

AUDIT DE QUALITÉ DU CODE & PERFORMANCE DE L'APPLICATION

Audit de qualité du code

Plusieurs outils ont été utilisés pour tester la qualité du code. Nous allons présenter des analyses de code réalisées sur SymfonyInsight et CodeClimate. Nous avons aussi mis en place des tests unitaires et fonctionnels pour vérifier la qualité du code mais aussi détecter les bugs et dépréciations.


SymfonyInsight & CodeClimate

Les analyses SymfonyInsight et CodeClimate n'ont pas révélé de problèmes majeurs sur le code de l'application.

SymfonyInsight



The screenshot shows the SymfonyInsight dashboard for a project named 'P8-OC-ToDoList #56' by user 'eddess'. The interface includes a sidebar with a 'Kudos!' badge, project statistics (11,496 lines of code, 0 suggestions), and a 'Last commit' section. The main area displays a message about new infrastructure, a summary of changes (-1 suggestion, 0 critical, 0 major, -1 minor, 0 info), and a notification about 15 upgrade issues. At the bottom, there are filters for 'No suggestion', 'Fixed (1)', and 'Ignored', along with a 'Stats' button.

eddess / **P8-OC-ToDoList #56** Analyzed a day ago by eddess, duration: a minute Edit configuration Analyze ▾

<  >

Stats
Lines of code: **11,496**
Nb of suggestions: **0**

Last commit
[Remove commented code](#) by eddesscy a day ago.

 SymfonyInsight
insight  Platinum medal



Your project is running on our new infrastructure!
Builds are now more reliable and faster.
You may need to adapt the `pre_composer_script` and `post_composer_script` options in your configuration. If you filter Git access by IP addresses, you will also need to update those: contact us for more details.
Give us your feedback or report issues at symfony.com/support!

Changes: -1 **suggestion** 0 critical 0 major -1 minor 0 info

SymfonyInsight detected **15 upgrade issues** in your project.
Upgrade your plan to see the details and how to upgrade

No suggestion Fixed (1) Ignored Stats

Your project is running on our new infrastructure!

 Open source ▼ Repositories 

eddesscv/P8-OC-ToDoList ★ Starred

Overview Progress Issues Code Trends Repo Settings

Last master build 1 min ago ↻ Refresh

Breakdown

99 FILES

MAINTAINABILITY

TEST COVERAGE

Codebase summary

MAINTAINABILITY

TEST COVERAGE

Repository stats

CODE SMELLS

DUPLICATION

OTHER ISSUES

Progress report 🔗 Compare on GitHub

Sat, April 09 — Sun, April 17

Maintainability Test Coverage

→ A

src/Security/LoginFormAuthenticator.php was added.

→ B

public/config.php was removed.

→ A

public/app_dev.php was removed.

→ A

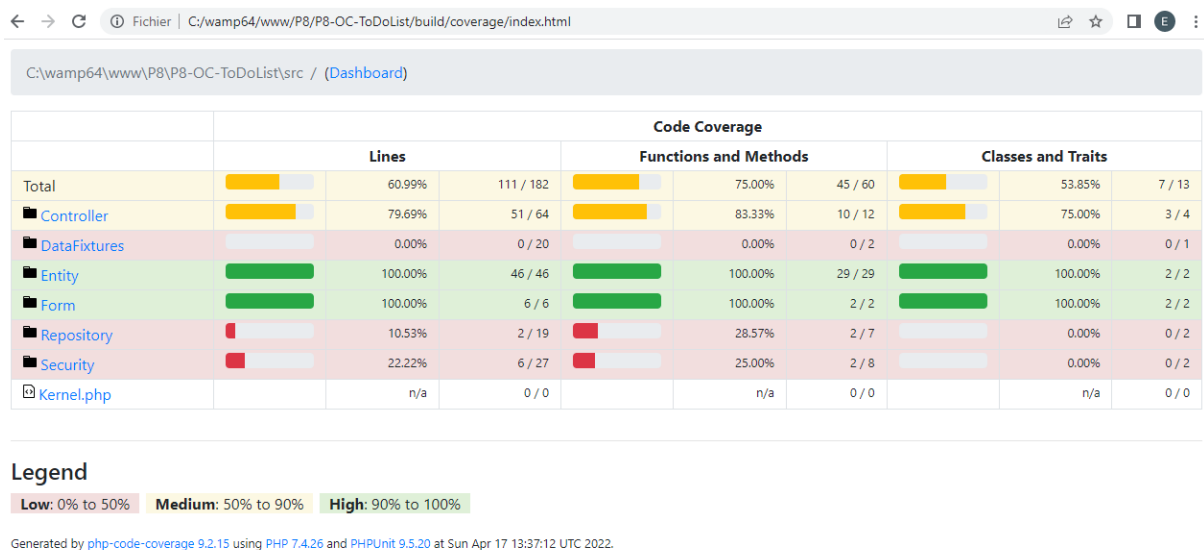
public/app.php was removed.

