

Audit de qualité du code & performance de l'application

Performances entre Symfony 3.1 et 6.4

Symfony 6.4 surpasse largement **Symfony 3.1** en termes de performances. Voici quelques points clés :

A. Vitesse:

- **Chargement des pages:** Symfony 6.4 est environ 2 fois plus rapide que 3.1 grâce à des optimisations du noyau et du système de routage.
- **Exécution des requêtes:** Le nouveau moteur de Twig offre une meilleure compilation et un cache plus efficace, réduisant le temps de traitement des requêtes.

B. Mémoire:

- **Utilisation de la mémoire:** Symfony 6.4 utilise moins de mémoire que 3.1, grâce à une meilleure gestion des dépendances et des objets.
- **Fuites de mémoire:** Les efforts de la communauté ont permis de réduire considérablement les fuites de mémoire dans Symfony 6.4.

C. Fonctionnalités:

- **Nouveau système de cache:** Symfony 6.4 introduit un système de cache plus flexible et performant, basé sur Symfony Cache.
- **Support PHP 8.1:** Symfony 6.4 tire parti des nouvelles fonctionnalités de PHP 8.1 pour offrir des performances accrues.

Qualité du code

A. version initiale

Après avoir récupéré le projet et l'avoir initialisé dans sa version de départ, j'ai effectué une première analyse du code à l'aide de Codacy.

Cet outil nous aide à identifier divers problèmes répartis en plusieurs catégories telles que le **style de code**, la **sécurité**, les **performances**, la **complexité**, et bien d'autres encore.

Voici le premier rapport de codacy avec le projet ToDoList sans aucun changement:

Quality evolution

Issues ? ●	Complexity ? ●	Duplication ? ●	Coverage ?
47 %	2 %	2 %	-

Issues breakdown



On remarque qu'il y a beaucoup de problèmes de sécurité, ainsi que des soucis dans la manière dont le code est écrit.

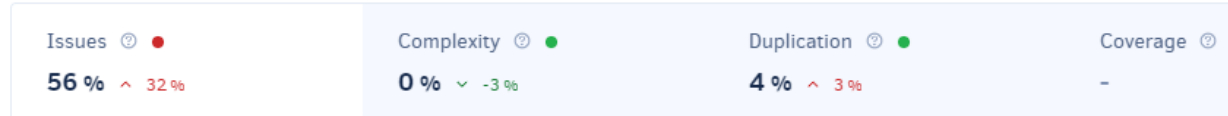
B. Nouvelle version

Après avoir effectué la mise à jour de Symfony de la version 3.1 à la version 6.4 et de PHP de la version 8.3, et après avoir utilisé PHP-CS-Fixer pour améliorer le style du code, voici le rapport de Codacy.

ToDoList B

old-front-phpunit ▾

Quality evolution



Issues breakdown



Nous constatons qu'il n'y a plus d'erreurs de sécurité. Certaines parties du code ne sont pas utilisées, et bien que le style du code ait encore besoin d'améliorations, il a tout de même été amélioré, ce qui m'a permis d'obtenir la note B dans Codacy.

