

Audit de qualité de code et de performance

I. Version old-todo-list :

1. Etat des lieux de la dette technique :

L'application « old-todo-list » dans sa version originale 3.1.6 de symfony présente beaucoup de manquements. Ces manquements impactent sur sa qualité. La qualité peut être perçue de différentes manières:

1.1 Qualité perçue par un développeur:

- L'application contient un fichier README.md qui n'explique pas les étapes d'installation du projet
- Elle ne contient pas un fichier expliquant aux futurs développeurs comment et quelles sont les règles à respecter pour contribuer
- La version 3.1.6 de symfony utilisée n'est plus stable voir obsolète
- Présente un bug qui empêche son démarrage. Lorsqu'on tente d'afficher la page d'accueil sur le navigateur, le server retourne l'erreur :

```
FatalThrowableError in : ErrorHandler.php line 364
```

- La version 3.3.7 de bootstrap utilisée est très ancienne
- Présente 0% de taux de couverture par les tests
- La logique métier est dans les « controllers »
- Les exceptions et les erreurs ne sont pas gérées
- Les propriétés "methods" et "requirements" ne sont pas ajoutées dans les annotations des routes
- Absence d'outil permettant de détecter les problèmes de normes de codage dans le code.

2.1 Qualité perçue par l'entreprise "ToDo & Co" :

- Absence de la sécurité et des rôles dans l'utilisation de l'application. N'importe quel utilisateur même anonyme peut gérer les tâches(CRUD), comme il peut gérer les utilisateurs(CRUD)

3.1 Qualité perçue par l'utilisateur :

- Absence de lien qui redirige vers la page « liste des utilisateurs »
- Absence de lien qui redirige vers la fonction permettant de supprimer un utilisateur
- Absence de lien qui redirige vers la page « modifier une tâche »
- Sur toutes les pages de l'application, absence de lien qui permet un retour vers la page d'accueil
- Sur la page d'accueil, on trouve un lien qui redirige vers la page « liste des tâches faites » et un autre « liste des tâches non faites » mais dans le projet il existe qu'une page « tasks » qui affiche les tâches faites et les tâches non faites.

2. Analyse de la qualité du code :

Pour l'analyse de la qualité du code, j'ai utilisé « codacy ». Les images suivantes montrent les résultats de l'analyse :

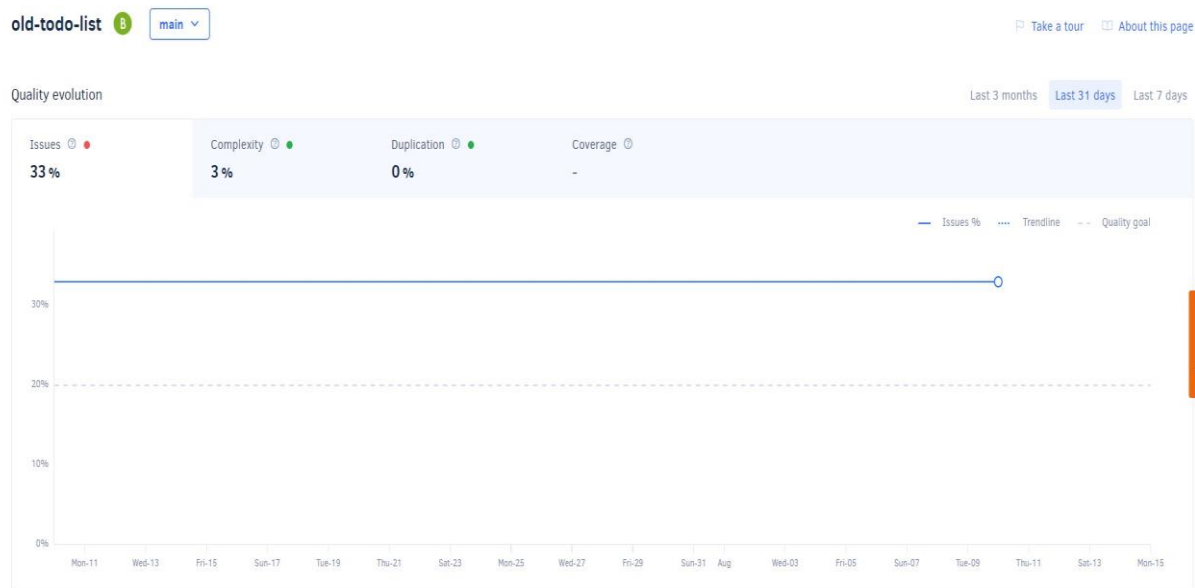


image1 : courbe d'évolution de la qualité du code (%) en fonction du temps (j) sur les 31 derniers jours

Commentaire : on voit que « old-todo-list » présente 33% d'issues, 3% de complexity et 0% de duplication ce qui lui permet d'obtenir le grade « B ».

