Audit de qualité

Présentation :

ToDo & Co est une startup dont le cœur de métier est une application permettant de gérer ses tâches quotidiennes. Le choix du développeur précédent a été d'utiliser le framework PHP Symfony pour créer l'application. L'application a dû être développée à toute vitesse pour permettre de montrer à de potentiels investisseurs que le concept est viable. Par la suite, ToDo & Co a réussi à lever des fonds pour permettre le développement de l'entreprise et surtout de l'application.

Le site a été créé sur sa première version sous Symfony 3. Pour avoir un site pouvant être mise à jour, propre et de bonne qualité suivant les Bests Pratices et le principe SOLID, son passage sous Symfony 4 est devenu une obligation. Après avoir réglé les problèmes de dépréciations et celle des librairies utilisés sur le framework. Son passage à Symfony 4.4 a été possible.

Après sa mise à jour sous symfony 4, de nouvelle fonctionnalités ont étaient mise en place et la modification des anomalies constatés lors de sa mise à jour ou demandé dans le cahier des charges ont étaient résolue. Des tests unitaires par la mise en place de PHPUnit et fonctionnelle avec PHPUnit permettant ainsi d'avoir un aperçu du coverage et Behat pour avoir un fonctionnement suivant celle faite par un utilisateur.

Plan :

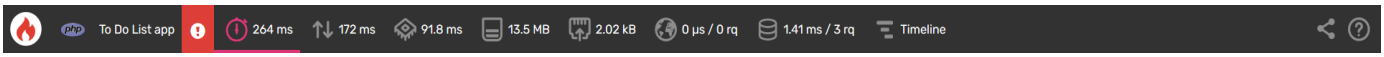
I) Présentation sur l'audit :

Cet audit a été créé pour présenter l'application. Elle présente les deux axes demandés : la performance et la qualité du code. Nous commencerons cette présentation par les généralités liées au fonctionnement de l'application. Dans un premier temps avec Backfire, pour la mesure des performances du site, puis la qualités du code avec Woorank, site d'audit SEO et Codacy, analyste de code automatisé. Pour finir une conclusion sur les points à améliorer représentant l'axe de conduite proposé pour l'amélioration du projet.

II) Audit généralisé :

a) Les performances du site :

Nous commençons cet audit avec une présentation des performances du site. En lui-même le suite fonction très bien.



Données d'execution de la page listant les tâches

Données d'execution de la page d'accueil

Chargement des pages

Page	Temps d'éxecution (ms)	Temps entrée/ sortie (ms)	CPU (ms)	Mémoire utilisé (MB)	Réseau (B)	Requêtes SQL (ms / rq)
Homepage (non connecté)	241	158	82.8	13.4	761	3.16 / 1
Homepage (connecté)	274	174	100	13.4	761	0.641 / 1
user_create	282	182	100	16.3	761	0.393 / 1
task_edit	279	180	98.7	16.1	1360	1.79/ 2
user_edit	285	183	102	16.3	761	0.385 / 1
task_list	233	151	81.7	13.5	2020	1.26/ 3
user_list	243	159	84.1	13.4	1430	0.766 / 2
login	209	129	80.4	9.66	n/a	n/a
task_toggle	322	218	104	12.7	n/a	4.8 / 3
logout	200	127	73.7	9.54	n/a	n/a
task_create	293	189	104	16	761	0.454 / 1

