

18/05/2020

# Audit de qualité et améliorations

## TO DO LIST

1  
2  
3



## Table des matières

Introduction .....	2
- 1 – Audit du code initial backend / frontend.....	2
1.1 Rapport CodeClimate : .....	2
1.2 Rapport PHPMetrics et PHPCS .....	3
1.3 Version de Symfony et PHP .....	5
1.4 La sécurité : .....	6
1.5 Mesure de performance BlackFire du code initial .....	6
1.5 Mesure de performance BlackFire du code initial (suite).....	7
1.6 Principes SOLID .....	9
1.7 Frontend .....	9
1.8 Conclusion .....	10
1.8.1 Backend.....	10
1.8.2 Frontend .....	10
- 2 – Améliorations du code initial backend / frontend .....	11
2.1 Migration de php et symfony.....	11
2.2 Amélioration de la sécurité.....	12
2.3 Amélioration de la qualité .....	17
2.4 Amélioration de la maintenabilité et des performances.....	20
- 3 – Améliorations du frontend .....	24
- 4 – Tests unitaires et fonctionnels .....	25

## Introduction

Rappel du projet, ToDo & Co est une startup dont le cœur de métier est une application de gestion des tâches quotidiennes.

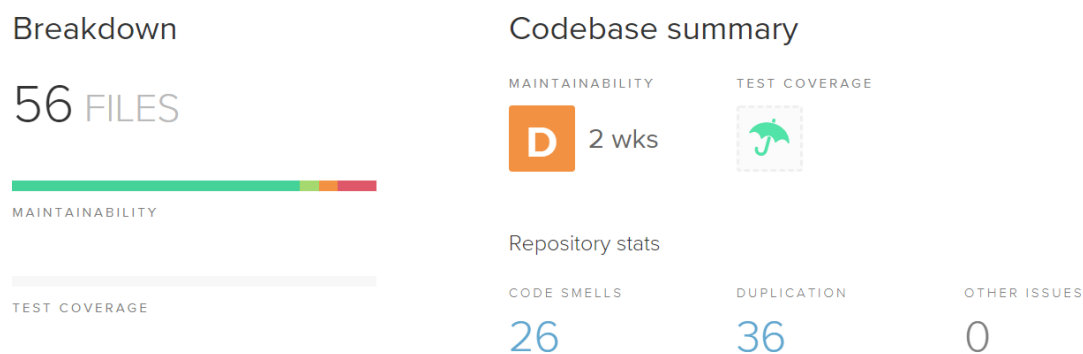
L'entreprise vient tout juste d'être créée, et l'application a dû être développée rapidement pour permettre de présenter à de potentiels investisseurs que le concept est viable (Minimum Viable Product ou MVP) avec le Framework Symfony dans sa version 3.1.

La mission qui m'a été confiée est de développer de nouvelles fonctionnalités, et d'analyser la qualité du code ainsi que ses performances.

Ce document explique l'audit réalisé ainsi que les points d'amélioration à réaliser.

### - 1 – Audit du code initial backend / frontend

#### 1.1 Rapport CodeClimate :



Des analyses de qualité de codes ont été effectuées avec CodeClimate.

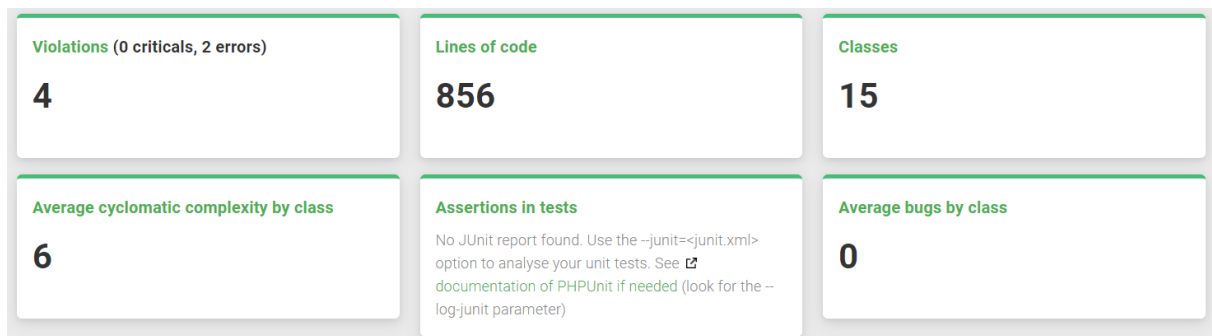
Ces analyses n'ont pas révélé de problèmes particuliers sur le code de l'application.

Il y a des remontées sur le code du framework Symfony, elles ont été ignorées, ainsi qu'une maintenabilité classée D qui doit être améliorée.

Les PSR1 et PSR2 sont respectés mais la PHPDoc est absente, elle est importante pour une meilleure compréhension et maintenance du code.

De plus elle servira aux futurs développeurs qui pourraient reprendre l'application pour la faire évoluer.

## 1.2 Rapport PHPMetrics et PHPCS



Le rapport montre un taux de complexité des classes de 6.07 qui doit être abaissé au maximum. PHPCS scan le code de chaque classe et donne un rapport très détaillé sur les éventuelles erreurs de code et corrections qui doivent être apportés.

Le rapport de phpcs sur les contrôleurs rapporte que l'indentation, ouverture et fermeture de parenthèses, annotations peuvent être corrigé grâce à l'outil intégré PHPCBF. Certaines corrections doivent être faite à la main. Voir [figure 1](#), [figure 2](#), [figure 3](#).

```
FILE: D:\Users\Jeromes2\Desktop\Developpement\Todoandco\src\AppBundle\Controller\DefaultController.php
-----
FOUND 5 ERRORS AFFECTING 5 LINES
-----
 1 | ERROR | [x] End of line character is invalid; expected "\n" but found "\r\n"
 2 | ERROR | [ ] Missing file doc comment
 8 | ERROR | [ ] Missing doc comment for class DefaultController
10 | ERROR | [ ] Missing short description in doc comment
12 | ERROR | [ ] Missing @return tag in function comment
-----
PHPCBF CAN FIX THE 1 MARKED SNIFF VIOLATIONS AUTOMATICALLY
-----
 17 |
 18 | }
 19 |
 20 | }

FILE: D:\Users\Jeromes2\Desktop\Developpement\Todoandco\src\AppBundle\Controller\SecurityController.php
-----
FOUND 12 ERRORS AFFECTING 11 LINES
-----
 1 | ERROR | [x] End of line character is invalid; expected "\n" but found "\r\n"
 2 | ERROR | [ ] Missing file doc comment
 9 | ERROR | [ ] Missing doc comment for class SecurityController
11 | ERROR | [ ] Missing short description in doc comment
11 | ERROR | [ ] Doc comment for parameter "$request" missing
13 | ERROR | [ ] Missing @return tag in function comment
21 | ERROR | [x] Opening parenthesis of a multi-line function call must be the last content on the line
24 | ERROR | [x] Closing parenthesis of a multi-line function call must be on a line by itself
27 | ERROR | [ ] Missing short description in doc comment
29 | ERROR | [ ] Missing @return tag in function comment
35 | ERROR | [ ] Missing short description in doc comment
37 | ERROR | [ ] Missing @return tag in function comment
-----
PHPCBF CAN FIX THE 3 MARKED SNIFF VIOLATIONS AUTOMATICALLY
-----
```

