuration du module pour connaître l'URL exacte. Avec `crontab` (Linux) et `curl`, l'appel ressemble à :

```
*/5 * * * * /usr/bin/curl -k --silent --output /dev/null
        https://monsite.com/modules/polarisprestaconnector/cron?
        guard=XXXX
```

Méthode Cron (CLI)

À privilégier lorsque vous avez accès à la ligne de commande. La synchronisation peut être déclenchée via la console Symfony de PrestaShop :

```
# Synchronisation incrémentale (seules les nouveautés depuis la
        dernière synchro)
php bin/console polarisprestaconnector:sync


# Resynchronisation complète de tout le catalogue
php bin/console polarisprestaconnector:sync --full
```

La commande est à lancer depuis la racine de votre installation PrestaShop. Les journaux d'audit s'affichent en temps réel dans le terminal. Elle est particulièrement utile pour :

- lancer une resynchronisation complète en production sans passer par l'interface web ;
- intégrer la synchronisation dans un pipeline CI/CD ou un script de déploiement ;
- déboguer une synchronisation qui échoue, en observant les logs dans le terminal.

Pour un déclenchement automatique via `crontab`, les deux méthodes sont équivalentes. La méthode CLI évite d'exposer l'URL de cron sur le réseau :

```
*/5 * * * * php /var/www/monsite/bin/console
        polarisprestaconnector:sync --quiet
```

L'option `--quiet` supprime les messages de statut tout en conservant le code de retour (`0` = succès, `1` = erreur), ce qui permet à votre supervision de détecter les échecs.

Bonnes pratiques

- Menez vos tests, en particulier lors d'un premier raccordement, sur un **site de préproduction**.
- Avant chaque changement sur un site de production, **sauvegardez** votre base de données et conservez cette sauvegarde.
- Sélectionnez les produits à publier de manière raisonnée, en commençant par une ou deux références aux caractéristiques distinctes, afin de mesurer l'impact de vos options. Avec peu de références, vous pourrez ajuster et multiplier les essais jusqu'à

trouver la configuration qui correspond à votre méthode de travail.
- Pensez à **désactiver l'exportation des clients** si vous ne voulez pas polluer votre backoffice avec des clients de test.

Actions de maintenance

Les actions de maintenance sont présentées sous forme de boutons dans le bandeau supérieur des pages de configuration :

- **Resynchroniser** : demande au module d'effectuer une synchronisation totale lors de sa prochaine connexion (re-téléchargement de toutes vos données depuis le départ) ;
- **Télécharger la configuration** : demande au module de télécharger tous les paramètres positionnés sur votre backoffice (tables, magasins Internet…) ;
- **Déclencher la tâche** : lance immédiatement la tâche automatisée de synchronisation ;
- **Mappings** : lorsqu'il est affiché, permet de définir les mappages des catégories et autres données ;
- **Déconnecter** : vous dissocie du backoffice pour effectuer un nouveau jumelage.

Configuration générale

Options globales

- **Synchronisation activée** : active ou désactive rapidement la synchronisation des données (par exemple pour mettre l'opération en pause). La désactiver empêche toute communication entre le site et le backoffice Polaris, dans un sens comme dans l'autre.
- **Mode de synchronisation** : le mode basique importe seulement les stocks, les prix et les promotions ; le mode complet crée et entretient en plus les fiches produits (référence, nom, description, photos, classification). Quel que soit le mode, les données exportées (commandes, clients…) sont toujours les mêmes.
- **Intervalles de synchronisation** : fréquence de mise à jour de vos informations. Plus la fréquence est élevée, plus la puissance de traitement nécessaire est grande. *Quelles que soient les valeurs indiquées, les synchronisations n'interviennent que lors de l'appel à la tâche cron ; les fréquences ne peuvent donc pas être plus basses que celle de cet appel.*

Accès maintenance Polaris

Le module embarque une fonctionnalité de **connexion SSO** réservée à l'équipe support technique VEGA. Quand elle est activée, un bouton **Connexion SSO VEGA** apparaît sur la page de connexion du backoffice PrestaShop, à côté du formulaire de login habituel.

Ce bouton lance une authentification OpenID Connect : une fois identifié, le technicien est automatiquement connecté avec un compte administrateur local dédié (`admin@vega-info.fr`), créé à la volée s'il n'existe pas. Cela permet au support d'intervenir rapidement sans demander d'identifiants ni créer de compte préalable.

Important : cette fonctionnalité **n'affecte en rien la connexion habituelle** ; vos utilisateurs, administrateurs et techniciens internes continuent à se connecter normalement. Seul le bouton additionnel est ajouté.