Performances

A. Mesures

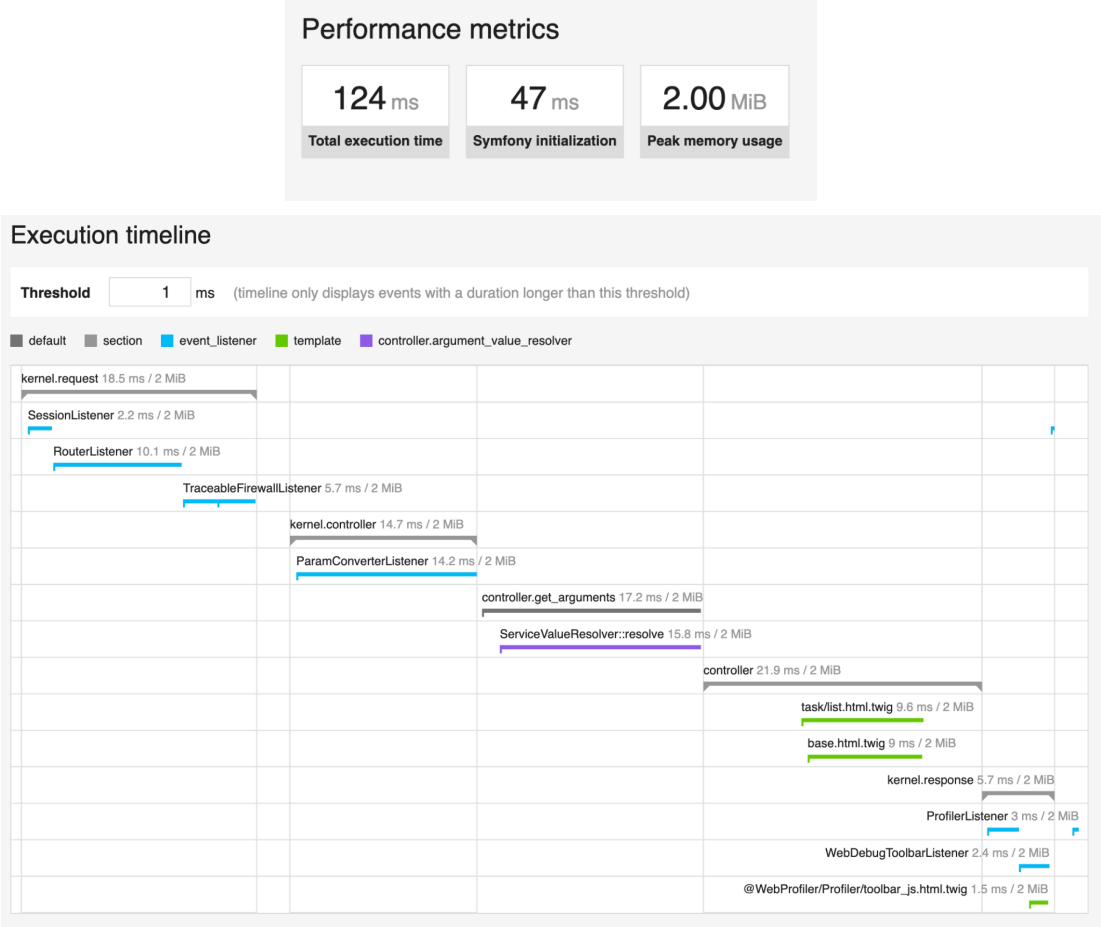
J'ai utilisé la barre d'outils Symfony pour comparer les performances des deux versions de l'application. Cet outil m'a permis d'obtenir des informations détaillées sur les requêtes effectuées, l'utilisation de la mémoire, le temps de chargement total et le temps d'exécution de chaque étape.