Issues breakdown

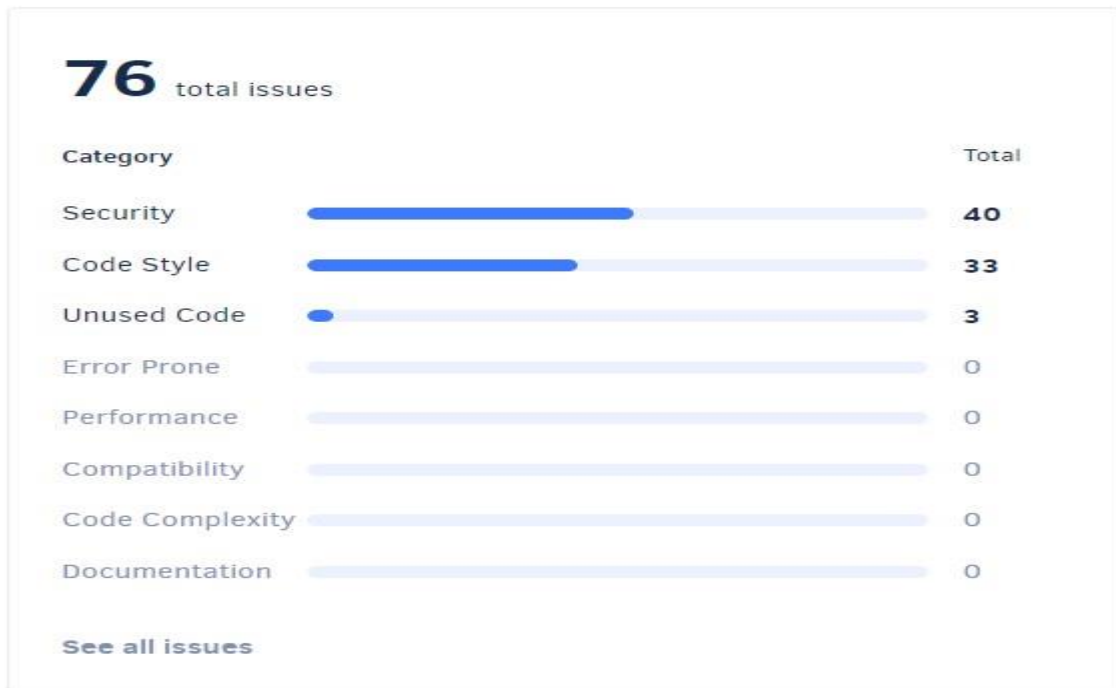


Image2 : nombre total des issues et leur répartition selon la category de l'issue

Commentaire : l'image2 montre que « old-todo-list » retourne un nombre total de 76 issues, 40 issues de la catégorie sécurité, 33 issues de la catégorie code style, 3 issues de la catégorie code inutilisé et 0 issue pour le reste des catégories.

On constate que la qualité du code n'est pas satisfaisante surtout le nombre important des issues liées à la sécurité. La sécurité d'une application est une priorité donc, il sera judicieux de réduire les erreurs liées à cette catégorie.

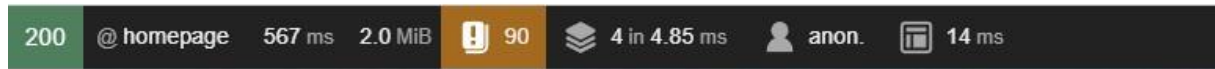
3. Analyse de la performance :

Pour l'analyse de la performance, j'ai utilisé le profiler de symfony. Le profiler se base sur certaines métriques pour mesurer la performance.

➤ Les métriques symfony :

- Le temps de rendu de la page en ms
- La consommation de mémoire en Mb
- Le nombre d'appels au cache
- Le temps de rendu des blocks twig en ms
- Nombre de requêtes SQL et le temps de leur exécution

- **Analyse de la « page d'accueil »** : le profiler symfony affiche les résultats suivant :



Commentaire : on voit un temps de rendu de page de 567ms. Une mémoire consommée de 2.0MiB et un temps de rendu de twig de 14ms. Le cache est appelé 4 fois en 4.85ms.

