



Websource

GUIDE PRATIQUE + CHECKLIST

Migrer vers PrestaShop

La méthode en 3 étapes pour passer en 9.2 en quelques heures,
sans perdre de commandes ni de positions SEO.

9.2

Ce que contient ce guide

Ce guide s'adresse aux marchands, chefs de projet et développeurs qui veulent migrer une boutique PrestaShop 1.6, 1.7 ou 8.x vers la 9.2. Chaque étape se termine par une checklist à cocher.

01	Pourquoi migrer maintenant	p. 3
02	La méthode en 3 étapes	p. 4
03	Choisir le bon chemin de migration	p. 5
04	Étape 1 — Audit automatisé	p. 6
05	Étape 2 — Migration en préproduction	p. 8
06	Étape 2 bis — Plan de tests complet	p. 10
07	Protéger le SEO	p. 11
08	Étape 3 — Bascule en production	p. 12
09	Après la migration : J+1, J+7, J+30	p. 13
10	Les 12 pièges qui font dérailler une migration	p. 14
11	La checklist complète à imprimer	p. 15

Avant de commencer

Les exigences techniques évoluent d'une version mineure à l'autre. Vérifiez toujours la page officielle « System requirements » de la documentation PrestaShop pour la version exacte que vous visez, et testez chaque commande de ce guide sur une copie avant de la lancer en production.

Le vrai coût, c'est de ne pas migrer

Une boutique qui tourne sur une version ancienne continue de fonctionner, jusqu'au jour où elle ne fonctionne plus. Le risque grandit chaque mois.

Rester en 1.6 ou 1.7, c'est...

- **Des failles de sécurité** qui ne sont plus corrigées, alors que les boutiques e-commerce sont des cibles privilégiées (skimming de cartes, injections SQL via des modules).
- **Une version de PHP obsolète** : la 1.6 et la 1.7 ne tournent pas sur PHP 8. Les hébergeurs finissent par imposer la mise à jour, et la boutique casse.
- **Des modules abandonnés** : les éditeurs arrêtent le support des anciennes versions, y compris pour les modules de paiement.
- **Des performances en retrait**, qui pèsent sur le taux de conversion et les Core Web Vitals.

Passer en 9.2, c'est...

- **Une base technique à jour** : PHP 8.1 et plus, Symfony 6.4, une version maintenue et corrigée.
- **Un back-office modernisé**, avec davantage de pages migrées vers Symfony.
- **Hummingbird**, le nouveau thème de référence, plus léger et plus accessible (le thème Classic reste disponible).
- **Une API d'administration** pour connecter ERP, PIM et outils marketing plus proprement.
- **L'écosystème à jour** : modules et thèmes récents, développés pour la 9.

Pourquoi les migrations font peur

Parce qu'elles sont souvent faites directement en production, sans inventaire préalable ni plan de retour arrière. Le problème n'est pas la migration elle-même, c'est l'absence de méthode. C'est exactement ce que ce guide corrige.

3 étapes, zéro improvisation

Le principe : tout ce qui peut casser doit casser en préproduction, jamais devant vos clients. Le jour J, on ne fait que rejouer une procédure déjà validée.

1

Audit automatisé

Inventaire de la version, de l'hébergement, des modules, des overrides, du thème et du code spécifique. On sait ce qui va casser avant de toucher à quoi que ce soit.

2

Migration en préprod

Copie exacte de la boutique, migration, corrections, puis tests complets : panier, paiement, e-mails, back-office, SEO.

3

Bascule en production

Procédure écrite et chronométrée, sauvegarde fraîche, bascule en moins de 10 minutes et plan de retour arrière prêt.

Qui fait quoi

Rôle	Responsabilités
Marchand	Liste des parcours critiques, validation fonctionnelle en préprod, choix de la date de bascule (période creuse).
Développeur / agence	Audit, migration, corrections de modules et du thème, rédaction de la procédure de bascule.
Hébergeur	Version de PHP, extensions, accès SSH, sauvegardes, éventuelle bascule DNS.
Éditeurs de modules	Versions compatibles 9.x des modules payants (à demander dès l'audit).