L'option **Autoriser l'accès maintenance aux équipes de Polaris**, dans la page **Options générales**, permet d'**activer ou de désactiver** cette fonctionnalité. Elle est **activée par défaut**, car nécessaire au support pour intervenir efficacement. Si vous la désactivez, le bouton disparaît et toute connexion par cette voie est refusée — vous devrez alors fournir vous-même les accès au support en cas d'incident.

Paramètres du stock

- **Stock de sécurité** : soustrait une ou plusieurs pièces du stock de chaque taille, afin de ne jamais être hors stock sur le site. Par exemple, avec « 1 » et 4 pièces en taille L, le site considérera qu'il en reste 3.
- **Visibilité des produits en stock** : détermine s'il faut afficher les produits dans le catalogue et les recherches lorsqu'ils n'ont plus de stock, ou les cacher. Un produit caché reste accessible via son URL pour ne pas nuire à votre référencement.
- **Composition du stock en ligne** : détermine quels magasins mettent leur stock à disposition pour la vente en ligne. Si aucun n'est coché, le module considère que tous les magasins participent.

Paramètres du catalogue

- **Fonctionnalités de catalogue** : activent le sélecteur de produits connexes ou l'affichage de la disponibilité en magasin (même un magasin ne participant pas au stock de vente en ligne, et le stock minimal n'influe pas sur la quantité présentée ici).
- **Attributs discriminants** : choisissez de créer un produit PrestaShop pour chaque produit Polaris, ou plutôt un produit PrestaShop par couleur ou matière différente. Sélectionnez alors les attributs discriminants. Dans le doute, laissez tout décoché.
- **Mappage des attributs de produit** : fait correspondre chaque attribut Polaris à un groupe d'attributs PrestaShop. Les attributs internes au groupe sont créés automatiquement selon les besoins (*par exemple, « coloris » créera rouge, bleu…*). Si vous n'affectez pas de groupe d'attributs en face d'un attribut Polaris,

l’information n’est pas synchronisée.

- **Mappage des caractéristiques de produit** : identique à l’option précédente, mais pour les caractéristiques.
- **Mappage des propriétés personnalisables** : à partir de Polaris 15, fait correspondre chaque propriété personnalisable Polaris (composition, entretien, origine, oui/non…) à une caractéristique PrestaShop. La valeur du produit y est reportée en texte libre (les cases à cocher deviennent « Oui »/« Non »). Laissez « — » (*Ne pas utiliser*, valeur par défaut) pour ignorer la propriété. La section n’apparaît que si votre Polaris expose des propriétés. Voir *Propriétés personnalisables* dans « Données manipulées ».
- **Mappage des couleurs** : sélectionnez un groupe d’attributs PrestaShop pour lequel un filtre de correspondance remplace la couleur fournie par une couleur générique, afin de limiter le nombre de couleurs présentées (*par exemple, « moutarde », « paille », « or » et « doré » → « jaune »*). Le module est livré avec une liste d’équivalences (noir, blanc, beige, gris, bleu, rouge, orange, marron, vert, jaune, violet, rose) et positionne « multicolore » lorsqu’il détecte plusieurs couleurs (ex. `noir&blanc`). Vous pouvez compléter cette liste dans `Catalogue > Mappages Polaris`.
- **Mappage des réductions** : classe les produits concernés par une promotion dans une catégorie PrestaShop spécifique — utile pour une page dédiée à un événement (ex. *Black Friday*). Les promotions sont automatiquement ajoutées ou retirées de cette liste lors de la synchronisation.
- **Options d’importation de produits** : choisissez quelles données sont mises à jour lors d’une synchronisation complète. Cette section n’est pas disponible en mode basique.
- **Nettoyage automatique des attributs non utilisés** : lorsqu’elle est activée, le module recherche et supprime une fois par semaine tous les attributs qui ne sont plus utilisés par aucun produit (activé ou non). Seuls sont concernés les groupes d’attributs utilisés dans le mapping des attributs de produit et celui des grilles de tailles.
- **Ne pas importer les produits sans stock** : ignore les produits sans stock en ligne (selon votre composition du stock en ligne) et ne les crée pas s’ils n’existent pas sous PrestaShop ; s’ils existent, ils sont mis à jour.
- **Format du nom de produit** : configure le formatage automatique du nom par une formule. Les balises s’écrivent `{{nom}} {{couleur}}` où `nom` et `couleur` sont des attributs. `nom` est le nom du produit tel que disponible sous Polaris. Les attributs vides sont ignorés. Vous pouvez insérer un séparateur entre les accolades, appliqué seulement si la variable n’est pas vide (ex. `{ - {nom}}`).