- **Analyse de la page « création d'une tâche »** :



Commentaire : on voit un temps de rendu de page de 504ms. Une mémoire consommée de 2.0MiB et un temps de rendu de twig de 25ms. Le cache est appelé 40 fois en 10.85ms

- **Analyse de la page « création d'un utilisateur »** :



Commentaire : on voit un temps de rendu de page de 525ms. Une mémoire consommée de 2.0MiB et un temps de rendu de twig de 42ms. Le cache est appelé 4 fois en 1.64ms.

D'après les résultats de l'analyse, « old-todo-list » est suffisamment performante mais qui peut être améliorée.

II. Version new-todo-list :

1. Analyse de la qualité du code :



Image1 : graphique d'évolution de la qualité du code (%) en fonction du temps (jours)

Issues breakdown

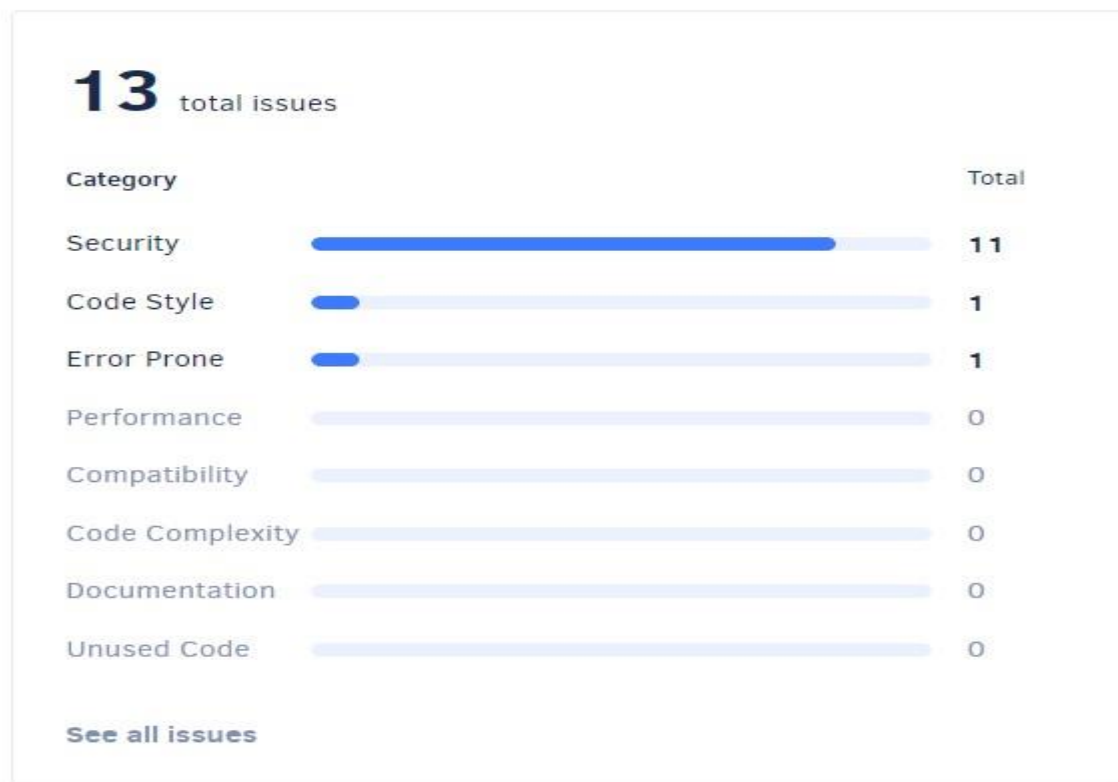


Image2 : nombre total des issues et leur répartition selon la category de l'issue

Commentaire : à ma première analyse, le graphe d'évolution de qualité montre 36% d'issues, 0% de complexity et 0% de duplication. Les 36% d'issues correspondent à un nombre d'issues de 150 sur Issues breakdown.

Sur Image1, on voit que le taux des issues baisse en fonction du temps jusqu'à ce qu'il atteigne 4%. Ce qui a permis d'attribuer un grade « **A** » au code.

Sur Image2, on voit que le nombre des issues a baissé jusqu'à 13 issues. 11 issues liées à la sécurité, 1 issue liée au code style, 1 issue liée à error prone et 0 issues ailleurs.

L'évolution de la qualité est dû à certaines erreurs que j'ai corrigées et d'autres (peu importantes) que j'ai ignorées.

