



ToDo & Co

Audit de qualité et de performances

Patrick Raspino

Kimealabs



Sommaire

1. Contexte

[1.1 Etat de l'application initiale](#)

[1.2 Besoins du client](#)

1.2.1 Corrections d'anomalies

1.2.2 Implémentations de nouvelles fonctionnalités

1.2.3 Implémentations de tests automatisés

2. Dette technique et fonctionnelle.

2.1 [Mise à jour de la version de Symfony](#)

2.2 [Ajout de fonctionnalités et amélioration UI/UX](#)

2.3 [Tests unitaires et fonctionnels](#)

3. Audit de qualité et de performances

[3.1 Qualité du code](#)

[3.2 Architecture](#)

[3.3 Performances initiales](#)

[3.4 Performances actuelles](#)

[3.5 Conclusions des résultats](#)

4. Bilan et axes d'améliorations

[4.1 Les performances](#)

[4.2 Les fonctionnalités](#)

1. Contexte

La société ToDo & Co vient de lever des fonds afin de permettre le développement de leur entreprise et notamment de leur application éponyme. En vue de l'amélioration et de l'analyse de leur application phare, Todo & Co a confié ces missions à notre société Kimealabs, laquelle est spécialisée dans les technologies Web.

1.1 Etat de l'application initiale

L'application initiale fonctionne autour du framework Symfony en version 3.1, laquelle n'est plus maintenue depuis juillet 2017 (End of security fixes). Il paraît impératif de commencer par mettre à jour le framework dans une version maintenue et sécurisée.

1.2 Besoins du client

1.2.1 Corrections d'anomalies

Une tâche doit être attachée à un utilisateur :

Actuellement, lorsqu'une tâche est créée, elle n'est pas rattachée à un utilisateur. Il vous est demandé d'apporter les corrections nécessaires afin qu'automatiquement, à la sauvegarde de la tâche, l'utilisateur authentifié soit rattaché à la tâche nouvellement créée.

Lors de la modification de la tâche, l'auteur ne peut pas être modifié.

Pour les tâches déjà créées, il faut qu'elles soient rattachées à un utilisateur "anonyme".

Choisir un rôle pour un utilisateur :

Lors de la création d'un utilisateur, il doit être possible de choisir un rôle pour celui-ci. Les rôles listés sont les suivants :

- rôle utilisateur (*ROLE_USER*) ;
- rôle administrateur (*ROLE_ADMIN*).

Lors de la modification d'un utilisateur, il est également possible de changer le rôle d'un utilisateur.

1.2.2 Implémentation de nouvelles fonctionnalités

Une liste de nouvelles fonctionnalités est demandée par le client. Bien que cette liste puisse paraître exhaustive, Todo & Co nous confie aussi la possibilité de proposer d'autres fonctionnalités.

Les demandes actuelles portent sur ces fonctionnalités :

Seuls les utilisateurs ayant le rôle administrateur (*ROLE_ADMIN*) doivent pouvoir accéder aux pages de gestion des utilisateurs.

Les tâches ne peuvent être supprimées que par les utilisateurs ayant créé les tâches en question.

Les tâches rattachées à l'utilisateur "anonyme" peuvent être supprimées uniquement par les utilisateurs ayant le rôle administrateur (*ROLE_ADMIN*).

1.2.3 Implémentation de tests automatisés

Il nous est demandé d'implémenter les tests automatisés (tests unitaires et fonctionnels) nécessaires pour assurer que le fonctionnement de l'application est bien en adéquation avec les demandes.

Ces tests seront implémentés avec **PHPUnit** et le bundle **DAMA**.

Tous les tests permettront de prouver, à minima, le fonctionnement des cas explicités dans ce document.

Il nous est demandé de fournir un rapport de couverture de code au terme du projet. Il faut que le taux de couverture soit supérieur à **70 %**.

2. Dette technique et fonctionnelle

En effet, l'application a une dette technique et fonctionnelle assez importante. Pour réduire cette dette, nous avons réalisé dans un premier temps les travaux suivants :

2.1 Mise à jour de la version de Symfony

Mise à jour de la version **3.1** du framework Symfony vers une version stable et sécurisé. Actuellement nous avons migré le framework sur la version **5.4** qui est une version **LTS** (Long Term Support), ce qui est un gage de stabilité dans le temps en terme de maintenabilité. Cette version est maintenue jusqu'en **novembre 2025**.