Paramètres des commandes

- choisir d'**importer et exporter les clients** et leurs informations nominatives vers le backoffice Polaris ;
- renseigner les **références Polaris (code-barre)** désignant les **frais de port** et les **frais d'emballage** dans les ventes exportées, via les options **référence pour les frais de port** et **référence pour l'emballage** (nous conseillons d'utiliser des produits de type service) ;
- **mapping des modes de règlement** : en face de chaque mode de règlement PrestaShop, renseignez le code de règlement Polaris à utiliser lors de l'exportation. Si aucun mode n'est précisé, le catch-all (*) est utilisé ; si le catch-all n'est pas non plus défini, la commande est ignorée.

À des fins de sécurité, dans les options d'exportation des commandes, vous devez définir :

- **l'âge maximal d'une commande** pour qu'elle (ou ses modifications) puisse être exportée ;
- **l'ID minimal qu'elle doit avoir**, pour ignorer en toute sécurité d'anciennes commandes (toute commande d'ID inférieur ne sera ni considérée ni exportée automatiquement). *Nous recommandons d'ajuster ce paramètre au moins une fois par an pour vous prémunir d'une exportation massive due à une erreur de manipulation ou à une édition de masse de votre base.*

Si vous souhaitez exporter manuellement les commandes sans automatisation, renseignez `-1` dans ce champ et utilisez le bouton `Exporter vers Polaris` depuis la page de chaque commande.

Si l'option **Exporter les annulations, modifications, remboursements et retours** est désactivée, seules les commandes validées sont envoyées. En l'activant, le module enverra aussi à Polaris toutes les mises à jour d'une commande après son premier envoi.

Mappages

Pour importer vos produits, certaines données ne peuvent pas être créées à la volée et doivent être **mappées** (« transposées »). Sans cette transposition, il serait impossible d'importer vos articles. Le mappage vous permet par exemple d'assigner la bonne marque et la bonne catégorie à chaque produit, en définissant une matrice qui, pour chaque donnée Polaris, fait correspondre une donnée PrestaShop.

Données mappables (liste non exhaustive) :

- les **grilles de tailles** → groupes d'attributs PrestaShop ;
- les **marques** → marques PrestaShop préexistantes ;

- les **catégories & RFS** → une ou plusieurs catégories PrestaShop préexistantes.

Tous les mappages fonctionnent de la même façon :

- si le mappage est défini pour une donnée précise, il est utilisé ;
- sinon, le mappage `wildcard (*)` est utilisé (souvent appelé *Tou(te)s les autres*) ;
- une **valeur spéciale** permet d'**ignorer** les produits utilisant cette donnée à l'importation.

Vous définissez vos mappages dans la page `Mapping Polaris` de la section `VENDRE \ Catalogue` du backoffice PrestaShop, également accessible par le bouton `Mappings` de la page de configuration du module.

> *À de rares exceptions près, vous ne pouvez pas créer manuellement des mappages, seulement les définir : leur création intervient automatiquement lors de la synchronisation. Tant que vous n'avez pas réalisé au moins une synchronisation, les grilles de mappage sont donc probablement vides.*
>
> *Si le module ne détecte pas toutes les catégories dynamiques ou certains paramètres de votre backoffice, cliquez sur le bouton « Télécharger » du bandeau supérieur : une connexion sera établie pour récupérer vos paramètres.*

Poids par défaut

Sur le mappage des catégories, vous pouvez spécifier un **poids par défaut** : lors de l'importation d'un produit appartenant à cette catégorie, si le backoffice ne précise pas de poids, le module lui affecte cette valeur. Si le produit appartient à plusieurs catégories ayant des poids par défaut différents, le module retient le poids le plus élevé.

À savoir : il faudra attendre (ou déclencher) une synchronisation pour appliquer vos modifications aux produits — cette synchronisation n'a pas besoin d'être totale.

Mappages périmés et nettoyage

Des clés de mappage peuvent ne plus être utilisées par aucun produit et parasiter l'interface. Un **utilitaire de nettoyage** retire de la base tous les mappages non définis et inutilisés. Il est automatiquement démarré lorsque vous téléchargez la configuration du backoffice via le bouton dédié de la barre d'outils.

Catégorie par défaut

L’option `Catégories mères des catégories par défaut` guide le module dans le choix de la catégorie par défaut. Le module recalcule la catégorie par défaut d’un produit lorsque :

- le produit n’en a pas encore ;
- celle positionnée n’appartient plus aux catégories du produit ;
- cette option est renseignée et la catégorie a changé ;
- cette option est renseignée et la catégorie par défaut n’a jamais été calculée.

Pour effectuer ce choix, le module parcourt toutes les catégories PrestaShop du produit et privilégie celles qui sont enfants (directes ou indirectes) des catégories mères sélectionnées ici. Parmi ces candidates, il retient la plus profonde (la plus spécifique). Si aucune catégorie du produit n’est enfant d’une catégorie mère sélectionnée, il retient simplement la plus profonde de toutes.

