

Contents

Dossier Bloc 4 – Maintenance et supervision	1
Mise à jour des dépendances	1
Processus	1
Fréquence	2
Système de supervision	2
Outils utilisés	2
Indicateurs suivis	2
Modalités de signalement	2
Collecte et consignation des anomalies	3
Processus	3
Informations consignées	3
Outils	3
Fiche de consignation d’une anomalie	3
Traitement d’une anomalie détectée	3
Exemple traité	3
Recommandations d’amélioration	4
Journal de version (extrait)	4
Exemple de problème résolu avec le support client	4

Dossier Bloc 4 – Maintenance et supervision

- Mise à jour des dépendances
- Système de supervision
- Collecte et consignation des anomalies
- Fiche d’anomalie rencontrée
- Traitement d’une anomalie détectée
- Recommandations d’amélioration
- Journal de version
- Exemple de problème résolu avec le support client

Mise à jour des dépendances

Processus

- Laravel :

- Mise à jour via `composer update` (test en local, staging, puis production)
- Vérification des changements majeurs dans `composer.lock`
- Tests automatiques après chaque mise à jour
- **Node.js** :
 - Mise à jour via `npm update` puis `npm run build`
 - Vérification des éventuels breaking changes
- **WordPress** :
 - Mises à jour via l'interface admin (core, plugins, thèmes)
 - Sauvegarde avant chaque mise à jour
 - Vérification de la compatibilité des plugins
- **Automatisation** :
 - Utilisation possible de Dependabot (GitHub) pour les alertes de sécurité et les PR automatiques

Fréquence

- Vérification hebdomadaire en développement
- Mises à jour de sécurité appliquées dès publication

Système de supervision

Outils utilisés

- **Laravel** : logs centralisés (fichiers, Sentry)
- **WordPress** : logs PHP, plugin de monitoring (Query Monitor)
- **UptimeRobot** : surveillance de la disponibilité HTTP(S)
- **Alertes** : email en cas d'erreur critique ou d'indisponibilité

Indicateurs suivis

- Taux d'erreurs 500/404
- Temps de réponse moyen
- Espace disque serveur
- Nombre d'utilisateurs actifs
- Disponibilité des services (API, site public, admin)

Modalités de signalement

- Alertes automatiques par email ou Slack
- Dashboard de suivi (Grafana, Sentry)
- Rapport hebdomadaire envoyé à l'équipe

Collecte et consignation des anomalies

Processus

- Signalement via GitHub Issues ou formulaire interne (menu “Support”)
- Collecte automatique des erreurs critiques via Sentry et logs Laravel
- Analyse quotidienne des logs d’erreur

Informations consignées

- Date et heure
- Description détaillée
- Gravité (bloquant, majeur, mineur)
- Environnement (prod, staging, local)
- Étapes de reproduction
- Capture d’écran ou log associé
- Statut (ouvert, en cours, résolu)

Outils

- GitHub Issues
- Sentry
- Tableur de suivi (optionnel)

Fiche de consignation d’une anomalie

Date	Description	Gravité	Environnement	Étapes de reproduction	Capture/log	Statut
2025-07-15	Erreur 500 lors de la création d’un article	Majeur	Production	1. Se connecter en admin 2. Aller sur /admin/articles 3. Cliquer sur “Créer”	voir log #123	Résolu

Traitement d’une anomalie détectée

Exemple traité

- **Anomalie** : Erreur 500 lors de la création d’un article
- **Diagnostic** : Problème de validation sur le champ “titre” (null non géré)
- **Correction** : Ajout d’une règle de validation dans le contrôleur Laravel

- **Test** : Ajout d'un test unitaire pour la création d'article sans titre
- **Déploiement** : Correction validée sur staging, puis déployée en production
- **Suivi** : Aucun nouvel incident signalé depuis la correction

Recommandations d'amélioration

- Mettre en place un monitoring temps réel plus avancé (Grafana, Prometheus)
- Automatiser les mises à jour de sécurité (Dependabot, WP-CLI)
- Renforcer la documentation des procédures de maintenance
- Ajouter des tests de charge réguliers
- Mettre en place un support utilisateur avec FAQ et base de connaissances

Journal de version (extrait)

Version	Date	Corrections / évolutions principales
1.3	2025-07-20	Correction bug création article, MAJ sécurité
1.2	2025-06-15	Ajout monitoring, optimisation requêtes
1.1	2025-05-10	Ajout tests unitaires, gestion des rôles
1.0	2025-04-01	Version initiale, base Laravel/WordPress

Exemple de problème résolu avec le support client

- **Signalement** : Un utilisateur ne parvient pas à réinitialiser son mot de passe (erreur inconnue)
- **Collecte** : Analyse des logs, reproduction du bug sur l'environnement de test
- **Communication** : Échanges par email avec l'utilisateur pour préciser le contexte
- **Correction** : Correction du bug dans le contrôleur de réinitialisation, ajout d'un test unitaire
- **Retour** : L'utilisateur confirme la résolution et la fluidité du process