

TODO_CO

Audit de qualité et de performance

Mars 2023

Sommaire

1. Objectifs de l'audit.....	3
2. Audit du code initial.....	3
2.1 Métriques.....	3
2.2 Qualité du code.....	3
2.3 Coût de la dette technique.....	5
3. Améliorations.....	6
3.1 Sécurité.....	6
3.2 Mise à jour de la version du code.....	6
3.3 qualité du code.....	7
4. Analyse de la performance.....	8

1. Objectifs de l'audit

ToDo & Co est une startup qui a développé une application de gestion des tâches quotidiennes. Le MVP qui nous a été présenté utilise le framework Symfony dans sa version 3.1

Notre mission a consisté à :

- Développer de nouvelles fonctionnalités
- Améliorer la qualité du code ainsi que les performances.
- Mettre en place des tests automatisés

2. Audit du code initial

2.1 Les métriques

Le score global obtenu après l'analyse Codacy est C, avec un total de 298 issues détaillées comme suit:

- 283 issues concernant PHP et 15 issues, le CSS.
- 17 issues critiques, 21 medium et 260 minors
- Les issues critiques concernent les fichiers: web/config.php et web/app_dev.php

2.2 Qualité du code

La qualité du code est globalement médiocre, nous présentons ci-dessous le rapport d'analyse :

Quality evolution

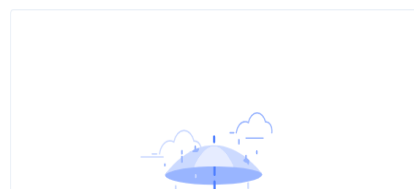
Last 3 months Last 31 days **Last 7 days**



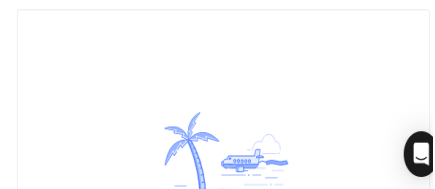
Issues breakdown



Coverage



Open pull requests



Les problèmes les plus importants concernent la sécurité, notamment l'utilisation des variables superglobales dans les fichiers : `web/app_dev.php` et `web/config.php`

Le niveau de sécurité initial est très insuffisant:

- Les rôles des utilisateurs ne sont pas spécifiés.
- Un utilisateur peut modifier n'importe quel compte ou n'importe quelle tâche sans être authentifié.

La majorité des erreurs signalées sont dues à la version très ancienne (3.1) de symfony.

web/app_dev.php

CRITICAL	Security	Direct use of \$_SERVER Superglobal detected.
13	if (isset(\$_SERVER['HTTP_CLIENT_IP']))	

CRITICAL	Security	Direct use of \$_SERVER Superglobal detected.
14	isset(\$_SERVER['HTTP_X_FORWARDED_FOR'])	

web/config.php

CRITICAL	Security	Direct use of \$_SERVER Superglobal detected.
13	if (!isset(\$_SERVER['HTTP_HOST'])) {	


CRITICAL	Security	Use of exit language construct is discouraged.
14	exit('This script cannot be run from the CLI. Run it from a browser.');	

CRITICAL	Security	Direct use of \$_SERVER Superglobal detected.
17	if (!in_array(@\$_SERVER['REMOTE_ADDR'], array(

2.3 Cout de la dette technique

Pour corriger les bugs critiques et mediums relevés par l'analyse du code nous aurons besoin d'au moins 4 heures et 10 minutes, c'est la somme des temps estimés pour chaque erreur comme dans l'exemple ci-dessous:

var/SymfonyRequirements.php

MEDIUM	Code Style	The method __construct() has a Cyclomatic Complexity of 11. The configured cyclomatic complexity threshold is 10.
 Saro0h 6 years ago		
Reported by PHP Mess Detector Time to fix: 30 minutes 🔗		
128	* @param string null \$helpText	The help text (when null, it will be inferred from \$helpHtml, i.e. stripped from HT
129	* @param bool \$optional	Whether this is only an optional recommendation not a mandatory requirement
130	*/	
131	public function __construct(\$cfgName, \$evaluation, \$approveCfgAbsence = false, \$testMessage = null, \$helpHtml = null, \$helpTex	
132	{	
133	\$cfgValue = ini_get(\$cfgName);	

3. Améliorations

3.1 Sécurité

- Nous avons créé des rôles pour les utilisateurs, qui sont attribués lors de la création:

rôle utilisateur (*ROLE_USER*) ;

rôle administrateur (*ROLE_ADMIN*).

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

Des permissions ont été mis en place avec les Voters afin que les tâches ne puissent être modifiées et supprimées que par l'utilisateur qui en est l'auteur.

3.2 Mise à jour de la version du code

Nous avons réalisé la nouvelle version de l'application avec php 8 et symfony 6.2, cela nous a permis d'intégrer des modifications significatives :

- Un système d'authentification plus fort a été implémenté avec le composant Auth, la création de l'entité User qui implémente `UserInterface` ainsi que `UserPasswordInterface` pour le hashage des mots de passe.
- Une fonctionnalité permettant l'inscription des utilisateurs a été ajoutée.
- Les entités Task et User sont liées par une relation `ManyToOne`, qui a permis de rattacher chaque tâche à l'utilisateur qui l'a créé. Désormais, le nom de l'auteur est affiché sur les tâches.
- Les controllers étendent désormais la classe `AbstractController` au lieu de la classe `Controller`.

