

Audit de Qualité du Code et de Performance

Application *ToDo & Co*

A - Contexte et Objectifs

Dans le cadre de la pérennisation de l'application *ToDo & Co*, les fondateurs ont exprimé le besoin de réaliser un **état des lieux de la dette technique existante**.

L'objectif de cet audit est donc de :

- évaluer la qualité du code existant,
- mesurer les performances de l'application,
- identifier les points bloquants,
- puis mesurer l'impact des améliorations apportées.

L'audit a été mené selon deux axes, conformément à l'énoncé :

1. **Qualité du code**
2. **Performance de l'application**

Chaque analyse est appuyée par des **métriques objectives** issues d'outils d'audit et de profiling, avec une comparaison **avant / après modification**.

B - Outils utilisés

Audit de qualité du code

- **Codacy**
 - Analyse statique
Détection des problèmes de maintenabilité
 - Identification des mauvaises pratiques

Audit de performance

- **Symfony Profiler**
 - Temps de réponse
 - Nombre de requêtes SQL
 - Consommation mémoire
 - Détection des requêtes N+1

Ces outils ont été utilisés **avant toute modification**, puis **après refactorisation**, afin de mesurer précisément l'évolution du projet.

1 - Audit de Qualité du Code

A - État initial (avant modifications)

L'analyse initiale met en évidence une **dette technique importante**, caractérisée par :

- Absence de tests automatisés
- Logique métier concentrée dans les contrôleurs
- Duplication de code dans les templates Twig
- Validation des données incomplète
- Manque de séparation des responsabilités

Métriques Codacy – AVANT

Ces résultats montrent un code fonctionnel, mais **difficilement maintenable et risqué à long terme (code de rang C selon codacy)**.



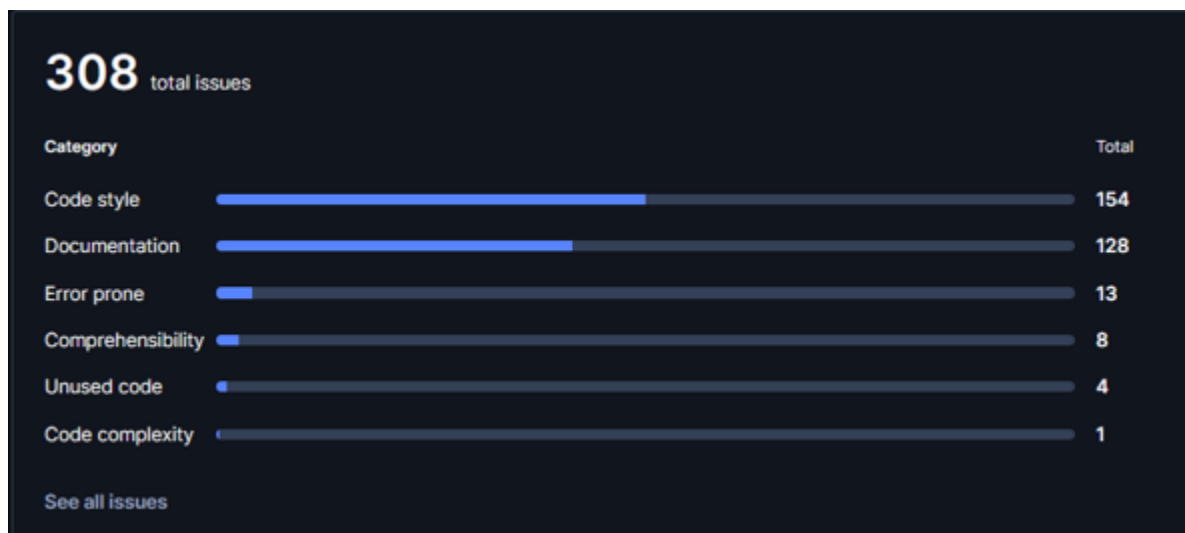
Améliorations apportées

Les actions suivantes ont été mises en place pour réduire la dette technique :

- Refactorisation des contrôleurs
- Introduction de **FormTypes** avec contraintes de validation
- Création de services dédiés
- Réduction de la duplication via des partials Twig
- Mise en place de tests fonctionnels

B - État final (après modifications)

Le code est maintenant **plus modulaire, plus facilement testable et prêt à évoluer. RANG B selon codacy.** Aucune erreur de type High ou Critical.



2 - Audit de Performance

A - État initial (avant modifications)

L'analyse via le Symfony Profiler a mis en évidence plusieurs problèmes de performance :

- Requêtes **N+1** sur les relations Doctrine
- Temps de réponse élevés sur certaines pages
- Consommation mémoire excessive
- Absence de cache HTTP

diverses erreurs rencontrées lors du test de l'application sur la partie inscription.