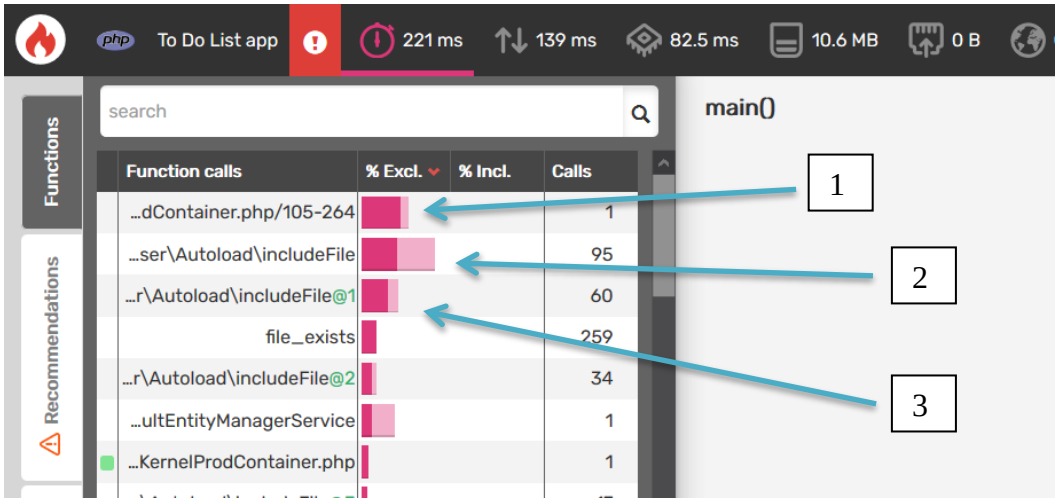
Fonctionnalités :

Création d'une tache	347	327	110	14.7	n/a	4.54 / 2
Supprimer une tache (task_delete)	265	172	92.4	12.8	n/a	0.787 / 2
Modifier une tache	376	233	143	14.6	n/a	0.790/ 2
Créer un utilisateur	1250	461	789	15.8	n/a	11.1 / 3
Modifier un utilisateur	1050	299	752	15.1	n/a	3.23 / 3

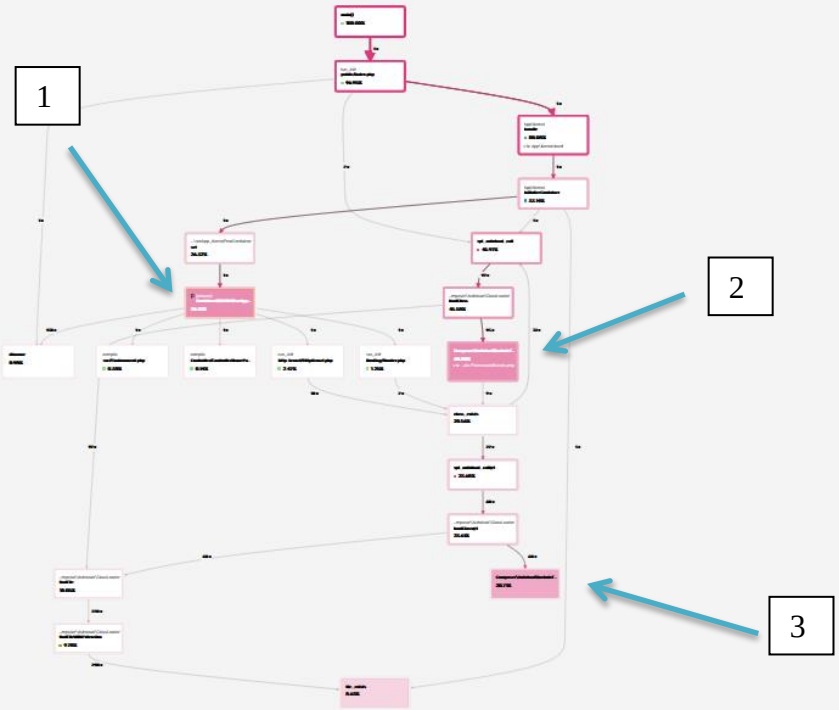
Comme vous pouvez le constater, l'exécution des pages est rapide. En moyenne nous somme à 300ms pour l'affichage des pages et 600 ms si il y a modification des données dans la base de données sachant qu'une personne attend en moyenne 3 secondes avant de changer de page.

b) Modifications possibles :

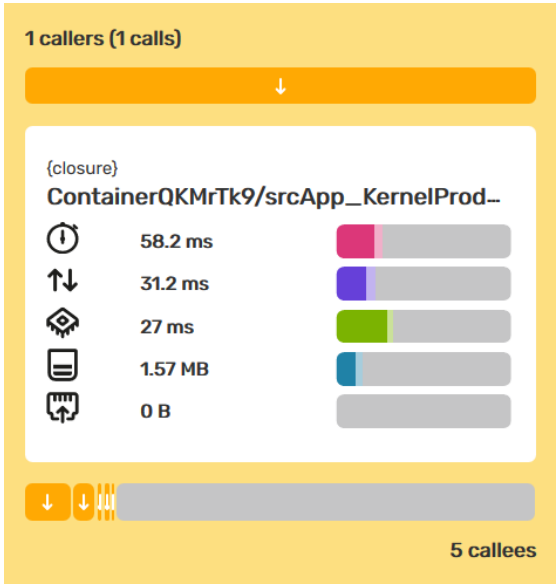
Sur les graphiques de temps exécution, d'utilisation CPU et la mémoire utilisé créer par Blackfire, nous nous apercevons quand même que des éléments peuvent être améliorés.