En effet, les alertes de la fonction **Dependabot** de la plateforme Github nous indique clairement que la version initiale ne pouvait être ni maintenue, ni sécuritaire :

<input type="checkbox"/> ⓘ 11 Open ✓ 0 Closed	Package ▾	Ecosystem ▾	Manifest ▾	Severity ▾	Sort ▾
<input type="checkbox"/> ⓘ Invalid HTTP method overrides allow possible XSS or other attacks in Symfony Critical					🔔 #3
#5 opened 20 hours ago • Detected in symfony/symfony (Composer) • composer.lock					
<input type="checkbox"/> ⓘ Forbid serializing AbstractAdapter and TagAwareAdapter instances Critical					🔔 #3
#3 opened 20 hours ago • Detected in symfony/symfony (Composer) • composer.lock					
<input type="checkbox"/> ⓘ Check service IDs are valid Critical					🔔 #3
#2 opened 20 hours ago • Detected in symfony/symfony (Composer) • composer.lock					
<input type="checkbox"/> ⓘ Twig may load a template outside a configured directory when using the filesystem loader High					🔔 #2
#11 opened 20 hours ago • Detected in twig/twig (Composer) • composer.lock					
<input type="checkbox"/> ⓘ Non-constant time comparison in UriSigner High					🔔 #3
#9 opened 20 hours ago • Detected in symfony/symfony (Composer) • composer.lock					
<input type="checkbox"/> ⓘ Deserialization of untrusted data in Symfony High					🔔 #3
#8 opened 20 hours ago • Detected in symfony/symfony (Composer) • composer.lock					
<input type="checkbox"/> ⓘ Improper authentication in Symfony High					🔔 #3
#6 opened 20 hours ago • Detected in symfony/symfony (Composer) • composer.lock					
<input type="checkbox"/> ⓘ Argument injection in a MimeTypeGuesser in Symfony High					🔔 #3
#4 opened 20 hours ago • Detected in symfony/symfony (Composer) • composer.lock					
<input type="checkbox"/> ⓘ Escape validation messages in the PHP templating engine Moderate					🔔 #3
#1 opened 20 hours ago • Detected in symfony/symfony (Composer) • composer.lock					
<input type="checkbox"/> ⓘ Deserialization of untrusted data in Symfony High Development					🔔 #1
#7 opened 20 hours ago • Detected in symfony/phpunit-bridge (Composer) • composer.lock					
<input type="checkbox"/> ⓘ Sandbox Information Disclosure Low					🔔 #2
#10 opened 20 hours ago • Detected in twig/twig (Composer) • composer.lock					

Dependabot nous proposé alors quelques « pull-request » confirmant qu’une mise à jour du framework Symfony était obligatoire :

<input type="checkbox"/> 3 Open ✓ 0 Closed		Author ▾	Label ▾	Projects ▾	Milestones ▾	Reviews ▾	Assignee ▾	Sort ▾
<input type="checkbox"/>	Bump symfony/symfony from 3.1.6 to 3.4.35 ✕ dependencies	#3 opened 20 hours ago by dependabot bot						
<input type="checkbox"/>	Bump twig/twig from 1.27.0 to 1.42.5 ✕ dependencies	#2 opened 20 hours ago by dependabot bot						
<input type="checkbox"/>	Bump symfony/phpunit-bridge from 3.1.6 to 3.4.26 ✓ dependencies	#1 opened 20 hours ago by dependabot bot						

2.2 Ajout de fonctionnalités et amélioration UI/UX

Dès la fin des corrections d’anomalies et de l’implémentation des nouvelles fonctionnalités demandées, nous avons ajouté des fonctionnalités, du design et l’ergonomie (UI/UX) :

A savoir qu’initialement le projet ne comporte qu’une rubrique permettant de voir toutes les tâches, terminées ou à faire, sur la même page (bouton Consulter la liste des tâches à faire). De plus il y avait un bouton inopérant (Consulter la liste des tâches terminées) :

To Do List app 

Créer un utilisateur

Se déconnecter

Bienvenue sur Todo List, l'application vous permettant de gérer l'ensemble de vos tâches sans effort !

