Les caractéristiques techniques de Portix V3

Architecture, sécurité, stockage JSON, PHP 8.3.3 à 8.5.x, mises à jour signées et fonctions techniques du cœur Portix.

Environnement

Portix V3 prend en charge PHP 8.3.3 à PHP 8.5.x. Il utilise les sessions PHP, JSON, OpenSSL et des extensions standards selon les fonctions activées, notamment cURL, PharData, DOM et les bibliothèques d’images.

Caractéristiques

- contrôleur frontal et routes explicites ;
- architecture par contrôleurs, services et gabarits ;
- stockage JSON transactionnel sans SQL obligatoire ;
- migrations de schéma avec sauvegarde de sécurité ;
- 53 thèmes natifs et générateur de thèmes ;
- impression A4 et PDF structurés ;
- moteur de recherche configurable par sources, correspondance et tri.

Recherche interne

Le moteur parcourt uniquement les contenus publiés et publics. L’administration choisit les articles, documents, FAQ et événements indexés, le mode « tous les mots », « au moins un mot » ou « expression exacte », ainsi que le tri, la limite de résultats et la taille des extraits.

Les correspondances dans les titres sont prioritaires. Les brouillons et les documents réservés aux membres sont exclus afin d’éviter toute exposition involontaire.

Réseau et mises à jour

Le client de mise à jour utilise exclusivement HTTPS et peut essayer cURL, le flux PHP ou une socket TLS. Les URL locales, privées ou réservées sont bloquées pour les téléchargements distants non autorisés.

Protection des fichiers

Les données, journaux, sauvegardes, paquets de mise à jour et zones de récupération sont protégés. L’importateur vérifie le manifeste, la version cible, les chemins et le SHA-256 de chaque fichier.

État courantLe moteur de recherche utilise directement le stockage JSON et les règles de confidentialité du cœur Portix.

Préparer un environnement cohérent

L’installation doit disposer d’une version PHP compatible, des extensions requises, d’un accès en écriture limité aux répertoires de stockage et d’un serveur Web correctement configuré. Les fichiers applicatifs et les données ne doivent pas être mélangés. Cette séparation facilite les mises à jour, les sauvegardes et le diagnostic en cas d’erreur.

Comprendre l’organisation de Portix

Le cœur de Portix reçoit la requête, identifie la route, applique les contrôles d’accès puis appelle le contrôleur correspondant. Les services regroupent les règles métier et les gabarits se chargent de l’affichage. Les modules ajoutent leurs propres routes et fonctions sans modifier directement les données des autres composants.

Mettre à jour sans perdre les données

Les archives d’importation doivent contenir uniquement les fichiers autorisés et un manifeste conforme. Les données locales, comptes, contenus et réglages restent en dehors du paquet automatique. Une mise à jour sérieuse commence par une sauvegarde, vérifie la version de départ, contrôle les empreintes puis déclenche les migrations prévues pour la version cible.

Contrôles après installation

Une fois Portix installé, il convient de tester la première connexion, le changement du mot de passe provisoire, l’accès à l’administration, les courriels, les PDF, les tâches planifiées et les principaux modules. Les permissions du système de fichiers et les journaux PHP doivent également être vérifiés avant l’ouverture du portail aux utilisateurs.

Des contrôles mesurables sur le cœur du portail

La version actuelle maintient un inventaire explicite des routes GET et POST, des formulaires protégés par CSRF, des alias historiques et des contrôleurs associés. Les archives de distribution sont vérifiées par extraction complète et les fichiers JSON sont contrôlés avant livraison.

PHP natif, mais avec une discipline de version

Le fonctionnement sans base SQL obligatoire reste un choix d’architecture. Il impose en contrepartie une attention stricte aux écritures atomiques, aux migrations, aux manifestes et à la séparation entre code de mise à jour et données runtime. Les mécanismes récents de synchronisation visent précisément à maintenir cette cohérence.