Sur ces graphiques vous avez une partie de la représentation des fonctions appelés. Représenté par des chiffres, vous avez les trois points mettant le plus de temps d'exécution, celle-ci sont des points à améliorer.

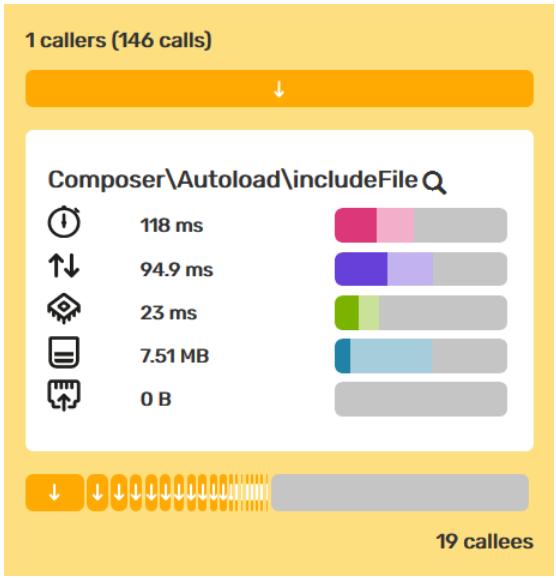


1) ContainerQKMrTk9\srcApp_KernelProdContainer.php\105-264 :



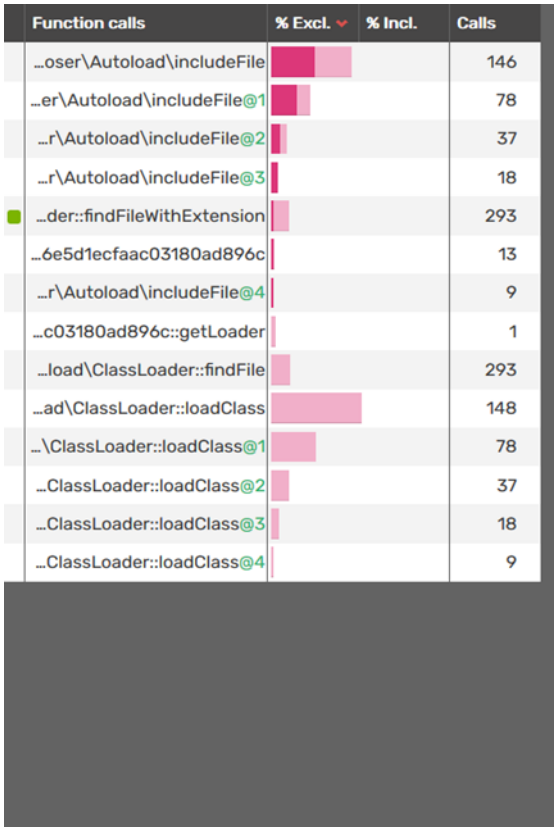
Le fichier « ContainerQKMrTk9\srcApp_KernelProdContainer.php \105-264 » est un fichier en cache contenant le conteneur de service. Même si il est mis dans le dossier cache, il est recréé à chaque demande. Comme vous pouvez le voir en ouvrant le fichier, la partie la plus consommatrice concerne la partie chargent les services.
Pour gagner en efficacité on pourrait utilisée un système de cache en octet comme OPCache permettant de mettre en cache le conteneur de service qui sera réutilisé permettant de gagner ce temps lors du prochaine demande.

2 et 3) Autoload\includeFile :



Permet le chargement des classes dans l'application. La seule solution pour optimiser le nombre d'appels à la fonction ***Autoload::includeFile*** serait de retirer des classes PHP.