Le controller DefaultController qui affiche la page d'accueil a été renommé HomepageController

- Les tests ont été réalisés avec Phpunit sur les dossiers : Controller, Entity, Form.

Les tests unitaires ont été réalisés sur les entités, Les controller et les formulaires ont été testés fonctionnellement

Le coverage global des tests est de 72%.

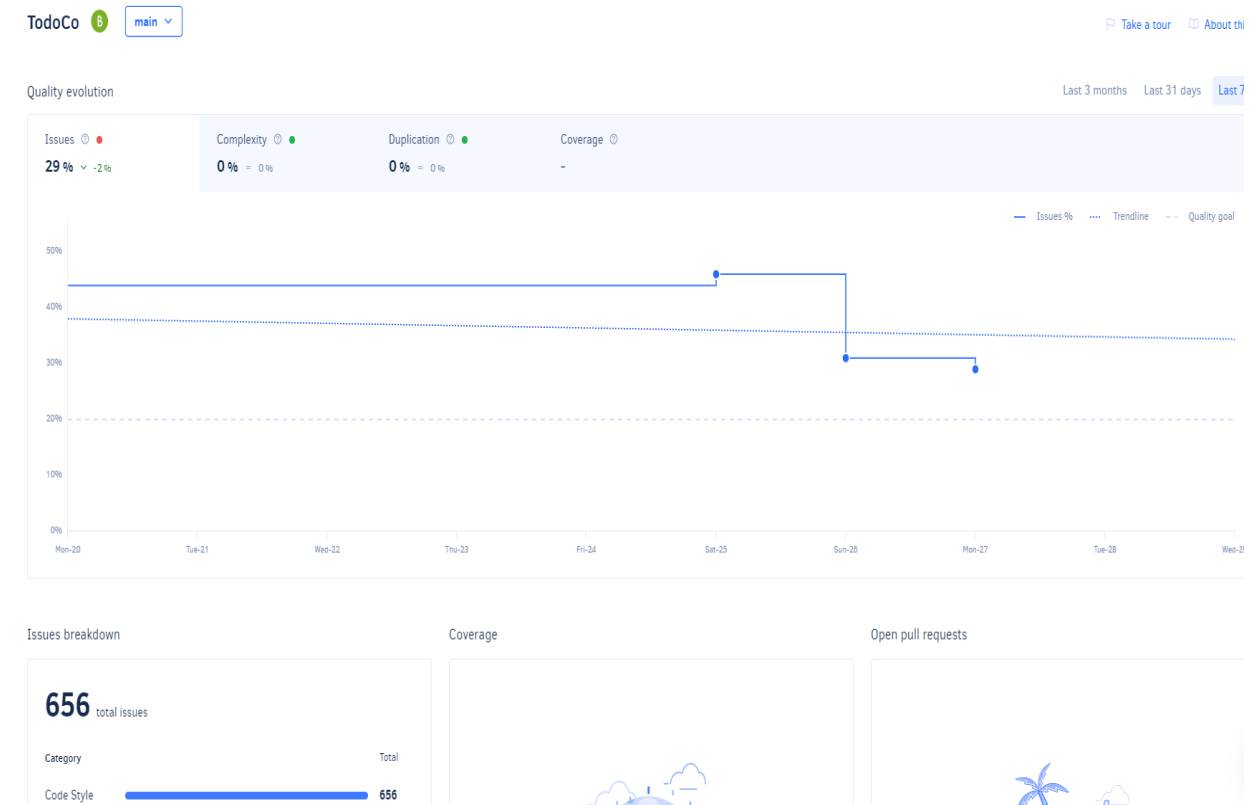
3.3 Qualité de code

La qualité du code a été améliorée passant de la note de C à la note B.

Il n'y a plus aucun problème de sécurité.

Les erreurs signalées sont mineures et ne mettent pas gravement en cause la qualité du code.

Le rapport d'analyse global est présenté ci-dessous:



4. Analyse de la performance

Nous avons utilisé l'outil de profilage de Symfony pour réaliser l'analyse de la performance.

Prenons l'exemple de la page d'accueil :



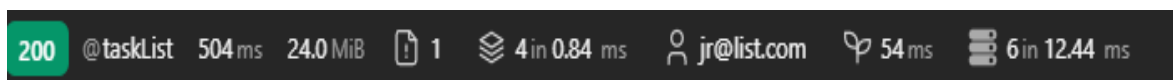
De gauche à droite, nous avons les données suivantes :

- Le status HTTP
- Le nom de la route
- Le temps total de chargement et de rendu : 457ms dont 140ms pour l'initialisation de la page
- La mémoire consommée

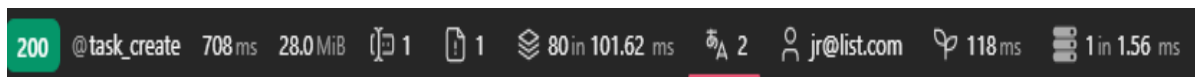
- Le nombre de dépréciation
- Les données concernant le cache
- Les informations de l'utilisateur connecté
- Le temps de rendu des blocs Twig
- Le nombre de requête à la base de données et le temps de rendu

Ci-dessous les analyses des principales pages :

Liste des tâches



Création d'une tâche



Liste des utilisateurs

