

Audit de Performance et de Qualité

1. Introduction

1.1 Contexte du projet

To Do & Co est une application de gestion de tâches, qui comporte notamment tout un système d'authentification et de sécurité.

Cet audit a pour objectif de mettre en avant les améliorations de performance et de qualité qui ont été apportées à la demande du client.

1.2 Méthodologie

Pour réaliser l'audit de la performance nous avons utilisé le profiler de symfony qui permet de récupérer les informations nécessaires sur la performance d'une application. Ces dernières sont, le Total Execution Time, qui permet de savoir en milliseconde le temps total d'exécution de l'application pour réaliser l'action demandée, ensuite nous avons le Peak Memory Usage, qui permet de savoir la mémoire serveur utilisée pour réaliser l'action demandée, et enfin il y a l'Execution Timeline, qui elle permet de mettre en avant à la fois le Total Execution Time et les différents composants exécutés, ce qui permet de savoir quel composant prend trop de temps à être exécuté.

Pour réaliser l'audit de qualité nous avons utilisé Symfony Insight, un outil qui permet d'analyser le code d'une application symfony depuis un repository git, et qui en fonction de cette analyse nous met en avant les différents défauts de qualité présents dans l'application.

2. État Initial

2.1 Performance

Page	Total Execution Time	Peak Memory Usage
Login Page	802ms	14.5MB
Landing Page	403ms	17.2MB

Page	Total Execution Time	Peak Memory Usage
Landing Page (after Login)	430ms	17.2MB
Listing Tasks Page	369ms	17.5MB
Listing Tasks Page (after deleting one)	405ms	17.5MB
Listing Tasks Page (after toggle one)	371ms	17.5MB
New User Page	439ms	20.8MB
New task Page	764ms	21.0MB

2.2 Qualité du Code

Après l'analyse sur le code initial du projet, Symfony Insight a détecté 7 problèmes critiques, 2 problèmes majeurs et 6 problèmes mineurs. Les différents problèmes sont : une incapacité à l'installation des dépendances, qui engendre une impossibilité au démarrage de l'application, il y avait aussi des dépendances avec des faille de sécurité connue etc..


4. État Final

4.1 Performance

Page	Total Execution Time	Peak Memory Usage
Login Page	89ms	4.0MB
Landing Page	144ms	4.0MB
Landing Page (after Login)	219ms	4.0MB
Listing Tasks Page	178ms	4.0MB
Listing Tasks Page (after deleting one)	142ms	6.0MB
Listing Tasks Page (after toggle one)	129ms	6.0MB
New User Page	167ms	4.0MB
New task Page	249ms	4.0MB

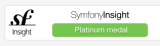
4.2 Qualité du Code

Après avoir mis à niveau l'application, j'ai pu relancer une analyse Symfony Insight. Dans cette analyse j'ai récupéré plusieurs erreurs mineures et une majeure. J'ai donc pu corriger les différents problèmes un par un, ou les ignorer pour certaines déjà corrigées, sans que Symfony Insight ne puisse s'en apercevoir.



Stats
Lines of code: **2,418**
Nb of suggestions: **0**

Last commit
[Merge pull request #21 from Davidouu/optimize-code-to-fit-symfonyinsight-godd-practices](#) by david.didnik 17 days ago.



Changes: no new suggestions!


No suggestion Fixed Ignored (2) Stats


Awesome!

Despite a very thorough analysis, SymfonyInsight couldn't find a single suggestion in the whole codebase of this project .

This is very rare, and is worthy of the Platinum medal.

Congratulations to all the developers of this project for such a high quality!

 david.didnik
57 commits

 david didnik
11 commits

5. Comparaison Avant/Après

5.1 Analyse de la Performance

Page	Total Execution Time (Avant)	Total Execution Time (Après)	Différence
Login Page	802ms	89ms	713ms
Landing Page	403ms	144ms	259ms
Landing Page (after Login)	430ms	219ms	211ms
Listing Tasks Page	369ms	178ms	191ms
Listing Tasks Page (after deleting one)	405ms	142ms	263ms
Listing Tasks Page (after toggle one)	371ms	129ms	242ms
New User Page	439ms	167ms	272ms
New task Page	764ms	249ms	515ms

Page	Peak Memory Usage (Avant)	Peak Memory Usage (Après)	Différence
Login Page	14.5MB	4.0MB	10.5MS
Landing Page	17.2MB	4.0MB	13.2MS
Landing Page (after Login)	17.2MB	4.0MB	13.2MS
Listing Tasks Page	17.5MB	4.0MB	13.5MS
Listing Tasks Page (after deleting one)	17.5MB	6.0MB	11.5MS
Listing Tasks Page (after toggle one)	17.5MB	6.0MB	11.5MS
New User Page	20.8MB	4.0MB	16.8MS
New task Page	21.0MB	4.0MB	17.0MS

6. Conclusion

6.1 Résumé des Améliorations

Suite aux optimisations effectuées, l'application a connu une amélioration significative du niveau de performance et du niveau de la qualité du code. Le temps d'exécution des pages a été drastiquement réduit, avec des gains de

plusieurs centaines de millisecondes, et la consommation de mémoire a été grandement réduite. En parallèle, le code a été refactorisé pour respecter les bonnes pratiques modernes, et des tests unitaires et fonctionnels ont été ajoutés pour garantir la robustesse de l'application.

6.2 Recommandations Futures

Pour maintenir ces améliorations, il est recommandé de continuer à surveiller les performances de l'application régulièrement, notamment en utilisant des outils comme le profiler Symfony. De plus, il est important de poursuivre le travail sur la qualité du code, en effectuant des revues de code systématiques et en utilisant des outils d'analyse comme Symfony Insight pour détecter et corriger les problèmes de manière proactive.