

Rapport d'audit du projet sync__woocommerce__odoo

Date : 13 juin 2025

1. Architecture et Structure

- Structure modulaire : `core/`, `utils/`, `scripts/`, `tests/`, `config/`.
- Dockerisation complète (`Dockerfile`, `docker-compose.yml`).
- Configuration centralisée par `.env`.
- Documentation technique générable (MkDocs).

2. Fonctionnalités

- Synchronisation WooCommerce - Odoo (automatique et manuelle).
- Dashboard Flask (audit, purge, synchronisation, monitoring).
- Audit CSV de chaque synchronisation.
- Gestion des doublons par SQLite locale.
- Monitoring Prometheus (`/metrics`), Sentry (erreurs critiques).
- Webhooks Flask pour synchronisation temps réel.
- Tests unitaires et d'intégration (CI GitHub Actions).

3. Sécurité

- Credentials dans `.env` (jamais versionnés).
- Sentry pour la surveillance des erreurs.
- Dashboard sans authentification (à renforcer pour la prod).
- Webhooks sans vérification de signature (à renforcer pour la prod).

4. Qualité logicielle

- Logs structurés (logger Python, audit CSV).
- Gestion d'erreurs robuste (`try/except`, logs, Sentry).
- Tests automatisés (unitaires, intégration, audit).
- CI/CD : pipeline GitHub Actions.
- Documentation technique (API, architecture, monitoring).

5. Observabilité et Monitoring

- Audit CSV pour chaque synchronisation.
- Endpoint `/metrics` pour Prometheus (uptime, nombre de sync).
- Sentry pour les erreurs critiques.
- Logs accessibles via `docker-compose logs`.

6. Déploiement et Maintenance

- Docker Compose pour déploiement simple.
- Script de purge pour la base locale et l'audit.
- Procédure de mise à jour documentée.
- Volume Docker pour persistance des données Odoo (si besoin).

7. Points forts

- Projet industrialisé, prêt pour la production.
- Monitoring et audit intégrés.
- Tests et CI automatisés.
- Documentation claire et extensible.
- Extensible (webhooks, dashboard, endpoints, etc.).

8. Axes d'amélioration

- Sécurité dashboard : ajouter une authentification.
- Sécurité webhooks : vérifier la signature WooCommerce.
- Logs centralisés : possibilité d'exporter vers ELK/Graylog.
- Déploiement cloud : exemple pour Azure, AWS, etc.
- Interface dashboard : enrichir avec plus de stats, logs, filtres.
- Tests avancés : mocks API externes, tests de charge.
- Alertes : emails ou Slack en cas d'erreur critique.

9. Résumé

Projet robuste, modulaire, monitoré et prêt pour la production. Quelques renforcements de sécurité et d'expérience utilisateur peuvent encore l'améliorer pour un usage en entreprise ou en SaaS.

Ce rapport peut être exporté en PDF via un outil comme Pandoc, VS Code ou un éditeur Markdown avec export PDF.