→ F

public/js/bootstrap.js was removed.

PHPUnit

Concernant PHPUnit, 30 tests ont été réalisés (avec 74 assertions) pour un rapport de couverture de l'application supérieur à 75%.



Conclusion

L'audit de qualité de code des deux projets (initial et amélioré) nous a permis d'observer que la montée de version a eu des bénéfices en terme de qualité de code mais aussi de sécurité. Ce n'est pas pour autant que le projet initial n'était pas sécurisé, mais la version de Symfony 3.1 n'était plus supportée et des failles de sécurité commençaient à apparaître, comme nous l'a rappelé GitHub lors de l'installation du projet.

Audit de performance de l'application

Les outils de développement de Symfony tel que le profiler influencent beaucoup sur les résultats.

Pour une analyse plus précise et détaillée des performances de notre application web nous allons utiliser Blackfire (<https://blackfire.io/>).

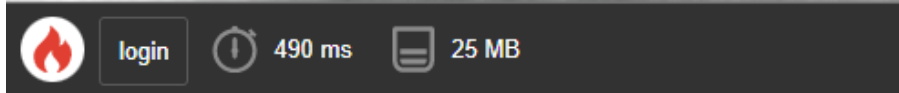
Analyse Blackfire

L'analyse de Blackfire donne : le temps de chargement du site, la mémoire utiliser 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. À savoir: avec la version payante des recommandations d'optimisation sont proposées.

Audit sans optimisation

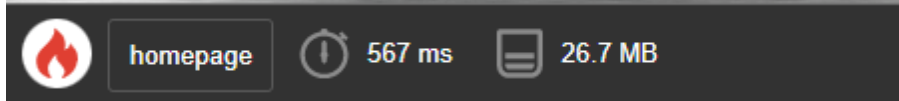
Page de login :

<https://blackfire.io/profiles/0a67ab21-602e-48c6-ac38-983f2e819602/graph>



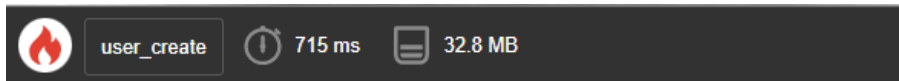
Page d'accueil (après login) :

<https://blackfire.io/profiles/462dd517-8b4f-406f-a125-c963890c15f4/graph>



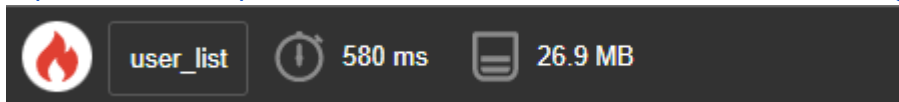
Page de création de nouvel utilisateur :

<https://blackfire.io/profiles/05ab5da6-833c-4d7d-a8af-8b5dd6c2e85a/graph>



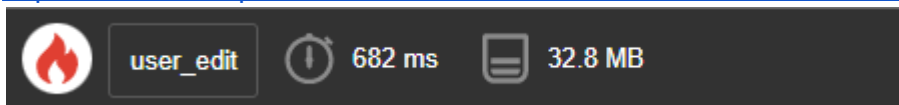
Page de liste d'utilisateurs:

<https://blackfire.io/profiles/3faaaa45-de25-4d19-919a-5934d62833fd/graph>



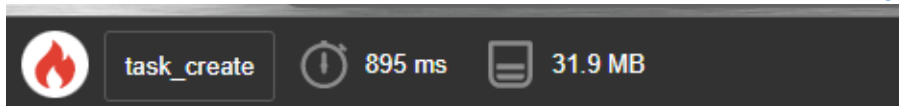
Page de modification utilisateur:

<https://blackfire.io/profiles/b8bd78d5-b252-4ee9-abda-4cd32d6f4959/graph>



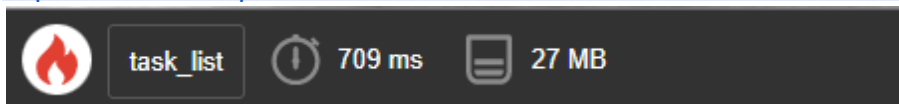
Page de création de nouvelle tâche :

<https://blackfire.io/profiles/7fb24256-1429-46a3-b08c-f48a7f5cb8ce/graph>



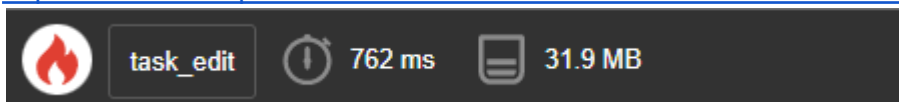
Page de liste de tâches :

