

Audit de qualité du code & performance de l'application

L'objectif de ce projet est l'amélioration d'une application existante du point de vue de la Qualité du code au sens large.

Il est demandé de réaliser:

- un état des lieux de la dette technique
- un audit sur la qualité du code et de la performance au terme du travail sur l'application.

1. Etat des lieux de la dette technique

a. Analyse du projet

L'installation du projet a permis de constater que ce dernier a été développé sur la base du framework Symfony en sa version 3.1. Afin de lever les erreurs présentes la commande "composer update" a été utilisée avec la modification suivante du fichier composer.json: "symfony/symfony": "3.*"

A partir de cette base, l'analyse a pu débuter. Des outils ont permis également d'établir un comparatif de l'application « avant/après » en indiquant la progression acquise en termes de qualité de code et de performance.

A cette étape seul le framework Symfony sur lequel repose l'application a fait l'objet d'une mise à jour.

i. Dépréciations

on peut constater qu'il a un certain nombre de dépréciation:

https://127.0.0.1:8000/
Method: GET HTTP Status: 200 IP: 127.0.0.1 Profiled on: Sun, 28 Mar 2021 16:38:22 +0000 Token: 316c2e

Last 10 Latest Search

Request / Response

Performance

Validator

Forms

Exception

Logs 11

Events

Routing

Cache

Security

Twig

Doctrine

E-Mails

Debug

Configuration

Log Messages

Info. & Errors 1 Deprecations 11 Debug 32 PHP Notices 0 Container 582

Log messages generated by using features marked as deprecated.

Time	Channel	Message
16:38:21	php	User Deprecated: Symfony\Component\HttpKernel\Kernel::loadClassCache() is deprecated since Symfony 3.3, to be removed in 4.0. Show context Show trace
16:38:21	php	User Deprecated: Symfony\Component\HttpKernel\Kernel::doLoadClassCache() is deprecated since Symfony 3.3, to be removed in 4.0. Show context Show trace
16:38:21	php	User Deprecated: Creating Doctrine\ORM\Mapping\UnderscoreNamingStrategy without making it number aware is deprecated and will be removed in Doctrine ORM 3.0. Show context Show trace
16:38:15	-	Using the unquoted scalar value "levent" is deprecated since Symfony 3.3 and will be considered as a tagged value in 4.0. You must quote it in "C:\wamp64\www\projet8-TodoListapp/config/config_dev.yml" on line 20. Show context Show trace
16:38:15	-	Using the unquoted scalar value "levent" is deprecated since Symfony 3.3 and will be considered as a tagged value in 4.0. You must quote it in "C:\wamp64\www\projet8-TodoListapp/config/config_dev.yml" on line 23. Show context Show trace
16:38:15	-	Using the unquoted scalar value "doctrine" is deprecated since Symfony 3.3 and will be considered as a tagged value in 4.0. You must quote it in "C:\wamp64\www\projet8-TodoListapp/config/config_dev.yml" on line 23. Show context Show trace
16:38:15	-	The "framework.trusted_proxies" configuration key has been deprecated in Symfony 3.3. Use the Request::setTrustedProxies() method in your front controller instead. Show context Show trace
16:38:15	-	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. Show context Show trace
16:38:15	-(2 times)	Autowiring-types are deprecated since Symfony 3.3 and will be removed in 4.0. Use aliases instead for "Psr\Log\LoggerInterface". Show context Show trace
16:38:15	-	Symfony\Component\HttpKernel\DependencyInjection\Extension::addClassesToCompile() is deprecated since Symfony 3.3, to be removed in 4.0. Show context Show trace

ii. Analyse fonctionnelle

Liste des routes disponibles :

```
PS C:\wamp64\www\ToDoList> php bin/console debug:router
```