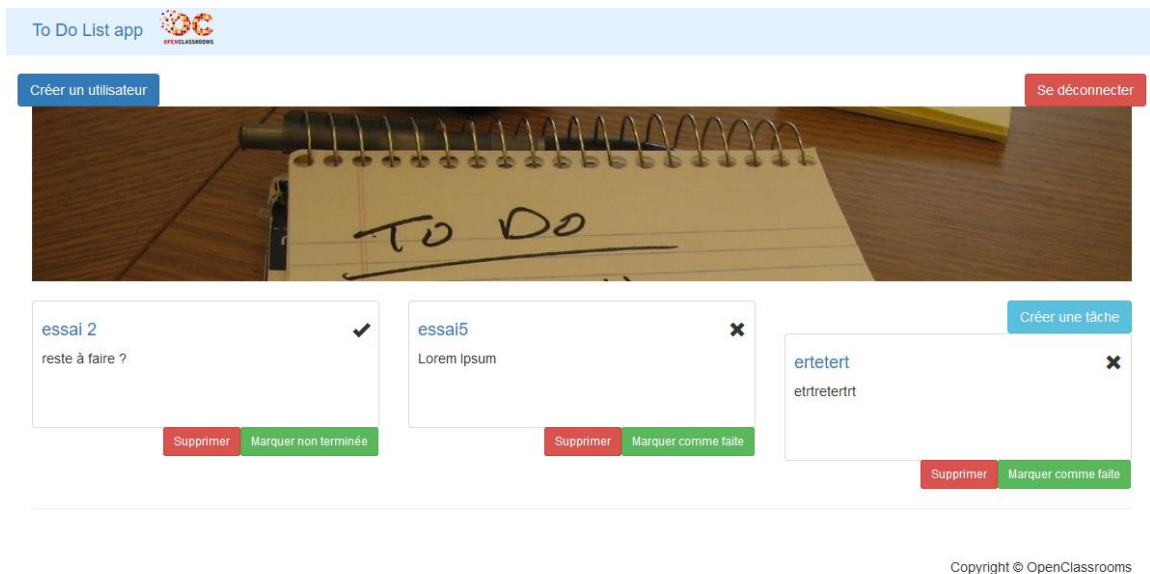


Créer une nouvelle tâche

Consulter la liste des tâches à faire

Consulter la liste des tâches terminées

La liste des tâches n'est aucunement organisée, aucun cadre graphique ne permet un affichage cohérent et fonctionnelle :



Aussi, nous avons ajouté/modifié ces éléments :

- Corrections des fonctions du bouton « toggle » permettant le passage du statut d'une tâche en « à faire » ou « marquée terminée ».
- Formulaire de mise à jour des utilisateurs permettant de laisser le mot de passe vide si inchangé.
- Mise à jour graphique avec Bootstrap version 5.
- Ajout du « Responsive », afin d'afficher l'application sur les mobiles.
- Ajout de liens vers la page d'accueil (n'existant pas).
 - Bouton « home » en entête du site.
 - Bouton « retour à l'accueil » sur toutes les pages.
- Ajout d'une page pour la catégorie des tâches terminées.
- Ajout d'une page pour la catégorie des tâches à faire.
- Modification de la page d'accueil.
- Messages informatifs de couleur pour toutes les actions réalisables.
- Affichage de la date de création d'une tâche ainsi que son auteur.
- Ajout d'une page 404 pour les URL inconnues.
- Ajout de validations de formats concernant notamment le fait de ne pouvoir s'inscrire avec un nom (ou un email) déjà utilisé en base de données.

2.3 Tests unitaires et fonctionnels

Afin de créer un ensemble de tests automatisés. Nous avons utilisé **PHPUnit** lié à des **fixtures**, lesquelles permettent de tester toutes les situations possibles concernant les utilisateurs, les tâches et les fonctionnalités de l'application.