Exemple : si vous sélectionnez « Hommes » comme catégorie mère et qu’un produit appartient à la fois à « Hommes > Vestes > Parkas » et à « Nouveautés », le module affecte « Parkas » comme catégorie par défaut (enfant de « Hommes » et plus profonde). Si l’option n’est pas renseignée, le module n’intervient pas sur la catégorie par défaut.

Options avancées

Le menu **Options avancées** regroupe plusieurs fonctionnalités non indispensables à un fonctionnement nominal, mais qui permettent d’aller plus loin dans l’intégration de votre catalogue Polaris ou de piloter la reprise d’un site existant. Vous y accédez par le bouton **Options avancées** du bandeau supérieur ou par l’onglet du même nom.

Mapping de produits

Cette section permet de **lier manuellement des produits existant déjà dans votre boutique PrestaShop** avec les fiches correspondantes sur Polaris. Sans cette opération, le module crée systématiquement de nouveaux produits à la première synchronisation : si vous aviez déjà un catalogue, le mapping de produits évite la duplication.

Le principe est de faire correspondre, pour chaque déclinaison PrestaShop, le **code-barre Polaris** (utilisé comme référence fournisseur) avec la déclinaison existante. Une fois le lien fait, le module reprend la main sur la fiche lors des synchronisations suivantes.

- Un avertissement est affiché à l’entrée de la page : modifier les mappages produits peut perturber la synchronisation si vous ne

savez pas ce que vous faites. Vous devez confirmer pour accéder à la grille (confirmation valable pour la journée).
- La grille présente une ligne par déclinaison PrestaShop : ID produit, désignation, ID de la déclinaison, nom de la variante, et un champ libre pour saisir l'**EAN Polaris** correspondant.
- Des filtres sont disponibles en haut de chaque colonne. Le filtre EAN accepte la valeur spéciale `vide` pour n'afficher que les déclinaisons sans EAN.
- La colonne **Lié** indique que la déclinaison est désormais administrée par le module (l'EAN saisi correspond à une référence connue dans Polaris).
- La colonne **Verrouillé** fige la fiche produit : le module ne touche plus à son contenu (titre, description, photos…) et ne synchronise que les prix et les stocks.

La saisie d'un EAN s'enregistre automatiquement à la perte de focus ou à l'appui sur `Entrée` : le champ vire au vert si l'enregistrement a réussi, au rouge en cas d'erreur.

> *Cette page est surtout utile lors d'une migration ou d'une reprise de catalogue. Pour les boutiques démarrant à zéro, elle n'a généralement aucune utilité.*

Redirections

Cette section permet de définir des **règles de redirection 302** appliquées lorsqu'une page n'est pas trouvée (erreur 404) — utile lors d'une refonte de site ou d'un changement de structure d'URL. L'écran d'édition se compose de deux blocs.

1. Création automatique de redirection produits. Vous saisissez un masque utilisé à chaque création ou modification de produit par le module. Le placeholder `%%ref%%` est remplacé par la référence du produit. Par exemple :

```
/product/(.*)-%%ref%%
```

…crée automatiquement, pour chaque produit, une règle depuis l'ancienne URL `/product/(.*)-<référence>` vers la nouvelle URL PrestaShop. C'est la façon la plus simple de basculer un site existant sans perdre votre référencement.

2. Édition manuelle du fichier de règles. Vous éditez directement `var/redirects.list`. Le format des règles et les directives spéciales (`category(<id>)`, `product(<id>)`, groupes nommés `{{slug}}`…) sont décrits à la section Redirections URL de la partie « Données manipulées ». Les règles sont testées dans l'ordre : la **première qui correspond** déclenche la redirection. Le compteur en haut du

formulaire indique combien de règles sont actives. Les expressions régulières invalides sont signalées à la sauvegarde et ignorées à l'exécution.

Méta-catégories

Les **méta-catégories** sont des catégories virtuelles définies entièrement dans le module, comme la **combinaison logique de plusieurs catégories Polaris existantes** (voir aussi la partie « Données manipulées »). Un exemple typique pour Polaris est `Jeans longues tailles`, qui combine le RFS et la longueur de jambe : une fois mappée, vous obtenez une catégorie PrestaShop regroupant automatiquement les produits correspondant à la combinaison souhaitée. L'écran de gestion se compose de deux blocs :

1. Liste des méta-catégories définies. Pour chacune, le tableau affiche son nom et la liste des catégories Polaris composantes (séparées par +). Un bouton **supprimer** est disponible par ligne : la suppression est définitive et efface aussi tous les mappages associés. Une resynchronisation sera nécessaire pour mettre à jour les produits concernés.

2. Formulaire de création. Vous saisissez un **nom** libre (qui apparaîtra comme libellé dans la grille de mapping des catégories) et les **catégories composantes** à cocher (au moins deux ; seules les catégories Polaris non virtuelles sont proposées — on ne combine pas une méta-catégorie avec une autre).

