

AUDIT DE QUALITE DU CODE ET DE PERFORMANCE DE L'APPLICATION

SOMMAIRE

1 – Comparatif entre la première version du projet et la deuxième

1.1 - Versions de PHP et Symfony utilisés.

1.2 - Analyse de la qualité du code avec Codacy et phpcs.

1.3 - Mesure des performances avec Blackfire

2 – Axes d'améliorations envisagés

2.1 - Sur la partie front

2.2 - Sur le code

2.3 - php 8

2.4 - mise en place du cache

1 – Comparatif entre la première version du projet et la deuxième

1.1 Versions de PHP et Symfony utilisées

V1

La version de Symfony de l'application est la 3.1.6.

Cette version a été mise en ligne en mai 2016 et maintenue jusqu'en janvier 2017.

Le projet tourne très bien sur la version 5.6 de PHP.

V2

Le projet a été upgradé sur la dernière version de Symfony maintenue et stable à savoir la 5.2.

Latest Stable Release

5.2.6

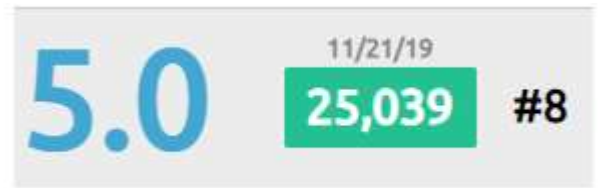
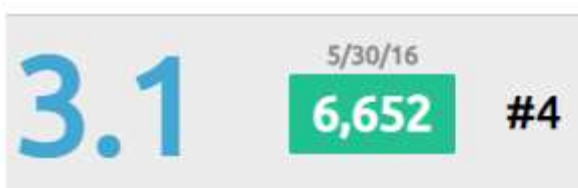
- First released in November 2020.
- **Recommended for most users.**
- Includes the latest features.
- It's easier to upgrade to newer versions.

Cette version du projet tourne sur la version 7.3 de PHP.

Il sera intéressant d'upgrader le projet vers la prochaine LTS (version qui sera maintenue le plus longtemps) de Symfony qui sera la 5.4.

Amélioration :

Passer sur la version 5 de Symfony couplée à la version 7.3 de PHP a permis, comme on peut le voir sur la capture suivante, d'obtenir une meilleure note au benchmark php. Les performances de l'application sont donc accrues.



La version 7.3 de PHP quant à elle apporte son lot de nouveautés comme le typage des variables et assure d'avoir la dernière version de php maintenu et bénéficier des correctifs de sécurité.

1.2 Analyse de la qualité du code avec Codacy et phpcs

Des analyses de la qualité du code ont été effectuées avec Codacy.

V1



Le projet a obtenu la note de C.

L'analyse ne montre pas de soucis particuliers dans le code de l'application. On note surtout des remontées sur le code interne de Symfony.

Issues breakdown



V2

Le projet obtient également la note de C.

On note qu'il y'a moins d'issues à traiter que sur la première version notamment sur la partie Security.



Des analyses sur le respect des normes psr-12 ont été réalisées avec l'outil Phpcs.

L'analyse du code de la deuxième version du projet montre des points à corriger.

```
FILE: ...rs\admin\Desktop\OC_Projet8\src\Controller\DefaultController.php
-----
FOUND 5 ERRORS AFFECTING 5 LINES
-----
```

```
FILE: ...s\admin\Desktop\OC_Projet8\src\Controller\SecurityController.php
-----
FOUND 8 ERRORS AND 2 WARNINGS AFFECTING 9 LINES
-----
```

```
FILE: ...Users\admin\Desktop\OC_Projet8\src\Controller\TaskController.php
-----
FOUND 29 ERRORS AND 4 WARNINGS AFFECTING 24 LINES
-----
```

```
FILE: ...Users\admin\Desktop\OC_Projet8\src\Controller\UserController.php
-----
FOUND 15 ERRORS AND 7 WARNINGS AFFECTING 16 LINES
-----
```

```
FILE: C:\Users\admin\Desktop\OC_Projet8\src\Form\UserType.php
-----
FOUND 12 ERRORS AFFECTING 11 LINES
-----
```

```
FILE: C:\Users\admin\Desktop\OC_Projet8\src\Form\TaskType.php
-----
FOUND 7 ERRORS AFFECTING 7 LINES
-----
```

Amélioration :

Les diverses remontées de l'analyse Codacy ainsi que ceux de l'analyse Phpcs pourront être corrigés afin d'avoir un code de qualité, répondant aux standards et ainsi être plus compréhensible par la communauté des développeurs travaillant sur le projet.

