Rapport qualité Project8_PhpUnit

Introduction :

Pour analyser la qualité du code, Codacy et Blackfire ont été utilisé.

Codacy est un bon logiciel d'analyse gratuit qui va suivre la qualité du code tout au long du développement du projet.

BlackFire est le logiciel principal d'analyse de performance de Symfony. Blackfire va montrer quelles sont les parties du code les moins performantes et suggérer les améliorations à faire afin de gagner en performance.

Codacy :

Rapport Codacy : https://app.codacy.com/project/Beskargam/Project8_PhpUnit/dashboard?bid=11981333

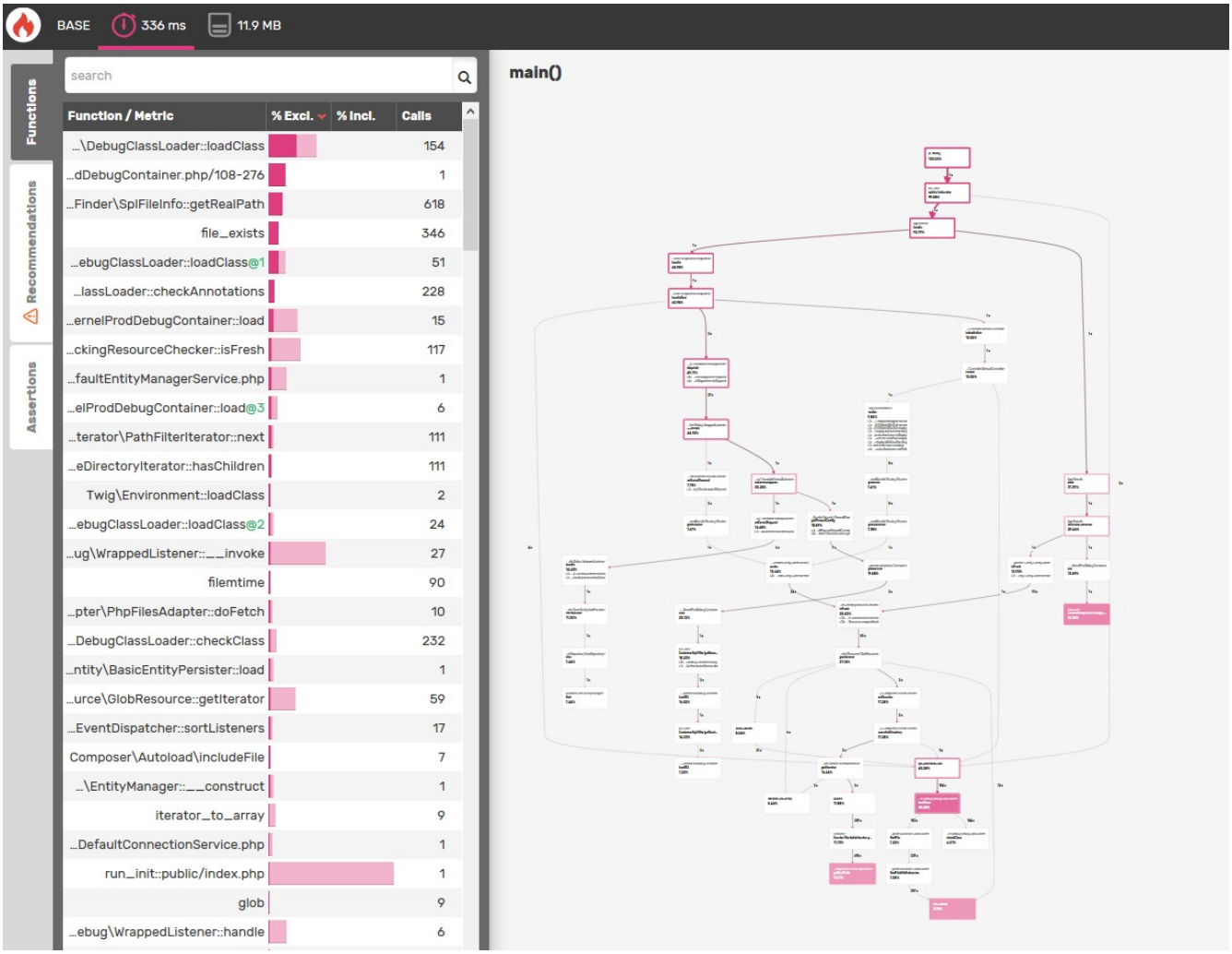
Bien que l'on puisse voir quelques alertes dans le rapport Codacy, celles-ci doivent être ignorées car elles concernent des fonctionnalités importantes au bon fonctionnement de Symfony.

