

# PolarisPrestaConnector - Documentation technique

## Manuel technique

La section suivante détaille les choix et le fonctionnement techniques du module PolarisPrestaConnector, ainsi que les données prises en compte.

Vous pouvez télécharger le module à l'URL suivante : <https://ftp.vega-net.net/dists/polarisprestconnector/stable/latest/polarisprestconnector.zip>

## Pré-requis techniques

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

- **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
- Environnement d'hébergement capable de communiquer vers l'extérieur sur le port de Polaris (souvent 3443). Par exemple, les offres mutualisées de os2switch ne sont pas compatibles.

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

## Fonctionnalités du module

Sous réserve du choix de mode de fonctionnement et paramétrages corrects du module, celui-ci 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 affecté aux différents critères. ;*

- la classification automatique des produits selon des règles de correspondances (mappage) ;
- la possibilité de filtrer les références importées depuis les règles de correspondances (si correspondance alors ne pas importer) ;
- le support du ramasse-mièes (catch-all) dans les règles de correspondances (si aucune correspondance alors...) ;
- le support de méta-catégories (mappage en fonction d'une combinaison logique de plusieurs catégories backoffice existantes et/ou de la marque) ;
- 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 commandés ;
- l'importation des clients avec adresse mail ;
- l'importation des bons d'avoir, d'acompte, et cadeaux des clients importés ;
- mise à disposition d'un relevé de la collecte de TVA facturée dans les commandes, par pays de livraison et par taux ;
- destruction automatique des photos des produits désactivés et redirigés n'ayant plus d'activité depuis plus d'un mois ;
- prise en charge d'une fonctionnalité de redirection d'URL pour les reprises de site ;

Note : dans la description ci-dessus, *importation* désigne le transfert de données du backoffice Polaris vers Prestashop. Par conséquent *Exportation* désigne le processus inverse. Consultez la suite du document pour savoir quelles données sont optionnelles et quelles données sont obligatoirement gérées.

## Modes de fonctionnement

Le module possède deux modes :

1. le **mode basique** : dans 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** : dans le mode complet, en plus d'effectuer toutes les actions du mode basique, le module crée et tient les fiches produits à jour 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 lors d'une synchronisation la donnée existe déjà sous Prestashop, celle-ci 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. Ces données sont toujours écrasées par le module si elles existent sous Polaris.\*

## Fonctionnement général

Le module répond au fonctionnement suivant :

- à intervalle régulier, il est nécessaire d'appeler la tâche cron présente à l'adresse  
`https://votresite.com/module/polarisprestaconnector/cron?guard=XXXX` ou bien en ligne de commande (*php bin/console prestaconnector:sync*) ; consultez la page de configuration du module pour connaître la valeur exacte du jeton d'accès (*guard*) ;
- c'est cette tâche qui va initier la communication entre le site Prestashop et votre backoffice et réaliser les opérations d'importation et d'exportation des données en se connectant aux API de votre backoffice ;
- il est nécessaire d'appeler cette tâche régulièrement ;
- la tâche ne resynchronise pas la totalité de vos données systématiquement mais, dans le domaine du possible et sauf ordre contraire ponctuel, cherche à travailler en incrémentiel - *ie.* de ne s'occuper que des données qui ont changées ;

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 de l'appel d'une tâche de synchronisation, il se connecte à nos serveurs pour vérifier si une nouvelle version est disponible. Dans un tel cas, il télécharge l'archive ZIP contenant la mise à jour et la déploie dans le répertoire `/updates/lastest`.

Il informe ensuite Prestashop de la disponibilité de la mise à jour. Les utilisateurs ayant le droit 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 du prochain appel 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, 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 qui contient une pré-version à des fins de tests, vous pouvez créer un fichier TESTING à la racine du site Prestashop. Les recherches de mises à jour ultérieur se feront sur la branche de développement qui contient toutes les dernières nouveautés.

**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, il faut préalablement avoir mappé vos magasins Polaris avec vos magasins listés sous Prestashop par l'intermédiaire du 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. Le module garde la trace du lien de parenté des produits créés de la sorte et incorpore un *hook* qui permet de générer un sélecteur pour naviguer entre ces produits.