Planning type

Phase	Boutique simple	Boutique avec code spécifique
Audit	30 min	0,5 à 1 jour
Migration préprod + corrections	1 à 3 h	2 à 5 jours
Tests	1 h	1 à 2 jours
Bascule	< 10 min	< 30 min

Mise à jour ou réinstallation ?

Il existe deux façons de migrer. Le bon choix dépend surtout de votre version de départ.

Version actuelle	Chemin recommandé	Pourquoi
8.x	Mise à jour directe vers 9.2 avec le module Update Assistant (ex-« 1-Click Upgrade »).	Chemin officiel et le plus rapide. Passez d'abord PHP en 8.1 minimum.
1.7.x	Mise à jour en 2 temps : 1.7 → 8.2 (sous PHP 7.4), puis PHP 8.1+ → 9.2.	La 1.7 ne tourne pas sur PHP 8.1 et la 9 l'exige. L'étape 8.x sert de pont.
1.6.x	Réinstallation d'une 9.2 propre + migration des données.	Trop d'écart d'architecture (Symfony, thèmes, modules) : la mise à jour directe est risquée.
Boutique très personnalisée	Réinstallation recommandée, quelle que soit la version.	Repartir propre coûte moins cher que traîner des années d'overrides.

Mise à jour (in-place)

- Conserve tous les ID, URLs, comptes et historiques.
- Rapide si les modules sont maintenus.
- Conserve aussi les défauts accumulés (tables orphelines, overrides).

Réinstallation + import

- Base saine, thème et modules récents.
- Demande un outil d'import fiable et une vérification des URLs.
- Conservez la `_COOKIE_KEY_` d'origine pour que les clients gardent leur mot de passe.

Données à migrer en cas de réinstallation

- ☐ **Catalogue** : produits, déclinaisons, catégories, caractéristiques, images, stocks, prix spécifiques
- ☐ **Clients** : comptes, adresses, groupes, mots de passe (clé de hachage conservée)
- ☐ **Commandes** : historique, statuts, factures, avoirs, messages
- ☐ **Contenu** : pages CMS, fabricants, fournisseurs, règles de panier, transporteurs
- ☐ **SEO** : URLs simplifiées, balises meta, en conservant les ID si possible

Savoir ce qui va casser, avant de commencer

L'audit prend de 30 minutes à une journée. C'est l'étape qui fait gagner le plus de temps sur tout le projet.

1. Version et environnement

```
# Version de PrestaShop (1.7 / 8.x)
grep "const VERSION" app/AppKernel.php
# Version de PrestaShop (1.6)
grep "_PS_VERSION_" config/settings.inc.php
# Version de PHP et extensions
php -v
php -m | grep -Ei "intl|gd|zip|curl|mbstring|pdo_mysql|fileinfo|dom|simplexml|openssl"
# Version de MySQL / MariaDB
mysql -V
```

2. Prérequis serveur pour la 9.x

Élément	Valeur cible
PHP	8.1 minimum (branche 8.x récente recommandée, voir la matrice officielle de la 9.2)
Base de données	MySQL 5.7+ ou MariaDB récente
Extensions PHP	intl, gd, zip, curl, mbstring, pdo_mysql, fileinfo, dom, simplexml, openssl, json
<code>memory_limit</code>	256 Mo minimum, 512 Mo pendant la migration
<code>max_execution_time</code>	300 s pendant la migration
Accès	SSH + accès direct à la base : indispensable pour aller vite

3. Inventaire des modules

```
# Tous les modules installés, avec version et statut
SELECT name, version, active FROM ps_module ORDER BY name;
```

Classez chaque module dans l'une de ces 4 catégories :

ÉTAPE 1 — AUDIT AUTOMATISÉ (SUITE)

Catégorie	Action
Natif	Mis à jour avec PrestaShop. Rien à faire.
Tiers maintenu	Télécharger la version compatible 9.x auprès de l'éditeur.
Tiers abandonné	Trouver une alternative ou le faire adapter.
Inutilisé	Désinstaller avant la migration : c'est le gain de temps le plus simple.