Figure 1

```

FILE: D:\Users\Jeromes2\Desktop\Developpement\Todoandco\src\AppBundle\Controller\TaskController.php
-----
FOUND 20 ERRORS AND 3 WARNINGS AFFECTING 18 LINES
-----
 1 | ERROR   | [x] End of line character is invalid; expected "\n" but found "\r\n"
 2 | ERROR   | [ ] Missing file doc comment
11 | ERROR   | [ ] Missing doc comment for class TaskController
13 | ERROR   | [ ] Missing short description in doc comment
15 | ERROR   | [ ] Missing @return tag in function comment
18 | WARNING | [ ] Line exceeds 85 characters; contains 131 characters
21 | ERROR   | [ ] Missing short description in doc comment
21 | ERROR   | [ ] Doc comment for parameter "$request" missing
23 | ERROR   | [ ] Missing @return tag in function comment
42 | WARNING | [ ] Line exceeds 85 characters; contains 87 characters
45 | ERROR   | [ ] Missing short description in doc comment
45 | ERROR   | [ ] Doc comment for parameter "$task" missing
45 | ERROR   | [ ] Doc comment for parameter "$request" missing
47 | ERROR   | [ ] Missing @return tag in function comment
62 | ERROR   | [x] Opening parenthesis of a multi-line function call must be the last content on the line
65 | ERROR   | [x] Closing parenthesis of a multi-line function call must be on a line by itself
68 | ERROR   | [ ] Missing short description in doc comment
68 | ERROR   | [ ] Doc comment for parameter "$task" missing
70 | ERROR   | [ ] Missing @return tag in function comment
76 | WARNING | [ ] Line exceeds 85 characters; contains 110 characters
81 | ERROR   | [ ] Missing short description in doc comment
81 | ERROR   | [ ] Doc comment for parameter "$task" missing
83 | ERROR   | [ ] Missing @return tag in function comment
-----
PHPCBF CAN FIX THE 3 MARKED SNIFF VIOLATIONS AUTOMATICALLY
-----

```

Figure 2

```

FILE: D:\Users\Jeromes2\Desktop\Developpement\Todoandco\src\AppBundle\Controller\UserController.php
-----
FOUND 12 ERRORS AND 5 WARNINGS AFFECTING 14 LINES
-----
 1 | ERROR   | [x] End of line character is invalid; expected "\n" but found "\r\n"
 2 | ERROR   | [ ] Missing file doc comment
11 | ERROR   | [ ] Missing doc comment for class UserController
13 | ERROR   | [ ] Missing short description in doc comment
15 | ERROR   | [ ] Missing @return tag in function comment
18 | WARNING | [ ] Line exceeds 85 characters; contains 131 characters
21 | ERROR   | [ ] Missing short description in doc comment
21 | ERROR   | [ ] Doc comment for parameter "$request" missing
23 | ERROR   | [ ] Missing @return tag in function comment
33 | WARNING | [ ] Line exceeds 85 characters; contains 109 characters
44 | WARNING | [ ] Line exceeds 85 characters; contains 87 characters
47 | ERROR   | [ ] Missing short description in doc comment
47 | ERROR   | [ ] Doc comment for parameter "$user" missing
47 | ERROR   | [ ] Doc comment for parameter "$request" missing
49 | ERROR   | [ ] Missing @return tag in function comment
57 | WARNING | [ ] Line exceeds 85 characters; contains 109 characters
67 | WARNING | [ ] Line exceeds 85 characters; contains 102 characters
-----
PHPCBF CAN FIX THE 1 MARKED SNIFF VIOLATIONS AUTOMATICALLY
-----

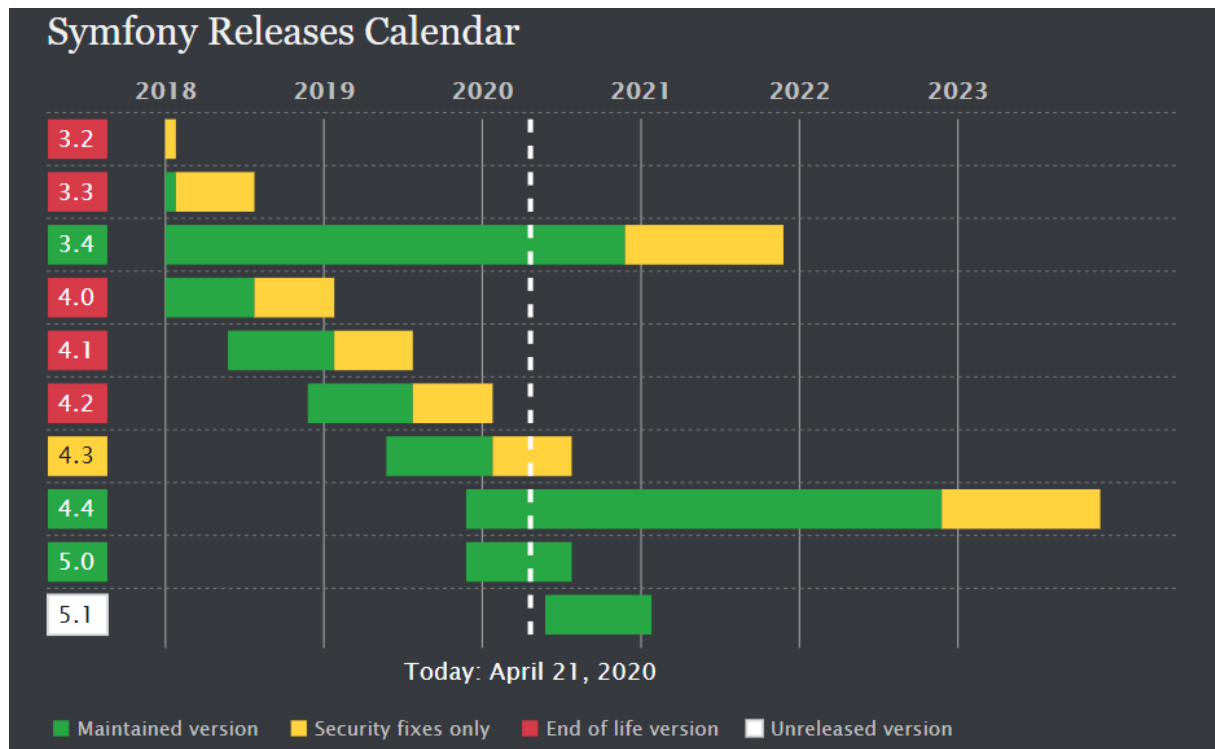
```

Figure 3

### 1.3 Version de Symfony et PHP

La version de Symfony initial de l'application est la 3.1.