Pour utiliser ce sélecteur, appeler le *hook* `{hook h='displayMultiProductSelector' product=$product}` dans la *template* (tpl) de la page produit à l'endroit où vous souhaitez générer le code HTML.

Vous pouvez maquiller le rendu du sélecteur par CSS et en modifier le rendu, comme par exemple cacher la variante affichée actuellement, modifier les couleurs, etc...

*Notez que 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 actuellement visualisé.*

## Données synchronisées

### Produits


Fonctionnement général :

- sur Polaris, certaines déclinaisons peuvent être discriminantes (par exemple, la couleur ou la matière), d'autres peuvent être des tailles (longueurs de jambe, bonnets, ...) ;
- chaque couple {produit - déclinaisons discriminantes} de Polaris qui est 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 lorsque celle-ci 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 la donnée n'est pas présente dans la colonne Prestashop, alors c'est que la donnée n'est pas synchronisée.*





Donnée Prestashop	Notes
Référence produit ⓘ	La référence utilisée est la 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 discriminante, est utilisée ici.
Nom	Construisable selon modèle à partir des informations
URL simplifiée	Déduit automatiquement du nom du produit lorsque 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 de vente de base est le prix le plus bas, toutes déclinaisons et tailles confondues.
Caractéristiques	Les caractéristiques sont définies par mappages de champs dynamiques dans Polaris ; voir la configuration pour plus de détails.
Catégories	Les catégories sont 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 qui inclut ce taux pour le magasin par défaut du site.

### Données synchronisée sur la déclinaison

*Quand la donnée est barrée dans la colonne Prestashop, alors c'est que la donnée n'est pas synchronisée avec cette version du module.*

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 concerné, à défaut, celle du modèle.
Association de photo	Le module rétablit les bonnes photos avec les bonnes déclinaisons automatiquement.

*Note : les déclinaisons de Prestashop correspondantes à des produits supprimés ou marqués comme non publiés sous Polaris sont automatiquement supprimées de Prestashop. Si un produit se retrouve sans déclinaisons de cette manière, alors il est désactivé si ce n'est pas déjà le cas.*

## Marques

Les marques ne sont pas créées automatiquement par le module, mais mappée.

Chaque fois que le module rencontre une nouvelle marque, il crée une ligne de mappage qu'il faudra configurer dans les options du module ou paramétrer un catch-all pour affecter la marque.

Une resynchronisation complète est nécessaire après modification du mappage.

## Caractéristiques, Tailles & Attributs

Les caractéristiques, tailles et autres attributs ne sont pas synchronisés au même titre que le reste des données.

Lorsque le module synchronise un produit référençant l'une de ses données et que la caractéristique ou le groupe d'attributs a été mappé, il regarde si une valeur équivalente existe. Si c'est le cas, il l'utilise ; autrement, il crée la valeur de caractéristique ou l'attribut et l'utilise.

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

## Catégories

Les catégories sont systématiquement mappées. A chaque nouvelle catégorie de produit sur le backoffice Polaris, une ligne de mapping est créée par le module. Quand celle-ci est renseignée, tous les produits appartenant à cette catégorie Polaris sont automatiquement assignés aux catégories Prestashop renseignées pour cette ligne.

Le module essaye au mieux possible de ne pas toucher aux associations que vous auriez fait manuellement, mais il est possible qu'il modifie les catégories d'un produit pensant défaire une association caduque si vous avez associé manuellement votre produit à l'une des catégories gérées par le mappage.

Aussi, une resynchronisation complète est 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 bien n'est plus associée au produit ;

Le module choisit systématiquement la catégorie la plus profonde (celle qui a le plus de parents). A 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 le backoffice : elles sont construites par le module comme la combinaison logique de plusieurs catégories dynamiques existantes. L'objectif est de permettre 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 pour les méta-catégories sont créées à la volée, au fur et à mesure que le moteur de synchronisation rencontre de nouvelles clés composites. Le comportement est identique à celui des catégories dynamiques ordinaires : une ligne de mappage vide apparaît dans la grille de mappage, en attente d'être renseignée par l'utilisateur. Tant qu'une clé composite n'est pas mappée (ou 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 de Prestashop `ps_stock_available` prévu à cet effet en passant par les objets internes pour déclencher tous les *hooks* du système afin que les modules se basant sur le changement des stocks puissent fonctionner correctement.