Nous avons couvert toutes les « routes » n'appartenant pas au framework Symfony :

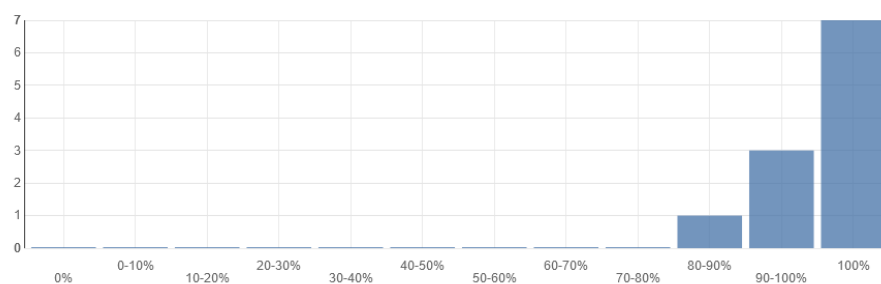
Actuellement, les codes de tests **couvrent 97.1%** des lignes de l'application, ce qui permet assez bien d'affirmer la qualité fonctionnelle de l'application et permettre de gagner du temps lors de futures mises à jour du code informatique.

		Lines	
Total	<div><div></div></div>	97.10%	134 / 138
Controller	<div><div></div></div>	98.36%	60 / 61
Entity	<div><div></div></div>	97.30%	36 / 37
Repository	<div><div></div></div>	100.00%	14 / 14
Security	<div><div></div></div>	92.31%	24 / 26
Kernel.php		n/a	0 / 0

		Functions and Methods	
Total	<div><div></div></div>	92.45%	49 / 53
Controller	<div><div></div></div>	90.91%	10 / 11
Entity	<div><div></div></div>	96.15%	25 / 26
Repository	<div><div></div></div>	100.00%	6 / 6
Security	<div><div></div></div>	80.00%	8 / 10
Kernel.php		n/a	0 / 0

		Classes and Traits	
Total	<div><div></div></div>	60.00%	6 / 10
Controller	<div><div></div></div>	75.00%	3 / 4
Entity	<div><div></div></div>	50.00%	1 / 2
Repository	<div><div></div></div>	100.00%	2 / 2
Security	<div><div></div></div>	0.00%	0 / 2
Kernel.php		n/a	0 / 0

Classes
Coverage Distribution

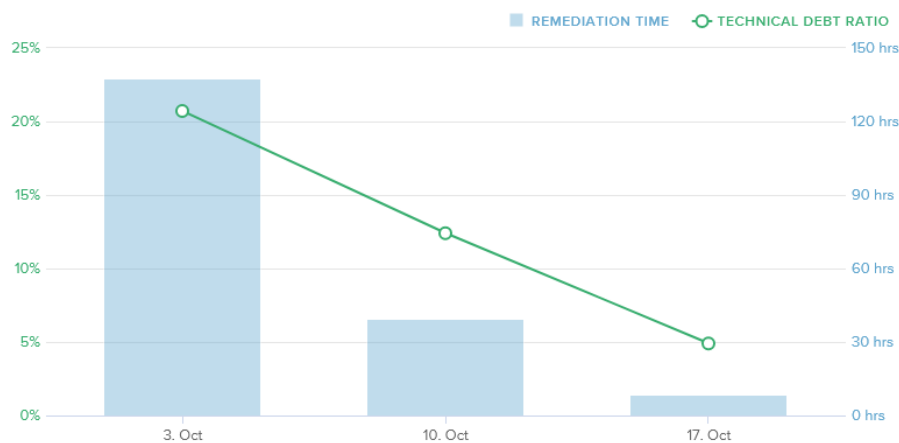


3. Audit de qualité et de performances

Utiliser **Blackfire.io** aurait produit un rapport de performances d'une qualité supérieure, hélas, ces outils devenus payant, nous avons utilisé le **profilier de Symfony** afin de mesurer les performances de l'application. Aucune inquiétude, les mesures livrées par cette fonctionnalité restent très bonnes et permettent de synthétiser des conclusions pertinentes.

3.1 Qualité du code

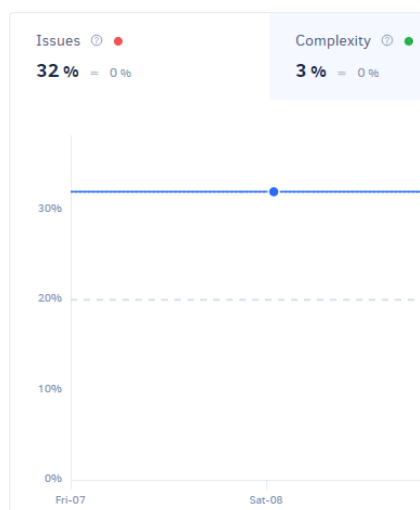
Outre les ajouts de fonctionnalités et le « redressement » de la charte graphique et ergonomique, nous avons évalué la qualité de l'application grâce à 2 outils, **Codacy** et **CodeClimate**.