Name	Method	Scheme	Host	Path
_preview_error	ANY	ANY	ANY	/_error/{code}.{_format}
_wdt	ANY	ANY	ANY	/_wdt/{token}
_profiler_home	ANY	ANY	ANY	/_profiler/
_profiler_search	ANY	ANY	ANY	/_profiler/search
_profiler_search_bar	ANY	ANY	ANY	/_profiler/search_bar
_profiler_phpinfo	ANY	ANY	ANY	/_profiler/phpinfo
_profiler_search_results	ANY	ANY	ANY	/_profiler/{token}/search/results
_profiler_open_file	ANY	ANY	ANY	/_profiler/open
_profiler	ANY	ANY	ANY	/_profiler/{token}
_profiler_router	ANY	ANY	ANY	/_profiler/{token}/router
_profiler_exception	ANY	ANY	ANY	/_profiler/{token}/exception
_profiler_exception_css	ANY	ANY	ANY	/_profiler/{token}/exception.css
homepage	ANY	ANY	ANY	/
login	ANY	ANY	ANY	/login
login_check	ANY	ANY	ANY	/login_check
logout	ANY	ANY	ANY	/logout
task_list	ANY	ANY	ANY	/tasks
task_create	ANY	ANY	ANY	/tasks/create
task_edit	ANY	ANY	ANY	/tasks/{id}/edit
task_toggle	ANY	ANY	ANY	/tasks/{id}/toggle
task_delete	ANY	ANY	ANY	/tasks/{id}/delete
user_list	ANY	ANY	ANY	/users
user_create	ANY	ANY	ANY	/users/create
user_edit	ANY	ANY	ANY	/users/{id}/edit
index	ANY	ANY	ANY	/

D'un point de vue fonctionnel:

- Aucune page d'accueil ne semble accessible sans passer par le système d'authentification.
- le lien du logo n'est pas actif
- Une redirection s'opère (Code 302) sur la page d'accueil « / » et nous conduit sur la route « /login ».
- La création d'utilisateurs est accessible depuis la page de login sans être authentifié.
- La création d'un nouvel utilisateur permet de se connecter via la page de login et d'accéder à la page d'accueil ainsi qu'aux liens suivant:
 - créer un utilisateur,
 - se déconnecter,
 - consulter la liste des tâches terminées,
 - consulter la liste des tâches à faire
 - créer une tâche.
 - Le lien associé au titre du site étant également non fonctionnel.
- La liste des utilisateurs est consultable (routes « /users » et « users/create ») mais aucun utilisateur n'existe en Base De Données.
Il est possible d'accéder au formulaire de création d'un utilisateur, de créer un utilisateur avec succès et de le modifier.

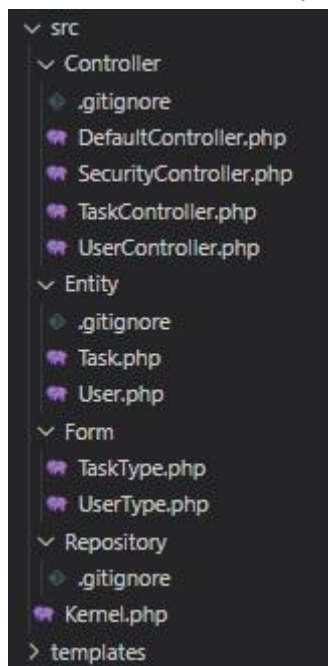
- les “Users”:
 - Création d’un utilisateur: une contrainte d’unicité existe sur l’e-mail mais n’empêche pas de soumettre plusieurs fois le même formulaire.
 - il n’est pas possible de supprimer un utilisateur.

- les “Task”:
 - La consultation de la liste des tâches à réaliser indique qu’aucune tâche n’existe en Base De Données.
 - Il est possible d’en créer, de les modifier et de les supprimer.
 - Aucune information sur l’auteur n’est associée à une tâche.
 - Il est possible de marquer une tâche comme « réalisée ». Cette dernière reste affichée dans la liste des tâches à faire au lieu de disparaître et d’être consultable dans la liste des tâches terminées.

iii. Analyse technique

1. Architecture du code

Le code métier respecte, en partie, une architecture de type Modèle, Vue, Controller.



La majorité du code métier est présent dans des Controller. L’application est développée en suivant des principes de la Programmation Orientée Objet.

