



**MARINE JOURDAN-  
DEVELOPPEUR PHP-SYMFONY**

# AUDIT DE CODE ET DE PERFORMANCE



# ANALYSE SYMFONY INSIGHT

Analyse effectuée via github et l'outil d'analyse en ligne symfony insight.

Effectué durant les différentes évolutions du code pour correspondre aux normes symfony et permettre un partage du projet efficace avec un socle de normes communes

## Évolutions symfonyInsight

- *Evolution de symfony : actuellement en 3.4 évolué en 5.4*
- *Supprimer les commentaires de code*
- *Ajouter le favicon personnalisé*
- *Typier les entités*
- *effectuer des migrations*

## Utilisation de PHP CS-FIXER

PHP CS Fixer est un outil permettant de vérifier et corriger le formatage du code PHP selon le code style défini dans la configuration du projet.

Le style code est une convention qui définit comment doit être écrit le code. Par exemple l'emplacement des accolades des structures de contrôle, des classes, des fonctions, les espaces autour d'un signe égal, etc.

Cela permet de faciliter la lecture du code pour tout le monde.

## Utilisation de symfony profiler

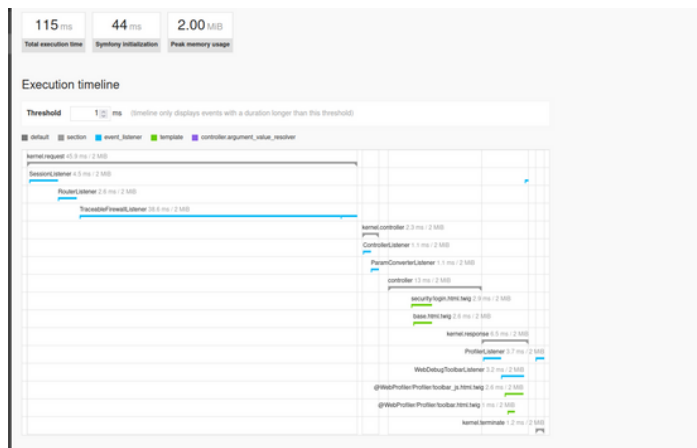
- Faire un point de l'existant
- Obtenir des solutions pérennes à l'amélioration de la performance
- Corriger les anomalies

Audit de code vise à améliorer la qualité de l'application, augmenter ses performances, accélérer l'expérience utilisateur, augmenter la sécurité de l'application, identifier et anticiper les problèmes de sécurité et de performance, permettre au code d'être réutilisable pour tous selon les normes en vigueur.

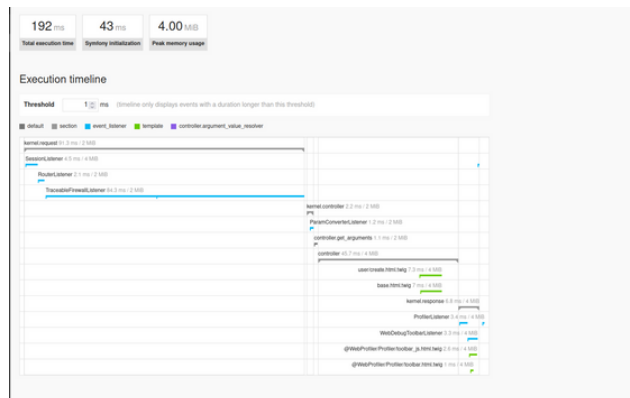
Il permet de collecter de nombreuses informations sur votre application durant son exécution : pour chaque requête utilisateur, le Web Profiler va créer un profil et l'application web fournit une interface soignée pour parcourir les informations de ce profil. Il permet de collecter de nombreuses informations sur votre application durant son exécution : pour chaque requête utilisateur, le Web Profiler va créer un profil et l'application web fournit une interface soignée pour parcourir les informations de ce profil.

- LE TEMPS DE RENDU DE LA PAGE;
- LA CONSOMMATION DE MÉMOIRE ;
- LE NOMBRE D'APPELS AU CACHE ;
- LE TEMPS DE RENDU DES BLOCS TWIG...