Nous pouvons voir que selon **CodeClimate**, la dette technique s'est réduite passant de **137 heures** pour atteindre maintenant quelques **8 heures**.

ToDoOCR **B** main ▾

Quality evolution



Initialement noté **B** par **Codacy**.

Quelques **32 %** des issues comportant des erreurs de sécurité ou de manquements aux bonnes pratiques (PSR), tels le nommage des classes, des variables, constantes ou utilisations de logiques trop compliqué au sein de méthodes ne permettant pas ni une bonne compréhension du code, ni même sa maintenabilité.

Actuellement, le code de l'application est noté **A** par Codacy, et les issues représentant un problème (ici mineures), ne représente plus de 3 – 4 %.

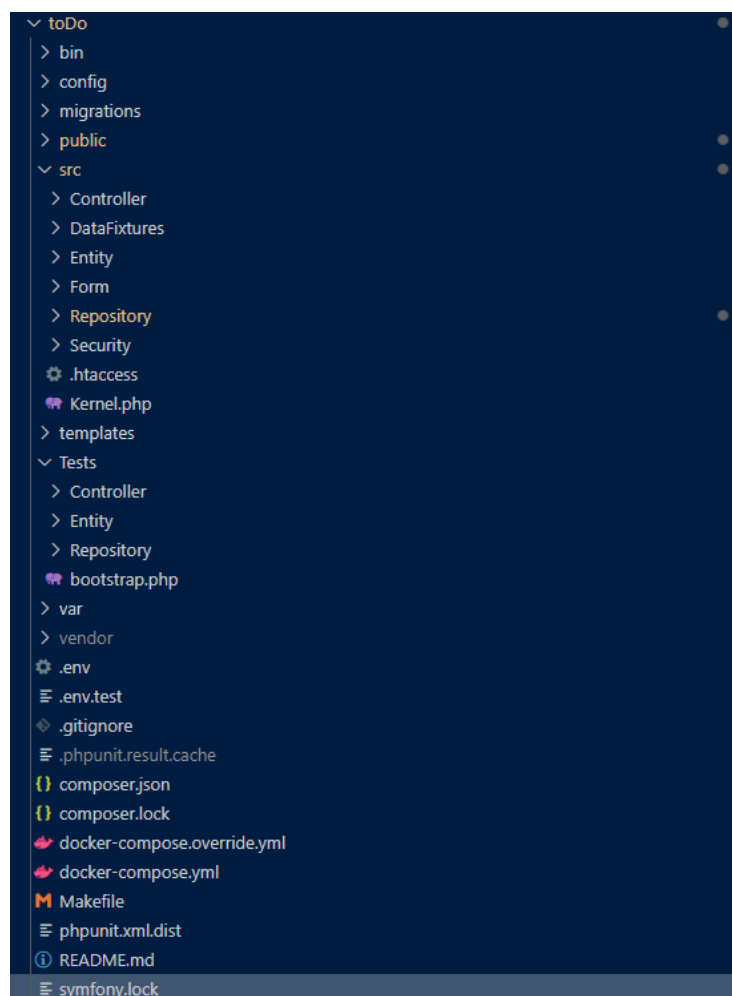
ToDoOCR A main ▾

Quality evolution



3.2 Architecture

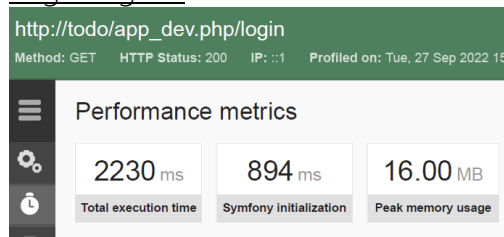
L'ancienne architecture du framework Symfony a été mise à jour. Vous pouvez retrouver ci-dessous la nouvelle « organisation » du framework.