4. Overrides et code spécifique

Les overrides sont la première cause de pages blanches après une migration. Listez-les tous :

```
# Overrides présents (hors fichiers index.php)
find override -name "*.php" ! -name "index.php"
# Modules maison ou modifiés : fichiers modifiés récemment
find modules -name "*.php" -newer config/defines.inc.php | head -50
# Appels à des méthodes supprimées ou dépréciées (exemples)
grep -rn "Tools::displayPrice\|Tools::jsonEncode\|Tools::jsonDecode" modules themes
override
```

Pour chaque override, posez trois questions : est-il encore utile ? Le besoin est-il couvert nativement en 9.2 ? Peut-on le remplacer par un hook ?

5. Thème

- ☐ Thème natif, acheté ou sur mesure ? Existe-t-il une version compatible 9.x ?
- ☐ Thème 1.6 : il doit être refait (le système de thèmes a changé depuis la 1.7)
- ☐ Thème 1.7 / 8 : vérifier les templates surchargés dans `themes/*/modules/` , ou repartir de Hummingbird

6. Volumes et points sensibles

- ☐ Nombre de produits, déclinaisons, clients, commandes (pour estimer la durée des scripts)
- ☐ Taille de la base et des tables de logs (`ps_connections` , `ps_guest` , `ps_log` ...) à purger
- ☐ Taille du dossier `img/` (durée des copies et des sauvegardes)
- ☐ Connecteurs externes : ERP, marketplaces, CRM, flux produits, webhooks
- ☐ Modules de paiement : version compatible et identifiants de test disponibles
- ☐ Tâches cron actives (liste et URLs)

Livrable de l'étape 1

Un tableau unique qui liste chaque module, override et connecteur, avec son statut (OK / à mettre à jour / à remplacer / à supprimer) et un responsable. S'il reste une ligne « inconnu », l'audit n'est pas terminé.

Casser en préprod, jamais en prod

1. Sauvegarde complète

```
# Base de données (sans verrouiller les tables InnoDB)
mysqldump --single-transaction --routines --triggers -u USER -p NOM_BASE > backup.sql
# Fichiers (sans le cache)
tar -czf backup-fichiers.tar.gz --exclude="./var/cache" .
```

Vérifiez que la sauvegarde se restaure réellement. Une sauvegarde jamais testée n'est pas une sauvegarde.

2. Création de la préproduction

- ☐ Sous-domaine dédié (ex. `preprod.maboutique.fr`) protégé par mot de passe HTTP
- ☐ Même version de PHP et de MySQL que la future production
- ☐ Restauration des fichiers et de la base, mise à jour du domaine dans `ps_shop_url`
- ☐ Paramètres de connexion à jour (`app/config/parameters.php` en 1.7+)
- ☐ Cache vidé : `rm -rf var/cache/*`

3. Neutraliser les effets de bord

À faire avant toute manipulation

- **E-mails** : désactiver l'envoi (Paramètres avancés > E-mail) ou rediriger vers un outil de capture type Mailpit.
- **Paielements** : passer tous les modules en mode test ou sandbox.
- **Crons et connecteurs** : couper les synchronisations ERP et marketplaces pour éviter les doublons de commandes ou de stock.
- **Indexation** : la protection par mot de passe empêche Google d'indexer la préprod.
- **Mode debug** : activer `_PS_MODE_DEV_` dans `config/defines.inc.php` pour voir les erreurs.