An exception occurred while executing 'INSERT INTO user (username, password, email) VALUES (?, ?, ?)' with params ["Charles", "\$2y\$13\$nc4NhjipPGabV7Lin8BXbugVjwXDx5qFDpV1gYj8QXAFhKjVDJqN.", "charles_test@gmail.com"]:

SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry 'Charles' for key 'UNIQ_8D93D649F85E0677'

500 Internal Server Error - UniqueConstraintViolationException
2 linked Exceptions:

- [Exception](#)
- [PDOException](#)

”

[3/3] **UniqueConstraintViolationException**: An exception occurred while executing 'INSERT INTO user (username, password, email) VALUES (?, ?, ?)' with params ["Charles", "\$2y\$13\$nc4NhjipPGabV7Lin8BXbugVjwXDx5qFDpV1gYj8QXAFhKjVDJqN.", "charles_test@gmail.com"]:

SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry 'Charles' for key 'UNIQ_8D93D649F85E0677' +

[2/3] **Exception**: SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry 'Charles' for key 'UNIQ_8D93D649F85E0677' +

[1/3] **PDOException**: SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry 'Charles' for key 'UNIQ_8D93D649F85E0677' +

Error: Maximum execution time of 120 seconds exceeded

500 Internal Server Error - FatalErrorException

”

Stack Trace

```
1. in vendor\doctrine\dbal\lib\Doctrine\DBAL\Driver\PDOStatement.php at line 118 -
    115.     {
    116.         try {
    117.             return parent::execute($params);
    118.         } catch (PDOException $exception) {
    119.             throw Exception::new($exception);
    120.         }
    121.     }
```

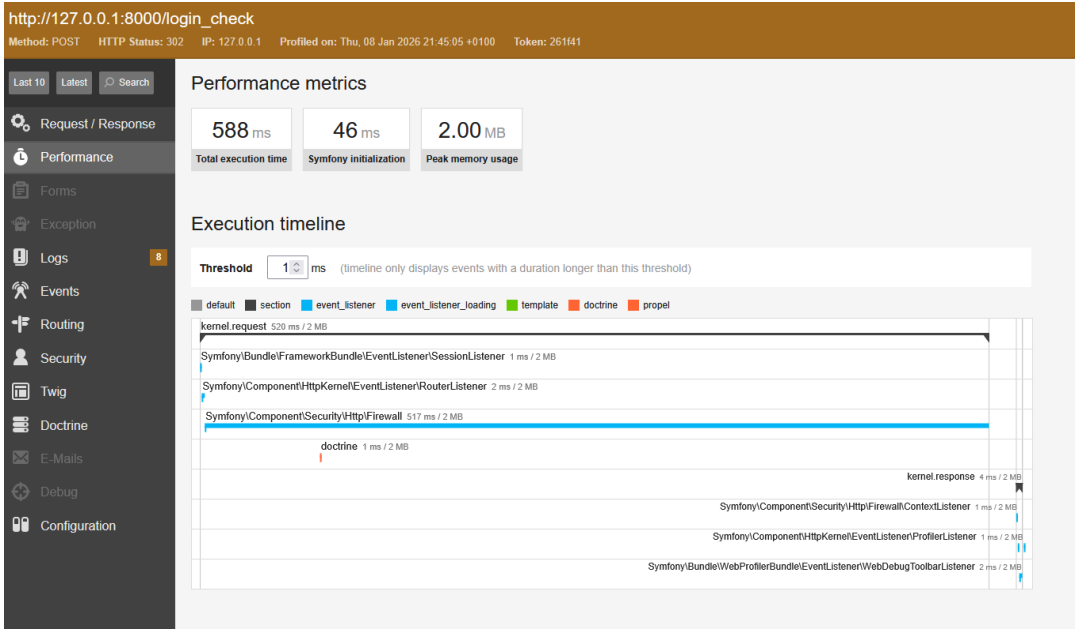
Screens des métriques du profiler –

AVANT :

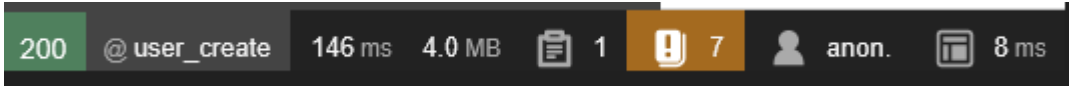
Login page:



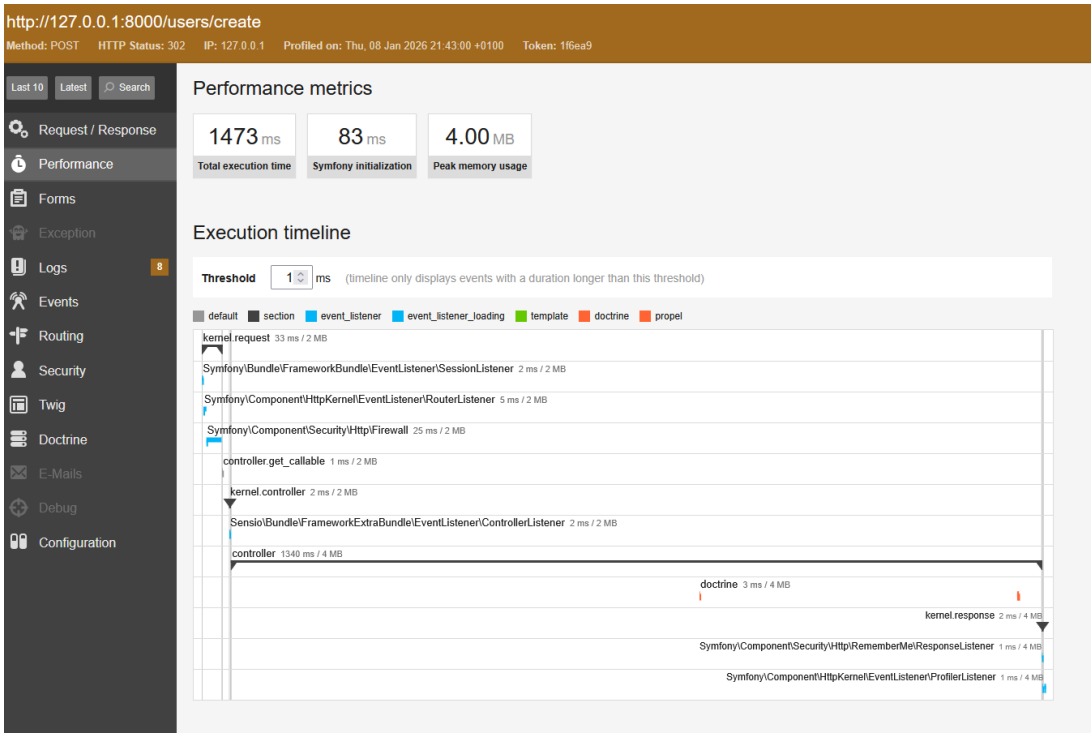
Login request :



Register page :



Register request :



Task page :