BlackFire :

Blackfire est un outil puissant qui va nous permettre d'améliorer le code afin de gagner en performance. Allons-y, améliorons Composer.

Composer est le manager d'injection de dépendance de Symfony. Il est puissant mais un peu lent.

Voyons cela :



Comme nous pouvons le voir, le Classloader est la partie du code la moins performante. C’est une action de Composer.

Cette fonction de Composer appelle chaque classe du projet afin de pouvoir les utiliser.

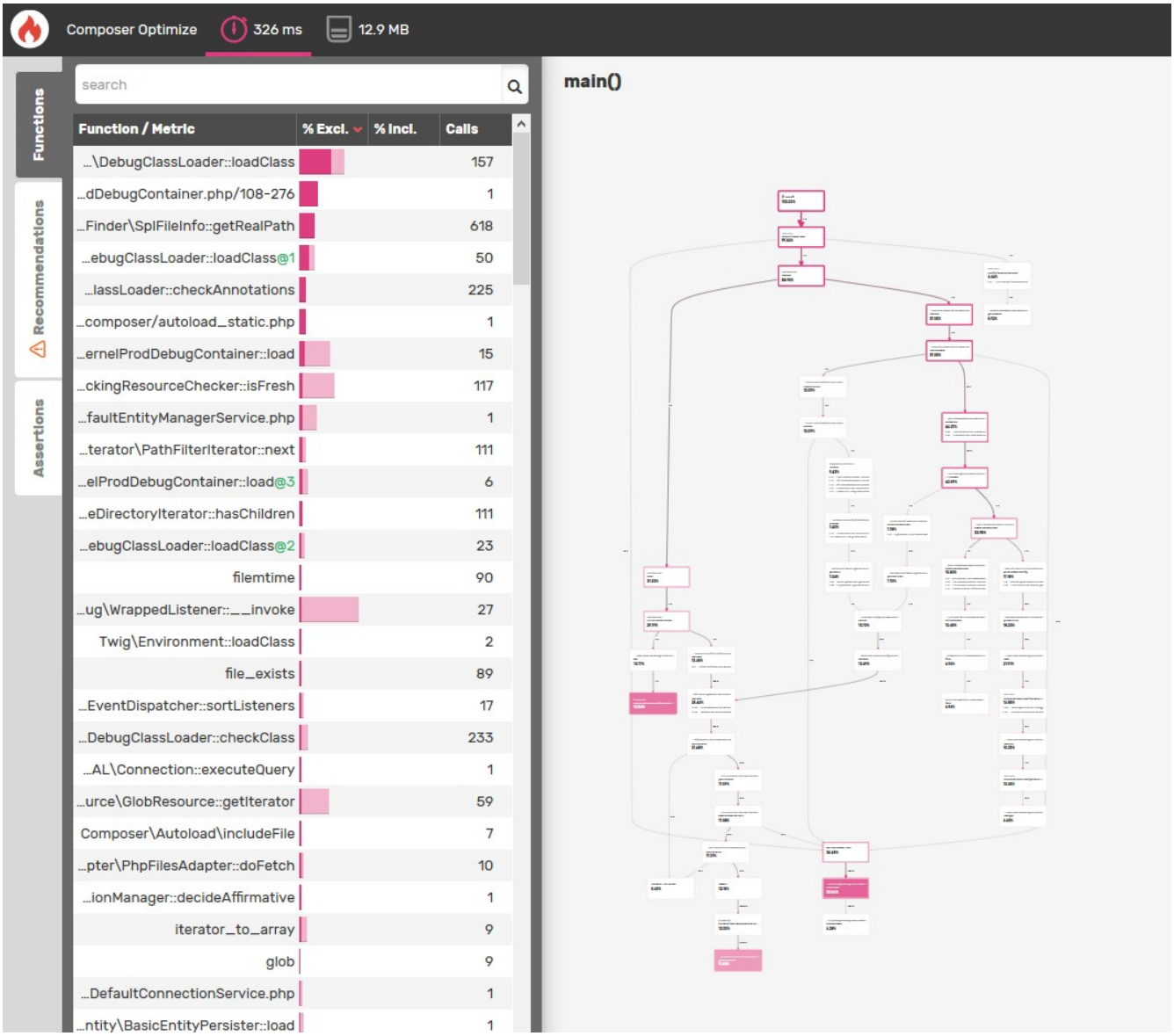
Nous pouvons améliorer cette action en créant un fichier où toutes les classes seront déjà enregistrées. De ce fait, Composer sait exactement dans quel emplacement la classe se trouve et n’aura pas besoin de regarder dans tous les fichiers du projet.