B. Optimisations

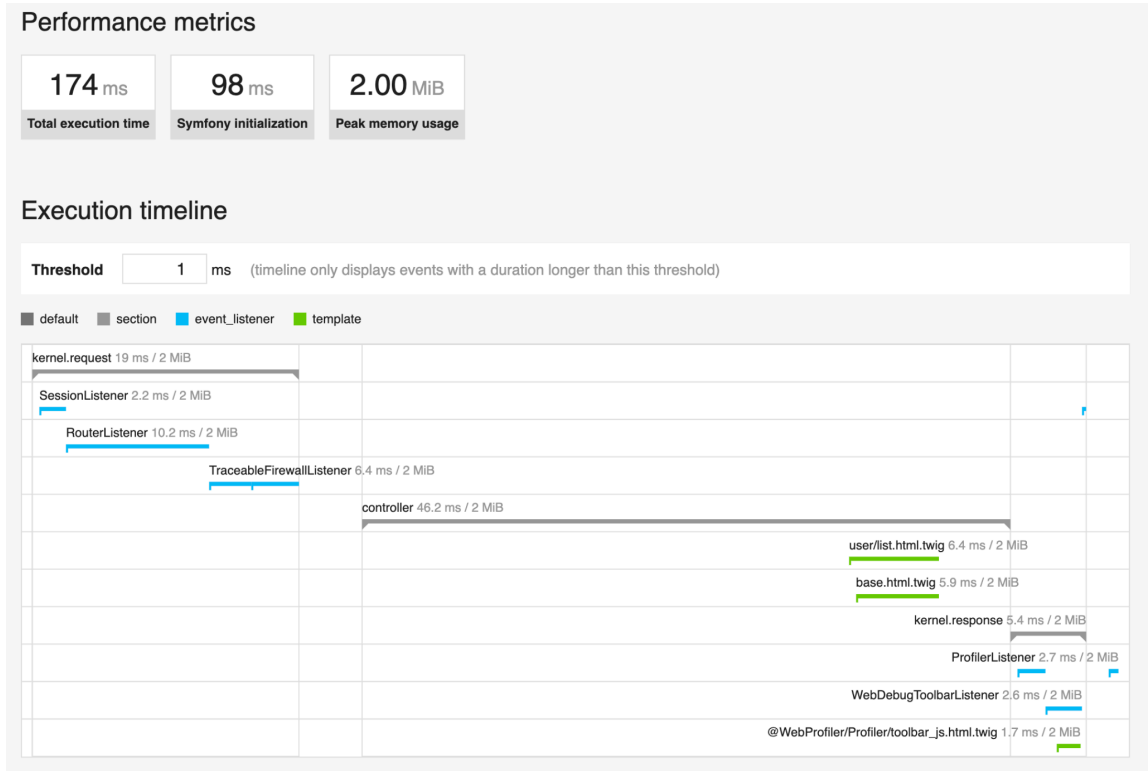
OPCache: J'ai activé l'extension OPCache dans la configuration de PHP. Cette extension améliore les performances en stockant un cache des scripts en mémoire. Cela permet d'éviter de charger et d'analyser le code PHP à chaque requête.