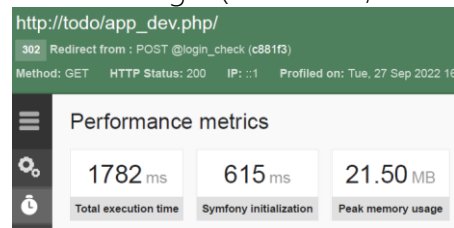
3.2 Performances initiales

3.3

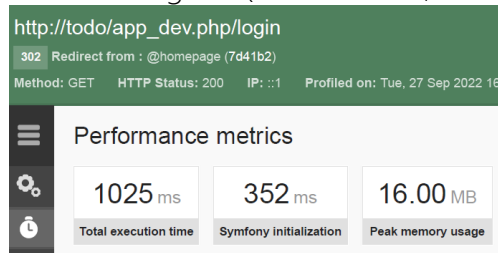
Page Login :



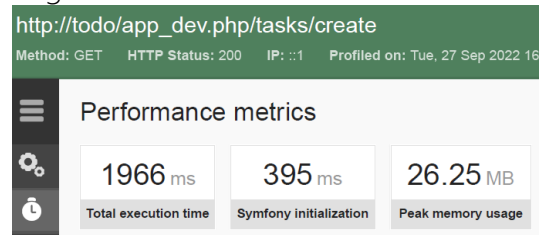
Fonction Login (connexion, -> accueil) :



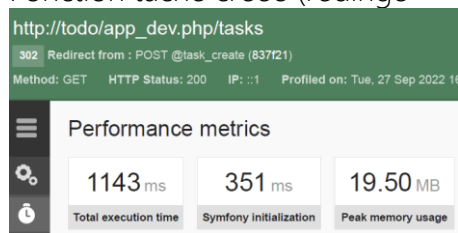
Fonction LogOut (déconnexion, -> login) :



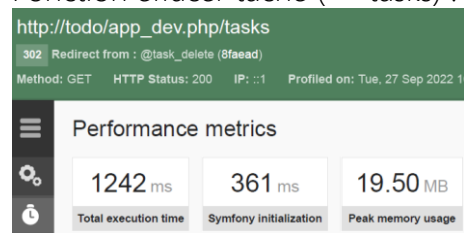
Page création de tâche :



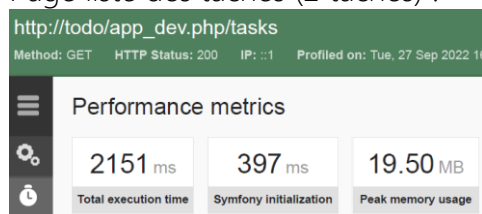
Fonction tâche créée (redirigé -> tasks) :



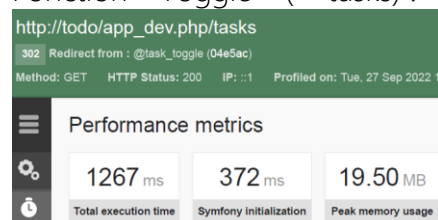
Fonction effacer tâche (-> tasks) :



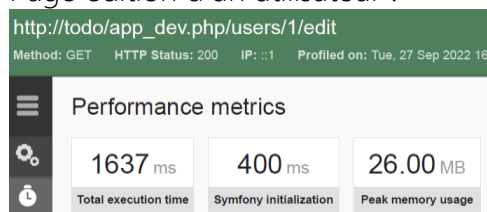
Page liste des tâches (2 tâches) :



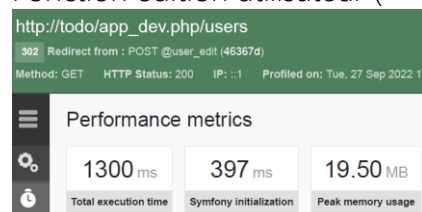
Fonction « Toggle » (-> tasks) :



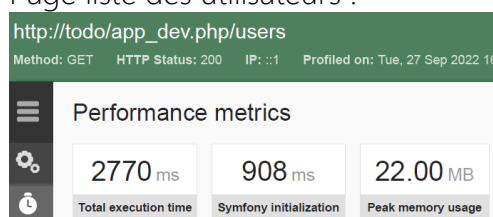
Page édition d'un utilisateur :



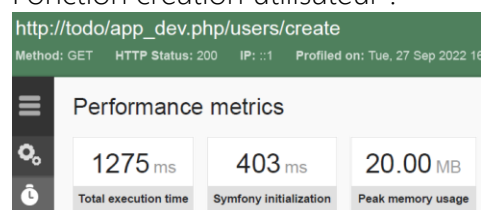
Fonction édition utilisateur (-> users) :