Nous pouvons créer ce fichier avec une simple commande de Composer dans l’invite de commande :

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

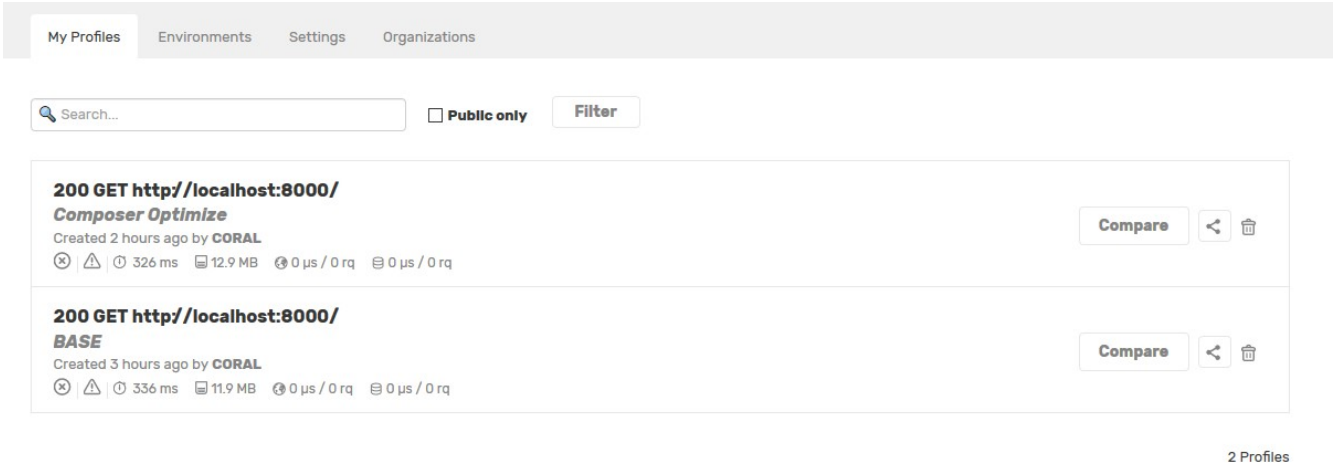
Bien ! Voyons ce que cette commande a amélioré dans notre projet.

Lançons une nouvelle analyse Blackfire :