## Page login (symfony 5.4)



## Page create (symfony 3.4)

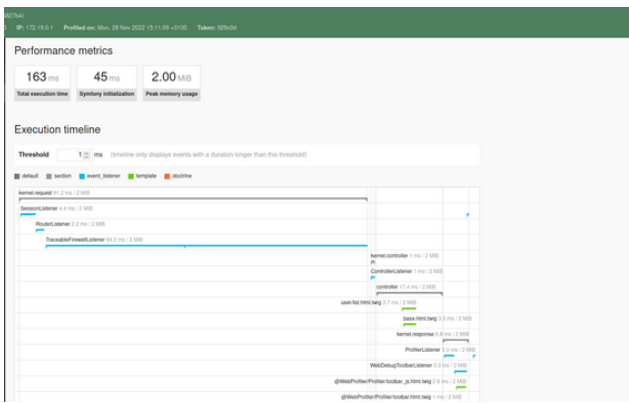


## Projet 8 -ToDoList

## Page users (symfony 3.4)

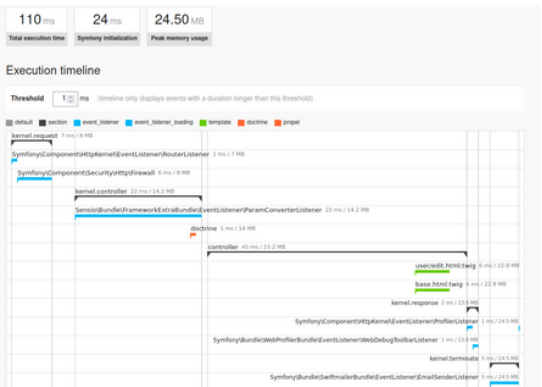


## Page users (symfony 5.4)

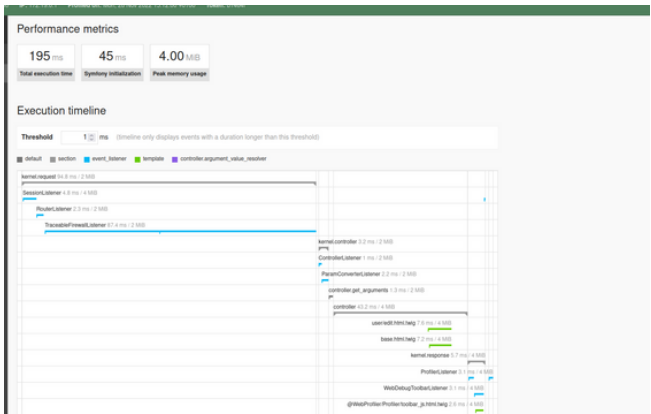


Réponse du kernel réduite  
la mémoire utilisée est elle énormément amoindrie entre les deux versions de symfony.  
Cependant, ici le controller en charge de l'affichage des users est plus rapide

## Page edit/user (symfony 3.4)



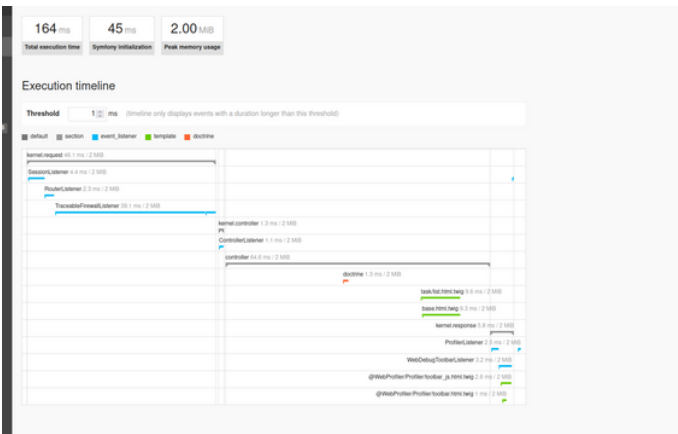
## Page edit/user (symfony 5.4)



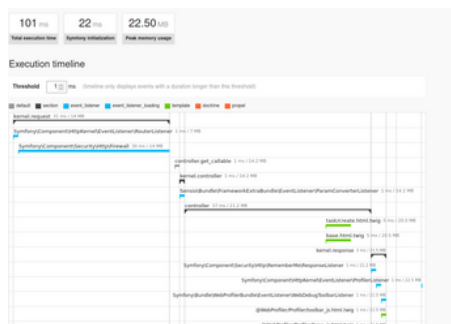
## Page tasks (symfony 3.4)