2. Sécurité

L'outil Local-PHP-Security-Checker qui vérifie si votre application PHP a des vulnérabilités de sécurité connues a été utilisé. On constate que l'ensemble des aspects semblent couverts:

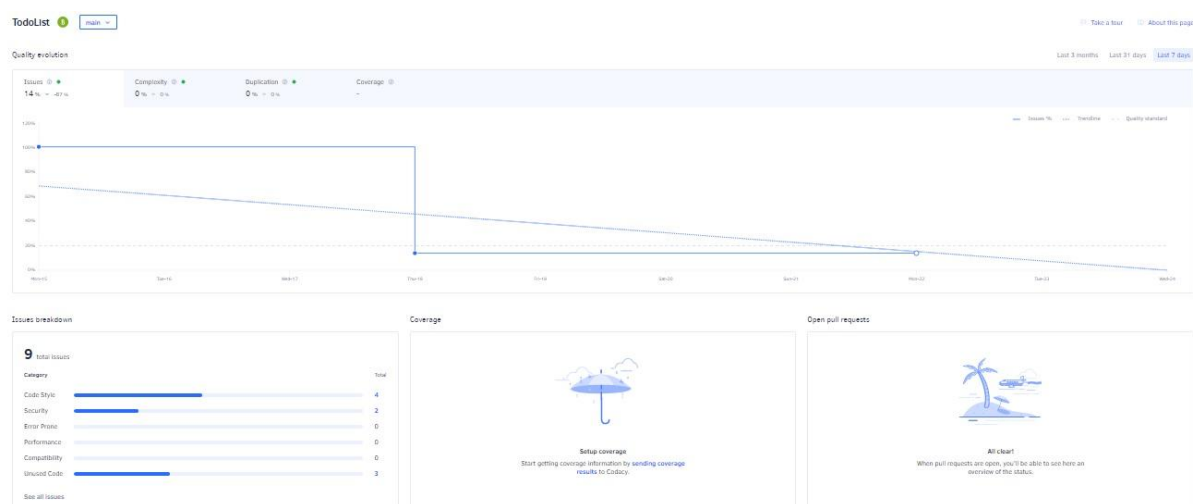
```
PS C:\wamp64\www\ToDoList> .\local-php-security-checker
Symfony Security Check Report
=====

No packages have known vulnerabilities.

Note that this checker can only detect vulnerabilities that are referenced in the security advisories database.
Execute this command regularly to check the newly discovered vulnerabilities.
```

3. Qualité du code

La qualité du code de l'application est suivie par l'outil Codacy. l'application obtient une note de Grade B La vue d'ensemble du rapport d'analyse de l'application est présentée ci-dessous :



Concernant les règles des PSR ainsi que celles de Symfony l'analyse de "PHP Code Sniffer" du dossier src indique :

fichier	nb erreur	nb alerte
DefaultController.php	5	
SecurityController.php	13	
TaskController.php	20	3
UserController.php	14	5

Task.php	28	
User.php	29	
TaskType.php	3	3
UserType.php	7	
Kernel.php	7	6

4. Tests : couverture du code de l'application

Le rapport de couverture du code de l'application par PHPUnit indique :



L'application n'est pas testée.

5. Schéma des tables en BDD

La structure actuelle des tables de la Base De Donnée:

todo-co user
id : int(11)
username : varchar(25)
password : varchar(64)
email : varchar(60)

todo-co task
id : int(11)
created_at : datetime
title : varchar(255)
content : longtext
is_done : tinyint(1)

Il n'y a pas de relation entre les utilisateurs et les tâches du fait que la consultation d'une tâche ne révélait aucun auteur associé. Les utilisateurs n'ont pas de rôle.