<https://blackfire.io/profiles/bb57e88b-66f7-43ec-9958-9e5093e12194/graph>



Page de modification de tâche:

<https://blackfire.io/profiles/ef1c9831-5780-4288-873b-939c7a6be6d7/graph>



Optimisations possibles

Optimisation de l'autoloader de Composer :

<https://getcomposer.org/doc/articles/autoloader-optimization.md>

Installer l'extension C pour Twig :

<https://twig.symfony.com/doc/1.x/installation.html#installing-the-c-extension>

Activation d'OPcache

<https://www.ekino.fr/publications/php-comment-configurer-utiliser-et-surveiller-opcache/>

Configuration de PHP realpath Cache

<https://symfony.com/doc/current/performance.html>

Varnish

https://symfony.com/doc/current/http_cache/varnish.html

L'optimisation réalisée dans cet exemple est celle de l'autoloader de **Composer**.

```
composer dump-autoload --no-dev --classmap-authoritative
```

Attention : l'optimisation ne doit pas être réalisée en environnement de développement !

Note: You **should not** enable any of these optimizations in **development** as they all will cause various problems when adding/removing classes. The performance gains are not worth the trouble in a development setting.

Le passage de votre machine locale à l'environnement de production peut se faire en changeant la variable d'environnement APP_ENV dans le fichier .env.local :

```
APP_ENV=prod
```

Ou vous pouvez utiliser la commande server:prod :

```
$ symfony server:prod
```

N'oubliez pas de le remettre sur dev à la fin de votre session de profilage :

```
$ symfony server:prod --off
```

Pour l'optimisation:

```
composer dump-autoload --no-dev --classmap-authoritative
```

Cette commande sert à mettre en cache les classes utiles à l'application, toutefois, si de nouvelles classes sont ajoutées, 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)

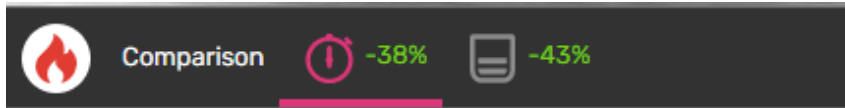
--classmap-authoritative crée un mappage de classe pour les classes compatibles PSR-0 et PSR-4 utilisées dans votre application, et empêche Composer d'analyser les classes qui ne se trouvent pas dans le mappage de classe

Comparaison après une optimisation :

On y observe qu'on gagne un léger gain sur le temps de chargement sur chaque page et utilisation de mémoire .

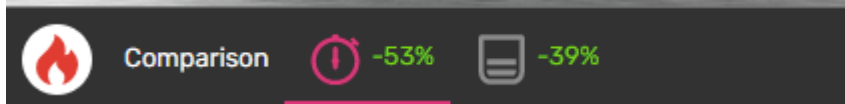
Page d'accueil (après login) :

<https://blackfire.io/profiles/compare/462dd517-8b4f-406f-a125-c963890c15f4...>



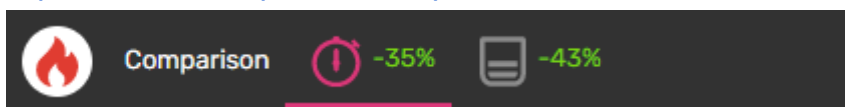
Page de création de nouvelle tâche :

<https://blackfire.io/profiles/compare/7fb24256-1429-46a3-b08c-f48a7f5cb8ce...>



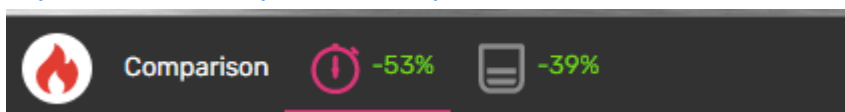
Page de liste d'utilisateurs:

<https://blackfire.io/profiles/compare/3faaaa45-de25-4d19-919a-5934d62833fd...>



Page de modification de tâche:

<https://blackfire.io/profiles/compare/ef1c9831-5780-4288-873b-939c7a6be6d7...>



D'autres recommandations pour améliorer la performance

- Optimisation des images :
 - Compresser les images
 - Privilégier le format SVG et JPEG
- Minifier les scripts
 - CSS
 - JS
- Opter pour Tailwind Css pour remplacer Bootstrap (design plus moderne, et poids plus léger)
- Réduire les requêtes externe
 - Remplacer JQuery par du JS Vanilla
- Utiliser un hébergeur web de qualité et fiable
- Opter pour un outil d'analyse de performance premium (Blackfire Premium)