

# Audit Qualité et Performance

## TODO & CO v2

### **1 - Améliorations appliquées sur l'application**

La première étape de ce projet fut d'upgrader Symfony (Version de base : 3.1). J'ai donc choisi de passer sur un Symfony 5.3, ceci m'a demandé un peu plus de travail mais le but étant de ne pas avoir à migrer également les nouvelles features que j'aurais développé à l'avenir en passant uniquement à un symfony 4.

#### **Pourquoi Symfony 5 ?**

En premier lieu, et c'est déjà à mes yeux une raison suffisante, Symfony 5 est actuellement la version de Symfony la mieux maintenue. Mais il faut aussi bien voir que migrer vers Symfony 5 permet de bénéficier d'améliorations de performance inédites. Ces améliorations ne sont plus intégrées dans les versions précédentes, y compris la version LTS Symfony 4.4.

Enfin, passer à Symfony 5, c'est avoir accès aux dernières fonctionnalités, mais aussi aux simplifications inédites identifiées par la communauté. Par exemple, Symfony 5.1 introduit un nouveau mécanisme plus flexible et plus simple, donc moins risqué et moins coûteux à maintenir, pour gérer les accès des utilisateurs.

## 2 - Audit de qualité du code (Codacy)



Grâce à l'analyse de Codacy, j'ai pu observer la qualité de mon code, celui-ci ayant obtenu un badge B, ce qui est tout à fait honorable, mais à l'avenir, nous pourrions améliorer celui-ci, avec l'aide notamment aux issues renvoyées par codacy.

## 3 - Analyse PHPUnit

		Lines		Code Coverage		Classes and Traits	
				Functions and Methods			
Total		99.43%	175 / 176	97.92%	47 / 48	88.89%	8 / 9
Controller		98.89%	89 / 90	92.86%	13 / 14	75.00%	3 / 4
DataFixtures		100.00%	26 / 26	100.00%	2 / 2	100.00%	1 / 1
Entity		100.00%	51 / 51	100.00%	29 / 29	100.00%	2 / 2
Repository		100.00%	9 / 9	100.00%	3 / 3	100.00%	2 / 2

Suite à la rédaction de mes tests, phpunit me renvoie donc un récapitulatif satisfaisant, en effet, plus de 90% de code coverage qu'il faudra maintenir, notamment grâce à ma documentation technique pour les développeurs qui intégreront le projet.

#### 4 - Blackfire

Route	Avant Opti	Après Opti
/	208ms / 10.9Mb	92.1ms / 8.25Mb
/login	177ms / 9.29Mb	62.4ms / 7.02Mb
/users/create	238ms / 14.6Mb	108ms / 10.4Mb
/users	192ms / 11Mb	84.5ms / 8.08Mb
/tasks/create	278ms / 14.4Mb	99.1ms / 9.58Mb
/tasks	260ms / 11.2Mb	106ms / 8.66Mb
/task/{id}/edit	244ms / 14.5Mb	100ms / 9.66Mb

Nous pouvons, grâce à Blackfire, observer une nette amélioration des performances de l'application.