Audit de qualité du code & performance de l'application Todo & Co

I Qualité du code :

Afin de tester la qualité du code, nous utiliserons Codacy. Il nous permettra de présenter des analyses de code à chaque push sur le repository.

De plus nous utiliserons également des tests unitaires et fonctionnels pour vérifier la qualité du code mais aussi détecter les bugs et dépréciations.

1. Codacy :

Le graphique ci-dessous présente des remarques ou problèmes techniques depuis le début développement.

1.1 Enjeux et objectifs :

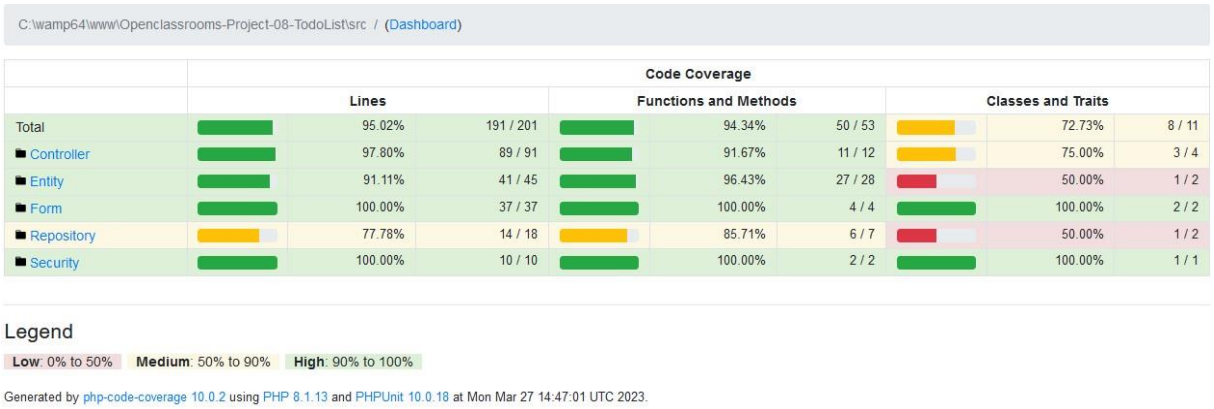
On remarque que la tendance des issues depuis le début est en chute, c'est ce que l'on recherche après la mise à jour du Framework et de la logique métier pour passer à la version 5.4lts.

Dans une optique du maintien de la qualité et de la sécurité, il est nécessaire de respecter les bonnes pratiques en vigueurs. Cela doit se faire à chaque modification du code.

Ainsi chaque commit doit être monitoré par Codacy.

1.2 PHP Unit :

PHP Unit va nous permettre de tester l'efficience du code, 29 tests ont été réalisés pour un rapport de couverture de l'application supérieur à 70%.



C:\wamp64\www\Openclassrooms-Project-08-TodoList\src / (Dashboard)

	Code Coverage								
	Lines			Functions and Methods			Classes and Traits		
Total	95.02%	191 / 201		94.34%	50 / 53		72.73%	8 / 11	
Controller	97.80%	89 / 91		91.67%	11 / 12		75.00%	3 / 4	
Entity	91.11%	41 / 45		96.43%	27 / 28		50.00%	1 / 2	
Form	100.00%	37 / 37		100.00%	4 / 4		100.00%	2 / 2	
Repository	77.78%	14 / 18		85.71%	6 / 7		50.00%	1 / 2	
Security	100.00%	10 / 10		100.00%	2 / 2		100.00%	1 / 1	

Legend

Low: 0% to 50% Medium: 50% to 90% High: 90% to 100%

Generated by php-code-coverage 10.0.2 using PHP 8.1.13 and PHPUnit 10.0.18 at Mon Mar 27 14:47:01 UTC 2023.

2.1 Enjeux et objectifs :

Les test vont nous permettre de tester toutes les fonctionnalités et d'affirmer s'il y a du code non utilisé.
Le but est de fournir un code maitrisé, optimisé, utile et qu'il réponde à nos attentes entièrement à nos attentes.