5) Performance de l'application

Pour disposer de point de comparaison avec la version apres modification, les deux routes principales ont été analysée:

- La page /users
- La page /tasks

200 GET https://127.0.0.1:8000/tasks

/tasks (version:3)

Created less than a minute ago by jpetitgas63@gmail.com

   0 rq 327 ms 17.4 MB 0 µs / 0 rq 0 µs / 0 rq

200 GET https://127.0.0.1:8000/users

/users (version:3)

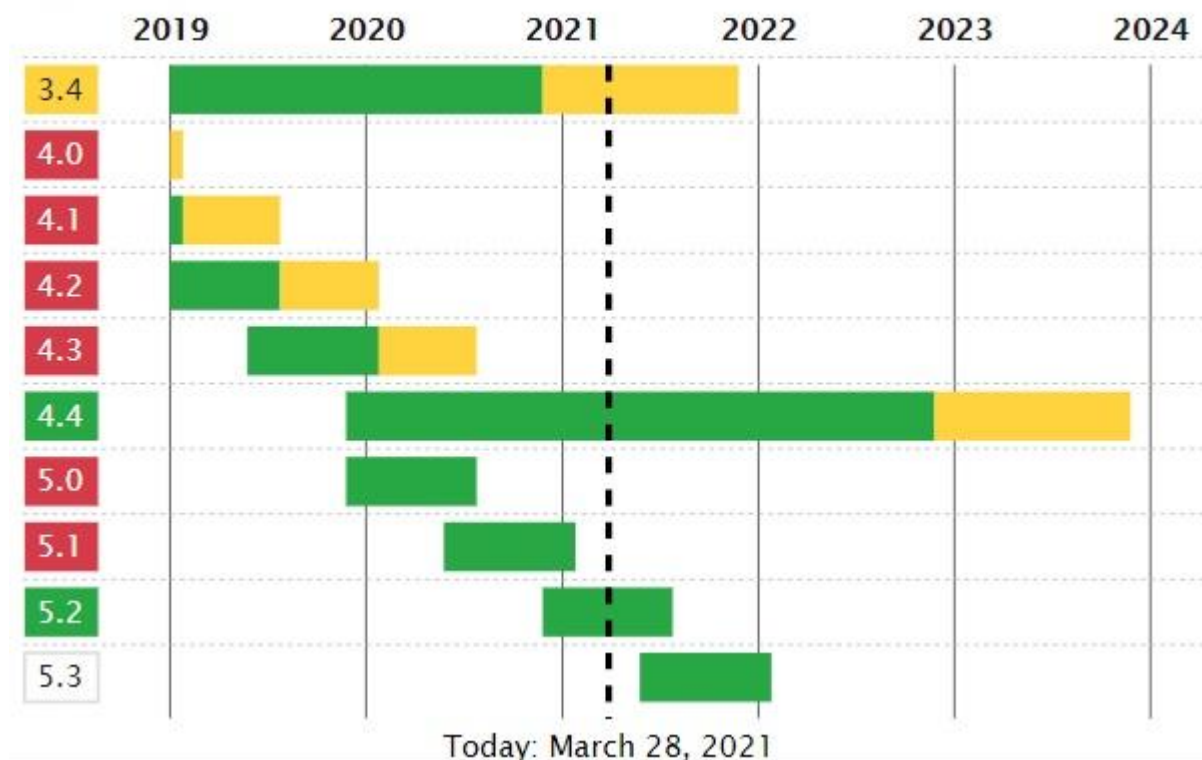
Created 2 minutes ago by jpetitgas63@gmail.com

   0 rq 349 ms 17.4 MB 0 µs / 0 rq 0 µs / 0 rq

Conclusion

A ce jour, la dernière version stable dite LTS (Long Term Support) du framework Symfony étant la version 4.4, la base du projet sera portée sur cette dernière.

Symfony Releases Calendar



Lors de l'implantation des nouvelles fonctions et des corrections demandés par le client, il sera nécessaire de faire évoluer l'application:

- Re-factorisé en vue de tendre au mieux vers le respect des principes SOLID
- La qualité du code doit être améliorée
- Couvrir le code par des tests unitaires et fonctionnelles
- mettre en place un plan d'améliorations fonctionnelles