Cette version a été mis en ligne en mai 2016 n'est plus maintenu depuis 2017 et n'apparaît plus dans la roadmap.



Passer sur la version 4.4 (LTS = Long Term Support) de Symfony permet de recevoir des mises à jour jusqu'en Novembre 2020 et des correctifs de sécurité jusqu'en Novembre 2022.

Seconde optimisation : Upgrader la version de PHP

La version de PHP utilisé est la 5.5.9, passer à la version 7.2 de PHP augmentera les performances au niveau de l'utilisation des ressources et offrira la possibilité d'utiliser les dernières fonctionnalités de PHP comme le typage des variables qui rendra le code plus stable.

Comparaison des performances de Symfony 3.1 avec PHP 5 et la version 4.4 de Symfony avec PHP 7.3 sur le site php benchmarks :

Release 3.1 6,652		PHP 5.6				VERSION Symfony 3.1	
Benchmark	Config	Concurrency 1	Concurrency 5	Concurrency 10	Concurrency 20	Score	
Hello world	?	3.4 ms 295 rq/sec	3.6 ms 1,373 rq/sec	7.3 ms 1,371 rq/sec	14.7 ms 1,364 rq/sec	2,748	i a </>
REST API	?	8.5 ms 117 rq/sec	6.5 ms 771 rq/sec	12.9 ms 776 rq/sec	26 ms 770 rq/sec	3,904	i a </>

Release 4.4 23,467		PHP 7.3				VERSION Symfony 4.4	
Benchmark	Config	Concurrency 1	Concurrency 5	Concurrency 10	Concurrency 20	Score	
Hello world	?	1.3 ms 791 rq/sec	1.1 ms 4,696 rq/sec	1.9 ms 5,232 rq/sec	3.8 ms 5,226 rq/sec	9,302	i a </>
REST API	?	2 ms 495 rq/sec	2 ms 2,455 rq/sec	3.7 ms 2,697 rq/sec	7.3 ms 2,746 rq/sec	14,165	i a </>

## 1.4 La sécurité :

Le niveau de sécurité initial est insuffisant, il permet d'accéder à certaines pages sans être authentifié. La page qui liste les utilisateurs /users et qui permet leur modification /users/{id}/edit est accessible sans authentification. Un correctif doit être apporté pour que cette page soit accessible seulement à un utilisateur ayant les droits ROLE\_ADMIN.

Des améliorations sont à apporter au formulaire d'authentification, il n'est pas protégé contre la faille CSRF.

Source : [https://symfony.com/doc/current/security/form\\_login\\_setup.html](https://symfony.com/doc/current/security/form_login_setup.html)

## 1.5 Mesure de performance BlackFire du code initial

### 1.5.1 Mesure de la page Home

```
D:\Users\Jeromes2\Desktop\Developpement\Todoandco (master -> origin)
λ blackfire curl http://127.0.0.1:8000/
Profiling: [#####] 10/10
Blackfire cURL completed
Graph URL https://blackfire.io/profiles/8df2f593-4be8-4de1-9b80-d526b0793326/graph
No tests! Create some now https://blackfire.io/docs/cookbooks/tests
3 recommendations https://blackfire.io/profiles/8df2f593-4be8-4de1-9b80-d526b0793326/graph?settings%5BtabPane%5D=recommendations

Wall Time      1.28s
I/O Wait       682ms
CPU Time       603ms
Memory         14.1MB
Network        n/a      n/a      n/a
SQL            n/a      n/a
```

### 1.5.2 Mesure de la page Login

```
D:\Users\Jeromes2\Desktop\Developpement\Todoandco (master -> origin)
λ blackfire curl http://127.0.0.1:8000/login
Profiling: [#####] 10/10
Blackfire cURL completed
Graph URL https://blackfire.io/profiles/1d148a56-bbbf-440e-986f-6d1f30d5e09e/graph
No tests! Create some now https://blackfire.io/docs/cookbooks/tests
4 recommendations https://blackfire.io/profiles/1d148a56-bbbf-440e-986f-6d1f30d5e09e/graph?settings%5BtabPane%5D=recommendations

Wall Time      660ms
I/O Wait       318ms
CPU Time       342ms
Memory         14.4MB
Network        n/a      n/a      n/a
SQL            n/a      n/a
```

### 1.5.3 Mesure de la page Logout

```
D:\Users\Jeromes2\Desktop\Developpement\Todoandco (master -> origin)
λ blackfire curl http://127.0.0.1:8000/logout
Profiling: [#####] 10/10
Blackfire cURL completed
Graph URL https://blackfire.io/profiles/677ed7a4-0bc8-41f2-8667-cd108f4f170e/graph
No tests! Create some now https://blackfire.io/docs/cookbooks/tests
3 recommendations https://blackfire.io/profiles/677ed7a4-0bc8-41f2-8667-cd108f4f170e/graph?settings%5BtabPane%5D=recommendations

Wall Time      582ms
I/O Wait       274ms
CPU Time       308ms
Memory         13.8MB
Network        n/a      n/a      n/a
SQL            n/a      n/a
```

## 1.5 Mesure de performance BlackFire du code initial (suite)

### 1.5.4 Mesure de la page Tasks

```
D:\Users\Jeromes2\Desktop\Developpement\Todoandco (master -> origin)
λ blackfire curl http://127.0.0.1:8000/tasks
Profiling: [#####] 10/10
Blackfire cURL completed 41ms
Graph URL https://blackfire.io/profiles/e62c0183-2ace-4a4f-bea8-2fae7ff2fca5/graph
No tests! Create some now https://blackfire.io/docs/cookbooks/tests
3 recommendations https://blackfire.io/profiles/e62c0183-2ace-4a4f-bea8-2fae7ff2fca5/graph?settings%5BtabPane%5D=recommendations
n/a n/a n/a
Wall Time 686ms n/a n/a
I/O Wait 349ms n/a n/a
CPU Time 337ms
Memory 14.1MB
Network n/a 1 n/a re n/a Fire de la page Home
SQL n/a n/a
```

### 1.5.5 Mesure BlackFire de la page Users

```
D:\Users\Jeromes2\Desktop\Developpement\Todoandco (master -> origin)
λ blackfire curl http://127.0.0.1:8000/users
Profiling: [#####] 10/10
Blackfire cURL completed
Graph URL https://blackfire.io/profiles/b0977e4f-9fe2-40df-8c1a-83153e3bbeb8/graph
No tests! Create some now https://blackfire.io/docs/cookbooks/tests
6 recommendations https://blackfire.io/profiles/b0977e4f-9fe2-40df-8c1a-83153e3bbeb8/graph?settings%5BtabPane%5D=recommendations
16 n/a n/a n/a
Wall Time 1.15s n/a n/a n/a
I/O Wait 572ms n/a n/a n/a
CPU Time 579ms n/a n/a n/a
Memory 20.5MB
Network 20 n/a n/a n/a login path: login
SQL 21 n/a n/a n/a check path: login check
```



### 1.5.6 Synthèse et conclusion des mesures

Mesure avant modifications			Mesure après modifications
Routes	Temps s ou ms	Mémoire mb	Voir la partie 2 Améliorations du code initial backend / frontend page 24
/	1.28 s	14.1 mb	
/login	660 ms	14.4 mb	
/logout	582 ms	13.8 mb	
/tasks	686 ms	14.1 mb	
/users	1.15 s	20.5 mb	
Environnement de developpement	Windows		

Grace aux données récoltées par BlackFire on peut comprendre pourquoi une page mettra 1.28 s pour s'afficher et consommera 14.1 mb de mémoire.