Page liste des utilisateurs :

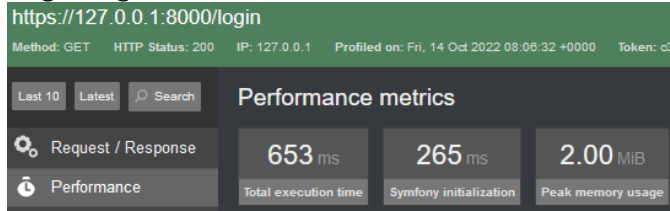


Fonction création utilisateur :

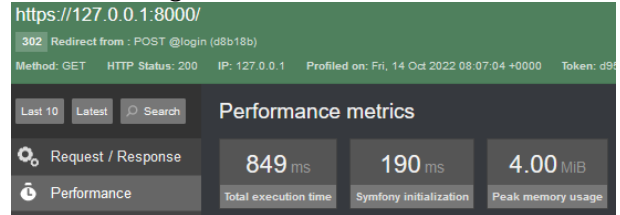


3.4 Performances actuelles

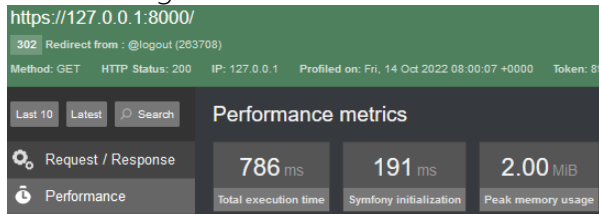
Page Login :



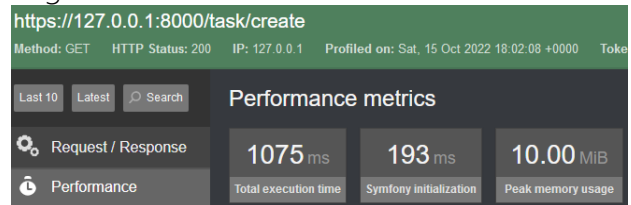
Fonction Login :



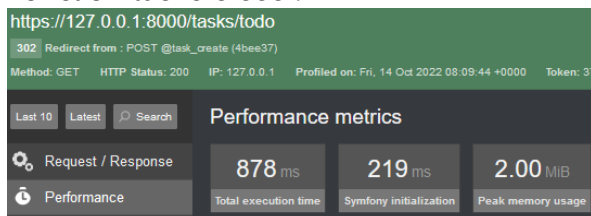
Fonction logOut :



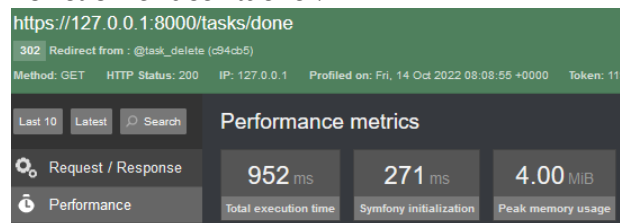
Page création de tâche :



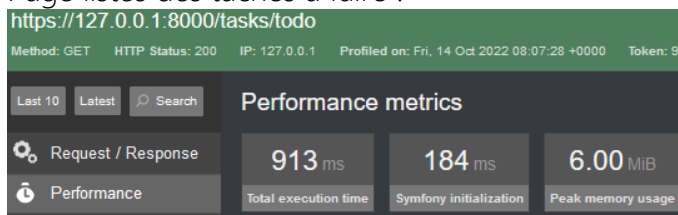
Fonction tâche créée :



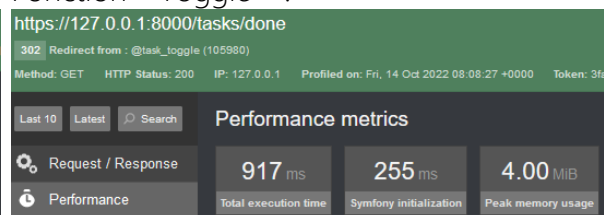
Fonction effacer tâche :



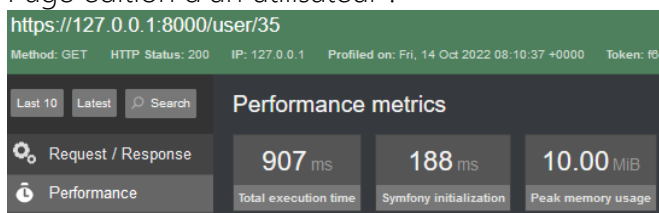
Page listes des tâches à faire :