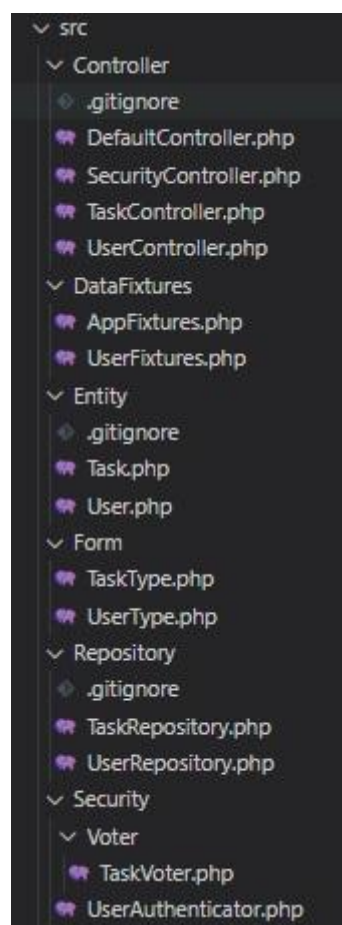
2. Audit de la qualité et des performances au terme du travail effectué.

Ce rapport de l'Audit de code fait suite aux corrections d'anomalies, de l'implémentation de fonctionnalités et d'améliorations du code, en accord avec les demandes de ToDo & Co.

a) Qualité du Code

i) Architecture

Le code métier a été réparti entre les controllers et les repository, en charge de persister les données en base de données.

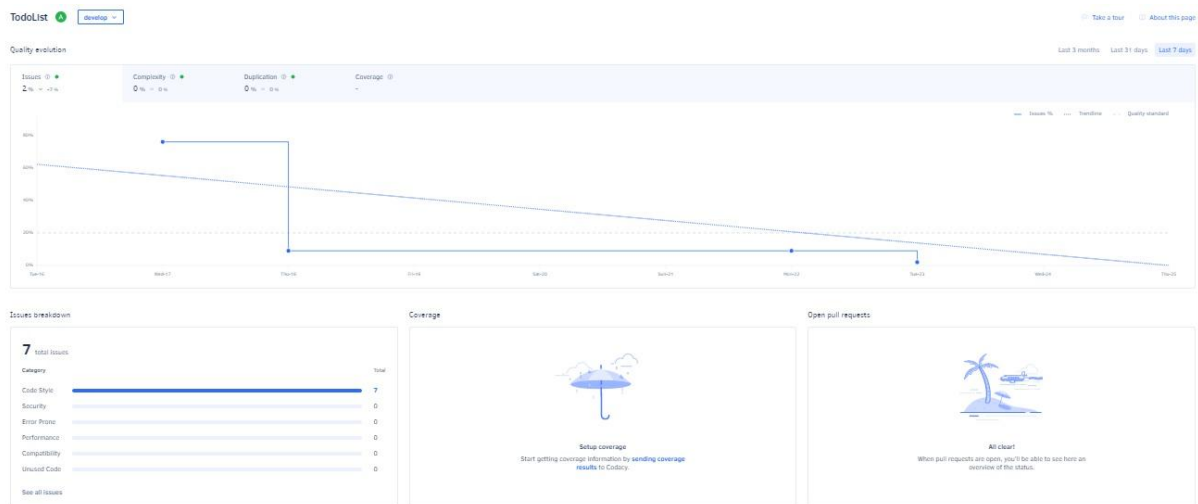


ii) Qualité:

L'analyse du code avec phpcs a fait ressortir un certain nombre d'erreurs qui ont été corrigées par l'outil PHP Coding Standards Fixer (PHP CS Fixer).

Les méthodes de chaque classe ont été documentées.

iii) Analyse par Codacy:



b) Taux de couverture du code

L'objectif fixé par ce projet était d'atteindre 70% de couverture de code.
Le rapport de couverture généré en local via PHPUnit actuel:

C:\wamp64\www\ToDoList\src / (Dashboard)

	Code Coverage							
	Lines			Functions and Methods			Classes and Traits	
Total	<div></div>	96.92%	189 / 195	<div></div>	92.98%	53 / 57	<div></div>	75.00% 9 / 12
Controller	<div></div>	98.89%	89 / 90	<div></div>	92.31%	12 / 13	<div></div>	75.00% 3 / 4
Entity	<div></div>	100.00%	50 / 50	<div></div>	100.00%	29 / 29	<div></div>	100.00% 2 / 2
Form	<div></div>	100.00%	10 / 10	<div></div>	100.00%	2 / 2	<div></div>	100.00% 2 / 2
Repository	<div></div>	100.00%	4 / 4	<div></div>	100.00%	2 / 2	<div></div>	100.00% 2 / 2
Security	<div></div>	87.80%	36 / 41	<div></div>	72.73%	8 / 11	<div></div>	0.00% 0 / 2

Legend













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

Generated by php-code-coverage 9.2.5 using PHP 7.4.9 with Xdebug 3.0.3 and PHPUnit 9.5.4 at Sun Mar 28 14:33:42 UTC 2021.

