



# ToDo & Co

## Dispositif de communication en ligne

Audit de qualité de l'application « ToDo List »

Développé par la startup « ToDo & Co »

---

Liste de diffusion :  
• Sébastien Reille

Version : 1.0

Date de la dernière mise à jour : 13/11/18

# SOMMAIRE

<b>1 Résumé du projet.....</b>	<b>3</b>
1.1 Description du besoin.....	3
1.1.1 Corrections d'anomalies.....	3
1.1.1.1 Une tâche doit être attachée à un utilisateur.....	3
1.1.1.2 Choisir un rôle pour un utilisateur.....	3
1.1.2 Implémentation de nouvelles fonctionnalités.....	4
1.1.2.1 Autorisation.....	4
1.1.2.2 Implémentation de tests automatisés.....	4
<b>2 Audit du code.....</b>	<b>5</b>
2.1 Security:Check.....	6
2.2 composer outdated.....	6
2.3 Analyse automatisée du code.....	7
2.4 Lecture du code.....	7
2.5 Test manuel de l'application.....	8
2.6 Vérification des dépendances.....	8
2.7 Tests de performance.....	9
<b>3 Actions misent en œuvre.....</b>	<b>10</b>
3.1 Mise à jour des composants.....	10
3.2 Analyse automatisée du code.....	11
3.3 Lecture du code.....	12
3.4 Test manuel de l'application.....	12
3.5 Vérification des dépendances.....	12
3.6 Tests de performances.....	13
<b>4 Conclusion.....</b>	<b>14</b>

# 1 Résumé du projet

---

ToDo & Co est une startup qui a développé une application « ToDo List » permettant de gérer un ensemble de tâche.

L'entreprise vient tout juste d'être montée, et l'application a dû être développée à toute vitesse pour permettre de montrer à de potentiels investisseurs que le concept est viable (on parle de Minimum Viable Product ou MVP). Suite à une levée de fond, l'entreprise peut avancer sur le développement de l'application.

Ainsi, pour reprendre le développement de l'application (développé avec le framework PHP Symfony), ToDo & Co demande dans un premier temps un audit de qualité du code, puis de corriger les points suivants :

- l'implémentation de nouvelles fonctionnalités.
- la correction de quelques anomalies.
- l'implémentation de tests automatisés.

## 1.1 Description du besoin

### 1.1.1 Corrections d'anomalies

#### 1.1.1.1 *Une tâche doit être attachée à un utilisateur.*

Actuellement, lorsqu'une tâche est créée, celle-ci n'est pas rattachée à un utilisateur. Il est demandé d'apporter les corrections nécessaires afin qu'automatiquement, à la sauvegarde de la tâche, l'utilisateur actuellement 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".

#### 1.1.1.2 *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.

## **1.1.2 Implémentation de nouvelles fonctionnalités**

### **1.1.2.1 Autorisation**

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 questions.

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

### **1.1.2.2 Implémentation de tests automatisés**

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

Ces tests doivent être implémentés avec **PHPUnit**, vous pouvez aussi utiliser Behat pour la partie fonctionnelle.

Vous prévoyez des données de tests afin de pouvoir prouver le fonctionnement dans les cas explicités dans ce document.

Il 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 Audit du code

---

L'ensemble de l'audit du code est réalisé avec les applications :

- [Codacy](#) : Pour vérifiez le style, la sécurité, la duplication, la complexité et la couverture du code. Le tout en suivant la qualité du code tout au long du développement.
- [Travis](#) : L'intégration continue est la pratique consistant à fusionner fréquemment de petits changements de code, plutôt que de fusionner en un grand changement à la fin d'un cycle de développement. L'objectif est de créer des logiciels plus sains en développant et en testant par petits incréments. C'est là que Travis CI intervient.

En tant que plate-forme d'intégration continue, Travis CI prend en charge le processus de développement en construisant et testant automatiquement les changements de code, fournissant ainsi une rétroaction immédiate sur le succès du changement.

