Audit de qualité du code & performance de l'application

Auteur : **Pavel KLIMOVICH**

Table des matières

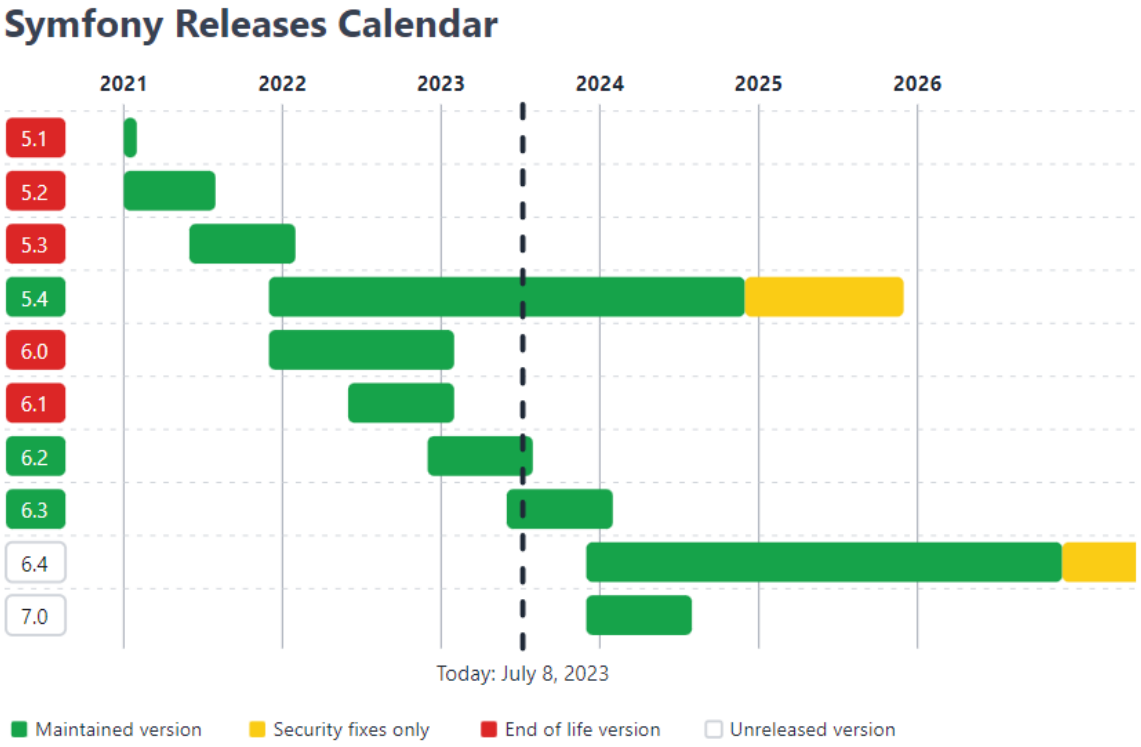
1. Introduction

L'application ToDo&Co est une startup spécialisée dans la gestion des tâches quotidiennes. Dans le cadre de son développement rapide, un audit de code et de performance a été réalisé. Ce document présente les résultats de cet audit et fournit des recommandations pour améliorer la qualité du code et les performances de l'application.

2. Version de Symfony et PHP

La version initiale de l'application Symfony est la 3.1. Cependant, cette version, mise en ligne en mai 2016, n'est plus maintenue depuis 2017 et ne fait plus partie de la roadmap.

L'application a été migré vers la version 5.4(LTS ou Long Term Support) en 2023. En passant à la version 5.4 de Symfony, on bénéficie de mises à jour jusqu'en novembre 2024 pour les bugs et novembre 2025 pour les failles de sécurité.