Blackfire nous donne les Temps d'accès, l'exécution processeur, l'occupation en mémoire, temps de la requête sql ainsi nous pouvons voir ce qui peut être amélioré pour rendre l'application plus réactive.

L'article « Vitesse de chargement : pourquoi et comment l'améliorer ? » nous informe que « La vitesse de chargement d'un site est la première impression donnée au visiteur. A l'heure où tout doit être instantané, même sur mobile, l'utilisateur n'attend plus. Si votre site met trop longtemps à charger, il est susceptible de quitter la page pour retourner à sa recherche, ce qui va augmenter votre taux de rebond\*. [Des études ont montré que 47% des utilisateurs s'attendent à ce qu'une page se charge en moins de 2 secondes et que 40% des internautes ferment une page si elle met plus de 3 secondes à s'afficher.](#) Diminuer le temps de chargement va permettre de retenir l'utilisateur plus longtemps, d'augmenter le nombre de conversions (achats, inscriptions) et de donner une image positive à votre marque. Ainsi, le visiteur sera plus enclin à revenir sur vos pages. »

Source : [Genius Interactive](#)

## 1.6 Principes SOLID

### 1.7 Frontend

Lors de nos tests de l'interface nous avons constaté un design manquant d'ergonomie et de fluidité.

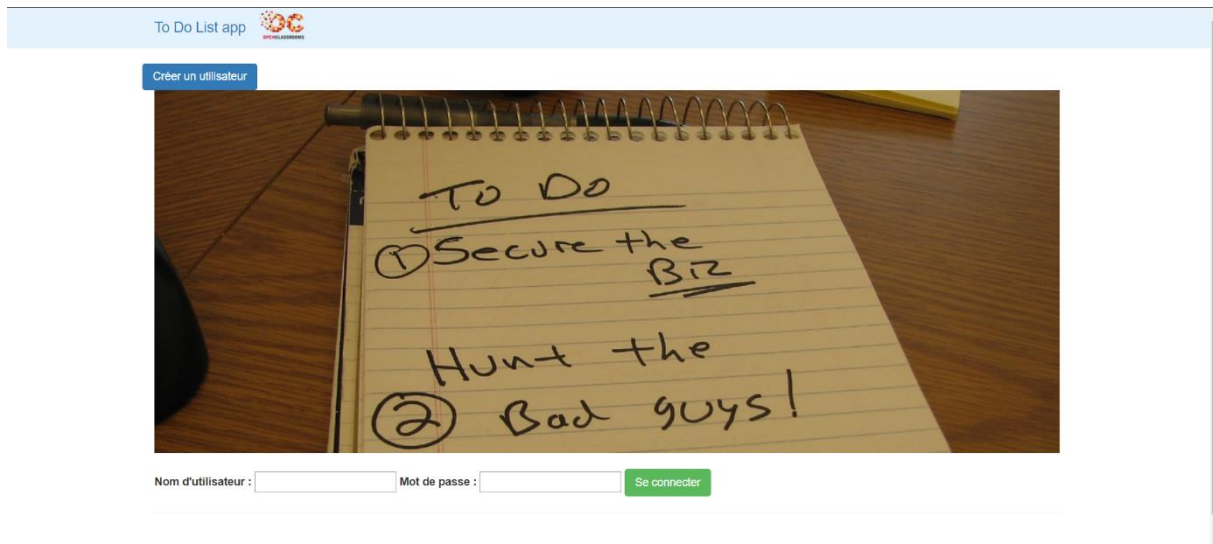


Figure 5 : page de login

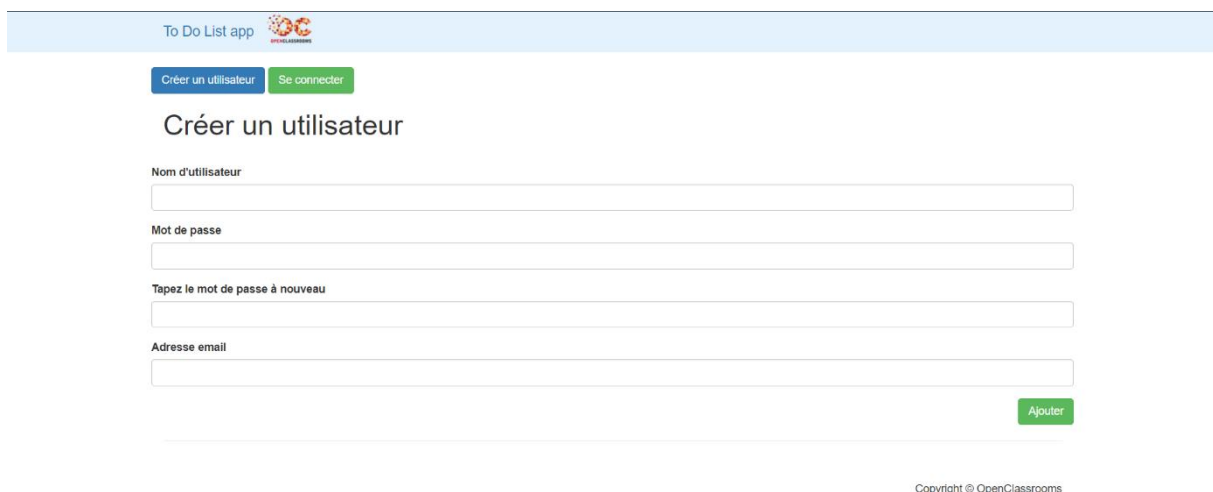


Figure 6 : page de création d'un utilisateur

Nous voyons sur la figure 5 ci-dessus le bouton « créer un utilisateur » est situé en haut de page au-dessus de l'image principale de la page et le formulaire de connexion sous l'image.

Sur la figure 6 le formulaire a des proportions trop importantes.

Un test sur la responsive nous montre qu'il faut l'améliorer.

Une mise en forme plus adaptée au standard actuel est nécessaire. Un design plus poussé rendra l'application plus attractive et intuitif à utiliser et offrira une meilleure expérience.