Vous pouvez définir quels magasins participent à l'aggrégation pour obtenir les stocks mis à disposition pour la vente en ligne, consultez notre section documentation pour connaître la procédure.

Vous pouvez vous appuyer sur la table `ps_pc_pol_stocks` pour calculer et afficher une disponibilité d'article au magasin dans votre propre module si vous le désirez.



Il est à noter que sur option, est déduit de ces stocks agrégés une quantité minimale de stock de sécurité pour chaque taille, mais cette quantité n'est jamais déduite des informations présentent dans la table `ps_pc_pol_stocks`.

## Promotions

Les promotions sont synchronisées sous la forme de prix spécifique sous Prestashop. Le module fait au mieux pour en créer le moins possible et de les regrouper lorsque la promotion concerne toutes les déclinaisons d'un produit.

De même le module ne crée que des promotions en % et ne touche pas aux promotions manuelles que le marchand pourrait créer lui-même. Pour cela, le module gère une copie de ses promotions dans la table `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 faisant partie de cette promotion soit automatiquement associé à cette catégorie Prestashop le temps de la promotion ; option particulièrement utile pour nommer ou présenter cette 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 mail valide.
- Les clients primo importés sous Prestashop depuis Polaris sont marqués faisant parti du groupe de clients "Clients Polaris".
- Les modifications clients sont exportées vers Polaris si ce dernier a passé au moins une commande sur le site Prestashop ou s'il fait parti du groupe de clients "Clients Polaris" et si ce n'est pas un compte invité et qu'il ne s'agit pas d'un compte écran créé automatiquement par une plateforme de 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	Raison sociale, en majuscules, si applicable

Nom	Nom en majuscule
Prénom	Prénom capitalisé
Civilité	Importé/Exporté seulement si un titre de civilité identique est présent sous Prestashop, sinon ignoré.
Inscription à la newsletter	Selon l'adhérence du client à la liste de diffusion système "Mailing par mail".
Mot de passe	Non synchronisé mais créé aléatoirement lors de la création du client sous Prestashop.
Adresse	Le système remonte l'adresse de facturation utilisée dans la dernière commande comme adresse principale. Si le client n'a jamais commandé, aucune information ne remonte. <i>Attention</i> : c'est l'adresse utilisée lors de la commande qui remonte et non les modifications faites par l'internaute à cette adresse <i>après</i> 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 et la compréhension du système, les clients ne sont exportés depuis Prestashop vers le backoffice que lorsqu'ils subissent une modification sur Prestashop. Ils ne sont pas reexportés en totalité lorsque 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 resynchronisation provoque par contre la rétribution de tous les clients depuis le backoffice.

## Bons

Les bons sont importés depuis le backoffice 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	Le montant est exprimé TTC ou 0 si le bon est en %
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 si désactivé ou plus valable sous Polaris
Sécable	Non, les bons Polaris ne sont pas sécables
Client	Lorsque l'importation client a été activé, les bons sont normalement reliés aux client. Sinon, les bons reliés à un client qui n'existe pas 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.*

### Carte prépayée (carte cadeau)

Les cartes prépayées sont importées depuis le backoffice 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 et le montant de la carte est réapprovisionné par le module.

Donnée	Notes
Code de la carte	Code EAN généré par Polaris
Montant disponible	Le montant est exprimé TTC ou 0 si le bon est en %
Quantité disponible	Toujours 1 quand le bon est mise à jour ou 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 le backoffice Polaris lors de la tâche automatisée de synchronisation dès qu'elles correspondent à toutes les conditions suivantes : - le statut actuel de la commande est considéré comme payée (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 est inférieure à 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 avec les paiements ; - 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 règlements devant être mappés.

