



Procédure de mise en production

Connecter Doli2Shop à un Dolibarr déjà peuplé

Doli2Shop · Mise à jour : 28/09/2026

Document générique. S'applique à toute installation où Shopify est alimenté par la migration d'une boutique existante (PrestaShop, WooCommerce, Magento...) alors que Dolibarr contient déjà l'historique commercial correspondant.

1. Le risque que cette procédure écarte

Lorsque vos commandes historiques sont reprises dans Shopify, elles y reçoivent des **identifiants Shopify neufs**, et leurs dates de création et de modification sont celles du **jour de la migration** — Shopify permet d'antidater la date de traitement, jamais ces deux-là.

Or Doli2Shop reconnaît une commande déjà importée à son **identifiant Shopify**. Une commande présente dans Dolibarr sous une autre origine ne porte aucun identifiant Shopify : le module ne peut pas savoir qu'il s'agit de la même vente. Sans précaution, elle est importée une seconde fois, à la date du jour, dans votre comptabilité.

Trois chemins peuvent importer une commande, et ils sont indépendants :

Chemin	Ce qu'il fait	Filtre de date
Webhook de commande	Traite toute commande inconnue comme une création	aucun
Tâche <code>Doli2ShopOrdersCatchup</code>	Réclame à Shopify les commandes modifiées récemment et importe celles absentes de Dolibarr	fenêtre glissante réglable
Tâche <code>Doli2ShopOrdersHistoricalImport</code>	Import historique volontaire	bornes de dates propres

2. Avant la migration — les réglages

- **Sens de la synchronisation de stock.** `DOLI2SHOP_SYNC_STOCKS_DIRECTION` doit valoir `dolibarr_to_shopify` si Dolibarr est votre référence de stock. La valeur par défaut est `both` : dans ce cas un webhook d'inventaire Shopify écrit aussi des mouvements de stock dans Dolibarr.

- **Commandes non réglées.** Par défaut, seules les commandes au statut financier *payé* sont importées. Si vos commandes en attente de paiement doivent entrer (paiement à terme B2B, virement), activez `DOLI2SHOP_SYNC_NON_PAID_ORDERS`.
- **Références produits.** Les SKU des variantes Shopify doivent être identiques aux références Dolibarr : c'est ce rapprochement qui évite la création de produits et de déclinaisons en double.
- **Courriels des tiers.** Le rattachement d'un client existant se fait sur le courriel porté par la **fiche société**, pas sur ses contacts, et dans la même entité. Un courriel qui ne vit que sur un contact, ou qui diffère de celui de Shopify, produira une société en double. Réconciliez-les avant la bascule : c'est un travail de données, pas de configuration.

3. Pendant la migration — ce qui doit rester éteint

Tant que la migration n'est pas terminée **et vérifiée**, les trois chemins d'import doivent être inactifs :

1. les **webhooks de commandes** — écran *Webhooks* de la configuration du module ;
2. la tâche planifiée `Doli2ShopOrdersCatchup` ;
3. la tâche planifiée `Doli2ShopOrdersHistoricalImport`.

Vérification en une requête (lecture seule) :

```
SELECT label, entity, status FROM llx_cronjob WHERE label LIKE 'Doli2Shop%';
```

Ces deux tâches doivent afficher `status = 0`. En Multicompany, vérifiez au passage que la colonne `entity` correspond bien à l'entité où le module est configuré : une tâche créée dans la mauvaise entité s'exécute à vide, sans erreur.

La synchronisation des produits et des stocks, elle, peut fonctionner pendant cette phase.

4. La bascule

1. Terminez la migration et vérifiez-la dans Shopify.
2. **Notez l'horodatage exact de fin de migration** — appelons-le T0.
3. Ramenez `DOLI2SHOP_CATCHUP_LOOKBACK_HOURS` à une valeur qui **ne remonte pas avant T0** : 1 ou 2 heures. La valeur par défaut est 168 (7 jours), et elle ferait entrer vos commandes migrées.
4. Activez d'abord les webhooks de commandes, puis la tâche de rattrapage.

5. Après la bascule — les vérifications

- Ouvrez l'écran des **événements webhook** : il liste ce qui entre, avec son statut.
- Contrôlez qu'**aucune commande antérieure à T0** n'a été créée dans Dolibarr.

- Ne remontez `DOLI2SHOP_CATCHUP_LOOKBACK_HOURS` à sa valeur habituelle (168) **qu'une fois T0 vieux de plus de sept jours**. Avant cela, la fenêtre rattraperait les commandes migrées.

6. Ce que cette procédure ne couvre pas

La fenêtre de rattrapage borne **la tâche planifiée**, pas les webhooks.

Après la bascule, toute **modification dans Shopify d'une commande migrée** — une édition, un remboursement, une correction d'adresse, l'ajout d'une étiquette — déclenche un webhook `orders/updated`. Le module ne connaît pas cette commande : il la crée. Vous obtenez alors une commande historique réimportée, datée du jour.

Ce trou ne se ferme pas par la procédure. Il se fermerait par un réglage de « **date de mise en production** » : toute commande Shopify créée avant la date choisie serait écartée de l'import, **quel que soit le chemin emprunté** — webhook de création, webhook de modification, tâche de rattrapage — la fonction d'import historique volontaire gardant ses propres bornes.



Ce réglage n'existe pas encore dans le module. Il fait l'objet d'un développement, proposé sur devis. En son absence, le seul recours est de surveiller les commandes migrées : ne pas les modifier dans Shopify après la bascule, et vérifier régulièrement qu'aucune commande antérieure à T0 n'est apparue dans Dolibarr.

7. Si un doublon est malgré tout créé

1. Repérez-le par l'écran des **événements webhook** : l'événement porte l'identifiant de la commande Shopify concernée.
2. Tant que la commande Dolibarr créée n'est ni validée ni facturée, sa suppression est sans conséquence comptable.
3. Si elle est déjà facturée, la correction relève de la procédure Dolibarr habituelle (avoir ou annulation selon votre pratique comptable) — le module n'a rien de particulier à faire.
4. Coupez la cause avant de nettoyer : sinon le webhook suivant recréera la même commande.