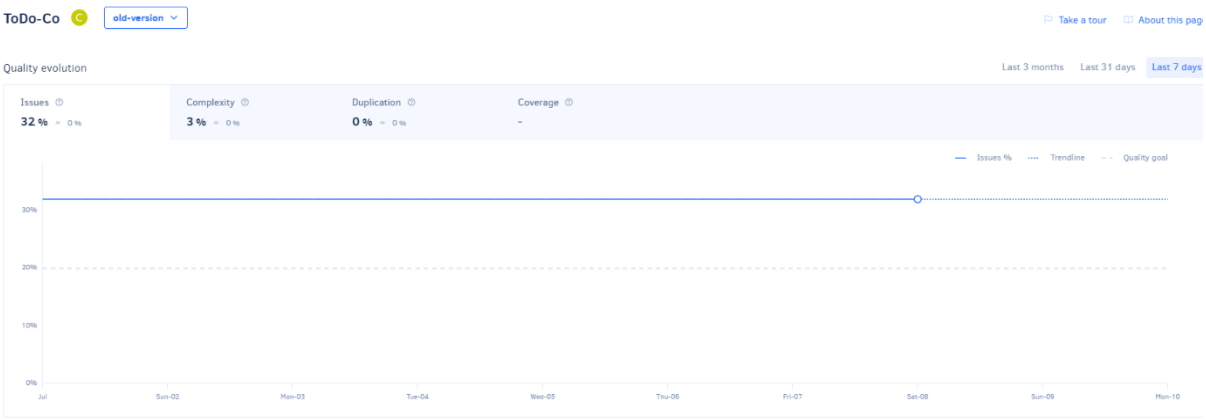
Cette migration permet de maintenir l'application à jour et de bénéficier des dernières améliorations et corrections de bugs. Il est également recommandé de prévoir une migration ultérieure vers la version 6.4 qui est aussi LTS. La version 6.4 offrira des améliorations supplémentaires, des fonctionnalités nouvelles et des correctifs de bogues par rapport à la version 5.4.

De plus, une seconde optimisation a consisté à mettre à niveau la version de PHP. Dans l'ancienne version, vous utilisez la version 5.5.9, mais en passant à la version 8.1 de PHP, nous avons améliorées les performances en termes d'utilisation des ressources. De plus, cette mise à jour à permit de profiter des fonctionnalités les plus récentes de PHP, telles que le typage des variables, ce qui rend le code plus stable et plus fiable.

3. Analyse de la qualité du code avec Codacy

3.1 Audit de l'ancienne version du projet.

L'audit de code de projet avec Codacy a révélé une note globale de C, ce qui indique que certaines améliorations sont nécessaires pour augmenter la qualité de code. Plusieurs problèmes ont été identifiés, principalement liés au style de code et à la complexité.



Tout d'abord, des violations des conventions de codage ont été détectées. Il est essentiel de suivre des normes de codage cohérentes pour garantir la lisibilité et la maintenabilité de code.

La complexité du code a également été évaluée et certains segments de code ont été jugés complexes. Une complexité excessive peut rendre le code difficile à comprendre et à maintenir. Il est conseillé de réorganiser ces parties de code pour les simplifier, en divisant les fonctions complexes en plusieurs fonctions plus petites ou en utilisant des structures de contrôle plus claires.

En résumé, pour améliorer la note globale de code, il est essentiel de prendre en compte les suggestions relatives au style de code, à la suppression des duplications et à la simplification de la complexité.

3.2 La version actuelle.

L'audit de code de la nouvelle version de ToDo&Co avec Codacy a révélé une excellente note globale de A, témoignant de la qualité élevée de code. Le code respecte les meilleures pratiques en termes de style, de lisibilité et de maintenabilité.

Toutes les conventions de codage sont scrupuleusement suivies, ce qui rend le code facilement compréhensible par les développeurs. Les noms de variables sont significatifs, la mise en forme est cohérente et les commentaires sont judicieusement utilisés pour expliquer les parties complexes du code.