- [Code Climate](#) : Révision automatisée du code pour la couverture des tests, la maintenabilité et plus encore pour gagner du temps et fusionner en toute confiance.

Pour y parvenir, nous testerons les commandes et vérifierons les points suivant :

- `php bin/console security:check`
- `composer outdated`
- Analyse du code avec les applications Codacy / Travis / Code Climate
- Lecture du code
- Test de l'application
- Vérification et correction des dépendances dépréciées remonté par le profiler de Symfony.

## 2.1 Security:Check

Vérification des vulnérabilités des packages installés pour le fonctionnement de Symfony avec la commande :

- `php bin/console security:check`

```
zohac@ubuntu-server ~/www/dev feature/Issue1 php bin/console security:check
Sat Oct 20 18:21:22 2018 (5216): [Info] pgsql extension is not loaded, Blackfire SQL analyzer will be disabled for pgsql SQL queries
Sat Oct 20 18:21:22 2018 (5216): [Info] oci8 extensions is not loaded, Blackfire SQL analyzer will be disabled for oci SQL queries

Symfony Security Check Report
=====

// Checked file: /home/zohac/www/dev/composer.lock

[ERROR] 1 packages have known vulnerabilities.

symfony/symfony (v3.1.10)
-----
* CVE-2018-14773: CVE-2018-14773: Remove support for legacy and risky HTTP headers
  https://symfony.com/blog/cve-2018-14773-remove-support-for-legacy-and-risky-http-headers
* CVE-2018-11407: CVE-2018-11407: Unauthorized access on a misconfigured LDAP server when using an empty password
  https://symfony.com/cve-2018-11407
* CVE-2017-16653: CVE-2017-16653: CSRF protection does not use different tokens for HTTP and HTTPS
  https://symfony.com/cve-2017-16653
* CVE-2017-16654: CVE-2017-16654: Intl bundle readers breaking out of paths
  https://symfony.com/cve-2017-16654
* CVE-2018-11386: CVE-2018-11386: Denial of service when using PDOSessionHandler
  https://symfony.com/cve-2018-11386
* CVE-2017-16790: CVE-2017-16790: Ensure that submitted data are uploaded files
  https://symfony.com/cve-2017-16790
* CVE-2018-11406: CVE-2018-11406: CSRF Token Fixation
  https://symfony.com/cve-2018-11406
* CVE-2018-11408: CVE-2018-11408: Open redirect vulnerability on security handlers
  https://symfony.com/cve-2018-11408
* CVE-2017-16652: CVE-2017-16652: Open redirect vulnerability on security handlers
  https://symfony.com/cve-2017-16652
* CVE-2018-11385: CVE-2018-11385: Session Fixation Issue for Guard Authentication
  https://symfony.com/cve-2018-11385

! [NOTE] This checker can only detect vulnerabilities that are referenced in the SensioLabs security advisories
! database. Execute this command regularly to check the newly discovered vulnerabilities.
```

L'installation de symfony 3.1 est obsolète, celle-ci sera mis à jour vers Symfony 3.4.

Symfony 3.4 est la version LTS (pour Soutien à Long Terme), fin du soutien en Novembre 2020.

## 2.2 composer outdated

La commande `composer outdated` affiche une liste des paquets installés qui ont des mises à jour disponibles. Il s'agit essentiellement d'un alias pour `composer show -lo`.

```
Sun Oct 21 19:07:22 2018 (18827): [Info] oci8 extensions is not loaded, Blackfire SQL analyzer will be disabled for oci SQL queries
doctrine/doctrine-cache-bundle 1.3.2 1.3.3 Symfony Bundle for Doctrine Cache
sensio/framework-extra-bundle v3.0.29 v5.2.1 This bundle provides a way to configure your controllers with annotations
sensiolabs/security-checker v4.1.8 v5.0.1 A security checker for your composer.lock
swiftmailer/swiftmailer v5.4.12 v6.1.3 Swiftmailer, free feature-rich PHP mailer
symfony/monolog-bundle v2.12.1 v3.3.0 Symfony MonologBundle
symfony/phpunit-bridge v3.4.17 v4.1.6 Symfony PHPUnit Bridge
symfony/swiftmailer-bundle v2.6.7 v3.2.3 Symfony SwiftmailerBundle
symfony/symfony v3.1.10 v4.1.6 The Symfony PHP framework
zohac@ubuntu-server ~/www/dev feature/Issue9
```

Ces composants seront mis à jour.

