Sécurisation des Clés API dans WPshop

Architecture de chiffrement AES-256 et exploitation des Salts WordPress • eoxia.com

Introduction : Vers une forteresse pour vos clés d'API

Dans le cadre du développement et de la maintenance de nos solutions e-commerce, la sécurité des données sensibles est une priorité absolue. Sur le projet **WPshop** (connecteur ERP / Dolibarr pour WordPress), nous introduisons une architecture de protection robuste pour la clé d'API Dolibarr, transformant le stockage des données en une véritable forteresse numérique.

Pour approfondir les mécanismes de sécurisation sous-jacents, vous pouvez consulter la documentation officielle sur WordPress Security APIs et les recommandations de l'OWASP concernant le Cryptographic Storage.

1. Chiffrement AES-256 en Base de Données

Avant qu'une clé d'API Dolibarr ne soit enregistrée dans la table `wp_options` de WordPress, elle subit un traitement cryptographique rigoureux via la fonction `openssl_encrypt`. Ce mécanisme s'appuie directement sur les clés de sécurité uniques de WordPress (les **Salt Keys** définies dans le fichier `wp-config.php`) pour servir de sel et de cadenas cryptographique.

En base de données, la clé n'apparaît plus jamais en clair : elle est convertie en une chaîne de caractères illisible, par exemple :

```
U2FsdGVkX1+... [Chaîne chiffrée AES-256 sécurisée par les Salts WP]
```

2. Déchiffrement à la volée & Transparence

Lorsque le plugin a besoin d'interagir avec l'ERP Dolibarr (pour tester la connexion, récupérer un statut ou synchroniser un catalogue de produits), il opère un déchiffrement silencieux et éphémère. La clé est décryptée strictement pour la durée de la requête HTTP, sans jamais être réécrite en clair dans les logs, le cache ou la base de données.

3. Masquage total dans l'interface d'administration

Côté interface utilisateur (back-office WordPress), l'onglet ERP a été repensé pour éliminer tout risque d'exposition visuelle ou d'inspection du DOM :

- La vraie clé d'API n'est plus transmise au navigateur web lors du chargement de la page.
- Le champ de saisie est transformé en un champ de type mot de passe (`type="password"`).
- Le code PHP force l'affichage de points de censure universels : `••••••••••••••••••••` .

- Même en inspectant le code source HTML de la page, un administrateur ou un script tiers ne peut récupérer la clé en clair.

4. Gestion intelligente de la sauvegarde

Le contrôleur de formulaire gère les modifications de manière granulaire afin d'éviter les écrasements accidentels :

Action de l'utilisateur	Comportement du Plugin WPshop
Clic sur "Enregistrer" avec les points (••••••)	Le système détecte que la valeur n'a pas changé et conserve l'ancienne clé chiffrée en base.
Champ laissé entièrement vide	Le plugin supprime purement et simplement la clé d'API enregistrée.
Saisie d'une nouvelle clé	La nouvelle valeur est capturée, chiffrée en AES-256 via les Salts WP, puis stockée de manière sécurisée.

> **Note technique Eoxia :** Cette implémentation garantit une conformité accrue avec les bonnes pratiques de développement pour eoxia.com, assurant la souveraineté et l'intégrité des données de nos clients e-commerce.