

Audit de code et de performance



I. Outils utilisés

- **SymfonyInsight** : Outil d'analyse en ligne de qualité du code PHP spécialisé Symfony.
- **Symfony Profiler** : Outil de développement inhérent au framework Symfony, utilisé ici l'analyse de performance à travers sa collecte de métriques à chaque requête.

II. Constat de l'existant

1. Framework Symfony

L'application fut originellement développée sous Symfony 3.1. Cette version n'est officiellement plus maintenue depuis 2017. De nombreuses dépendances sont également dépréciées. Il est donc nécessaire d'effectuer une migration de version vers la version actuelle, Symfony 6.3.

2. PHP

Le projet utilise la version 5.6 de PHP sortie il y a maintenant 9 ans. Aujourd'hui obsolète et possédant des vulnérabilités de sécurité, le projet doit passer au minimum sur PHP 8.1. De nombreuses corrections de bugs, optimisations et fonctionnalités furent ajoutées depuis PHP 5.6.4. Un historique est disponible [ici](#).

3. Anomalies

a. Anomalies signalées

- Actuellement, une tâche n'est pas rattachée à l'utilisateur qui la crée.
- Les tâches existantes sans auteur devraient être considérées comme « anonymes ».
- La création et modification de compte doit permettre d'assigner un rôle d'utilisateur ou d'administrateur.

b. Anomalies relevées

- Absence de lien vers la page d'accueil
- Liens vers des pages d'administration (création de compte, liste utilisateur) affichés à tous les utilisateurs sans contrôle d'accès
- Plusieurs liens présents sur la page d'accueil attachés à aucune route
- Absence de certains messages flash à la suite de soumission de formulaires
- Absence de token CSRF dans le formulaire de connexion
- Méthodes HTTP non spécifiées sur les routes

4. Analyse du code par SymfonyInsight










SymfonyInsight est un outil en ligne d'analyse automatisée de code PHP spécialisé sur Symfony. Il remonte les différents problèmes que peut avoir l'application analysée, notamment sur la performance, la sécurité et la maintenabilité.

The screenshot displays the SymfonyInsight web interface. On the left sidebar, a score of 28/100 is shown with a medal icon, and a recommendation to reach 4.3 days to get the Platinum Medal. Below this is a search bar and a 'Severity' filter showing 4 Critical, 2 Major, 6 Minor, and 2 Info issues. A 'Risk' section lists 1 Data leak, 5 Productivity, 1 Reliability, 5 Reputation, 2 Security, and 0 Uninsured. The 'Developer' section shows 6 Collective, 7 Saro0h, and 1 Ylevo. The 'Stats' section indicates 2,536 lines of code and 16 suggestions. The 'Last commit' section shows a commit by Ylevo 15 days ago. The main content area shows a summary of changes (+16 suggestions, 6 critical, 2 major, 6 minor, 2 info) and a message about 2 upgrade issues. Below this is a list of suggestions, including 'Your project must not rely on dependencies with known security issues', 'Your project must not expose sensitive infrastructure configuration', 'Your project must use a custom favicon instead of the default one', 'Your project should use Doctrine migrations', 'Your project should not contain "FIXME" comments', 'Your project should not contain commented code', 'Your project should not use an .htaccess file', 'Web applications should contain a site.webmanifest file', 'Your project composer.json file should not raise warnings', and 'Text files should end with a valid new line character.' Each suggestion has a 'Read doc' link and a severity label (Critical, Major, Minor, Info).

Une note de 28/100 fut attribuée à l'application et aucune médaille. On peut remarquer plusieurs infractions de niveau critique, notamment de sécurité sur ses dépendances dépréciées.

5. Analyse de performance pré-correctifs

Relevée des métriques de temps d'exécution et de consommation de mémoire par Symfony Profiler sur différentes routes de l'application :

200	@ homepage	149 ms	17.2 MiB	 7	 15 in 4.75 ms
200	@ login	125 ms	14.8 MiB	 7	 5 in 0.91 ms
200	@ task_list	171 ms	17.5 MiB	 7	 27 in 8.68 ms
200	@ task_create	256 ms	24.0 MiB	 1  8	 77 in 23.63 ms
200	@ user_list	163 ms	17.5 MiB	 7	 15 in 5.39 ms
200	@ user_create	174 ms	18.8 MiB	 1	
200	@ user_edit	188 ms	21.2 MiB	 1  8	 15 in 6.32 ms

III. Correctifs et réponses apportées

1. Mise à jour de Symfony, PHP et Bootstrap

Une migration vers Symfony 6.3 fut effectuée ainsi qu'une mise à jour des différentes dépendances à l'aide de Composer. Le projet requiert désormais au minimum PHP 8.1. Bootstrap fut également mis à jour en passant de la version 3.3.7 à la version 5.3.1.

2. Anomalies corrigées

- Chaque tâche nouvellement créée est désormais rattachée à l'utilisateur authentifié, son auteur.
- Les anciennes tâches sont considérées comme « anonymes ».
- Il est maintenant possible de sélectionner le rôle de l'utilisateur à sa création et modification.
- Un lien vers la page d'accueil a été ajouté.
- Les liens vers les pages d'administration sont uniquement affichés si l'utilisateur connecté a le rôle d'administrateur.
- Les routes ont leur méthode HTTP correctement renseignée.
- Des messages flash notifiant l'utilisateur du succès de la soumission d'un formulaire ont été rajoutés.
- Un token CSRF est désormais présent dans le formulaire de connexion.