Version : 1.0

Date de la dernière mise à jour : 13/11/18





## 2.3 Analyse automatisée du code

Visible à l'adresse : <https://github.com/zohac/ToDoList/tree/Issue1#todolist>

### ToDoList

code quality **B** coverage 0% build passing

Le projet n'obtient que la note de B sous codacy.

	Code Coverage								
	Lines			Functions and Methods			Classes and Traits		
Total		0.00%	0 / 139		0.00%	0 / 34		0.00%	0 / 8
 Controller		0.00%	0 / 60		0.00%	0 / 12		0.00%	0 / 4
 Entity		0.00%	0 / 60		0.00%	0 / 20		0.00%	0 / 2
 Form		0.00%	0 / 19		0.00%	0 / 2		0.00%	0 / 2
 AppBundle.php		n/a	0 / 0		n/a	0 / 0		n/a	0 / 0

Legend

Low: 0% to 50%

Medium: 50% to 90%

High: 90% to 100%

Il n'y a aucune couverture de code par des tests unitaires et fonctionnels.

## 2.4 Lecture du code

Une lecture du code fait remonter les points suivant :

- Manque des assertions dans les entités. [#Validation](#)
- Manque des contraintes dans les formulaires.
- Manque un pattern pour l'utilisation de mot de passe plus complexe.
- Manque de commentaires dans le code.
- Utiliser l'injection de dépendance plutôt que le container. [#Fetching Services](#)
- Ajouter des services pour simplifier les controllers.
- Ajouter les méthodes HTTP (*POST/GET/...*) dans les annotations des routes.
- Dans l'annotation des routes, ajouter des contraintes (*requirements*).

## 2.5 Test manuel de l'application

Une test de l'application fait remonter les points suivant :

- Un bouton pour « consulter la liste des tâches terminées » non fonctionnel
- Manque un bouton pour consulter la liste des utilisateurs
- Le lien « To Do List app » dans la barre de navigation ne redirigeant vers rien
- Erreur de chargement de jQuery (non présent dans le dossier web/js)

## 2.6 Vérification des dépendances.

Après mise à jour de Symfony vers la version 3.4.\*, on vérifie le code déprécié avec le profiler de Symfony.

Log Messages		
Info. & Errors	1	Deprecations 6
Debug	33	PHP Notices 0
Container	575	

Log messages generated by using features marked as deprecated.

Time	Channel	Message
20:40:21	php	User Deprecated: Symfony\Component\HttpKernel\Kernel::loadClassCache() is deprecated since Symfony 3.3, to be removed in 4.0. <a href="#">Show context</a> <a href="#">Show trace</a>
20:40:21	php	User Deprecated: Symfony\Component\HttpKernel\Kernel::doLoadClassCache() is deprecated since Symfony 3.3, to be removed in 4.0. <a href="#">Show context</a> <a href="#">Show trace</a>
20:40:21	php	User Deprecated: The "Sensio\Bundle\FrameworkExtraBundle\Configuration\Route" annotation is deprecated since version 5.2. Use "Symfony\Component\Routing\Annotation\Route" instead. <a href="#">Hide context</a> <a href="#">Show trace</a> <pre>[{"exception" =&gt; ErrorException {#406 ▶}]</pre>
19:42:04	-	The "framework.trusted_proxies" configuration key has been deprecated in Symfony 3.3. Use the Request::setTrustedProxies() method in your front controller instead. <a href="#">Show context</a> <a href="#">Show trace</a>
19:42:04	-	Not setting "logout_on_user_change" to true on firewall "main" is deprecated as of 3.4, it will always be true in 4.0. <a href="#">Show context</a> <a href="#">Show trace</a>
19:42:04	-	Enabling the "sensio_framework_extra.router.annotations" configuration is deprecated since version 5.2. Set it to false and use the "Symfony\Component\Routing\Annotation\Route" annotation from Symfony itself. <a href="#">Show context</a> <a href="#">Show trace</a>



## 2.7 Tests de performance

Les tests de performance sont réalisés avec l'application [Blackfire](#).