> *La composition d'une méta-catégorie ne peut pas être modifiée après création (cela invaliderait tous les mappages déjà positionnés) : pour la modifier, supprimez-la, recréez-la, puis remappez-la.*
>
> *Les méta-catégories n'apparaissent dans la grille de mapping qu'après la synchronisation suivant leur création, le temps que le module rencontre les combinaisons réellement présentes. Pour un résultat immédiat, déclenchez une resynchronisation totale.*

Synchronisation du catalogue produits

La synchronisation du catalogue est réalisée à intervalles réguliers par la tâche automatisée, ou appelée manuellement (cf. Installation), à condition d'avoir paramétré le mode « Synchronisation du catalogue & du stock ». Pour les produits déjà existants, un bouton de synchronisation est ajouté dans la barre d'outils des fiches produits compatibles (à côté du bouton d'aide) : il permet de resynchroniser ce produit pour refléter des modifications récentes ou vérifier que la synchronisation fonctionne.

Audit et journalisation

La synchronisation produit un **fichier d'audit par jour**, stocké dans le répertoire `var/logs` de PrestaShop. Il contient la journalisation des actions du module et est consultable depuis les options du module.

Informations synchronisées

Vous sélectionnez les informations à synchroniser dans l'onglet « Catalogue » de la page de configuration (référence, nom du produit, marques, etc.). Une donnée synchronisée est systématiquement écrasée par sa valeur du backoffice Polaris lorsqu'elle existe et que le module synchronise le produit ; sinon, elle est simplement reportée à la création de la fiche.

> *Par sécurité, le module modifie la page d'édition des produits de PrestaShop pour vous empêcher de modifier une donnée maintenue à jour depuis le backoffice (elle serait susceptible d'être remplacée, vous faisant perdre vos modifications). Les champs gérés par le module sont **grisés et de couleur orange**, uniquement sur les produits créés et maintenus par le module.*

Disponibilité des pièces dans les boutiques

Le module embarque 3 options pour afficher les stocks par magasin sur vos pages produits, afin que vos internautes voient d'un coup d'œil si le produit consulté est disponible dans les magasins proches de chez eux. Vous devez préalablement avoir mappé vos magasins Polaris avec vos magasins PrestaShop (mapping des magasins, dans les options).

1. **Afficher la disponibilité sur les thèmes compatibles** (le plus simple) : activez l'option « Afficher la disponibilité par magasin » dans `Paramètres du catalogue`. Vous réglez le seuil de quantités déterminant si la pièce est disponible (vert) ou en « dernières pièces » (orange). Le hook utilisé pour injecter la disponibilité est `displayProductAdditionalInfo` ; si l'information n'apparaît pas au bon endroit, ajustez l'ordre via `Personnaliser / positions` de PrestaShop.
2. **Afficher la disponibilité par hook dans les templates** : appelez le hook `hookDisplayStoreStocks` directement dans le template de la page produit. Il calcule et affiche un tableau habillable par CSS, avec une puce de couleur (vert, orange, rouge) selon la disponibilité de la taille sélectionnée dans chaque magasin.
3. **Calculer la disponibilité par hook dans les templates** : appelez le hook `hookBuildStoreStocks` directement dans le template. Il charge les informations de disponibilité ; vous calculez ensuite le rendu de votre choix à partir de la donnée `stores_stocks` peuplée

pendant l’appel.

6. FAQ — Comment faire ?

Avant toute manipulation, lorsque vous rencontrez un problème, **pensez à consulter le journal d’audit du module**. Il contient toutes les décisions et actions prises par le système pour chaque élément à importer ou exporter ; la raison d’un problème de synchronisation s’y trouve généralement, en clair. Le journal est accessible depuis le menu « Audit Polaris » de l’administration, ou via l’icône journal de l’encart du module sur le tableau de bord.

J’ai fait ma première synchronisation et aucun produit n’a été créé

- Vérifiez que vous êtes bien en mode **complet** : en mode basique, le module ne crée aucune fiche produit.
- Vérifiez que vos mappings de **marque** et de **grille de tailles** sont correctement définis. Par défaut, le module est réglé pour ignorer les produits de toutes les marques et grilles de tailles, soit tous les produits.
- Si tout est correct, consultez le **journal d’audit** : la raison y est généralement indiquée lisiblement.

J’ai corrigé mes mappings mais mes produits ne s’importent / ne se déplacent toujours pas

Certains mappings doivent être resynchronisés depuis le backoffice Polaris. Or la synchronisation est incrémentielle : elle ne prend en compte que les produits créés ou modifiés depuis la dernière synchro. La solution est de **tout resynchroniser** pour forcer le module à tout reprendre depuis le départ.