A l'issue de l'exportation, la commande se voit affublée d'un des deux statuts suivants, créés automatiquement par le module lorsqu'ils n'existent pas :

- Exportée vers Polaris : la commande a bien été exportée sous Polaris, sans erreur ;
- Erreur d'exportation vers Polaris : la commande n'a pas été exportée sous Polaris. L'erreur est présente dans le journal de Prestashop et dans l'audit de synchronisation ;

*Le système est prévu pour retenir l'envoi de la commande dès qu'il détecte une modification sur la commande et qu'elle obéit aux conditions préalablement indiquées ci-dessus.*

Voici les données exhaustives prises en charge par cet 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 ici qui est pris en compte et non la date de la commande, essentiellement pour des questions fiscales.
Type de la vente	Les commandes exportées sont typées comme "Vente Internet" sous Polaris
Magasin	C'est le magasin de référence Internet qui est spécifié ici comme magasin ayant réalisé la vente
Produits	Pour chaque produit, est exporté : <ul style="list-style-type: none"><li>- le code barre Polaris ; à défaut le Gencode s'il est présent</li><li>- la quantité</li><li>- le prix de vente TTC consenti</li><li>- le montant TTC de la remise, si existante. (les remises sont typées en Promotion)</li><li>- le taux de TVA pratiqué sur le site</li></ul>
Les règlements	La liste exhaustive des règlements, mappés

	selon la configuration
Remise panier	La remise panier est considérée comme un mode de règlement et à ce titre, suit le mappage. (par défaut RE).
Client	Si le client n'est pas un compte invité et si l'option d'exportation des clients vers Polaris a été activée, alors le client est liée à la vente.

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

### **Modification, annulation, retours & remboursement**

Le module ne traite les modifications, annulations, retours & remboursements qu'à la condition que l'option éponyme ait été activée dans sa configuration. Autrement, seule la primo version de la commande est exportée vers LCV.

*Note : en raison d'une très grande liberté d'édition des commandes sous Prestashop, il est souvent possible d'arriver à un résultat non importable dans LCV, notamment en cas de remboursement partiel ou de modifications successives de la commande. Dans ce cas, une erreur d'exportation se produit et il faudra 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 sur une commande, 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, alors le module n'exportera pas cette commande ;
- ensuite, s'il y a déjà eu une ou plusieurs modifications pour cette commande qui ont été exportées, le module annule ces modifications en les passant sur le backoffice en sens inverse ;
- puis, si la commande se trouve dans un état exportable ou si la commande n'est plus dans un état exportable, mais l'a été un jour et a déjà été synchronisée avec le backoffice, alors, le module exportera une vente qui reflète la différence entre la commande d'origine et la commande à l'heure de l'exportation ;

## **Redirections URL**

Le module PrestaConnector 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 URLs PrestaShop cibles. Ce mécanisme repose sur un fichier texte de règles (`redirects.list`) et s'articule en deux volets : la gestion manuelle via l'onglet **Redirections** du module, et la génération automatique lors des synchronisations produits.

## 1. Fichier de règles `redirects.list`

Le fichier est stocké dans le répertoire du site :

```
/var/redirects.list
```

S'il n'existe pas, le module le crée automatiquement avec un en-tête commenté décrivant le format.

### Format

Chaque ligne active représente une règle de redirection :

```
<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 section 3).
- Les lignes vides et les lignes 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}}
```

## 2. Saisie manuelle via l'onglet "Redirections"

L'onglet **Redirections** du back-office du module permet de saisir des règles manuellement. Chaque règle saisie est ajoutée en fin de fichier `redirects.list` via `appendRule()`.

## Comportement important

**Si une règle avec le même masque existe déjà dans le fichier, elle n'est pas modifiée.**

Pour modifier une destination existante, il faut **supprimer manuellement la ligne** dans `redirects.list`, puis ressaisir la règle depuis l'onglet.

Ce comportement est intentionnel : il protège les règles ajoutées manuellement ou automatiquement contre un écrasement involontaire.

## 3. Groupes capturés et directives spéciales

### Groupes nommés

Les groupes nommés dans le masque regex `((?P<nom>...))` peuvent être réutilisés dans la destination via la syntaxe `{{nom}}` :

```
/produit/(?P<ref>[A-Z0-9\-\ ]+)\.html | /fr/produit/{{ref}}
```

### 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()`) :

```
/fr/catalogue/new/femme/chaussures\.awp | category(51)
```

### 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()`) :

```
/fr/catalogue/new/homme/mocassins\.awp | product(102)
```

## 4. Génération automatique lors de la synchronisation

### Principe

À chaque synchronisation d'un **nouveau produit** depuis la source externe, le module peut générer automatiquement une ligne de redirection dans `redirects.list`, en appliquant un pattern configuré dans l'onglet **Redirections**.