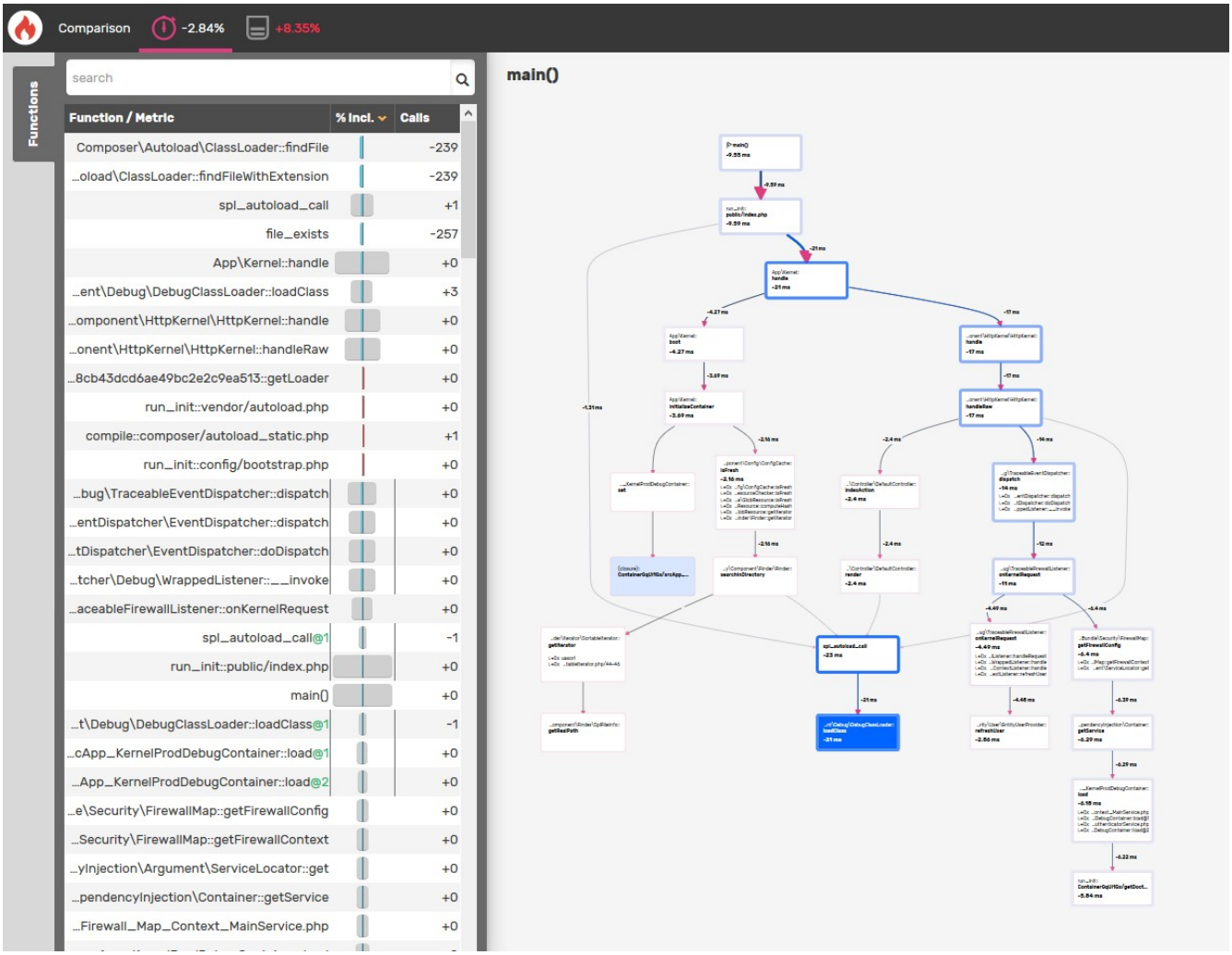


On peut observer une amélioration. Nous avons gagné 10ms lors du chargement de la page.

Peut-être pouvons nous mieux voir les améliorations en comparant cette analyse avec la précédente.



Et, voici les changements :



Super ! Nous avons gagné 2.84 % en vitesse de chargement et nous avons perdu 8.35% de mémoire vive.

Pas d’inquiétudes. En réalité, améliorer la performance d’une application est souvent une histoire de compromis. Gagner en performance mais utiliser plus de mémoire vive.

Sur internet, nous avons besoin de sites internet très rapides. La mémoire vive ne coûte presque rien au niveau matériel pour votre commerce.

Autres :

Nous pouvons améliorer les performances sans même utiliser de logiciel.

Simplement, paginer les tâches au lieu de charger 100-200 tâches en même temps. L’application pourrait charger que 10 ou 20 tâches et paginer le reste des tâches.