## 1.8 Conclusion

L'audit de l'application ToDoList nous montre qu'il y a des améliorations à réaliser au niveau backend et frontend. Je propose l'amélioration des points suivant :

### 1.8.1 Backend

Points d'améliorations	Backend
<b>Symfony</b>	Passage à la version 4.4
<b>Php</b>	Passage à la version 7.3
<b>Sécurité</b>	Voir la documentation de la sécurité
<b>SOLID</b>	Mise en place des principes SOLID
<b>PSR</b>	Respect des règles PSR
<b>Maintenabilité</b>	Rendre le code plus facilement maintenable
<b>Documentation</b>	Mise en place de la documentation

### 1.8.2 Frontend

	Frontend
Points d'améliorations	
<b>Design</b>	Amélioration des interfaces
<b>Responsive</b>	Amélioration de la responsive
<b>Contenu</b>	Ajout d'une page terminée et d'une page d'erreur

## - 2 – Améliorations du code initial backend / frontend

### 2.1 Migration de php et symfony

Mise à jour de l'environnement de développement et migration vers la version de Symfony 4.4 LTS (long time service).

Points d'améliorations traités	
<b>Symfony</b>	Passage à la version 4.4
<b>Php</b>	Passage à la version 7.3

----- Symfony -----		
Version	4.4.7	2.1 Amélioration du backend
Long-Term Support	Yes	
End of maintenance	11/2022	
End of life	11/2023	Rappel des améliorations backend à apporter :
-----		
Kernel	-----	
Type	d'améli App\Kernel	Backend
Environment	Symfony dev	Passage à la version 4.4
Debug	Php true	Passage à la version 7.3
Charset	UTF-8	
Cache directory	./var/cache/dev (10.6 MiB)	Voir la documentation de la sécurité
Log directory	./var/log (1.6 MiB)	Respecter les principes SOLID
	PSR	Respecter les règles PSR
-----		
PHP	Maintenabilité	Rendre le code plus facilement maintenable
-----		
Version	7.3.12	
Architecture	64 bits	
Intl locale	fr_FR	

Figure 5 Capture d'écran de l'environnement de développement et la version de Symfony de l'application

## 2.2 Amélioration de la sécurité

Point d'amélioration traité	
Sécurité	Voir la documentation de la sécurité

Cet audit de qualité et d'améliorations est joint avec une documentation sur le fonctionnement et l'implémentation de la sécurité dans une application Symfony que vous trouverez dans le dossier : [Todoandco\docs\audit\Sécurité\Documentation\\_sécurité\\_de\\_symfony.pdf](#)

Les améliorations apportées dans l'application :

- Modification du SecurityController :

Suppression des méthodes sans retour « Login\_Check » et « Logout » qui indique seulement la route pour la vérification du login et celle de la déconnexion. Les routes ont été ajoutées dans le fichier « LoginFormAuthenticator.php » pour la connexion (/login) security.yaml pour la déconnexion (/logout).

- Modification du formulaire d'authentification Login\_Form:

Comme mentionné dans l'audit de code initiale le formulaire ne respecte pas les bonnes pratiques de Symfony.

Modifications apportées :

Vérification du CSRF Token au moment de l'authentification : permet de protéger les données envoyées par un utilisateur.

```
<input type="hidden" name="_csrf_token" value="{ { csrf_token('authenticate') } }">
```

Figure 6 Capture du champs caché « \_csrf\_token »

Afin que le processus fonctionne il a fallu également ajouter la classe « LoginFormAuthenticator.php » qui étend de la classe « AbstractFormLoginAuthenticator.php » qui permet de vérifier toutes les données reçues comme le nom d'utilisateur, le mot de passe crypté.

- Correction de l'anomalie à l'ajout d'une tâche :  
Ajout d'une relation entre l'entité Task et User.

```
/**
 * @var User
 *
 * @ORM\ManyToOne(targetEntity="App\Entity\User", inversedBy="task")
 * @ORM\JoinColumn(onDelete="cascade")
 */
private $user;
```

Figure 7 Capture de la relation ManyToOne dans le fichier « src/entity/Task »

```
/**
 * @var Task
 *
 * @ORM\OneToMany(targetEntity="App\Entity\Task", mappedBy="user", orphanRemoval=false)
 */
private $task;
```

Figure 8 Capture de la relation OneToMany dans le fichier « src/entity/User »

Maintenant lors de l'ajout d'une tâche celle-ci est rattaché à l'utilisateur connecté (donc son auteur).

```
if ($form->isSubmitted() && $form->isValid()) {

    $task->setUser($this->tokenStorage->getToken()->getUser());

    $this->repository->save($task);
}
```

Figure 9 Capture de la partie de la méthode qui rattache l'utilisateur à une tâche

Avec cette relation entre l'utilisateur et ses tâches celui-ci peut visualiser, modifier, supprimer et marquer uniquement ses tâches. Une vérification est faite avant chaque action, le système vérifie que la tâche appartient bien à l'utilisateur connecté.

```
if ($this->authorization->isGranted( attributes: TaskVoter::DELETE, $task) === true) {
```

Figure 10 Capture de la partie de la méthode qui vérifie si l'utilisateur connecté est bien l'auteur de la tâche

- Type « Hinting » :  
La version initiale de PHP (5) ne permettait pas de typer les retours de méthodes ou des paramètres. Le typage a été ajouté suite au passage de la version 7 de PHP.
- « Validation password » :  
Ajout de la validation sur le mot de passe utilisateur car aucun été paramétré initialement.

- La commande « CreateAdminCommand » dans le dossier (App/Command/) pour faciliter l'ajout d'un administrateur à l'application en ligne de commande.

```
D:\Users\Jeromes2\Desktop\Developpement\Todoandco (tests)
λ php ./bin/console app:create-admin admin password admin@email.fr
Admin Creator
=====

You are about to create an admin-user.
Username: admin
Password: password
Email: admin@email.fr
Role: ROLE_ADMIN
Admin successfully created
```

Figure 11 Capture d'écran de la commande de création d'un administrateur

- Optimisation des paramètres de PHP :  
Les recommandations de Symfony sont de modifier quelques paramètres du fichier PHP.ini pour optimiser le chargement.

Réglage du RealPath et de l'OP Cache

```
1 ; php.ini
2 ; maximum memory that OPcache can use to store compiled PHP files
3 opcache.memory_consumption=256
4
5 ; maximum number of files that can be stored in the cache
6 opcache.max_accelerated_files=20000
```

Figure 12 Capture de la configuration de php.ini de la documentation de Symfony

Source : [Documentation Symfony opcache](#)

- Mise en place du cache avec REDIS :

Afin d'améliorer les performances de l'application, la mise en cache des données est primordiale.