Par contre, comme nous pouvons voir, on peut optimiser l'application en optimisant Composer.



Function calls	% Excl.	% Incl.	Calls
…oser\Autoload\includeFile			146
…er\Autoload\includeFile@1			78
…r\Autoload\includeFile@2			37
…r\Autoload\includeFile@3			18
…der::findFileWithExtension			293
…6e5d1ecfaac03180ad896c			13
…r\Autoload\includeFile@4			9
…c03180ad896c::getLoader			1
…load\ClassLoader::findFile			293
…ad\ClassLoader::loadClass			148
…\ClassLoader::loadClass@1			78
…ClassLoader::loadClass@2			37
…ClassLoader::loadClass@3			18
…ClassLoader::loadClass@4			9

Ceci est tous les fichiers concernant Composer. On pourrait optimiser une partie de c'est fichier en utilisant avant sa mise en production, la commande **composer dump-autoload –o**. Celle-ci devrait faire disparaitre les appelles a findFileWithExtension et findFile de la classe ClassLoader. Celle-ci traitera aussi la class file_exists (en quatrième position) permettant de retiré 9% de temps d'exécution.

c) La qualité du code :

Voici le tableau des choses à améliorer ou bonne chose d'après le site « WOORANK », site de test SEO. Celle-ci se compose de 19 éléments divisés en 6 parties.

Contenu :

Balise titre	14 caractère(s)	Idéalement, le titre devrait contenir entre 20 et 70 caractères espaces compris. Le titre doit être suffisamment explicite et contenir les mots clés les plus importants.
Méta description	Manquant	Les meta description devraient contenir entre 70 et 160 caractères espaces compris. Les Metas descriptions vous permettent d'influencer la façon dont vos pages sont décrites et affichées dans les résultats de recherche. Veillez à ce que chacune de vos pages ait une meta

		description suffisamment explicite et qu'elle contienne vos mots clés les plus importants. Une bonne description agit comme une publicité potentielle et encourage le lecteur à cliquer sur votre site.
Titre <h1> a <h6> dans le block « body »	Manquant dans tous les fichiers templates	Votre site n'utilise pas les balises HTML (<H1> à <H6>). Placez vos mots clés dans les titres de préférence. Assurez-vous que le premier niveau (<H1>) contient vos mots clés les plus importants. Pour un meilleur référencement, n'utilisez qu'un seul titre <H1> par page.
Attribut « alt » pour les images	Balise « alt » mis dans les balise « img »	Le texte alternatif a pour but de décrire vos images aux moteurs de recherche (et aux malvoyants)
Balise « lang »	Présent dans fichier base.html.twig	Vous avez précisé la langue de votre page

Indexation :

Robot.txt	Manquant	Le site ne dispose pas d'un fichier robots.txt, ce qui peut poser problème. Un fichier robots.txt permet d'empêcher l'accès des moteurs de recherche à certains dossiers et pages spécifiques. Ce fichier précise également où se trouve

		la sitemap XML.
SitemapXML	Manquant	En spécifiant votre sitemap XML dans votre fichier robots.txt, vous permettez aux moteurs de recherche et autres robots d'exploration d'accéder à votre sitemap de façon plus efficace chaque fois qu'ils accèdent à votre site.
Paramètres d'url	Les URLs sont"propres".	Les paramètres d'URL servent à suivre le comportement des utilisateurs sur le site (ID de session), les sources du trafic (ID de référents) ou à donner aux utilisateurs le contrôle sur le contenu d'une page (tri et filtrage).

Responsive :

Optimisation mobile	Ce site est parfaitement optimisé pour les smartphones.	Les sites compatibles avec les dispositifs mobiles (ou "mobile friendly") permettent aux utilisateurs d'accomplir des tâches courantes et d'utiliser un design ou un modèle cohérent sur tous les supports mobiles
Utilisation tactile	Parfait, les principaux boutons et liens sont suffisamment grands et ont suffisamment d'espace entre eux.	Parfait. Vos boutons et liens sont assez grands pour qu'un utilisateur appuie facilement dessus et suffisamment espacés pour qu'il puisse naviguer sans crainte d'appuyer sur un

		autre bouton par inadvertance.
Lisibilité	Parfait, le texte de cette page web est lisible sur un appareil mobile.	Au moins 60 % des textes de votre page ont une taille de police de 12 ou plus.
Fenêtre d'affichage méta	La fenêtre d'affichage est configurée. Le contenu rentre bien dans la fenêtre d'affichage spécifiée.	Etant donné que la largeur (en pixels CSS) d'une fenêtre d'affichage peut varier, le contenu de votre page ne doit pas uniquement se référer à la largeur d'une fenêtre d'affichage spécifique pour bien s'afficher.