3. Nouvelles fonctionnalités

a. Contrôle d'accès et autorisations

- Les tâches sont désormais supprimables uniquement par leur auteur. La seule exception étant les tâches dites « anonymes » qui peuvent être supprimées par les administrateurs.
- Les pages d'administration (gestion des utilisateurs) sont maintenant restreintes aux administrateurs du site.

Le contrôle d'accès a été mis en place dans le fichier **security.yaml** situé dans le dossier **config/packages** en utilisant l'option **access_control**. Plus d'informations à ce sujet sont disponibles dans la documentation sur l'authentification située dans le dossier **docs**.

Un **Voter** Symfony a été créé pour gérer les permissions vis à vis des tâches.



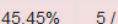


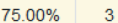


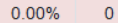


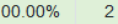


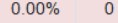


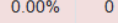
b. Mise en place de tests automatisés

De nombreux tests avec PHPUnit ont été implémentés pour valider et assurer le bon fonctionnement de l'application à long terme. Les entités ont été testées à l'aide de tests unitaires et des tests fonctionnels sont utilisés pour le comportement des différentes routes de l'application.

Dans un terminal à la racine du projet, vous pouvez exécuter ces tests en utilisant la commande : **php vendor/bin/phpunit**

Pour générer le rapport de couverture des tests : **vendor/bin/phpunit --coverage-html public/test-coverage**

Le rapport est ensuite disponible dans le dossier **public/test-coverage**.

	Code Coverage							
	Lines			Functions and Methods			Classes and Traits	
Total		89.16%	148 / 166		82.35%	42 / 51		45.45% 5 / 11
■ Controller		97.26%	71 / 73		83.33%	10 / 12		75.00% 3 / 4
■ Entity		92.31%	36 / 39		92.86%	26 / 28		0.00% 0 / 2
■ Form		100.00%	34 / 34		100.00%	2 / 2		100.00% 2 / 2
■ Repository		14.29%	2 / 14		33.33%	2 / 6		0.00% 0 / 2
■ Security		83.33%	5 / 6		66.67%	2 / 3		0.00% 0 / 1
📄 Kernel.php		n/a	0 / 0		n/a	0 / 0		n/a 0 / 0

Legend

Low: 0% to 50% Medium: 50% to 90% High: 90% to 100%

4. Analyse de performance post-correctifs

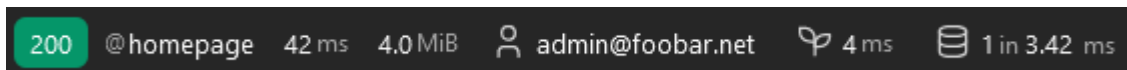
Des optimisations ont été effectuées en suivant la [documentation officielle](#) de Symfony. Configurer Opcache notamment améliore sensiblement le temps d'exécution et la consommation mémoire de toutes les routes de l'application, comme le montre le comparatif ci-dessous.

Page d'accueil

Avant :

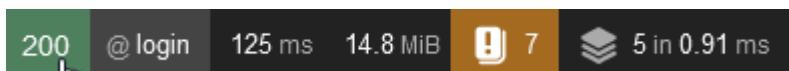


Après :



Formulaire de connexion

Avant :



Après :

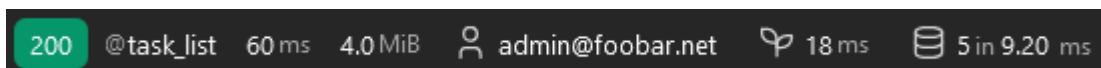


Liste des tâches

Avant :



Après :



Formulaire de création de tâche

Avant :

200	@task_create	256 ms	24.0 MiB	1	8	77 in 23.63 ms
-----	--------------	--------	----------	---	---	----------------

Après :

200	@task_create	64 ms	6.0 MiB	1	admin@foobar.net	12 ms	1 in 3.22 ms
-----	--------------	-------	---------	---	------------------	-------	--------------

Liste des utilisateurs

Avant :

200	@user_list	163 ms	17.5 MiB	7	15 in 5.39 ms
-----	------------	--------	----------	---	---------------

Après :

200	@user_list	52 ms	4.0 MiB	admin@foobar.net	4 ms	2 in 3.75 ms
-----	------------	-------	---------	------------------	------	--------------

Formulaire de création d'utilisateur

Avant :


200	@user_create	174 ms	18.8 MiB	1
-----	--------------	--------	----------	---

Après :

200	@user_create	60 ms	4.0 MiB	1	admin@foobar.net	8 ms	1 in 1.47 ms
-----	--------------	-------	---------	---	------------------	------	--------------

5. Analyse SymfonyInsight post-correctifs

En éliminant les dépréciations, en réduisant la dette technique et en appliquant les bonnes pratiques PHP & Symfony, l'analyse SymfonyInsight ne trouve aucune correctif supplémentaire à suggérer, et se voit forcée d'accorder la médaille de platine à l'application.



Stats

Lines of code: 844

Nb of suggestions: 0

Changes: -5 suggestions -1 critical -4 major 0 minor 0 info

No suggestion Fixed (5) Ignored

Stats

Excellent!

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!