Le cache REDIS à été mis en place pour optimiser les requêtes Doctrine et obtenir un affichage rapide de la liste des tâches. Plusieurs caches sont gérés par Redis, le « metadataCache », le « queryCache » et le « resultCache ». Paramétrage du « metadataCache » et « queryCache » se situant fichier App/Config/doctrine.yaml.

```
orm:
  auto_generate_proxy_classes: true
  naming_strategy: doctrine.orm.naming_strategy.underscore_number_aware
  auto_mapping: true

  mappings:
    App:
      is_bundle: false
      type: annotation
      dir: '%kernel.project_dir%/src/Entity'
      prefix: 'App\Entity'
      alias: App
  metadata_cache_driver:
    id: redis
  query_cache_driver:
    id: redis
  result_cache_driver:
    id: redis
```

Figure 13 Capture de la configuration de redis pour Doctrine



Création du fichier App/Config/redis.yaml ou l'on indique l'adresse du serveur Redis.

```
snc_redis:
  clients:
    default:
      type: predis
      alias: default
      dsn: redis://127.0.0.1:6379
    doctrine:
      type: predis
      alias: doctrine
      dsn: redis://127.0.0.1:6379
  doctrine:
    metadata_cache:
      client: doctrine
      entity_manager: default
      document_manager: default
    result_cache:
      client: doctrine
      entity_manager: default
    query_cache:
      client: doctrine
      entity_manager: default
```

Figure 14 Capture d'écran de la configuration de Redis

Pour la mise en cache des données récupérer avec Doctrine il l'indiqué dans une requête (DQL ou avec le QueryBuilder).

```
/**
 * @return int|mixed|string
 */
public function listTask()
{
    return $this->createQueryBuilder( alias: 't')
        ->orderBy( sort: 't.createdAt', order: 'DESC')
        ->getQuery()
        ->enableResultCache( lifetime: 0)
        ->useQueryCache( bool: true)
        ->setQueryCacheLifetime( timeToLive: 60)
        ->setResultCacheId( id: 'tasks')
        ->getResult();
}
```

Figure 15 Capture du requête DQL pour la mise en cache dans le fichier TaskRepository

## 2.3 Amélioration de la qualité

Point d'amélioration traité	
SOLID	Mise en place des principes SOLID
PSR	Respect des règles PSR

Les classes TaskController et UserController comme nous l'avons vu dans la page 9, de ce document s'occupait de toutes les tâches. Maintenant, les contrôleurs s'occupent uniquement d'un rôle (create, read, delete, update), c'est le principe de single responsibility principle, la logique métier des méthodes traitant les données de formulaires a été déplacé dans des Handler (App/Handler).

Les méthodes s'occupant d'afficher la liste des tâches ou des utilisateurs on dorénavant leur propre contrôleur.

Les contrôleurs étendaient de la classe Controller (classe dépréciée sur les nouvelles versions de Symfony elle et remplacer par la classe AbstractController).

La classe Controller a donc été supprimé, mais pour de meilleure performance je n'ai pas étendu la classe vers sa remplaçante mais plutôt choisi d'injecter les services vraiment utiles aux contrôleurs.

Ajout de la PHPDOC pour une meilleure lisibilité du code.

Source : [Documentation de Symfony](#)

```
class TaskController extends Controller
{
    /**
     * @Route("/tasks", name="task_list")
     */
    public function listAction()
    {
        return $this->render('task/list.html.twig', ['tasks' => $this->getDoctrine()->getRepository('AppBundle:Task')->findAll()]);
    }

    /**
     * @Route("/tasks/create", name="task_create")
     */
    public function createAction(Request $request)
    {
        $task = new Task();
        $form = $this->createForm(TaskType::class, $task);

        $form->handleRequest($request);

        if ($form->isValid()) {
            $em = $this->getDoctrine()->getManager();

            $em->persist($task);
            $em->flush();

            $this->addFlash('success', 'La tâche a été bien été ajoutée.');
```

Figure 16 TaskController avant modification

```

public function createTask(
    Request $request,
    TaskCreateHandler $taskCreateHandler
) {
    $task = new Task();

    $form = $this->formFactory->create( type: TaskType::class, $task)
        ->handleRequest($request);

    if ($taskCreateHandler->handle($form, $task)) {

        return new RedirectResponse(
            $this->urlGenerator->generate( name: 'task_list'),
            status: RedirectResponse::HTTP_FOUND
        );
    }

    return new Response(
        $this->twig->render(
            name: 'task/create.html.twig', [
                'form' => $form->createView()
            ]
        ), status: Response::HTTP_OK
    );
}

```

Figure 17 TaskCreateController après modification

```

public function handle(FormInterface $form, Task $task)
{
    if ($form->isSubmitted() && $form->isValid()) {

        $task->setUser($this->tokenStorage->getToken()->getUser());

        $this->repository->save($task);

        $this->messageFlash->getFlashBag()->add('success', 'La tâche a bien été ajoutée.');
```

Figure 18 TaskCreateHandler en lien avec le TaskCreateController

## Vérification de la qualité du et du respect des PSR avec PHP\_CS\_FIXER

```
C:\Users\Jeromes2\AppData\Roaming\Composer\vendor\bin
λ php-cs-fixer fix D:/Users/Jeromes2/Desktop/Developpement/ToDoandco/src
Loaded config default.
Using cache file ".php_cs.cache".
 1) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Command\AdminCreateCommand.php
 2) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Controller\TaskCreateController.php
 3) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Controller\TaskDeleteController.php
 4) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Controller\TaskDoneController.php
 5) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Controller\TaskEditController.php
 6) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Controller\TaskListController.php
 7) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Controller\TaskSwitchController.php
 8) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Controller\UserCreateController.php
 9) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Controller\UserDeleteController.php
10) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Controller\UserEditController.php
11) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Controller\UserEditPasswordController.php
12) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Controller\UserListController.php
13) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\DataFixtures\UserFixtures.php
14) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Form\TaskType.php
15) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Form\UserCreateType.php
16) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Form\UserEditPasswordType.php
17) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Form\UserEditType.php
18) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Form\UserType.php
19) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Handler\TaskCreateHandler.php
20) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Handler\TaskEditHandler.php
21) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Handler\UserCreateHandler.php
22) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Handler\UserEditHandler.php
23) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Handler\UserEditPasswordHandler.php
24) D:\Users\Jeromes2\Desktop\Developpement\ToDoandco\src\Voter\TaskVoter.php

Fixed all files in 0.292 seconds, 12.000 MB memory used

C:\Users\Jeromes2\AppData\Roaming\Composer\vendor\bin
```

Figure 19 Capture d'écran utilisation de PHP\_CS\_FIXER

[Source : PHP\\_CS\\_FIXER](#)

## 2.4 Amélioration de la maintenabilité et des performances

Point d'amélioration traité	
Maintenabilité	Rendre le code plus facilement maintenable
Documentation	Mise en place de la documentation

### 2.4.1 Analyse CodeClimate

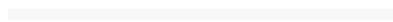
Avec les différentes modifications apportées au projet le nombre de fichiers passe de 56 à 102, ajout de nouveaux contrôleurs, listener, handler, repository ainsi qu'un code maintenable plus facilement. Le code venant de librairies externes et celle de Symfony a été ignoré lors des mesures.