Blackfire permet à tous les développeurs de vérifier et d'améliorer continuellement les performances de l'application, tout au long de son cycle de vie, en obtenant de données de mesure.

Ensembles des adresses testées :

- "/", [name="homepage"](#)
- "/login", [name="login"](#)
- "/tasks", [name="task\\_list"](#)
- "/tasks/create", [name="task\\_create"](#)
- "/tasks/{id}/edit", [name="task\\_edit"](#)
- "/users", [name="user\\_list"](#)
- "/users/create", [name="user\\_create"](#)
- "/users/{id}/edit", [name="user\\_edit"](#)

<b>200 GET http://192.168.0.25/users/create</b> <i>ToDo List /users/create</i> Created less than a minute ago by <b>zohac</b> 60.6 ms 2.34 MB 0 µs / 0 rq 0 µs / 0 rq	Compare
<b>200 GET http://192.168.0.25/users/1/edit</b> <i>ToDo List /users/{id}/edit</i> Created 2 minutes ago by <b>zohac</b> 62.3 ms 2.36 MB 0 µs / 0 rq 0 µs / 0 rq	Compare
<b>200 GET http://192.168.0.25/users</b> <i>ToDo List /users</i> Created 3 minutes ago by <b>zohac</b> 43.4 ms 1.76 MB 0 µs / 0 rq 0 µs / 0 rq	Compare
<b>200 GET http://192.168.0.25/tasks/1/edit</b> <i>ToDo List /tasks/{id}/edit</i> Created 4 minutes ago by <b>zohac</b> 64.4 ms 2.34 MB 0 µs / 0 rq 0 µs / 0 rq	Compare
<b>200 GET http://192.168.0.25/tasks/create</b> <i>ToDo List /task/create</i> Created 6 minutes ago by <b>zohac</b> 60.5 ms 2.31 MB 0 µs / 0 rq 0 µs / 0 rq	Compare
<b>200 GET http://192.168.0.25/tasks</b> <i>ToDo List /tasks</i> Created 16 minutes ago by <b>zohac</b> 48.8 ms 1.8 MB 0 µs / 0 rq 0 µs / 0 rq	Compare
<b>200 GET http://192.168.0.25/</b> <i>ToDo List /</i> Created 24 minutes ago by <b>zohac</b> 43.8 ms 1.75 MB 0 µs / 0 rq 0 µs / 0 rq	Compare
<b>200 GET http://192.168.0.25/login</b> <i>ToDo List /login</i> Created 26 minutes ago by <b>zohac</b> 34.1 ms 1.4 MB 0 µs / 0 rq 0 µs / 0 rq	Compare

Version : 1.0

Date de la dernière mise à jour : 13/11/18

# 3 Actions mises en œuvre

---

## 3.1 Mise à jour des composants

La version de Symfony 3.1 n'est plus maintenue par SensioLab, une mise à jour est effectuée vers la version 3.4.18. C'est la dernière version LTS de Symfony.

Un passage vers la version 4 n'est pas souhaitable à ce stade du développement de l'application. En revanche une branche de développement spécifique pourrait être maintenue, pour une mise en production lors de la prochaine version 4.4 Lts de Symfony.

Suite à la mise à jour de Symfony, les différentes dépendances du projet sont mise à jour.

## 3.2 Analyse automatisée du code

Suite à la corrections d'anomalies et l'implémentation de nouvelles fonctionnalités, la qualité du code et la couverture du code par des tests automatisés se sont améliorés.