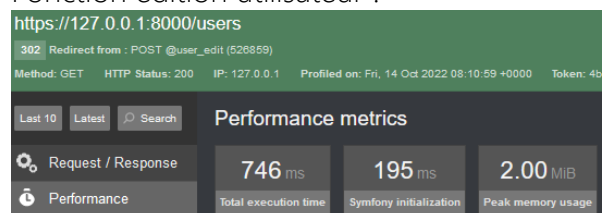
Fonction « Toggle » :



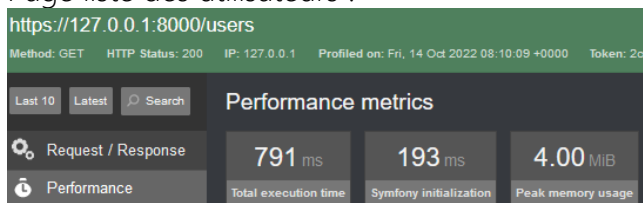
Page édition d'un utilisateur :



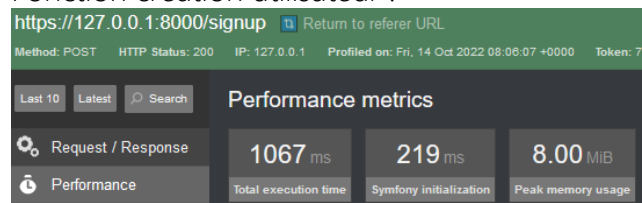
Fonction édition utilisateur :



Page liste des utilisateurs :



Fonction création utilisateur :



3.5 Conclusions des résultats

Il semble évident que les temps d'exécutions et d'occupations mémoires se sont vus très réduits.

Les temps « d'initialisation de Symfony » sont aussi très largement inférieurs par rapport à la version initiale.

Par rapport à des analyses plus détaillées de ces « metrics », ces améliorations proviennent essentiellement du fait de la mise à jour du framework Symfony, lequel n'appelle plus, ni de la même façon, ni au même nombre, ses services internes. Le nombre d'appels à « *l'Autoloader* » est forcément différent de l'ancienne version.

En moyenne, nous pouvons constater une amélioration de **35 à 45%**. A ajouter à cela la maintenabilité, la sécurité et les évolutions facilitées de l'application.

4. Bilan et axes d'améliorations

4.1 Les performances

Les performances et la qualité de l'application, bien que largement améliorées grâce à la mise à jour du framework Symfony, ainsi qu'à l'applications des bonnes pratiques de développement (PSR, refactor) peuvent être encore améliorées dans un futur.

Propositions d'améliorations futures pour les performances :

- Utilisation d'un cache permettant d'économiser les appels à la base de données.
- Utilisation de Opache de PHP
- Optimiser « l'Autoloader » grâce à l'instruction CLI suivante :
`composer dump-autoload --no-dev --classmap-authoritative`
A exécuter à chaque modification/ajout de contrôleurs ou autre.

4.2 Les fonctionnalités

De nouvelles fonctionnalités pourraient être intéressantes, autant sur le plan de l'ergonomie que sur un plan conceptuel :

- Ne plus permettre l'inscription en tant qu'administrateur (seul un « super-administrateur » pourrait donner des privilèges à certains utilisateurs).
- Affichages des tâches uniquement liées à l'utilisateur (et toutes pour les administrateurs).
- Ajouter une date prévisionnelle de fin (**Due date**) pour une tâche et en avertir le créateur par email 24h avant échéance. Et même une alerte en cas de dépassement de cette date.
- Ajouter un contrôle de validation par email et Captcha lors de l'inscription.
- Créer une page « profil » permettant à chaque utilisateur de modifier certaines données ou caractéristiques de son compte.
- Créer un système de pagination en cas d'un certain nombre de tâches à afficher dans les listes « à faire » et « terminées ».
- Permettre à un administrateur d'afficher la liste des tâches liées à un utilisateur.
- « Upgrader » le framework Symfony vers sa version **6.1**.
- Mise en place d'une intégration continue grâce à **Travis C.I.**