#### Breakdown

102 FILES



MAINTAINABILITY



TEST COVERAGE

#### Codebase summary

MAINTAINABILITY



4 days

TEST COVERAGE



#### Repository stats

CODE SMELLS

6

DUPLICATION

11

OTHER ISSUES

0

Figure 20 Capture d'écran nouvelle mesure CodeClimate

## 2.4.2 Analyse PHPMetrics

Le nouveau rapport de PHPMetrics montre une très bonne maintenabilité des classes (représenté par les cercles vert), un résultat de complexité des classes passant de 6.07 à 2 grâce au respect du principe SOLID : une classe une responsabilité. *Deux erreurs apparaissent elles demandent de passer les classes entity et repository en classe abstraite (ce qui est impossible pour le bon fonctionnement de l'application, je pense donc à un bug de PHPMetrics)*

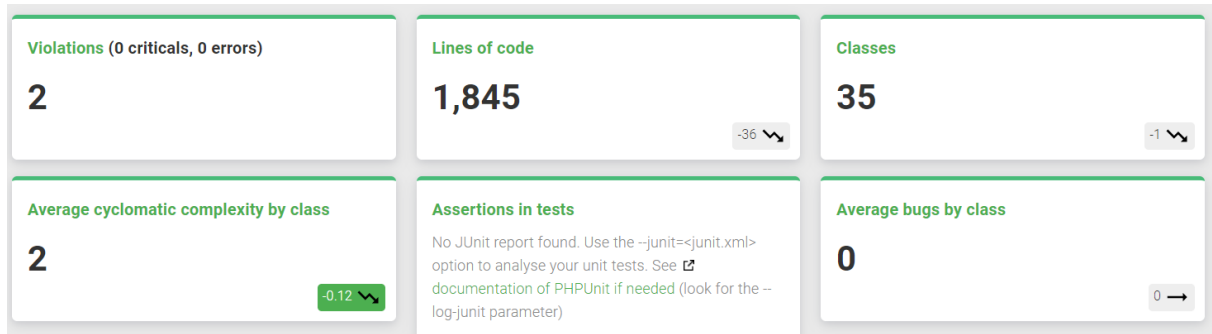


Figure 21 Capture d'écran nouvelle mesure PHPMetrics partie 1

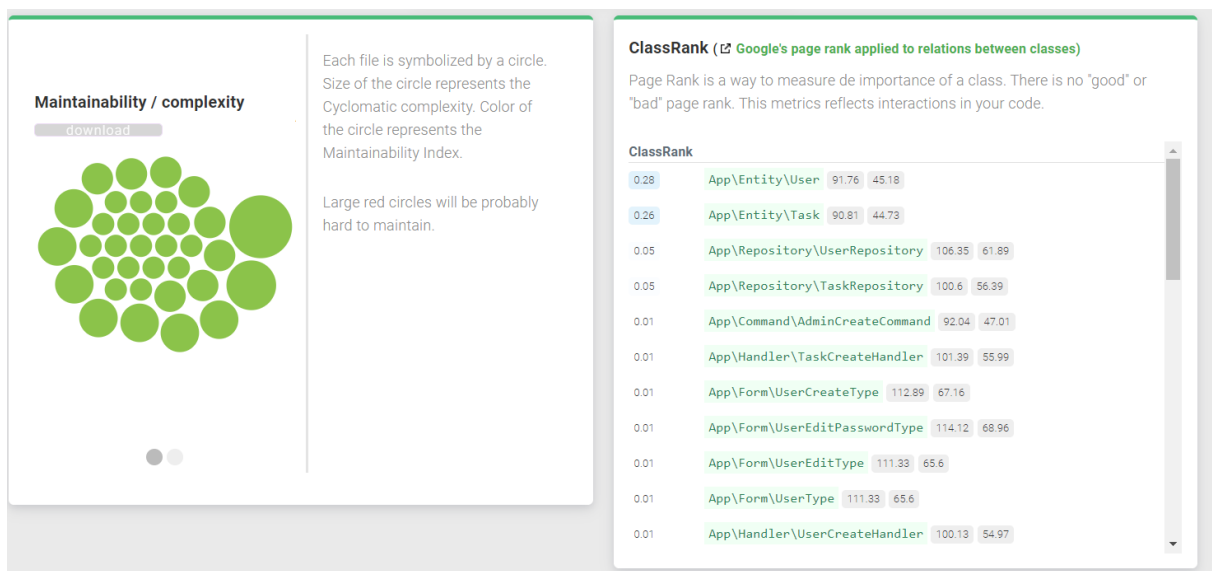


Figure 22 Capture d'écran nouvelle mesure PHPMetrics partie 2

### 2.4.3 Analyse Blackfire

La nouvelle analyse montre un léger gain au niveau du temps d'accès des différentes routes. Le passage à PHP 7 et Symfony 4.4 a eu son effet. L'application gagne en fluidité, la navigation est plus agréable surtout lors de l'affichage des tâches grâce à l'implémentation du cache REDIS, et des optimisations du fichier PHP.INI préconisé par Symfony.

```
D:\Users\Jeromes2\Desktop\Developpement\Todoandco (tests)
λ blackfire curl http://127.0.0.1:8000/
Profiling: [#####] 10/10
Blackfire cURL completed.
Graph https://blackfire.io/profiles/f803e4e2-4e0b-411d-827a-50390e689538/graph
No tests! Create some now https://blackfire.io/docs/cookbooks/tests
No recommendations
Wall Time 326ms
Memory 16.8MB
Network n/a n/a n/a
SQL n/a n/a
```

Figure 23 Mesure de la route /

```
D:\Users\Jeromes2\Desktop\Developpement\Todoandco (tests)
λ blackfire curl http://127.0.0.1:8000/login
Profiling: [#####] 10/10
Blackfire cURL completed
Graph https://blackfire.io/profiles/9d4d4576-1d49-445c-b77e-7bfd244631b3/graph
No tests! Create some now https://blackfire.io/docs/cookbooks/tests
No recommendations
Wall Time 336ms
Memory 17.6MB
Network n/a n/a n/a
SQL n/a n/a
```

Figure 24 Mesure de la route /login

```
D:\Users\Jeromes2\Desktop\Developpement\Todoandco (tests)
λ blackfire curl http://127.0.0.1:8000/logout
Profiling: [#####] 10/10
Blackfire cURL completed
Graph https://blackfire.io/profiles/72454a1a-af34-4e82-a7f1-16905a9d48c7/graph
No tests! Create some now https://blackfire.io/docs/cookbooks/tests
No recommendations
Wall Time 304ms
Memory 16.8MB
Network n/a n/a n/a
SQL n/a n/a
```