200

@ task_list

139 ms

2.0 MB

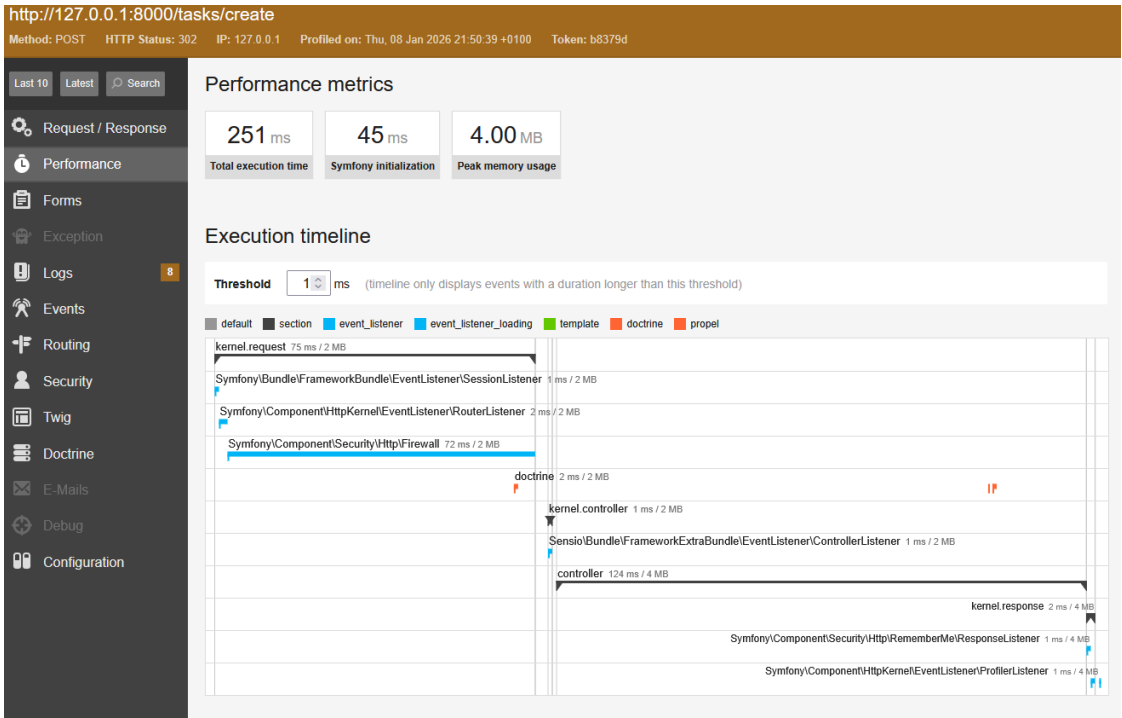
 7

 Tester

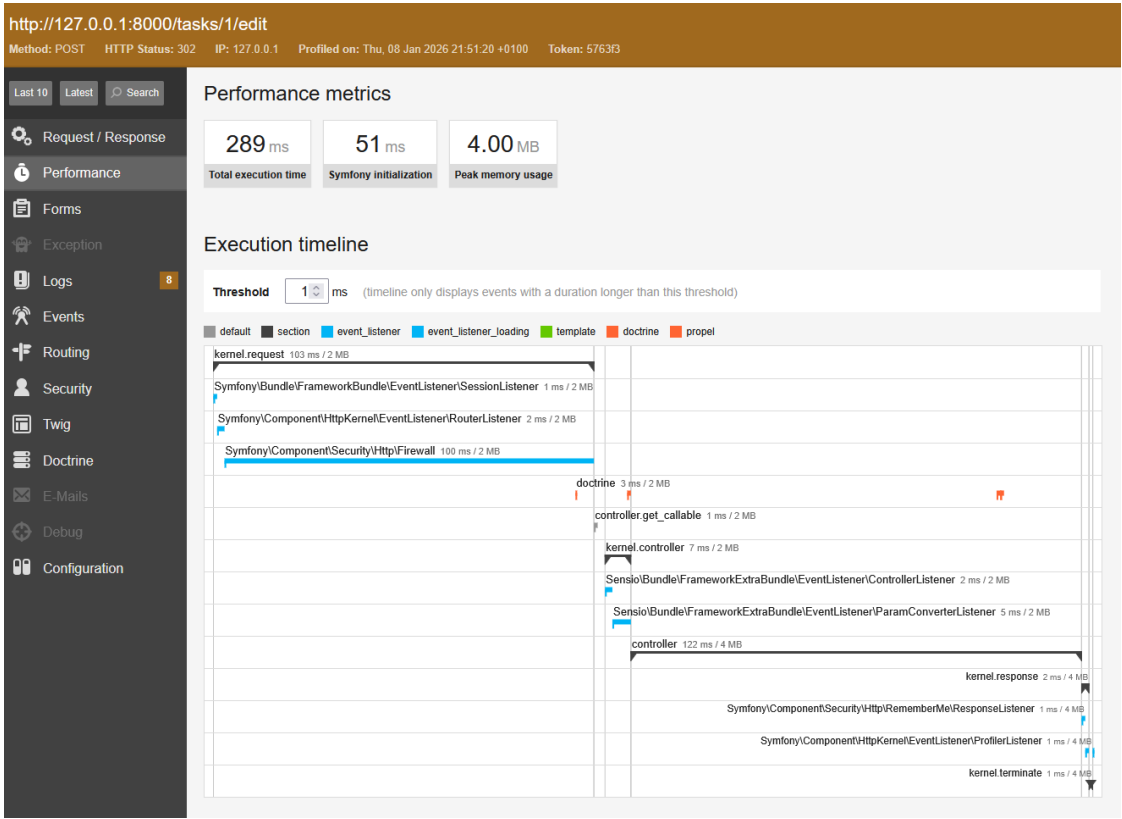
 3 ms

 2 in 1.08 ms

Task create:



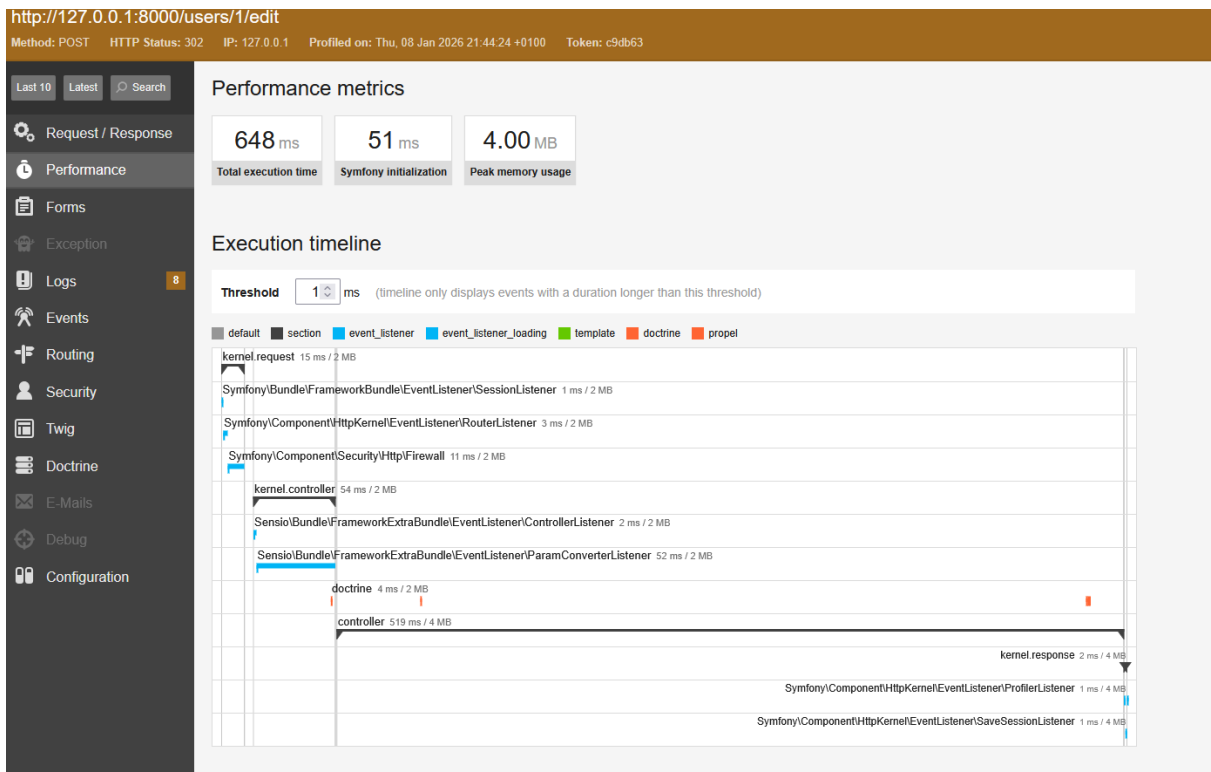
Task edit :



users page request :



user edit :

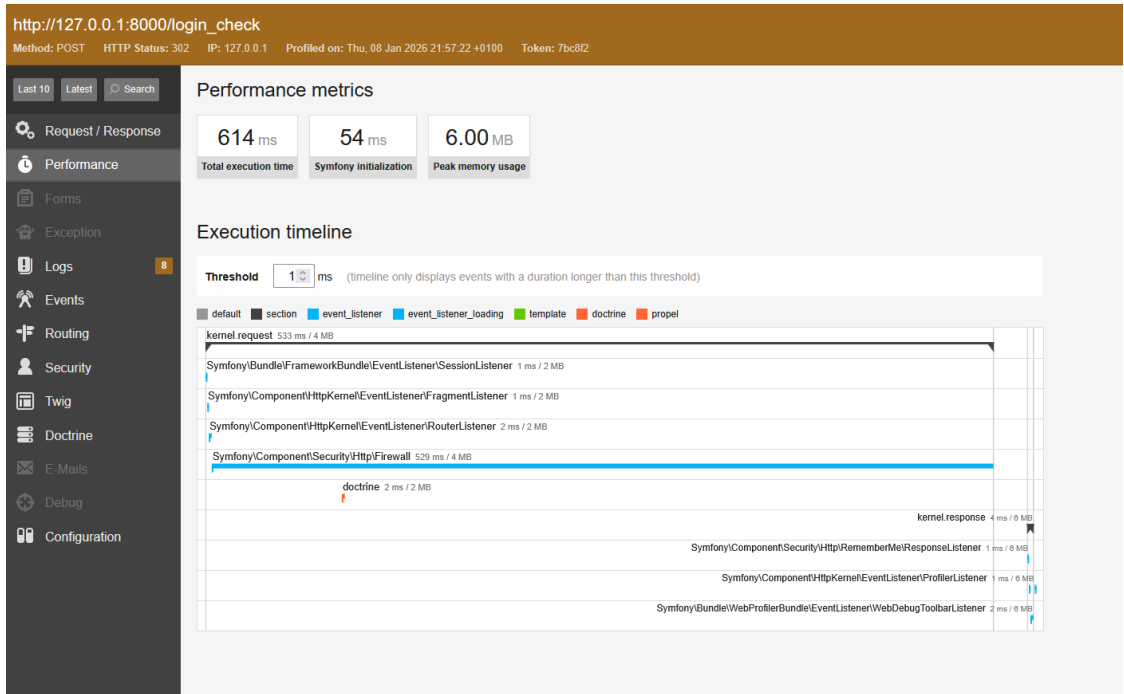


APRÈS :