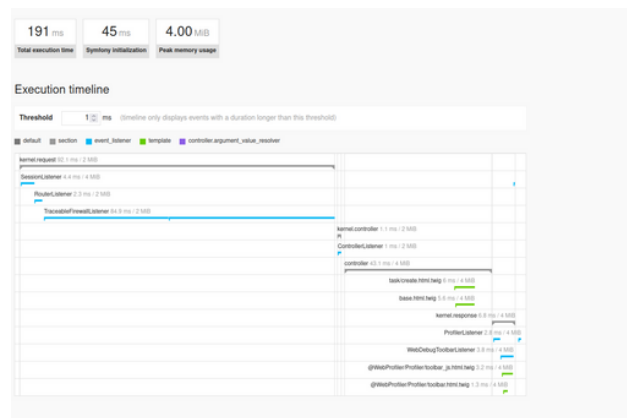
## Page tasks (symfony 5.4)



## Page task/create (symfony 3.4)



## Page task/create (symfony 5.4)



Conclusion du passage de symfony 3.4 à 5.4 : Une évolution notable dans la kernel response, la mémoire utilisée et le chargement des données doctrine.

La version 5 et les bundles installés suite à la migration ont ralenti le temps de chargement du kernel est des dépendances.





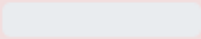

L'application a grossi, le système de gestion des utilisateurs a été mis en place ce qui donne globalement un temps de chargement identique ou plus long de l'affichage plus précisément.

Cependant, la mémoire a été énormément réduite ce qui permet d'avoir un site beaucoup plus léger.






### DÉVELOPPEMENT POSSIBLE:

- OPTIMISER LE CHARGEUR AUTOMATIQUE DE COMPOSER EN MODE PROD  
LE CHARGEUR DE CLASSE UTILISÉ LORS DU DÉVELOPPEMENT DE L'APPLICATION EST OPTIMISÉ POUR TROUVER DES CLASSES NOUVELLES ET MODIFIÉES. DANS LES SERVEURS DE PRODUCTION, LES FICHIERS PHP NE DOIVENT JAMAIS CHANGER, SAUF SI UNE NOUVELLE VERSION DE L'APPLICATION EST DÉPLOYÉE.
- LORS DE LA CRÉATION D'UNE ENTITÉ, IL EST RECOMMANDÉ DE METTRE EN PLACE DES INDEXES SQL AFIN D'ACCÉLÉRER LE TEMPS DE RECHERCHE SUR LES COLONNES DE LA BASE DE DONNÉES.
- RESULT CACHE : CE CACHE GARDE EN MÉMOIRE LE RÉSULTAT DES REQUÊTES ENVOYÉES À LA BASE DE DONNÉES AFIN DE NE PAS AVOIR À L'INTERROGER UNE SECONDE FOIS PAR LA SUITE
- SÉLECTIONNER PLUSIEURS ENTITÉS DANS UNE REQUÊTE DOCTRINE

# RAPPORT DE COUVERTURE DE CODE SYMFONY 5.4 (APRÈS MIGRATION)






		Lines
Total		76.13%
Controller		76.92%
Entity		90.20%
Form		100.00%
Migrations		0.00%
Security		64.00%
AppBundle.php		n/a

TAUX DE COUVERTURE GLOBALE:  
76,13%






		Lines	
Total		87.69%	57 / 65
DefaultController.php		100.00%	1 / 1
SecurityController.php		66.67%	4 / 6
TaskController.php		82.86%	29 / 35
UserController.php		100.00%	23 / 23

Legend

## LES CONTROLLEURS




		Lines	
Total		90.20%	46 / 51
 Task.php		85.00%	17 / 20
 User.php		93.55%	29 / 31

## LES ENTITÉS

		Lines	
Total		100.00%	6 / 6
 TaskType.php		100.00%	2 / 2
 UserType.php		100.00%	4 / 4

Legend

## LES FORMULAIRES

		Lines	
Total		64.00%	16 / 25
 LoginFormAuthenticator.php		64.00%	16 / 25

Legend

## LES POINTS DE SECURITÉ (LOGIN)