Comment tout resynchroniser ?

Par défaut, le module ne resynchronise que les données ayant changé depuis la dernière synchronisation. Pour tout resynchroniser (catalogue, stocks et prix), cliquez sur le bouton **Resynchroniser** de la page de configuration. La resynchronisation totale sera effectuée lors de la prochaine synchronisation programmée.

La synchronisation ne semble pas fonctionner

Consultez d’abord le **journal de PrestaShop** : s’il contient des erreurs relatives à la synchronisation, la raison y sera indiquée. Résolvez-les en priorité. Sinon, consultez le **journal d’audit** du

module. Les causes les plus fréquentes sont :

- la tâche cron n’est pas appelée (le journal d’audit est alors souvent vide) ou retourne une erreur ;
- un mode de règlement, des frais de port ou d’emballage non définis dans la configuration ;
- une marque, une catégorie ou une grille de tailles non mappée ou interdite ;
- le mode de synchronisation mal défini ou mis sur pause.

Empêcher une fiche produit d’être synchronisée

Lorsque le module synchronise les fiches produits, une **icône de verrouillage** permet de l’empêcher de modifier une fiche. Si l’icône est un cadenas rouge, le module ignore cette fiche et n’en synchronise que les prix et les stocks. Cliquez sur l’icône pour verrouiller/déverrouiller. Le verrouillage libère aussi tous les champs en édition.

Resynchroniser une fiche produit

Lorsque le module synchronise les fiches produits, une **icône de synchronisation** apparaît sur la fiche : elle déclenche une resynchronisation de ce produit. Cette synchronisation ne concerne que ses propriétés, pas son stock ni ses promotions (pour ceux-ci, procédez à une synchronisation complète ou attendez son déclenchement).

Exporter ou réexporter une commande manuellement

Les commandes ne se réexportent jamais tant que le système ne détecte pas d’erreur. Trois cas peuvent se présenter :

Cas n°1 — Erreur d’exportation. La commande s’est vue affubler de l’état `Erreur d'exportation Polaris`. Consultez le journal d’audit pour connaître la raison, corrigez le problème, puis affectez un autre état à la commande (elle sera reprogrammée pour la prochaine synchronisation) ou appuyez sur le bouton `Réexporter vers Polaris` pour une tentative immédiate.

Cas n°2 — Perte au niveau du backoffice Polaris. Vous avez perdu la commande sur Polaris (retour en arrière, désastre, incident). Ce cas n’est pas solutionnable avec le module : vous devez ressaisir manuellement les commandes perdues.

Cas n°3 — La commande ne s’exporte pas automatiquement. La commande, pourtant validée et payée, n’a ni l’état `Exportée vers Polaris`, ni `Erreur d'exportation Polaris`. Vérifiez le journal d’audit ; en général il s’agit de l’une de ces causes :

- la commande est trop âgée (plus de 90 jours) ;
- un des modes de règlement n’est pas mappé ;
- il n’y a pas d’article à livrer (tout a été remboursé) ;
- le statut n’est pas un statut valide (payée, facturée et livrable) ;
- le montant payé ne correspond pas au total de la commande.

Gérer les retours & remboursements

Quand vous remboursez partiellement une commande (par exemple dans le cadre d’un retour), passez par le bouton `Remboursement` de PrestaShop et cochez `générer un avoir`, même si vous prévoyez de recréditer la carte bancaire. Sous PrestaShop, un avoir est un document qui atteste fiscalement de l’annulation d’une partie ou de la totalité de la facture, et non un bon utilisable pour une autre commande. Vous pouvez ensuite recréditer la carte ou générer un bon de réduction.

Si un avoir a été généré, le remboursement est exporté vers Polaris sous forme de remboursement. Si vous marquez la commande comme annulée, remboursée (ou tout autre état non valide), ou si vous ne générez pas d’avoir, le module annule l’intégralité du précédent débit exporté vers Polaris pour le resaisir dans sa version modifiée (ou ne pas le resaisir s’il s’agit d’une annulation complète). Si un bon de réduction a été émis sous PrestaShop, il est importé comme une remise négative sous Polaris.

Pourquoi un de mes comptes client PrestaShop n’est-il pas exporté vers Polaris ?

Le module n’envoie vers Polaris que les clients ayant effectué au moins un achat, qui ne sont pas des clients marketplace (Amazon, Spartoo…), ou bien appartenant au groupe `Clients Polaris`. Si votre client ne remonte pas, c’est probablement que vous n’êtes dans aucun de ces cas. Puisqu’il n’est pas exporté, ses modifications ne remontent pas non plus.