### ToDoList



	Code Coverage							
	Lines			Functions and Methods			Classes and Traits	
Total		98.15%	212 / 216		94.12%	48 / 51		83.33% 10 / 12
Controller		93.55%	58 / 62		75.00%	9 / 12		50.00% 2 / 4
DataFixtures		100.00%	26 / 26		100.00%	4 / 4		100.00% 1 / 1
Entity		100.00%	46 / 46		100.00%	26 / 26		100.00% 2 / 2
Form		100.00%	63 / 63		100.00%	5 / 5		100.00% 3 / 3
Repository		100.00%	5 / 5		100.00%	1 / 1		100.00% 1 / 1
Security		100.00%	14 / 14		100.00%	3 / 3		100.00% 1 / 1
AppBundle.php		n/a	0 / 0		n/a	0 / 0		n/a 0 / 0

Legend

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

Sont exclus de l'analyse du code les fichiers de base de Symfony. Les dossiers et fichiers suivants sont exclus :

- var/\*\*
- app/AppCache.php
- app/AppKernel.php
- app/autoload.php
- var/SymfonyRequirements.php
- web/app\_dev.php
- web/app.php
- web/config.php

## 3.3 Lecture du code

Les corrections suivantes ont été apporté au code de l'application :

- Mise en place des assertions dans les entités. [#Validation](#)
- Mise en place des contraintes dans les formulaires.
- Mise en place d'un pattern pour l'utilisation de mot de passe plus complexe.
- Utilisation de commentaires dans le code.
- Utilisation l'injection de dépendance plutôt que le container dans les controllers. [#Fetching Services](#)
- Ajout des méthodes HTTP (*POST/GET/...*) dans les annotations des routes.
- Dans l'annotation des routes, ajout des contraintes (*requirements*).

A prévoir pour une amélioration du code :

- Ajouter des services pour simplifier les controllers.

## 3.4 Test manuel de l'application

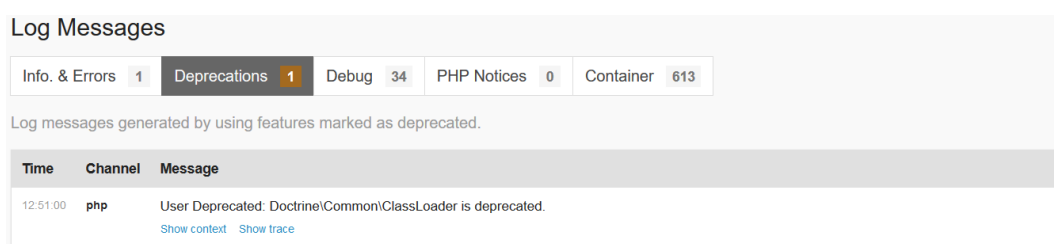
Suite aux tests, la correction suivante à été apporté :

- Ajout d'un CDN( Content Delivery Network ) pour jQuery et Bootstrap.

## 3.5 Vérification des dépendances

Les dépendances dépréciées révélé par le profiler de symfony sont supprimées.

Une seule reste à ce stade non résolu.



The screenshot shows the 'Log Messages' section of the Symfony Profiler. It has tabs for 'Info. & Errors' (1), 'Deprecations' (1), 'Debug' (34), 'PHP Notices' (0), and 'Container' (613). The 'Deprecations' tab is selected. Below the tabs, it says 'Log messages generated by using features marked as deprecated.' A table with columns 'Time', 'Channel', and 'Message' displays one message: 'User Deprecated: Doctrine\Common\ClassLoader is deprecated.' with a timestamp of '12:51:00' and channel 'php'. There are links for 'Show context' and 'Show trace'.

Le paquet Doctrine Common sera divisé en petits paquets et le composant ClassLoader sera supprimé, c'est pourquoi la dépréciation est affichée. Voir <https://github.com/doctrine/common/issues/826> et <https://www.doctrine-project.org/2018/07/12/common-2-9-and-dbal-2-8-and-orm-2-6-2.html>.

Ainsi, dans les nouvelles version de symfony, l'obsolescence devrait disparaître.