4. Nettoyage avant migration

- ☐ Désinstaller les modules classés « inutilisés » pendant l'audit
- ☐ Purger les tables de statistiques et de logs volumineuses
- ☐ Supprimer les paniers abandonnés très anciens
- ☐ Désactiver temporairement les overrides non indispensables (Paramètres avancés > Performances)

5. Lancer la migration

Chemin « mise à jour » (8.x, 1.7.x)

- Installer la dernière version du module **Update Assistant**.
- Lancer sa vérification de prérequis et corriger chaque point bloquant.
- En 1.7 : mettre d'abord à jour vers la 8.x sous PHP 7.4, contrôler, puis passer PHP en 8.1+.
- Lancer la mise à jour vers 9.2. Pour les grosses boutiques, préférer le mode ligne de commande (CLI) du module, qui ne dépend pas des timeouts du navigateur.
- Consulter les logs de mise à jour avant de passer à la suite.

Chemin « réinstallation » (1.6, boutique très personnalisée)

- Installer une 9.2 vierge, avec les mêmes langues et devises.
- Reporter la `_COOKIE_KEY_` de l'ancienne boutique.
- Importer les données avec un outil de migration éprouvé, en conservant les ID.
- Installer les versions 9.x des modules et le nouveau thème.
- Réappliquer la configuration : transporteurs, taxes, statuts, paiements.

6. Corrections après migration

- ☐ Réactiver les modules un par un, en vérifiant les logs à chaque fois (`var/logs/`)
- ☐ Remplacer les overrides par des hooks quand c'est possible
- ☐ Adapter les templates du thème aux nouveaux noms de variables et de hooks
- ☐ Régénérer les miniatures d'images et le fichier `.htaccess`
- ☐ Reconstruire les index de recherche
- ☐ Vider le cache et désactiver le mode debug avant les tests finaux

7. Écrire la procédure de bascule pendant la préprod

Chaque commande lancée en préprod doit être notée dans un document unique, dans l'ordre, avec sa durée. Idéalement, faites un deuxième passage complet sur une copie fraîche de la production en suivant uniquement ce document. S'il fonctionne sans improvisation, la procédure est prête.

Le secret des migrations « en quelques heures »

La rapidité ne vient pas d'un module miracle : elle vient d'une procédure répétée au moins une fois de bout en bout. Le jour J, il n'y a plus de surprise, seulement des étapes connues et chronométrées.

Tester comme un client, puis comme un admin

Front-office

- ☐ Page d'accueil, catégories, filtres à facettes
- ☐ Fiche produit : déclinaisons, prix, stock, images
- ☐ Recherche interne
- ☐ Création de compte, connexion avec un ancien compte
- ☐ Mot de passe oublié
- ☐ Ajout au panier, codes promo, règles de panier
- ☐ Calcul des frais de port pour chaque zone
- ☐ Affichage mobile et tablette
- ☐ Pages CMS, formulaire de contact, bannière cookies

Tunnel de commande

- ☐ Commande invité et commande client connecté
- ☐ Chaque moyen de paiement, en mode test
- ☐ Paiement refusé ou annulé : retour correct au panier
- ☐ Retour de la banque (IPN / webhook) : statut mis à jour
- ☐ Facture PDF générée et correcte (TVA, mentions légales)

E-mails

- ☐ Confirmation de commande
- ☐ Changements de statut (expédiée, livrée)
- ☐ Création de compte, mot de passe
- ☐ Alertes internes (nouvelle commande, stock)
- ☐ Rendu mobile et liens dans les e-mails

Back-office

- ☐ Connexion de chaque employé et droits d'accès
- ☐ Création et modification d'un produit
- ☐ Traitement d'une commande de bout en bout
- ☐ Remboursement et avoir
- ☐ Imports CSV habituels
- ☐ Pages de configuration de chaque module

Technique

- ☐ Aucune erreur dans `var/logs/` ni dans les logs PHP
- ☐ Crons exécutés avec succès
- ☐ Synchronisations ERP et marketplaces testées
- ☐ Temps de chargement égal ou meilleur qu'avant
- ☐ Scores Lighthouse / Core Web Vitals comparés

Migrer sans perdre une seule position

Une migration mal préparée peut faire chuter le trafic organique pendant des mois. La règle : chaque URL qui existait avant doit répondre après, soit à l'identique, soit par une redirection 301.

