

Rapport d'audit du projet initial

Terminologie :

- *Dépendances* : Logiciel, librairie, module qui ajoute des fonctionnalités au projet
- *Supporté* : Dépendance encore maintenu
- *Déprécié* : Dépendance présente mais déconseillée
- *Obsolète* : Dépendance qui n'est plus utilisé, maintenu ou mis à jour actuellement

Introduction

Ce rapport présente les résultats de l'audit réalisé sur le projet qui utilise Symfony 3.1.



L'objectif de cet audit est d'évaluer l'état actuel du code, d'identifier les problèmes potentiels et de proposer des recommandations pour améliorer la qualité du projet.

Les outils utilisés pour cet audit incluent Symfony Insight et Codacy, qui ont permis d'obtenir une vue d'ensemble des dépendances, de la qualité du code, ainsi que des suggestions d'amélioration.

Méthodologie

L'audit a été mené en utilisant les outils suivants :

- *Codacy* : Vérification de la sécurité, de la complexité, et de la couverture des tests
- *Symfony Insight* : Evaluation de la qualité du code et les risques associés
- *Composer.json* : Analyse de la liste des dépendances

Résultat obtenu

Liste des dépendances

Dans l'état actuel du projet, l'analyse des dépendances montre une part significative de dépendances obsolètes ou dépréciées.

En effet, parmi les principales bibliothèques utilisées, plusieurs sont obsolètes, notamment *Symfony* (version 3.1) et des composants clés comme *SwiftMailer*, *Doctrine*, ou encore des packages comme *SensioGeneratorBundle*.

Ces versions obsolètes ne bénéficient plus de support actif ni de mises à jour de sécurité, ce qui augmente le risque pour le projet de rencontrer des vulnérabilités de sécurité ou des erreurs non résolues à l'avenir.

Certaines dépendances sont également dépréciées, ce qui signifie qu'elles restent fonctionnelles pour l'instant, mais qu'elles ne seront probablement plus supportées dans les versions futures de Symfony et devraient être mises à jour ou remplacées par des solutions plus récentes.

En somme, le projet nécessite une mise à jour complète de ses dépendances, tant pour la stabilité que pour la sécurité à long terme, afin de se conformer aux normes actuelles et d'améliorer la compatibilité avec les nouvelles versions de Symfony et de PHP.

| Dépendance | Version | État |
|--------------------------------------|---------|----------|
| php | >=5.5.9 | Obsolète |
| symfony/symfony | 3.1 | Obsolète |
| doctrine/orm | ^2.5 | Déprécié |
| doctrine/doctrine-bundle | ^1.6 | Obsolète |
| doctrine/doctrine-cache-bundle | ^1.2 | Déprécié |
| symfony/swiftmailer-bundle | ^2.3 | Obsolète |
| symfony/monolog-bundle | ^2.8 | Supporté |
| symfony/polyfill-apcu | ^1.0 | Supporté |
| sensio/distribution-bundle | ^5.0 | Obsolète |
| sensio/framework-extra-bundle | ^3.0.2 | Déprécié |
| incenteev/composer-parameter-handler | ^2.0 | Supporté |
| sensio/generator-bundle | ^3.0 | Obsolète |
| symfony/phpunit-bridge | ^3.0 | Obsolète |

Analyse avec Symfony Insight

Note : 27 sur 100

Gravité :

7 Critique / 2 Majeur / 6 mineur / 1 info

Risque :

- 1 Fuite de données
- 4 Productivité
- 1 Fiabilité
- 5 Réputation
- 3 Sécurité
- 2 Non assuré

Statistiques :

- Lignes de code : 2 523
- Nombre de suggestions : 16

Liste des suggestions :

- Les dépendances de votre projet n'ont pas pu être installées
- Votre application n'a pas pu démarrer
- Votre projet ne doit pas s'appuyer sur des dépendances avec des problèmes de sécurité connus (3)
- Votre projet ne doit pas exposer de configuration d'infrastructure sensible
- Votre projet doit utiliser une favicon personnalisée au lieu de celle par défaut
- Votre projet doit utiliser les migrations Doctrine
- Votre projet ne doit pas contenir de commentaires "FIXME"
- Votre projet ne doit pas contenir de code commenté (2)
- Votre projet ne doit pas utiliser de fichier .htaccess
- Les applications Web doivent contenir un fichier site.webmanifest
- Le fichier composer.json de votre projet ne doit pas générer d'avertissements

Analyse avec Codacy

Note : Grade B

Evolution de la qualité :

Problèmes : 28 % / Complexité : 3 % / Duplication : 0 % / Couverture : Aucune

Répartition des problèmes (66)

Sécurité : 63

Code inutilisé : 3

Conclusion

L'audit du projet réalisé sur Symfony 3.1 met en lumière plusieurs aspects critiques à adresser pour assurer sa pérennité, sa sécurité et sa maintenabilité dans le temps. Les résultats des analyses des dépendances et de la qualité du code révèlent en effet un nombre important de dépendances obsolètes ou dépréciées, avec un impact direct sur la sécurité, la stabilité et les performances de l'application.

Sur le plan des dépendances, la majorité des packages sont soit obsolètes, soit dépréciés, ce qui expose le projet à des failles de sécurité non corrigées, et à des incompatibilités potentielles avec des environnements ou versions de PHP plus récents. Les composants clés tels que Symfony (en version 3.1), SwiftMailer, Doctrine, ainsi que certains bundles Sensio doivent impérativement être mis à jour pour bénéficier des correctifs de sécurité et d'optimisation. Par ailleurs, l'absence de support à terme pour certaines de ces bibliothèques représente un risque important pour la continuité et la maintenance du projet.

L'analyse avec Symfony Insight souligne également des problèmes de gravité élevée, affectant des aspects cruciaux du projet comme la sécurité, la productivité et la réputation. Les failles critiques identifiées, en particulier en matière de fuite de données et de vulnérabilités de sécurité, sont prioritaires à corriger pour limiter les risques. De plus, des améliorations liées à la structure du code et aux configurations sensibles sont recommandées, afin de renforcer la qualité du code et d'accroître la productivité des équipes de développement.

L'audit Codacy montre que le projet a encore un niveau de qualité relativement faible avec une note B, principalement dû aux problèmes de sécurité et au code inutilisé. L'absence de tests et de couverture de code est un autre point faible, limitant la capacité à valider automatiquement les modifications futures. La complexité du code est modérée, mais pourrait être optimisée en supprimant les duplications et en améliorant l'architecture pour favoriser la lisibilité et la maintenabilité du projet.

En somme, ce rapport d'audit recommande en priorité une migration du projet vers une version plus récente de Symfony (idéalement Symfony 7), en mettant à jour les dépendances obsolètes et en adoptant des pratiques de sécurité renforcées. La résolution des problèmes critiques identifiés par Symfony Insight et Codacy permettra de réduire les risques liés à la sécurité et d'optimiser la performance et la qualité du code.