In fine la pratique la plus optimisé est le test-driven développement ou « TDD » qui renvoie à une technique de développement logiciel qui vise à réduire les anomalies d'une application en favorisant la mise en œuvre fréquente de test tout au long du développement.

II Performance de l'application :

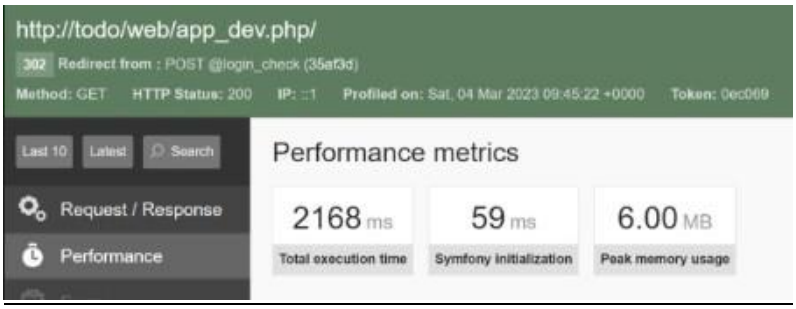
2.1 Enjeux et objectifs :

Les performances sont un critère primordiale pour l'expériences utilisateur. Les attentes en termes de temps de chargement sont de plus en plus élevées. Il faut rappeler que les connexions à une application peuvent être fortement variables d'un endroit à un autre et que l'affluence sur celle-ci peux la faire ralentir de manière importante.

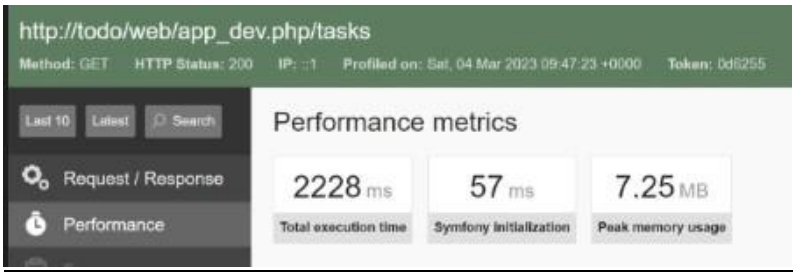
Nous ferons ici les tests sur le profiler de Symfony sur quelques pages principales, la tendance sera la même pour toutes les requêtes.

2.2 Audit sans optimisation avec Symfony 3.1 :

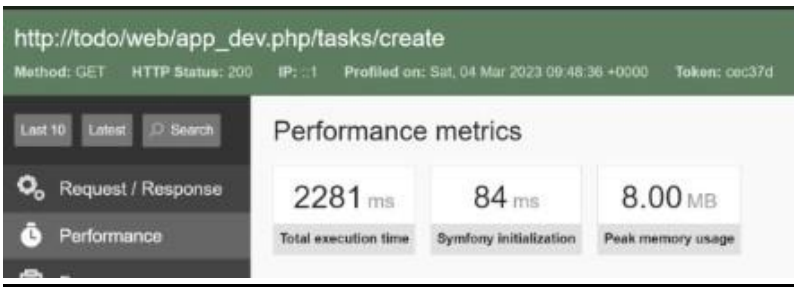
Soumission du formulaire de connexion :



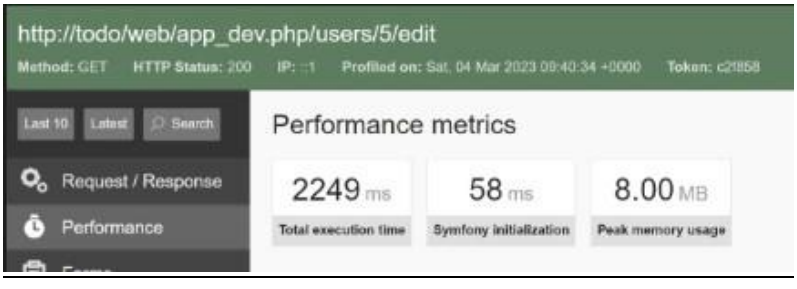
Page liste des taches :



Page d'une création d'une tache :



Page d'Edition d'un utilisateur :



