

21/04/2020

Audit de qualité

Application ToDoList

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	10
1.8 Conclusion	11
1.8.1 Backend.....	11
1.8.2 Frontend	11

Introduction

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 :

Breakdown

56 FILES



TEST COVERAGE

Codebase summary

MAINTAINABILITY



2 wks

TEST COVERAGE



Repository stats

CODE SMELLS

26

DUPLICATION

36

OTHER ISSUES

0

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

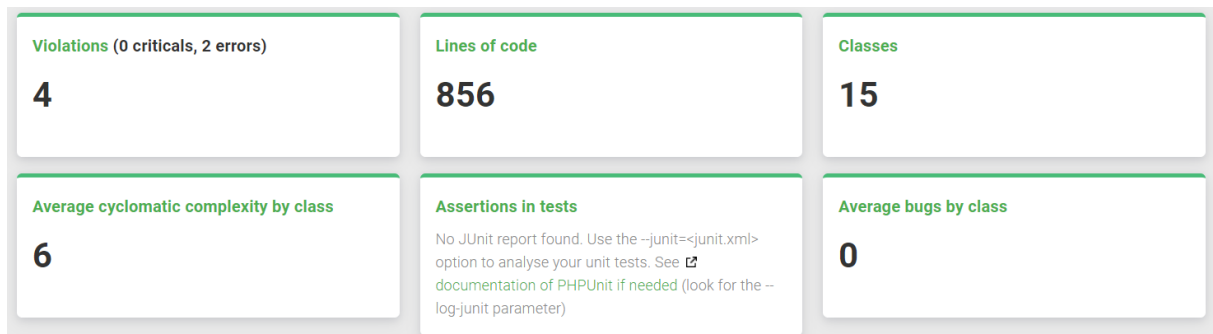
Ces analyses n'ont pas révélé de problèmes particuliers sur le code de l'application.

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
-----

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
-----

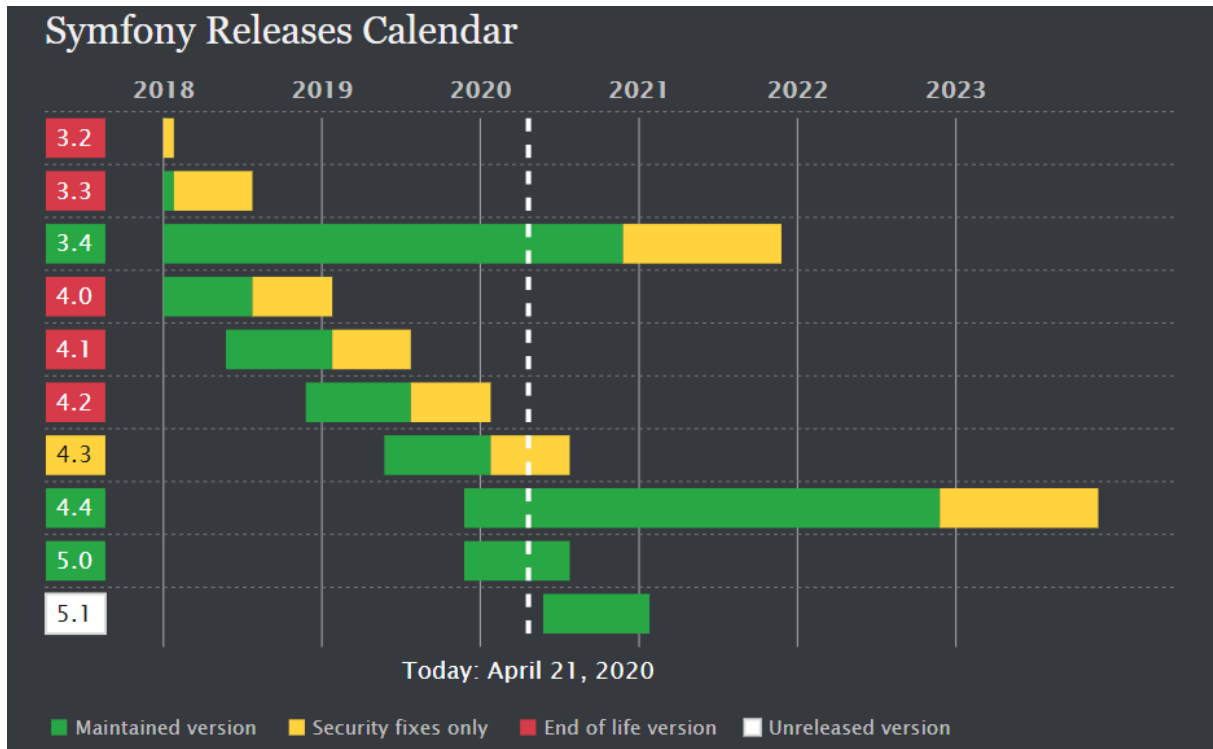
```

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

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	<div>2,748</div>
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	<div>3,904</div>

Release 4.4

23,467

PHP

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	<div>9,302</div>
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	<div>14,165</div>

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

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/setup.html
Profiling: [#####] 10/10 du code initial
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      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      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      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
I/O Wait 349ms
CPU Time 337ms
Memory 14.1MB
Network n/a 1 n/a re 1 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
I/O Wait 572ms
CPU Time 579ms
Memory 20.5MB
Network 20 n/a n/a n/a login path login
SQL 21 n/a n/a login path login check
```

1.5.6 Synthèse et conclusion des mesures

Mesure avant modifications			Mesure après modifications	
Routes	Temps s ou ms	Mémoire mb		
/	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

Un des principes SOLID dit qu'une classe, une fonction ou une méthode doit avoir une et une seule responsabilité. Le code sera implémenté en respectant certains des principes.