C. Version de base

1. Taks

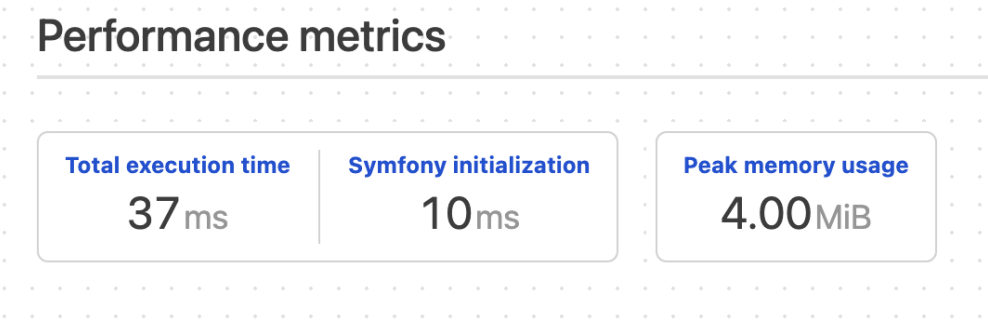


2. Users

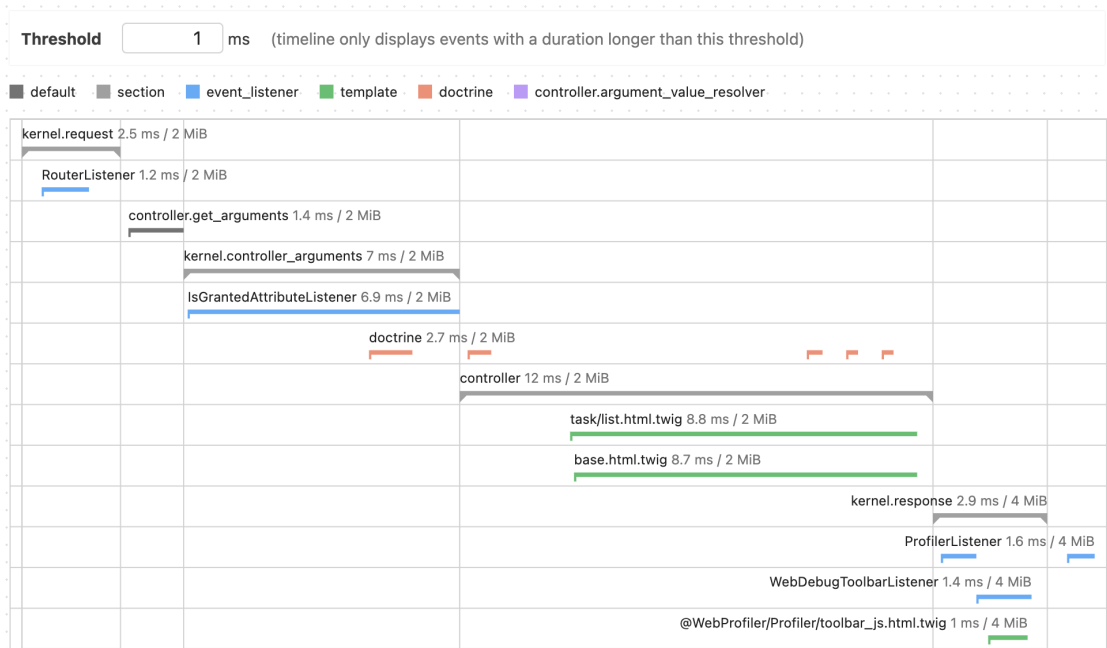


D. Version final

1. Tasks



Execution timeline



2. Users

Performance metrics

Total execution time

22ms

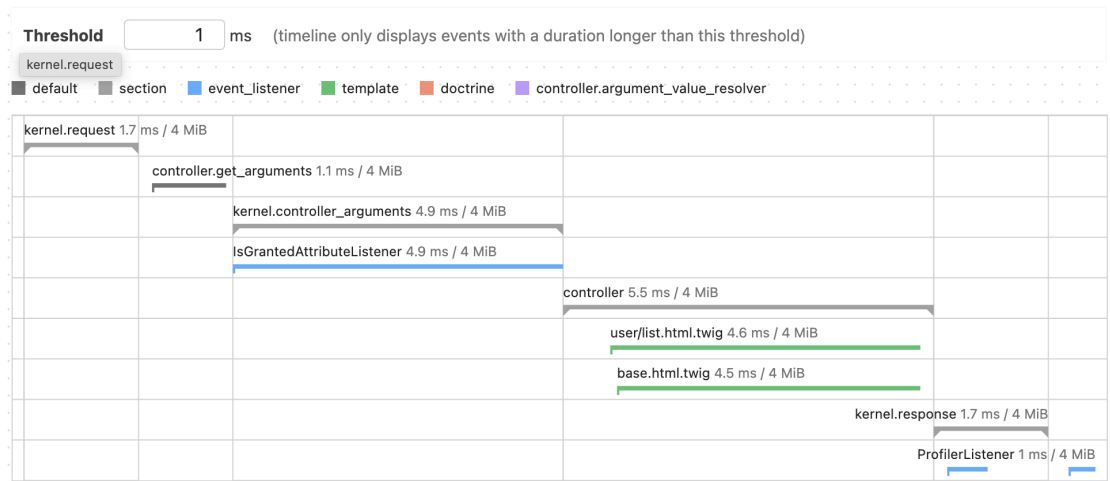
Symfony initialization

7ms

Peak memory usage

4.00MiB

Execution timeline



Les performances globales du projet Symfony ont été bien améliorées.

Le temps total d'exécution pour l'utilisateur a été réduit de 174 ms à 22 ms, et le temps d'initialisation de Symfony a diminué de 98 ms à 7 ms. Cependant, l'utilisation de la mémoire a augmenté entre les versions.

De même, pour la tâche, le temps total d'exécution a été réduit de 124 ms à 37 ms, et le temps d'initialisation de Symfony a diminué de 47 ms à 10 ms. L'utilisation de la mémoire a aussi connu une augmentation entre les versions.