Codacy n'a détecté aucune duplication dans le code, ce qui témoigne de la capacité à réutiliser efficacement les fonctions et les classes existantes. Cette pratique permet d'éviter la redondance de code et facilite la maintenance à long terme.

La complexité du code est également bien maîtrisée, avec des sections claires et concises. Le code est organisé de manière logique et évite les constructions complexes. Cela rend le code plus facile à comprendre, à tester et à maintenir.

En résumé, le projet a obtenu une note A exemplaire grâce à attention portée au style de code, à la suppression des duplications et à la gestion efficace de la complexité. Il faut continuer à appliquer ces bonnes pratiques de codage pour garantir une qualité élevée à travers le projet et faciliter la collaboration avec d'autres développeurs.

4. Analyse des performances

Dans cette section, nous allons vous présenter un comparatif de performance entre l'application initiale, telle qu'elle était au début de la reprise du projet, et l'application améliorée, correspondant à la version finale.

Un audit de performance a été réalisé à l'aide de profiler de Symfony. Le test a été réalisé sur quatre pages représentatives de l'application :

• La page d'accueil

• La page de login

• La page des tâches à réaliser

• La liste des utilisateurs

Nous avons effectué une analyse approfondie des performances des deux versions de l'application afin de mesurer les améliorations réalisées. Les résultats obtenus démontrent clairement les avantages de l'application améliorée par rapport à sa version initiale.

En termes de vitesse d'exécution, nous avons constaté une nette amélioration dans l'application finale. Les temps de chargement des pages ont été réduits de manière significative, offrant ainsi une expérience utilisateur plus réactive et fluide. Les requêtes vers la base de données ont également été optimisées, ce qui a permis d'accélérer le traitement des données.

Du point de vue des ressources utilisées, l'application améliorée a également montré des résultats prometteurs. Nous avons réussi à optimiser l'utilisation du processeur, de la mémoire et d'autres ressources système, ce qui a conduit à une consommation réduite et à une meilleure efficacité globale.

En ce qui concerne la stabilité et la fiabilité, l'application améliorée a montré une résistance accrue aux erreurs et aux dysfonctionnements. Les problèmes identifiés dans la version initiale ont été résolus, réduisant ainsi les risques de pannes et de comportements inattendus.

En résumé, le comparatif de performance entre l'application initiale et l'application améliorée démontre clairement les progrès réalisés. Les améliorations apportées ont permis d'optimiser la vitesse d'exécution, d'utiliser de manière plus efficace les ressources système, et d'améliorer la stabilité et la fiabilité de l'application finale.

4.1 Application initiale

Page	Temps de chargement (ms)	Mémoire consommée (mb)
Accueil	489	28
Connexion	660	32
La liste des tâches	724	34
La liste des utilisateurs	957	36

4.2 Application finale

Page	Temps de chargement (ms)	Mémoire consommée (mb)
Accueil	284	22
Connexion	289	24
La liste des tâches	324	24
La liste des utilisateurs	417	24

4.3 Conclusion

On constate une diminution du temps de chargement et de la mémoire consommée. Suite à l'optimisation du projet, on constate une amélioration globale du temps de chargement. En comparant les profils avec l'application initiale/finale, on constate un gain important de performance dans la rapidité d'affichage des pages sans consommation supplémentaire de mémoire significative.

5. Actions menées sur le projet ToDo&Co

Les actions suivantes ont été menées sur le projet initial pour :

• Montée de version majeur et de ses dépendances pour l'application.

• Correction des différents points relevés dans l'état des lieux initial de l'application afin de réduire sa dette technique et de pérenniser ses futures évolutions.

• Correction des d'anomalies et implémentation des nouvelles fonctionnalités demandées dans le cahier des charges.

• Écriture d'une série de tests unitaires et fonctionnels afin de garantir le bon fonctionnement de l'application.

• Mise en place d'un environnement de développement facilitant l'analyse de la qualité du code et de la performance de l'application.