Reprise d’un site existant

Pour mettre en place la synchronisation, vous devez établir la correspondance entre vos articles PrestaShop et vos références backoffice. **Procédez impérativement par étapes :**

1. configurez d’abord le module en **mode basique** pour prévenir toute suppression de données involontaire ;
2. vérifiez votre **mapping de produits** ;
3. une fois le mapping réalisé, vous pouvez éventuellement passer en **mode complet** pour gérer les fiches produits.

Pour mapper vos produits, dans le *mapping des produits* de la page de configuration, faites correspondre à chaque ligne de produit PrestaShop la référence backoffice **EAN** de l'article. La case **LOCK**, cochée par défaut, verrouille la fiche en l'état pour empêcher le module de remplacer ses données par celles du backoffice (en mode complet, seuls les prix, promos et stocks sont alors entretenus). Décochée, la fiche devient administrée par le module et vous risquez de perdre tout ou partie des informations saisies manuellement. **Ne la décochez qu'en sachant exactement ce que vous faites, et après une sauvegarde.**

Site de préproduction

Vous pouvez raccorder un site de préproduction au backoffice Polaris avec la même clé, pour tester des modifications sans risque. Pour éviter une seconde clé d'accès, initialisez la base de préproduction à partir d'une copie de la base de production : elle contient l'empreinte de la clé d'accès et héritera des mêmes données. **Dans tous les cas, ajustez le paramètre `ID minimal d'exportation` des options d'exportation de commandes à `-1`** pour éviter toute exportation de commandes de test sur le backoffice Polaris.

7. FAQ — Résolution des problèmes

Première connexion — « Failed to connect »

Si vous obtenez cette erreur lors de la tentative d'association (immédiatement ou après quelques secondes), vérifiez que votre hébergeur autorise la **sortie applicative sur le port de contact** (présent dans le message d'erreur, généralement **3443**). C'est notamment le cas chez OS2Switch, où il faut déverrouiller ce port dans le firewall applicatif.

Première connexion — « Timeout »

Si, après avoir soumis les premières informations, vous obtenez une erreur « Timeout » après un long moment, vérifiez que le **backoffice est en ligne et fonctionnel**. Dans 90 % des cas, le poste qui héberge Polaris est éteint ou coupé d'Internet au moment de la requête.

Exportation des ventes — « Débit déséquilibré »

Il est possible qu'une commande ne soit pas exportée vers Polaris avec un message ressemblant à :

```
Erreur API Polaris : impossible de créer la vente : 0x0010000A3 :
```

```
Objet métier : Une des données
n'est pas correcte : MetierDebit : 2 Débit déséquilibré dans l'objet
piu_Ventes.MetierDebit
```

En règle générale, cela se produit lorsque la commande comporte un **bon (voucher)** qui existe sous Polaris mais dont le montant est différent (souvent inférieur) de celui enregistré sous PrestaShop et utilisé dans la commande (édition manuelle du bon sur l'une des plateformes avant utilisation, bug…).

Le programme est conçu pour ne pas importer cette vente, car l'événement n'aurait jamais dû se produire et son acceptation poserait des problèmes fiscaux. La seule solution consiste à **ressaisir manuellement la vente sous Polaris**.

Importation — la synchronisation est terminée, aucune fiche produit n'a été créée

Fréquent à la mise en place d'un site. Si aucune fiche ne se crée lors de votre première importation :

- vérifiez que le pont est en **mode complet** (en mode basique, il ne crée pas de fiche : il met seulement à jour stock et prix) ;
- vérifiez que les **grilles de tailles** et les **marques** des produits à importer ont bien été mappées (sinon, le comportement par défaut est de les ignorer) ;
- dans tous les cas, consultez les **journaux d'audit** pour connaître la raison d'une non-importation.

Importation — un produit devrait s'importer mais ne le fait pas

Cherchez d'abord la raison dans le journal d'audit. Si votre produit n'y figure pas, c'est que votre backoffice ne l'exporte pas : contactez votre support habituel. Sinon, la raison y sera indiquée. En général, pour qu'un produit s'importe, il faut :

- que le module soit en **mode complet** ;
- que le produit ait du **stock** dans les magasins constitutifs du stock en ligne (sur configuration, les produits sans stock en ligne peuvent être ignorés) ;
- que sa **marque** et sa **grille de tailles** soient mappées ;
- qu'aucune de ses **catégories** ne soit filtrée dans le mappage.

8. Documentation technique (webmestres)

Cette partie s'adresse aux webmestres souhaitant comprendre l'organisation interne du module installé, ses points d'entrée et les tables qu'il crée en base de données. Elle n'est pas nécessaire à l'utilisation courante du module.