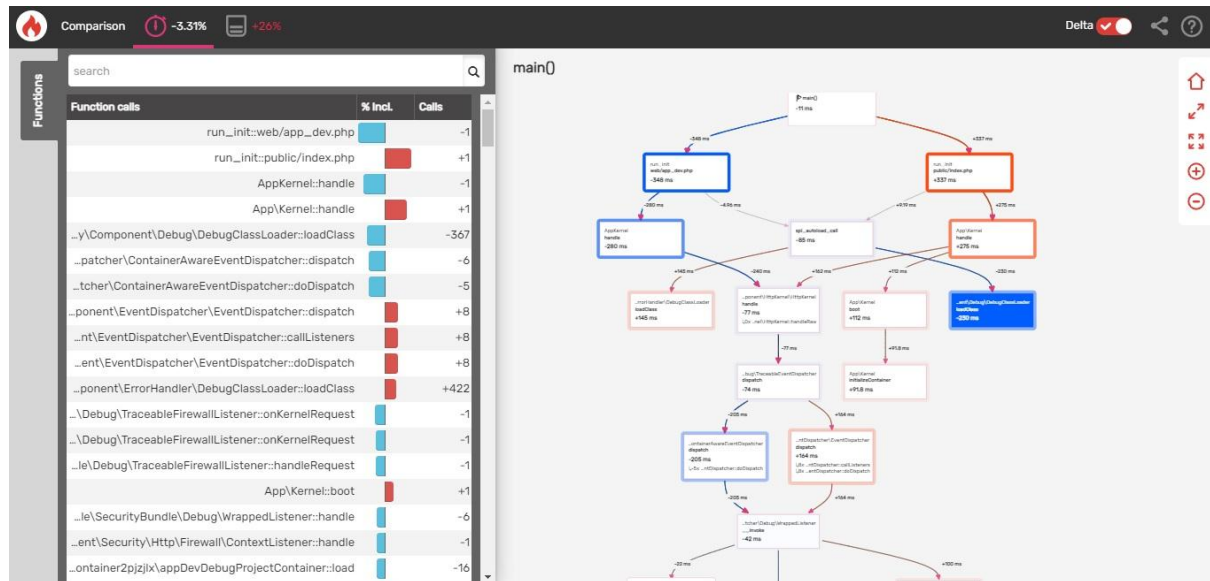
c) Performance de l'application

Les analyses ont été effectuées sur l'environnement de développement en local (window 10 et wamp). Elles ne reflètent donc pas les performances réelles de l'application en production.