Figure 25 Mesure de la route /logout

```

D:\Users\Jeromes2\Desktop\Developpement\Todoandco (tests)
λ blackfire curl http://127.0.0.1:8000/tasks
Profiling: [#####] 10/10
Blackfire cURL completed
Graph https://blackfire.io/profiles/76270204-dd07-4876-aeb7-7fce3e9ea6c2/graph
No tests! Create some now https://blackfire.io/docs/cookbooks/tests
No recommendations

Memory      20.6MB
Wall Time   442ms
Network     n/a      n/a      n/a
SQL         n/a      n/a

```

Figure 26 Mesure de la route /tasks

```

D:\Users\Jeromes2\Desktop\Developpement\Todoandco (tests)
λ blackfire curl http://127.0.0.1:8000/users
Profiling: [#####] 10/10
Blackfire cURL completed
Graph https://blackfire.io/profiles/c38f2555-24b1-4d38-9579-6713e5eeef4d/graph
No tests! Create some now https://blackfire.io/docs/cookbooks/tests
No recommendations

Memory      20.6MB
Wall Time   416ms
Network     n/a      n/a      n/a
SQL         n/a      n/a

```

Figure 27 Mesure de la route /users

Routes	Mesure avant modifications		Mesure après modifications	
	Temps s ou ms	Mémoire mb	Temps s ou ms	Mémoire mb
/	1.28 s	14.1 mb	326 ms	16.8 mb
/login	660 ms	14.4 mb	336 ms	17.6 mb
/logout	582 ms	13.8 mb	304 ms	16.8 mb
/tasks	686 ms	14.1 mb	442 ms	20.6 mb
/users	1.15 s	20.5 mb	416 ms	20.6 mb
Environnement de développement	Windows		Windows	



### - 3 – Améliorations du frontend

Les points suivants ont été améliorés :

- ✓ L'application est basée sur la version 3 de bootstrap, je l'ai remplacé par la version 4 via CDN qui améliore le chargement et qui apporte les dernières nouveautés du langage CSS offrant un design plus moderne avec des fonctionnalités plus poussée.
- ✓ Suppression du code inutile dans le fichier Shop-homepage.css.
- ✓ Lorsqu'un administrateur est connecté deux liens apparaissent dans la barre de navigation : gérer les utilisateurs (si l'utilisateur a le rôle « admin ») et créer un utilisateur.
- ✓ Réorganisation de la page d'accueil. Alignement du titre, de l'image et des boutons.
- ✓ Implémentation du nouveau logo d'Openclassrooms
- ✓ Réorganisation de l'affichage d'une tâche. Les boutons sont maintenant dans le cadre.
- ✓ Ajout d'une bannière obligatoire pour mentionner aux visiteurs que l'application utilise des cookies.
- ✓ Amélioration de l'ergonomie avec des titres de page dynamique.
- ✓ Ajout de la page tâche terminé :
- ✓ Le bouton « consulter les tâches terminées » était bien présent mais la page n'existe pas. Ajout d'une information : la date à laquelle la tâche a été marquée comme terminée.
- ✓ Ajout d'une page d'erreur.
- ✓ Réorganisation générale de l'application pour la rendre responsive.

## - 4 – Tests unitaires et fonctionnels

Réalisation des tests unitaires et fonctionnels avec PHP Unit.

La couverture des tests unitaires et fonctionnels avec PHPUnit couvre 100% du code.

Les tests suivants ont été réalisés :

- Réponses et redirection de chaque méthode des contrôleurs.
- Scénario d'ajout, de modification et suppression d'une tâche.
- Scénario d'ajout, de modification et suppression d'un utilisateur.

```
D:\Users\Jeromes2\Desktop\Developpement\Todoandco (tests)
λ php ./bin/phpunit
PHPUnit 7.5.20 by Sebastian Bergmann and contributors.

Testing Project Test Suite
..... 65 / 88 ( 73%)
..... 88 / 88 (100%)

Time: 17.51 seconds, Memory: 54.00 MB

OK (88 tests, 106 assertions)
```

Figure 28 Résultat des tests unitaires et fonctionnels

		Code Coverage					
		Lines		Functions and Methods		Classes and Traits	
Total		96.49%	385 / 399	91.95%	80 / 87	84.38%	27 / 32
Command		100.00%	31 / 31	100.00%	3 / 3	100.00%	1 / 1
Controller		95.57%	151 / 158	85.00%	17 / 20	76.92%	10 / 13
Entity		100.00%	44 / 44	100.00%	28 / 28	100.00%	3 / 3
Form		100.00%	53 / 53	100.00%	5 / 5	100.00%	5 / 5
Handler		100.00%	48 / 48	100.00%	10 / 10	100.00%	5 / 5
Listener		100.00%	8 / 8	100.00%	4 / 4	100.00%	1 / 1
Repository		100.00%	28 / 28	100.00%	9 / 9	100.00%	2 / 2
Security		72.73%	8 / 11	75.00%	3 / 4	0.00%	0 / 1
Voter		77.78%	14 / 18	25.00%	1 / 4	0.00%	0 / 1

Figure 29 Résultat graphique des tests

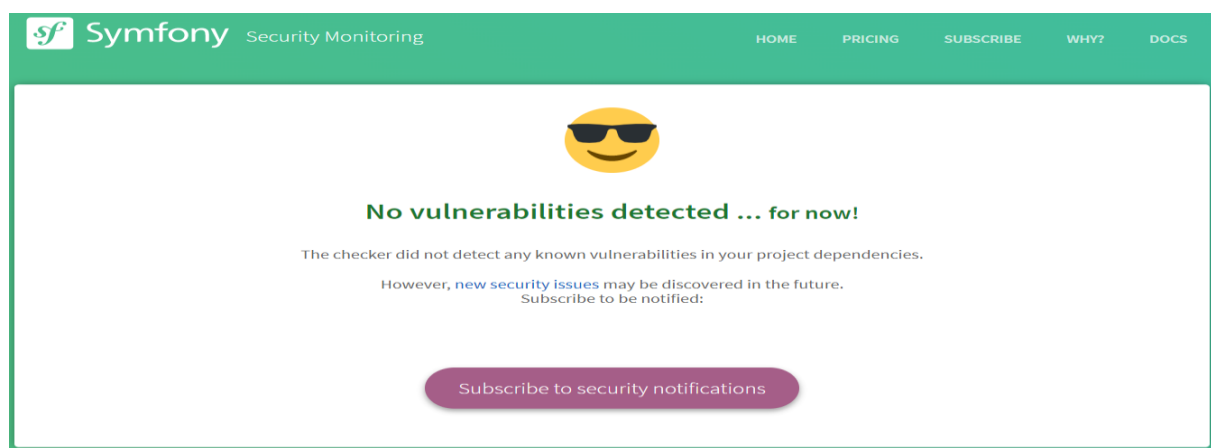


Figure 30 Tests de sécurité du fichier composer.json