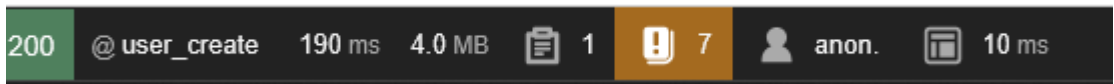
Login page :



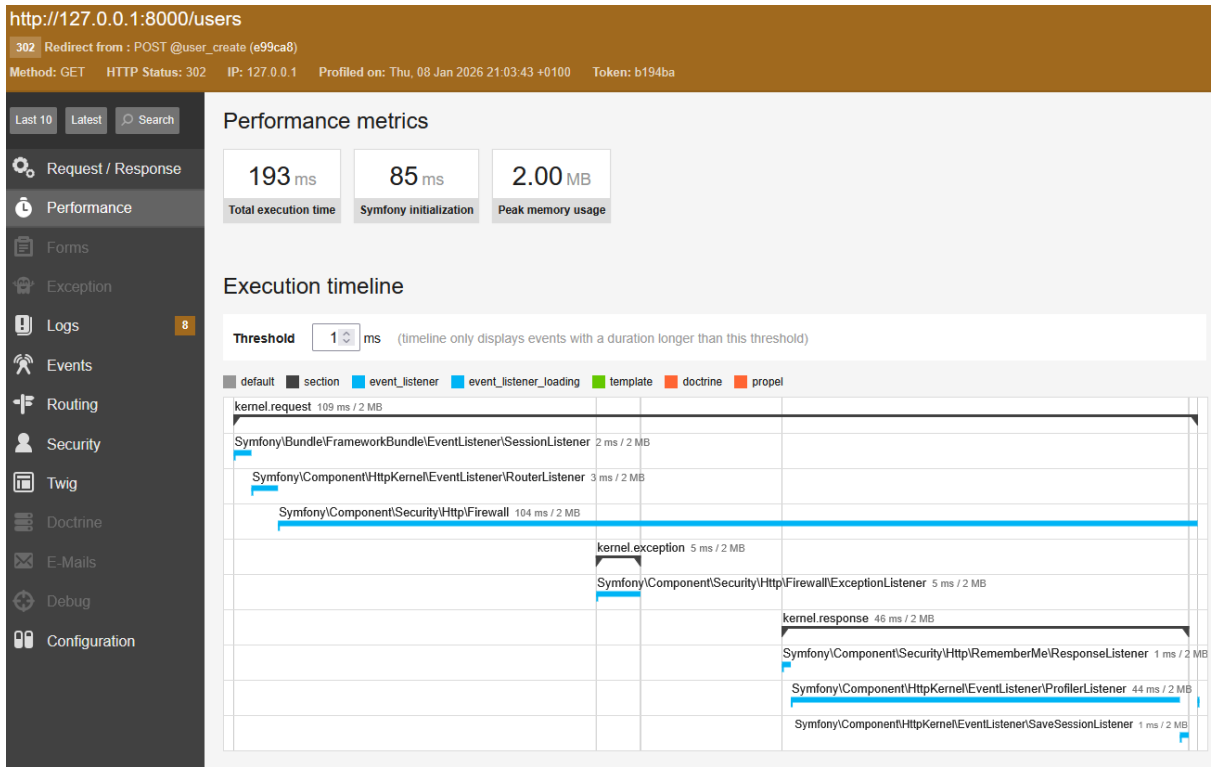
Login request :



Register page :