Avant la migration

- ☐ Crawler toute la boutique (Screaming Frog ou équivalent) et exporter la liste des URLs
- ☐ Exporter les pages qui génèrent le plus de trafic depuis Google Search Console
- ☐ Relever les réglages d'URL (Préférences > Trafic & SEO) : format des URLs produit, catégorie, CMS
- ☐ Sauvegarder le `robots.txt`, le `.htaccess` et le sitemap actuels
- ☐ Noter les balises title, meta description et les données structurées des pages clés

En préproduction

- ☐ Recrawler la préprod et comparer les deux listes d'URLs
- ☐ Toute URL en 404 doit être corrigée ou redirigée en 301 vers son équivalent le plus proche
- ☐ Vérifier les balises canonical, hreflang (multilingue) et les données structurées Product
- ☐ Contrôler que les images gardent les mêmes URLs, ou qu'elles sont redirigées

Après la bascule

- ☐ Vérifier que la production n'a pas hérité d'un `noindex` ou d'un `robots.txt` bloquant
- ☐ Régénérer le sitemap et le soumettre dans Search Console
- ☐ Suivre les erreurs d'exploration et les 404 pendant 30 jours
- ☐ Comparer impressions, clics et positions sur les pages clés à J+7 et J+30

L'erreur la plus fréquente

Changer le format des URLs « tant qu'on y est ». Si vous voulez modifier la structure des URLs, faites-le dans un second temps, une fois la migration stabilisée, avec un plan de redirection dédié.

Moins de 10 minutes, avec un plan B

Choisissez un créneau creux (souvent tôt le matin en semaine), jamais un vendredi soir ni une veille de soldes. Prévenez l'équipe et le support client.

Quand	Action
J-7	Procédure validée sur une copie fraîche. Date de bascule communiquée. Versions 9.x des modules téléchargées.
J-1	Gel des modifications : plus de changement de catalogue, de configuration ou de module. TTL DNS abaissé si changement de serveur.
H-30 min	Crons et synchronisations coupés. Sauvegarde complète (fichiers + base) vérifiée.
H0	Mode maintenance activé (votre IP en liste blanche). Exécution de la procédure, étape par étape, chronométrée.
H+10 min	Tests express (voir ci-dessous). Si tout est vert : maintenance désactivée, crons réactivés.
H+1 h	Surveillance des logs, des premières commandes et des e-mails envoyés.

Tests express avant réouverture (10 minutes)

- ☐ Accueil, une catégorie, une fiche produit s'affichent sans erreur
- ☐ Connexion avec un compte client existant
- ☐ Commande de bout en bout avec un paiement réel de faible montant, puis remboursement
- ☐ E-mail de confirmation reçu
- ☐ La commande apparaît dans le back-office avec le bon statut
- ☐ Aucune nouvelle erreur dans les logs

Plan de retour arrière

Décidez à l'avance du critère d'abandon (par exemple : paiement impossible après 20 minutes de diagnostic). Le retour arrière consiste à restaurer la sauvegarde de H-30 min. Il doit avoir été testé et prendre moins de 15 minutes. Avoir ce plan, c'est ce qui permet de basculer sereinement.

J+1, J+7, J+30

Une migration n'est terminée que lorsque les chiffres sont revenus au niveau d'avant, ou mieux.

J+1 Stabilisation

- ☐ Relire les logs PHP et PrestaShop des dernières 24 h
- ☐ Vérifier que toutes les commandes passées ont bien un paiement associé
- ☐ Contrôler les stocks et les synchronisations externes
- ☐ Confirmer que les crons ont tourné
- ☐ Recueillir les retours du support client

J+7 Contrôle

- ☐ Comparer le taux de conversion avec la semaine précédente
- ☐ Vérifier les 404 et les erreurs d'exploration dans Search Console
- ☐ Comparer les Core Web Vitals
- ☐ Supprimer la préproduction ou la protéger durablement
- ☐ Supprimer les dossiers de sauvegarde accessibles depuis le web

