

Audit de qualité du code
&
performance de l'application

1) Version de Symfony de base

La version de Symfony du projet utilisée est la 3.1.

D'après la documentation de symfony, elle n'est plus maintenue :



Symfony 3.1 is no longer maintained. Consider upgrading to Symfony 6.1.

Status: UNMAINTAINED	Source Code
Latest patch version: 3.1.10	Documentation
Released on: May 2016	MIT License
End of bug fixes: January 2017	Changelog
End of security fixes: July 2017	New features

2) Dette technique de l'application

- Ancienne version de Symfony (v 3.1)
- Ancienne version de Doctrine (v 2.5)
- Ancienne version Bootstrap (v 3.3.7)
- Architecture de l'application
- Absence de contrainte de validation
- Absence de test unitaire et de test fonctionnel
- Revoir la fonctionnalité de sécurité de symfony
- Pages d'erreurs non personnalisées
- Absence de Voter

3) Plan pour réduire la dette technique

- Migration vers Symfony 6.0.8
- Migration vers Doctrine 2.12.2
- Migration vers Bootstrap 5.2.0
- Réorganisation de l'architecture de l'application
- Ajout des contraintes de validation
- Ajout des tests unitaire et fonctionnel
- Modification de la fonctionnalité de sécurité de symfony

- Ajout des pages d'erreur personnalisées
- Ajout Voter

4) Audit de qualité

Pour avoir un code de qualité, il faut respecter les règles suivantes :

- Respecter les PSR (PHP Standards Recommendations)
- Respecter le principe SOLID

Afin, de respecter à maximum les PSR. J'ai utilisé un composant de composer qui est `php_codesniffer`.

Voici une analyse de qualité du code via Codacy :

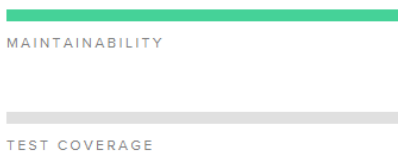


Selon Codacy, la note globale du site est « A », c'est la meilleure note obtainable. Dans la globalité, la qualité du code est de très bonne qualité.

Voici une analyse de la maintenabilité de l'application via CodeClimate :

Breakdown

116 FILES



Codebase summary

MAINTAINABILITY

A 0 mins

TEST COVERAGE



Repository stats

CODE SMELLS

0

DUPLICATION

0

OTHER ISSUES

0

Selon Code Climate, la maintenabilité de l'application est optimale, car il n'y a pas de mauvaises pratiques (Code Smells), duplications et autres problèmes. Ce qui représente les meilleures notes possibles à avoir sur son site.

5) Audit de performance

L'analyse de performance a été faite avec Blackfire et en environnement de production.

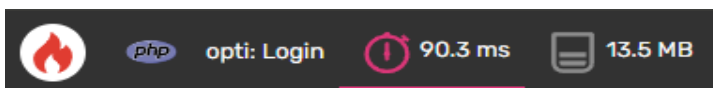
L'analyse de Blackfire donne : le temps de chargement du site, la mémoire utilisée pour son chargement, les fonctions utilisées et son nombre d'appels. Chaque fonction est détaillée avec son temps d'exécution et la mémoire utilisée. Un schéma est aussi consultable pour suivre le plan d'exécution du site, avec une couleur spécifique pour les fonctions les plus gourmandes. Les tests ici présents, sont faits avec la licence gratuite de blackfire, ce qui limite l'analyse du site.

Se connecter « **/login** » :

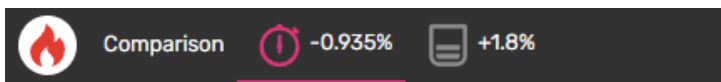
- Sans optimisation



- Avec optimisation

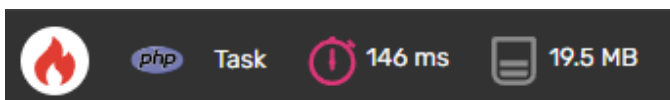


- Comparaison

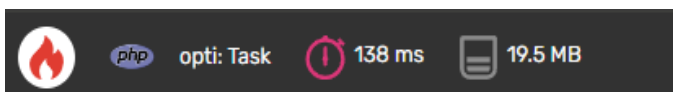


Une tâche « **/tasks/{id}** » :

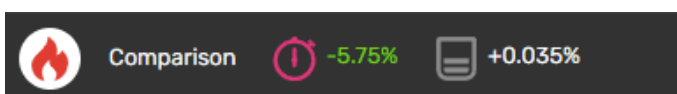
- Sans optimisation



- Avec optimisation

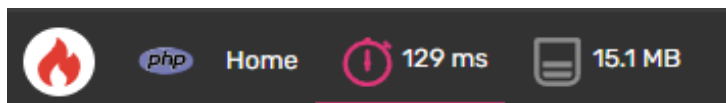


- Comparaison

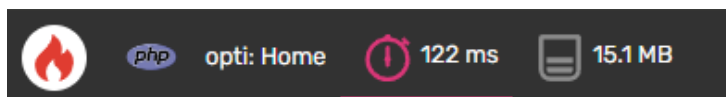


Page d'accueil « / » :

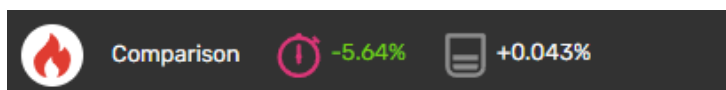
- Sans optimisation



- Avec optimisation



- Comparaison



Liste des tâches « /tasks » :

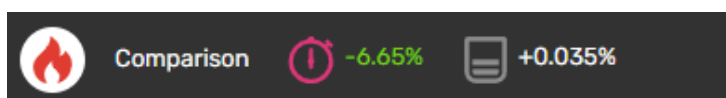
- Sans optimisation



- Avec optimisation



- Comparaison



6) Optimisation

Pour optimiser cette partie de l'application il faut exécuter une simple commande dans le terminal à la racine du projet :

```
composer dump-autoload --no-dev -a
```

Cette commande sert à mettre en cache les classes utiles à l'application, toutefois, si de nouvelles classes sont ajoutés il faudra absolument relancer cette même commande !

- **--no-dev** exclut les classes qui ne sont nécessaires que dans l'environnement de développement (c'est-à-dire les dépendances require-dev et les règles autoload-dev)
- **--classmap-authoritative** ou **-a** crée un mappage de classe pour les classes compatibles PSR-0 et PSR-4 utilisées dans votre application, et empêche Composer d'analyser les classes qui ne se trouvent pas dans le mappage de classe