Ce pattern utilise les mêmes mécanismes que les règles manuelles (groupes nommés, directives spéciales).

### ***Comportement***

- Si la règle correspondant au produit **n'existe pas encore** dans le fichier → elle est ajoutée automatiquement.
- Si la règle **existe déjà** → elle est conservée telle quelle, sans modification.

Ce comportement garantit que les corrections manuelles apportées à une règle existante ne sont jamais écrasées lors des synchros suivantes.

## **5. Reconstruire les redirections sur des produits déjà existants**

La génération automatique ne s'applique qu'aux **nouveaux produits** détectés lors d'une synchronisation. Pour des produits déjà synchronisés antérieurement, les redirections n'ont pas été générées.

Pour forcer la reconstruction des redirections sur l'ensemble du catalogue, lancer une **resynchronisation totale** en ajoutant le paramètre suivant à l'URL de la tâche :

`&rebuild-redirects=1`

Exemple d'URL complète :

`https://www.monsite.fr/modules/prestaconnector/cron.php?  
token=XXXX&full=1&rebuild-redirects=1`

⚠ Cette opération ajoute uniquement les règles manquantes. Les règles déjà présentes dans `redirects.list` ne sont pas modifiées.

## **6. Application des redirections**

Au moment où PrestaShop reçoit une requête, le module intercepte l'URL entrante et la confronte séquentiellement à chaque règle du fichier. La première règle dont le masque correspond déclenche la redirection (HTTP 301) vers la destination calculée.

Si aucune règle ne correspond, la requête suit son cheminement normal dans PrestaShop.

## **Erreurs fréquentes**

### **Première connexion**

**Failed to connect**



Si vous avez cette erreur lors de la tentative d'association immédiatement ou après quelques secondes d'attente, en général, vérifiez que votre hébergeur autorise la sortie applicative sur le port de contact (présent dans le message d'erreur, mais généralement 3443).

C'est notamment le cas chez OS2Switch où il faut déverrouiller ce port dans le firewall applicatif.

### **Timeout**

Lors de la tentative d'association, si après avoir soumis les premières informations, vous obtenez une erreur du style "Timeout" après un long moment, vérifiez que le backoffice est en ligne et fonctionnel.

Dans 90% des cas, le poste qui accueille Polaris est éteint ou coupé d'internet au moment où la requête a été effectuée.

## **Exportation des ventes**

### **"Débit déséquilibré"**

Il est possible lors d'une synchronisation qu'une commande ne soit pas exportée vers Polaris avec un message qui ressemble 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 Erreur lors  
de la synchronisation de la commande SCPBNLQHT : 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 votre commande comporte un bon (voucher) qui existe sous Polaris mais dont le montant est différent (en général inférieur) de celui enregistré sous Prestashop et utilisé dans la commande (édition manuelle du bon sur l'une des deux plateformes avant utilisation, bug, ...).

Le programme est conçu pour ne pas importer cette vente puisque l'évènement n'aurait jamais du se produire et que son acceptation pose des problèmes fiscaux.

La seule solution dans ce cas consiste à ressaisir manuellement la vente sous Polaris.

## **Importation**

**La synchronisation est terminée, aucune fiche produit n'a été créée.**

Il est fréquent à la mise en place d'un site de tomber sur cette erreur. Si aucune fiche produit ne se crée lors de votre première importation :

- vérifiez que le pont est en mode complet. En effet, en mode basique, le pont ne crée pas de fiche produit : il se contente de mettre à jour le stock et les prix ;
- vérifiez ensuite que les grilles de tailles ainsi que les marques des produits que vous essayez d'importer ont bien été mappés. Dans le cas contraire, le comportement par défaut du module est de les ignorer.
- dans tous les cas, consultez les journaux d'audit pour connaître la raison d'une non-importation.

### **Un produit devrait s'importer mais ne le fait pas.**

Un produit de votre backoffice devrait s'importer sur Prestashop mais ne le fait pas, vérifiez ces points après avoir cherché dans le journal d'audit la raison de la non importation. Si votre produit n'y ait pas, c'est que votre backoffice ne l'exporte pas. Contactez votre support habituel pour comprendre pourquoi. Dans le cas contraire, la raison 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 la grille de tailles utilisées soient mappées ;
- qu'aucune de ses catégories ne soit filtrées dans le mappage.