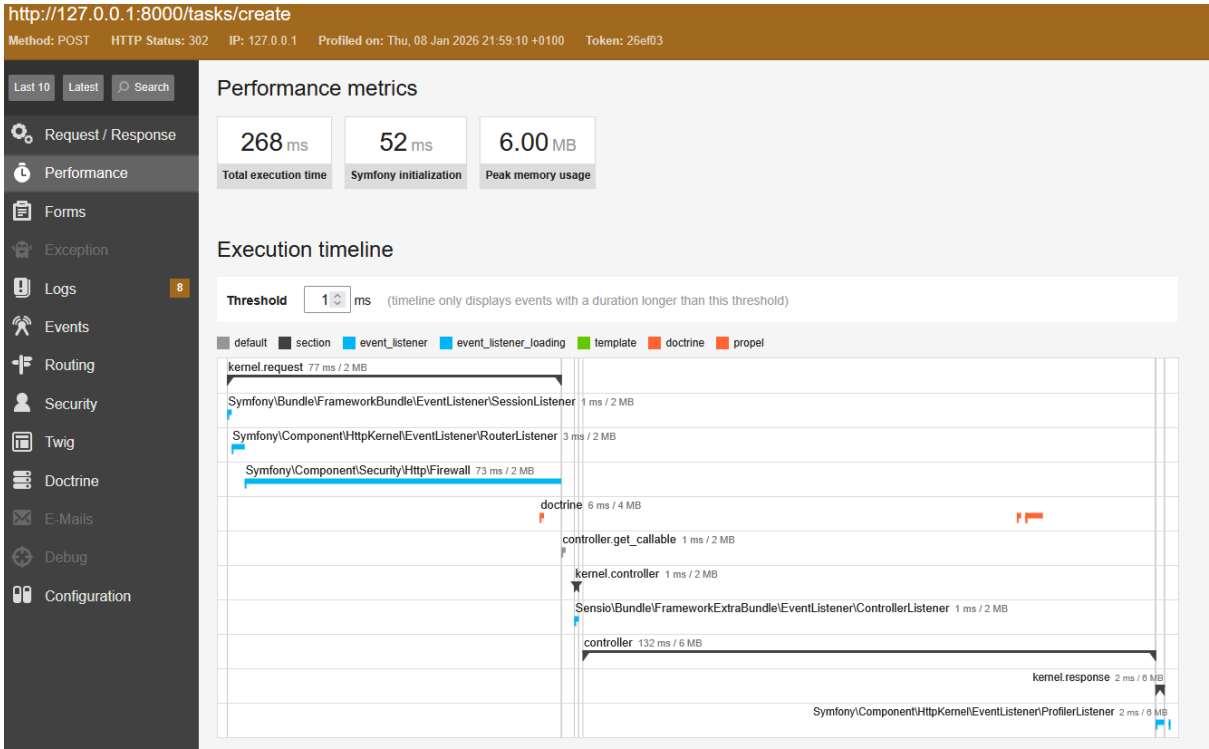
Register request :



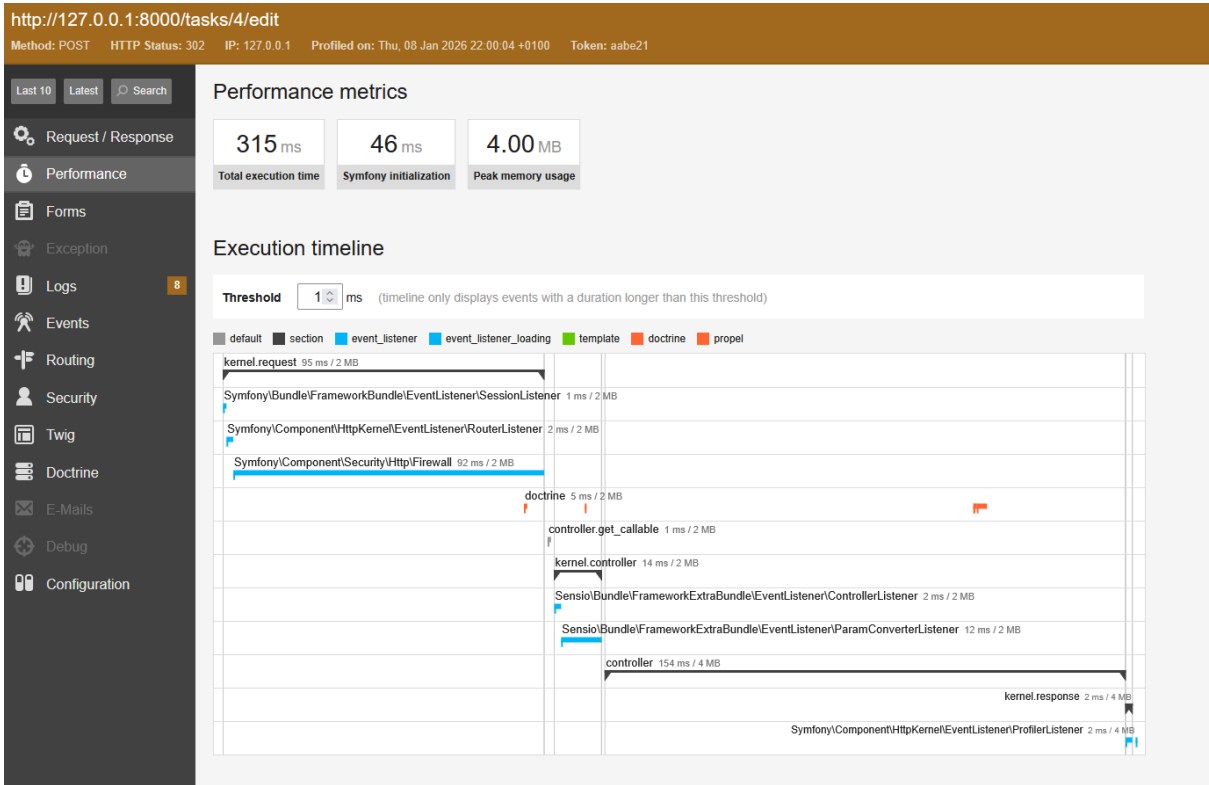
Task page :