- **Issues :**

1. Code style :

- [no-duplicate-headings] Do not use headings with similar content (5:1) => titre en double dans le fichier README.md pour corriger, j'ai supprimé le doublant.
- [no-consecutive-blank-lines] Remove 1 line after node => deux sauts de ligne consécutifs. Pour corriger, j'ai supprimé une ligne
- Expected indentation of 2 spaces (indentation) : reported by StyleInt, j'ai reconfiguré spaces=1 dans le pattern Specify Indentation.

2. Javascript Code : le pattern Prettier vue:Prettier renvoi plusieurs erreurs comme :

- Replace ``./css/bootstrap.min.css`` with ``"./css/bootstrap.min.css``
- Replace ``.....addEntry('app','./assets/app.js`` with ``..addEntry("app","./assets/app.js"`` => pour ce pattern, codacy ne respecte pas encore cette norme donc j'ai désactivé le pattern de l'analyse.

3. PHP code :

- Avoid using short method names like `Version20220626132043::up()`. The configured minimum method name length is 3 => j'ai ignoré le pattern « Prohibit short method names »
- The class `TaskControllerTest` has 12 non-getter- and setter-methods. Consider refactoring `TaskControllerTest` to keep number of methods under 10. Pour corriger, j'ai reconfiguré nombre de méthodes à 12 dans pattern: Excessive methods

1- Analyse de la performance :

➤ Analyse de la « page d'accueil » :

200	@ app_homepage	303 ms	2.0 MiB	 1	 n/a	 23 ms
-----	----------------	--------	---------	---	---	---

Commentaire : on voit un temps de rendu de page de 303ms. Une mémoire consommée de 2.0MiB et un temps de rendu de twig de 23ms.

➤ Analyse de la page « création d'une tâche » :

200	@ app_task_create	458 ms	8.0 MiB	 1	 1	 2	 admin1	 31 ms	 2 in 4.35 ms
-----	-------------------	--------	---------	---	---	---	--	---	--

Commentaire : on voit un temps de rendu de page de 458ms. Une mémoire consommée de 8.0MiB et un temps de rendu de twig de 31ms et 2 requêtes SQL sont appelées en 4.35ms.

➤ Analyse de la page « liste des tâches faites » :

200	@ app_task_list_done	372 ms	4.0 MiB	 1	 admin1	 11 ms	 3 in 3.98 ms
-----	----------------------	--------	---------	--	---	--	---

Commentaire : on voit un temps de rendu de page de 372ms. Une mémoire consommée de 4.0MiB et un temps de rendu de twig de 11ms et 3 requêtes SQL sont appelées en 3.98ms.

➤ Analyse de la page « liste des tâches non faites » :

200	@ app_task_list_not_done	377 ms	4.0 MiB	 1	 admin1	 11 ms	 3 in 4.67 ms
-----	--------------------------	--------	---------	---	--	---	--

Commentaire : on voit un temps de rendu de page de 377ms. Une mémoire consommée de 4.0MiB et un temps de rendu de twig de 11ms et 3 requêtes SQL sont appelées en 4.67ms.

➤ Analyse de la page création d'un utilisateur :

200	@ app_user_create	471 ms	8.0 MiB	 1	 1	 2 in 0.04 ms	 5	 admin1	 30 ms	 3 in 4.00 ms
-----	-------------------	--------	---------	---	---	--	---	--	---	--

Commentaire : on voit un temps de rendu de page de 471ms. Une mémoire consommée de 8.0MiB et un temps de rendu de twig de 30ms et 3 requêtes SQL sont appelées en 4ms.

➤ Analyse de la page liste des utilisateurs :



Commentaire : on voit que le temps de rendu de la page « liste des utilisateurs » est de 423ms. La mémoire consommée est 2.0Mb. 3 requêtes sql appelées en 3.76ms et le temps de rendu de twig est de 12ms.

Conclusion :

Si l'on compare les résultats de l'analyse de code entre « old-todo-list » de 36% d'issues et « new-todo-list » de 4% d'issues, on constate une amélioration considérable de la qualité du code de « new-todo-list ».

De même, si l'on compare les résultats d'analyse de performance entre ces deux versions, on constate que sur « new-todo-list » le temps de rendu des pages a diminué sur toutes les pages ainsi le temps de rendu des blocks twig a vu une diminution sur la plupart des pages. Ceci explique l'amélioration de « new-todo-list » en terme de performance.