Organisation des fichiers livrés

Le module est installé dans `modules/polarisprestaconnector/` de votre PrestaShop. Son organisation principale :

Emplacement	Rôle
`polarisprestaconnector.php`	Classe principale du module (installation, hooks, configuration).
`controllers/front/cron.php`	Point d'entrée HTTP de la synchronisation (vérifie le jeton `guard` puis délègue au contrôleur de synchronisation).
`src/`	Logique métier : orchestrateur de synchronisation (`Syncer`) et modules par domaine (`SyncerProducts`, `SyncerStocks`, `SyncerOrders`, `SyncerCustomers`, `SyncerDiscounts`, `SyncerVouchers`, `SyncerPrepaidCards`), opérations PrestaShop réutilisables (`*Manager`), objets de transfert (`Models/`), contrôleurs d'administration (`Controller/`), commandes CLI (`Command/`) et formulaires (`Forms/`).
`config/`	Déclaration des routes (`routes.yml`) et des services Symfony (`services.yml`).
`views/`	Templates Twig de l'administration, feuilles de style et scripts.
`db/`	Scripts de migration de schéma (`upgrade-*.php`).
`docs/`	La présente documentation.
`var/redirects.list`	Fichier de règles de redirection d'URL (situé à la racine du site).
`var/logs/`	Journaux d'audit quotidiens de la synchronisation (à la racine du site).

La synchronisation est **incrémentale par défaut** : les positions de reprise sont mémorisées en configuration et réinitialisées lors d'une resynchronisation totale (`--full` ou `&full=1`).

Points d'entrée

- **Cron HTTP** (à appeler le plus souvent possible) : `https://votresite.com/modules/polarisprestaconnector/cron?guard=XXXX`. Le jeton `guard` protège l'accès (toute requête sans le bon jeton est refusée). Paramètres optionnels : `&full=1` (resynchronisation totale) et `&rebuild-redirects=1` (reconstruction des redirections). Consultez la page de configuration pour l'URL et le jeton exacts.
- **CLI (console Symfony)** : `php bin/console polarisprestaconnector:sync [--full] [--quiet]`, à lancer depuis la racine de PrestaShop. Une commande utilitaire de diagnostic est également disponible (`polarisprestaconnector:tools`).
- **SSO de maintenance (OpenID Connect)** : une route dédiée permet aux équipes support de se connecter via le bouton **Connexion SSO VEGA** de la page de login (voir « Accès maintenance Polaris »).

Hooks d'intégration au thème

Pour enrichir vos pages produits, le module expose notamment les hooks suivants, exploitables dans vos templates :

- `displayMultiProductSelector` — sélecteur de produits connexes (variantes scindées) ;
- `displayProductAdditionalInfo` — disponibilité par magasin sur les thèmes compatibles ;
- `displayStoreStocks` (`hookDisplayStoreStocks`) — rendu du tableau de disponibilité par magasin ;
- `buildStoreStocks` (`hookBuildStoreStocks`) — chargement des données de disponibilité (`stores_stocks`) pour un rendu personnalisé ;
- `displayProductPriceBlock` — intégration au bloc de prix produit.

Tables de base de données

Le module crée ses propres tables, toutes préfixées par `ps_pc_pol_` (le préfixe `ps_` correspond au préfixe de votre installation PrestaShop) :

Table	Rôle
`ps_pc_pol_configuration`	Configuration du module sous forme de paires clé/valeur JSON (options, jeton `guard`, positions de reprise incrémentale, empreinte de la clé d'accès au backoffice).
`ps_pc_pol_image`	Correspondance entre les photos du backoffice et les images PrestaShop (`id_product` / `id_image`).
	Stocks ventilés par déclinaison et par

`ps_pc_pol_stocks`	magasin, source de l'agrégation vers `ps_stock_available` et de l'affichage de la disponibilité par magasin.
`ps_pc_pol_promotion`	Copie des promotions gérées par le module (code, libellé, dates, catégorie optionnelle).
`ps_pc_pol_promotion_detail`	Détail des promotions par produit/déclinaison (lien vers les prix spécifiques PrestaShop).
`ps_pc_pol_product`	Données étendues issues du backoffice pour chaque produit (catégorisation backoffice, propriétés étendues, catégorie par défaut calculée).
`ps_pc_pol_orders_updates`	Suivi de l'export des commandes (révisions, empreinte exportée, état d'envoi) pour gérer les modifications et ré-exports.

Les **bons**, **cartes prépayées** et **clients** ne disposent pas de table dédiée : ils sont matérialisés par des objets PrestaShop natifs (règles panier `cart_rule` pour les bons et cartes, clients standard pour les clients).

Le schéma n'est pas créé à l'installation : il est entièrement géré par les scripts de migration `db/upgrade-*.php`, exécutés dans l'ordre des versions (la création initiale des tables se trouve dans `db/upgrade-0.3.php`). La mise à jour du schéma est **automatique** : elle est rejouée lors d'un changement de version du module (hook du tableau de bord et démarrage de synchronisation).