## 3.6 Tests de performances

Optimisation de l'autoloader de composer avec la commande :

- `composer dump-autoload -o`

<b>200 GET http://192.168.0.25/users/create</b> <i>To Do List app</i> Created less than a minute ago by zohac ⓧ ⚠ ⌚ 57.4 ms 📄 2.47 MB 🕒 0 µs / 0 rq 🗑 0 µs / 0 rq	Compare ↩ 🗑
<b>200 GET http://192.168.0.25/users/1/edit</b> <i>To Do List app</i> Created 1 minute ago by zohac ⓧ ⚠ ⌚ 57.4 ms 📄 2.48 MB 🕒 0 µs / 0 rq 🗑 0 µs / 0 rq	Compare ↩ 🗑
<b>200 GET http://192.168.0.25/users</b> <i>To Do List app</i> Created 1 minute ago by zohac ⓧ ⚠ ⌚ 39.6 ms 📄 1.81 MB 🕒 0 µs / 0 rq 🗑 0 µs / 0 rq	Compare ↩ 🗑
<b>200 GET http://192.168.0.25/tasks/1/edit</b> <i>To Do List app</i> Created 1 minute ago by zohac ⓧ ⚠ ⌚ 50.9 ms 📄 2.29 MB 🕒 0 µs / 0 rq 🗑 0 µs / 0 rq	Compare ↩ 🗑
<b>200 GET http://192.168.0.25/tasks/create</b> <i>To Do List app</i> Created 4 minutes ago by zohac ⓧ ⚠ ⌚ 46.4 ms 📄 2.25 MB 🕒 0 µs / 0 rq 🗑 0 µs / 0 rq	Compare ↩ 🗑
<b>200 GET http://192.168.0.25/tasks</b> <i>To Do List app</i> Created 5 minutes ago by zohac ⓧ ⚠ ⌚ 48.2 ms 📄 2.01 MB 🕒 0 µs / 0 rq 🗑 0 µs / 0 rq	Compare ↩ 🗑
<b>200 GET http://192.168.0.25/</b> <i>To Do List app</i> Created 5 minutes ago by zohac ⓧ ⚠ ⌚ 34.4 ms 📄 1.72 MB 🕒 0 µs / 0 rq 🗑 0 µs / 0 rq	Compare ↩ 🗑
<b>200 GET http://192.168.0.25/login</b> <i>To Do List app</i> Created 12 minutes ago by zohac ⓧ ⚠ ⌚ 22.1 ms 📄 1.48 MB 🕒 0 µs / 0 rq 🗑 0 µs / 0 rq	Compare ↩ 🗑

En optimisant l'autoloader les gains de performance sont les suivant :

Adresse	Temps d'exécution	Mémoire
/login	-35 % (-12ms)	+5,86 %
/	-21 % (-9,42ms)	-1,86 %
/tasks	-1,27 % (-619µs)	+11,4 %
/tasks/create	-23 % (-14ms)	-2,64 %
/tasks/{id}/edit	-21 % (-13ms)	-1,92 %
/users	-8,89 % (-3,86ms)	+2,28 %
/users/{id}/edit	-7,86 % (-4,89ms)	+5,11 %
/users/create	-5,34 % (-3,24ms)	+5,23 %

Version : 1.0

Date de la dernière mise à jour : 13/11/18

# 4 Conclusion

---

Après les modifications apportées à l'application nous obtenons une qualité de code supérieure, ainsi qu'une couverture du code de plus 98 %, comme nous le montre l'analyse du code par Codacy et CodeClimate.

Au niveau des performances, un léger gain a été obtenu en optimisant l'autoloader de composer.

Quelques points peuvent encore être améliorés :

- **Application :**
  - Un bouton pour « consulter la liste des tâches terminées » non fonctionnel
  - Manque un bouton pour consulter la liste des utilisateurs
  - Le lien « To Do List app » dans la barre de navigation ne redirigeant vers rien
  - Personnaliser les pages d'erreurs (Erreur 403/404/500 etc...).
  - Modifier le favicon.
- **Performances :**
  - En production : réappliquer la commande suivante `composer dump-autoload -o`
  - Mise en place d'un système de cache HTTP
  - Optimisation des images