J+30 Bilan

- ☐ Comparer le trafic organique, les positions et le chiffre d'affaires
- ☐ Lister la dette technique restante (overrides conservés, modules à remplacer)
- ☐ Planifier la veille des mises à jour mineures 9.x
- ☐ Documenter la procédure pour la prochaine mise à jour

Rester à jour, ensuite

Une fois en 9.x, appliquez les versions mineures et correctifs régulièrement, en passant chaque fois par la préprod. Une mise à jour mineure prend alors quelques minutes, au lieu d'un projet de plusieurs semaines tous les cinq ans.

12 erreurs qui font dérailler une migration

#	Erreur	Comment l'éviter
1	Migrer directement en production	Toujours passer par une préprod identique.
2	Sauvegarde jamais restaurée	Tester la restauration avant le jour J.
3	Oublier de changer la version de PHP au bon moment	Suivre l'ordre : 1.7 → 8.x sous PHP 7.4, puis PHP 8.1+, puis 9.x.
4	Garder tous les overrides	Les auditer et les remplacer par des hooks.
5	Modules installés mais inutilisés	Les désinstaller avant de migrer.
6	Perdre la <code>_COOKIE_KEY_</code>	La conserver pour que les clients gardent leur mot de passe.
7	Préprod indexée par Google	Protection par mot de passe HTTP.
8	E-mails envoyés aux vrais clients depuis la préprod	Désactiver ou capturer les e-mails.
9	Commandes en double dans l'ERP	Couper les synchronisations en préprod.
10	Changer les URLs en même temps	Une seule évolution à la fois.
11	Commandes perdues pendant la bascule	Mode maintenance et gel des modifications.
12	Pas de critère d'abandon	Définir à l'avance quand et comment revenir en arrière.

La checklist complète

1 Audit

- ☐ Version de PrestaShop, PHP et MySQL relevées
- ☐ Chemin choisi : mise à jour ou réinstallation
- ☐ Prérequis 9.2 validés chez l'hébergeur
- ☐ Modules classés (natif / maintenu / abandonné / inutilisé)
- ☐ Versions 9.x des modules tiers obtenues
- ☐ Overrides listés et décidés
- ☐ Thème évalué
- ☐ Connecteurs et crons listés

2 Préproduction

- ☐ Sauvegarde complète faite et restauration testée
- ☐ Préprod créée, protégée par mot de passe
- ☐ E-mails, paiements, crons neutralisés
- ☐ Nettoyage (modules, logs, paniers)
- ☐ Migration lancée, logs relus
- ☐ Modules réactivés un par un
- ☐ Thème adapté
- ☐ Procédure de bascule écrite et rejouée

2 bis Tests & SEO

- ☐ Parcours client complet
- ☐ Chaque moyen de paiement testé
- ☐ Tous les e-mails vérifiés
- ☐ Back-office et droits validés
- ☐ Crawl avant / après comparé, 0 URL en 404
- ☐ Redirections 301 en place
- ☐ Performances égales ou meilleures

3 Bascule

- ☐ Date et créneau creux choisis
- ☐ Gel des modifications à J-1
- ☐ Sauvegarde fraîche à H-30 min
- ☐ Mode maintenance activé
- ☐ Procédure exécutée, chronométrée
- ☐ Tests express au vert
- ☐ Maintenance désactivée, crons réactivés

J+ Suivi

- ☐ J+1 : logs, commandes, stocks
- ☐ J+7 : conversion, Search Console
- ☐ J+30 : bilan trafic et chiffre d'affaires



Websource

BESOIN D'UN COUP DE MAIN ?

Votre boutique en 9.2, en quelques heures.

- Audit de votre boutique et devis au forfait
- Migration complète en préproduction, tests inclus
- Bascule en production accompagnée, plan de retour arrière prêt
- Suivi SEO et technique après la migration

websource.fr

Agence web à Aix-en-Provence · sites, e-commerce, applications