2.3 Apres upgrade vers la version 5.4 lts et optimisation :

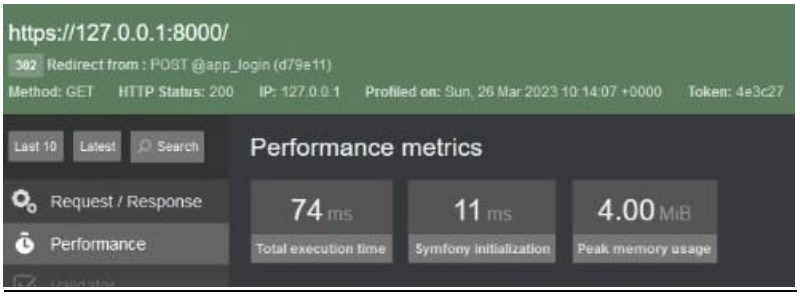
- Mise à jour de la version du framework Symfony 3.1 => 5.4lts
- Mise à jour de php 5.6 vers 8.1
- Mise à jour de bootstrap v3 => v5
- Correction des dépréciations de code dans la logique métier tel que l'accès a certaine dépendance que sont les Repositories :

```
public function listAction(): Response
{
    return $this->render('user/list.html.twig',
    ['users' => $this->getDoctrine()->getRepository('AppBundle:Task')->findAll()]);
}
```

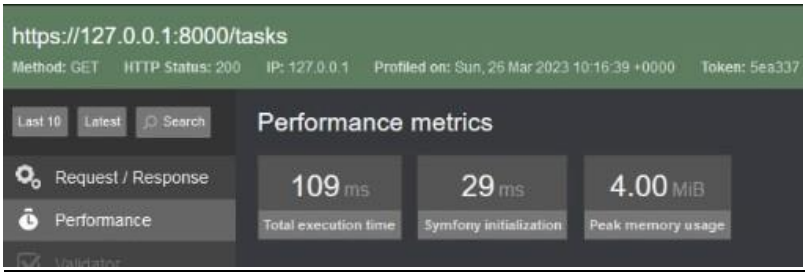
et AuthenticationUtils :

```
public function index(): Response
{
    // get the login error if there is one
    $authenticationUtils = $this->get('security.authentication_utils');
```

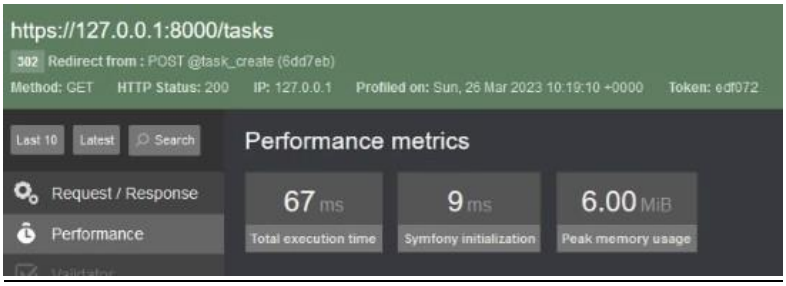
Soumission du formulaire de connexion :



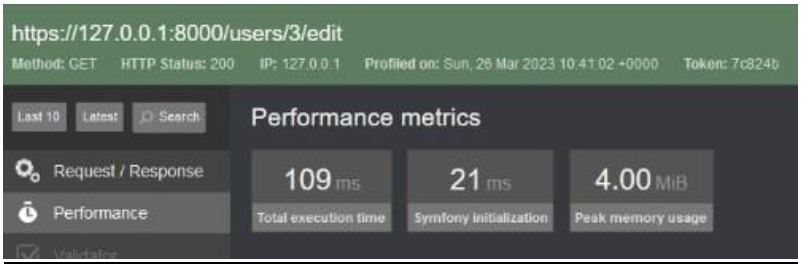
Page liste des taches :



Page d'une création d'une tache :



Page d'Edition d'un utilisateur :



Le temps de chargement des pages a été diminué entre 15 et 20 fois et le pic de mémoire vive a été diminué de près de 2 fois malgré une refonte du design des templates, on a inexorablement une expérience utilisateur bien meilleur.