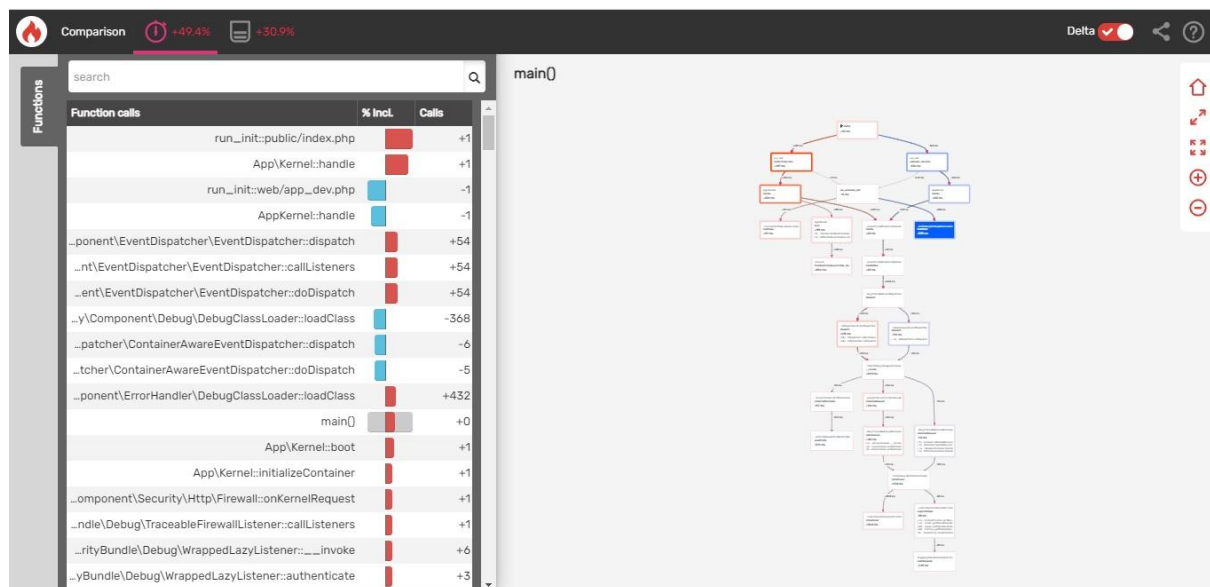
L'objectif ici est d'établir un référentiel afin de pouvoir en déduire un gain ou une perte de performance du fait de remaniement ultérieur de l'application.

200 GET https://127.0.0.1:8000/users <i>/users (version: 4)</i> Created less than a minute ago by jpetitgas63@gmail.com    0 rq 337 ms 21.9 MB 0 µs / 0 rq 0 µs / 0 rq
200 GET https://127.0.0.1:8000/tasks <i>/tasks (version : 4)</i> Created 1 minute ago by jpetitgas63@gmail.com    0 rq 488 ms 22.8 MB 0 µs / 0 rq 0 µs / 0 rq
200 GET https://127.0.0.1:8000/tasks <i>/tasks (version:3)</i> Created 4 minutes ago by jpetitgas63@gmail.com    0 rq 327 ms 17.4 MB 0 µs / 0 rq 0 µs / 0 rq
200 GET https://127.0.0.1:8000/users <i>/users (version:3)</i> Created 5 minutes ago by jpetitgas63@gmail.com    0 rq 349 ms 17.4 MB 0 µs / 0 rq 0 µs / 0 rq

Comparatif Blackfire – route /users - Application d'origine et application après modification



Comparatif Blackfire – route /tasks - Application d'origine et application après modification



Les analyses effectuées avec Blackfire nous permettent de voir l'évolution des critères de performance de l'application. Il s'agit du temps d'exécution de la requête pour obtenir la page ainsi que de la mémoire allouée à la génération de cette page.

d) Plan d'amélioration proposé

Ce projet qui vise l'amélioration d'une application existante a permis la correction d'anomalies, l'implémentation de nouvelles fonctionnalités et d'améliorer la qualité de cette application.

De ce travail émane un plan d'amélioration, non-exhaustif, qui permettra de fournir aux collaborateurs du projet des pistes d'améliorations à entreprendre:

i) Améliorations d'ordre fonctionnel

- reprendre le design de l'application.
- Ajouter un menu de navigation
- Ajouter des fonctionnalités au système de gestion des tâches (délai de réalisation sur chaque tâche, mise en place d'alerte à l'expiration d'une tâche...)
- Système de pagination des tâches.
- Ajouter aux tâches un système de tag
- Système de filtrage des tâches (par catégorie, auteur...).
- Edition d'un User : on ne devrait pas avoir à renseigner à chaque édition d'un User son mot de passe.

ii) Améliorations d'ordre technique

- Ajouter l'activation de nouveau compte User
- Tests : refactoriser les tests pour réduire le niveau de répétition de code.
- Le code n'est pas uniformisé à l'anglais

iii) Améliorations d'ordre organisationnel

- Définir plus précisément les règles à suivre au niveau de PHP Coding Standards Fixer