Task create :



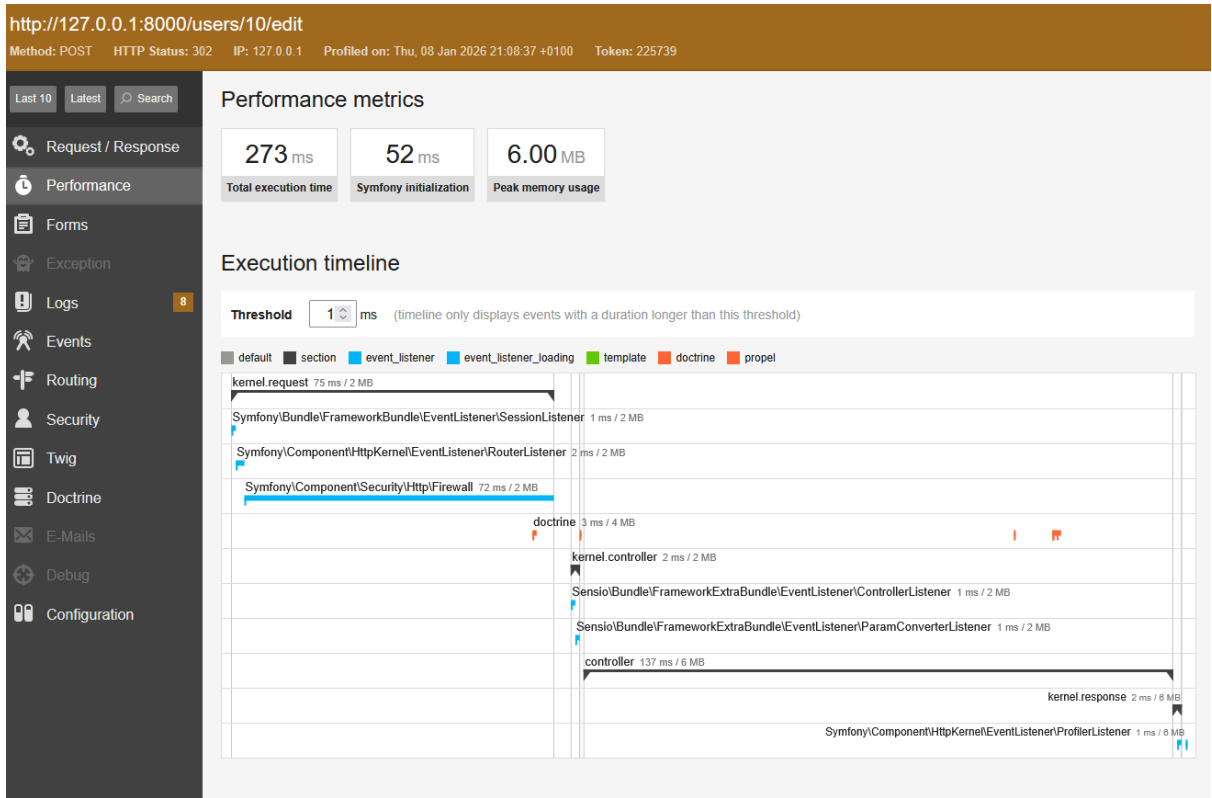
Task edit :



users page request :



user page edit :



État final (après modifications)

L'application est désormais **plus rapide, plus stable et plus scalable**. De plus, les anomalies ont été corrigées. Les erreurs fatales aussi ainsi que les 404.

Comparaison Avant / Après

Critère	Avant	Après
Qualité du code	Moyenne	Élevée
Dettes techniques	Importante	Fortement réduite
Couverture de tests	0 %	~85 %
Requêtes SQL	Élevées	Optimisées

Conclusion

Cet audit a permis de mettre en évidence une dette technique réelle, mais maîtrisable. Les améliorations apportées ont permis de :

- réduire significativement la dette technique,
- améliorer la lisibilité et la maintenabilité du code,
- augmenter la fiabilité grâce aux tests,
- optimiser les performances globales de l'application.

L'application *ToDo & Co* dispose désormais de **bases techniques saines**, favorisant sa pérennité et son évolution future.

Pistes d'amélioration

- Migration vers une version plus récente de Symfony
- Augmentation de la couverture de tests (tests d'intégration)
- Mise en place d'un outil de monitoring en production
- Optimisation avancée via cache applicatif (Redis)

Audit réalisé par : *Renaud*

Date : Décembre 2025

Statut : Conforme aux attentes de l'audit demandé