Principes		Définitions
S	Responsabilité unique (Single responsibility principle)	Une classe, une fonction ou une méthode doit avoir une et une seule responsabilité
O	Ouvert/fermé (Open/closed principle)	Une entité applicative (class, fonction, module ...) doit être ouverte à l'extension, mais fermée à la modification
L	Substitution de Liskov (Liskov substitution principle)	Une instance de type T doit pouvoir être remplacée par une instance de type G, tel que G sous-type de T, sans que cela ne modifie la cohérence du programme
I	Ségrégation des interfaces (Interface segregation principle)	Préférer plusieurs interfaces spécifiques pour chaque client plutôt qu'une seule interface générale
D	Inversion des dépendances (Dependency inversion principle)	Il faut dépendre des abstractions, pas des implémentations

Source : [Wikipédia](#)

Dans l'application, le fichier « **TaskController** » n'implémente pas le principe (SRP) qui explique qu'une classe n'a qu'une seule responsabilité. Le contrôleur de tâches implémente plusieurs responsabilités comme indiqué dans la figure 4.

```
/**
 * @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 ajoutée.');
```

Figure 4

1.7 Frontend

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

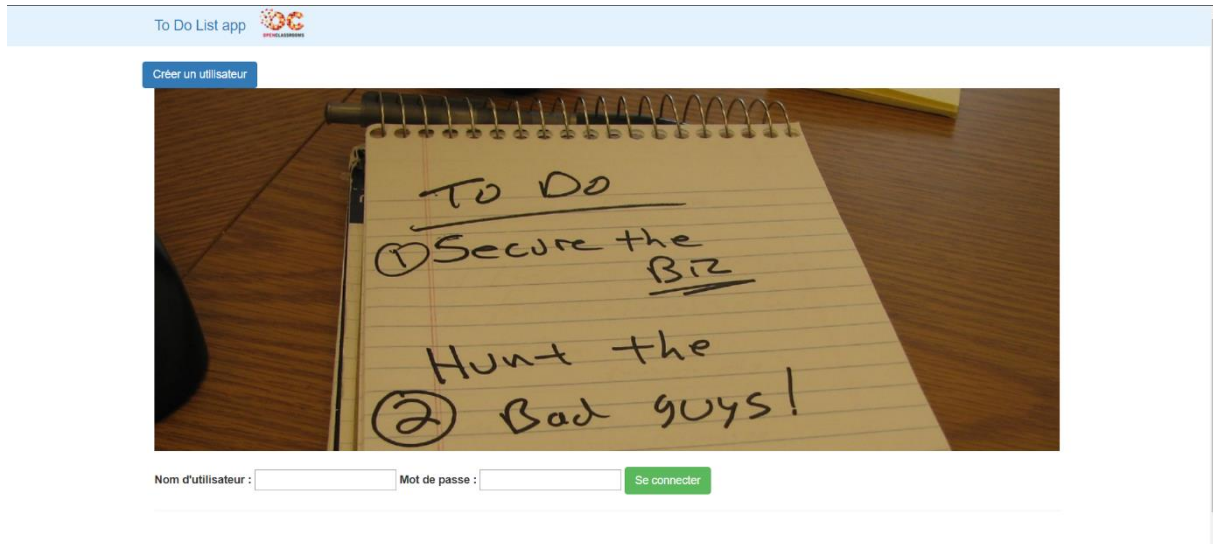


Figure 5 : page de login

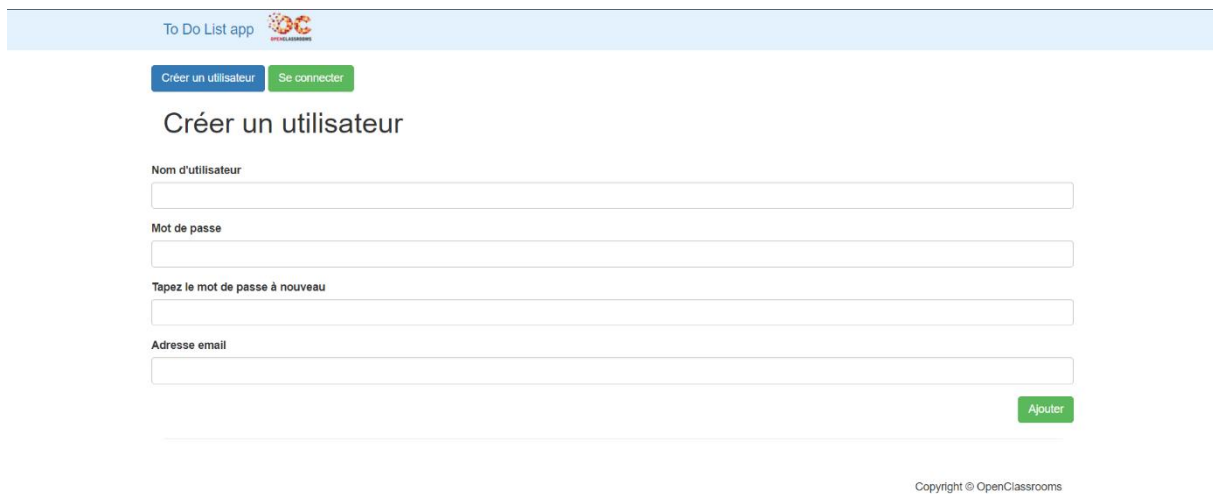


Figure 6 : page de création d'un utilisateur

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 adapté 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

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

1.8.2 Frontend

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