1.3 Mesure des performances de l'application avec Blackfire

Des mesures de la performance de l'application ont été effectuées avec le logiciel Blackfire.

Voici un tableau comparatif entre la première version et la deuxième version du projet.

Pour les deux versions, la mesure des métriques a été réalisée en environnement de production avec une optimisation de l'autoloader de composer comme le préconise blackfire.

ROUTES	VERSION 1		VERSION 2			
	Temps d'exécution (ms)	Mémoire (mb)	Temps d'exécution (ms)		Mémoire (mb)	
homepage	303	15.7	253	↘16.5%	16	↗1.91%
task_list	303	15.9	258	↘14.85%	16.2	↗1.89%
task_create	325	17.9	312	↘4%	18.7	↗4.47%
task_edit	328	17.8	318	↘3.04%	18.6	↗3.91%
task_toggle	276	14.8	225	↘18.48%	13.5	↘8.78%
task_delete	306	15.8	293	↘4.25%	18.7	↗18.35%
user_list	301	15.7	251	↘16.61%	16.3	↗3.82%
user_create	404	19.8	356	↘11.88%	20.3	↗2.53%
user_edit	847	17.8	336	↘60.33%	19.4	↗8.99%

Amélioration

On observe une **baisse des temps d'exécution** sur l'ensemble des routes passées au crible sur la version 2 d'environ **-19.70%** de moyenne pour l'ensemble de l'application mais une augmentation de la **mémoire consommée de +4.12%** en moyenne. Cette faible augmentation s'explique par l'enrichissement du Framework en fonctionnalités.

2 – Axes d'améliorations envisagées

2.1 Sur la partie front de l'application

Séparer la liste des tâches terminées des tâches en cours. Actuellement, il y a une page qui affiche toutes les tâches sans distinction et l'autre qui n'affiche que les tâches terminées. Il serait intéressant de n'afficher que les tâches en cours sur la première page. Ces dernières mélangées avec les tâches terminées n'offrent que peu d'intérêt et permettrait un affichage plus rapide de la page.

Mise en place une pagination ou un scroll infini. Sur les pages listant les tâches ou les utilisateurs, il serait intéressant de mettre en place un système de pagination ou de scroll infini et ainsi n'affichait les items que par petit paquet en fonction du besoin via des appels ajax.

Mise en place de moteurs de recherche. La mise en place de moteurs de recherche sur la date de création, la date d'échéance ou même maintenant qu'une tâche est reliée à un utilisateur permettrait de n'afficher que les tâches dont l'utilisateur a besoin et ainsi optimiser l'ergonomie.

2.2 Sur le code

Les principales remontées concernant le respect des psr de l'outil phpcs ont été corrigées.

De même que le projet obtient la note de B maintenant sur Codacy.

2.3 PHP 8

PHP 8 est disponible depuis la fin de l'année 2020 et apporte son lot d'optimisations et de nouvelles fonctionnalités.

Voici quelques-unes des nouveautés de cette version qui pourraient être intéressantes :

- PHP JIT pour Compilation Just In Time. Le compilateur JIT traduira l'opcode en code machine pour une exécution directe dans le CPU. Peu intéressante pour notre application, le gain porterait surtout sur l'apprentissage automatique, le rendu 3d, 2d et l'analyse des données. J'en parle quand même ici car selon la RFC de JIT un effort supplémentaire est prévu pour améliorer JIT pour les applications de la vie réelle.

- Constructor Property Promotion, permet de déclarer plus simplement les propriétés d'une classe grâce à une nouvelle syntaxe.

- Une nouvelle expression de correspondance (match) en plus de switch.

- des nouvelles fonctions

- etc ...

2.4 Mise en place d'un système de gestion du cache

Symfony intègre un composant permettant la mise en place d'un système de cache. Le but ici pour notre application serait de mettre en cache les résultats des différentes requêtes sur la base de données afin de ne pas réinterroger cette dernière à chaque appel mais d'accéder directement aux résultats mis en cache.

Les différents caches auront une durée de validité et seront effacés à la fin de ce délai ou en cas de modification des données enregistrées. Un nouveau sera alors recréé au premier appel à la base.

Le but ici est d'accroître les performances de l'application en réduisant les délais d'affichage des pages.