Données structurées :

Schema .org	Manquant	Schema.org est un ensemble de syntaxes ou vocabulaires lisibles par des machines (entre autres Google) et utilisés pour donner une signification aux informations d'une page Web.
Protocole Open Graph	Manquant	Facebook a créé le protocole Open Graph afin de permettre l'intégration de n'importe quelle page Web avec sa plate-forme. D'autres réseaux sociaux ont également adopté le protocole vous permettant de contrôler et d'améliorer l'apparence du contenu de vos pages lorsqu'elles sont partagées sur les réseaux sociaux.
TwitterCard	Manquant	Les Twitter Cards sont conçues pour offrir aux utilisateurs une expérience multimédia complète lorsqu'un tweet contient des

		liens vers le contenu d'un site. Twitter propose différents types de cartes qui peuvent afficher des aperçus de contenu, des vidéos et augmenter le trafic vers votre site Web.

Branding :

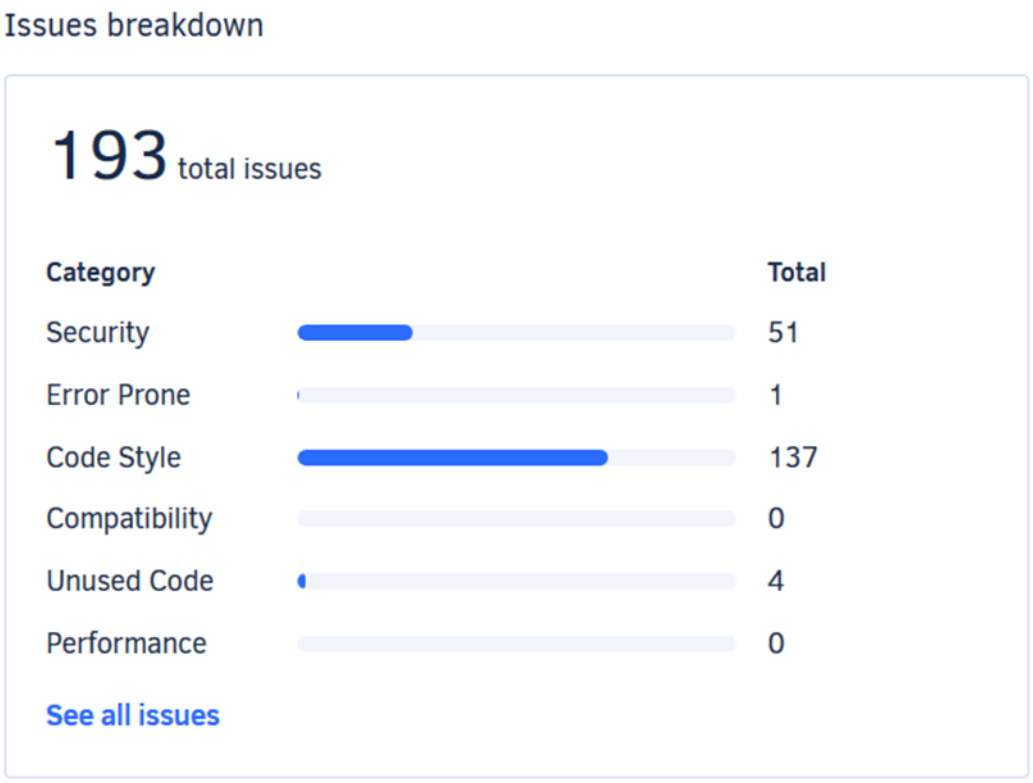
Favicon	Parfait, ce site a un favicon	Les Favicons sont de petites icônes qui apparaissent à côté du nom ou de l'URL de votre site dans un navigateur.
Page 404 personnalisée	Votre site dispose bien d'une page d'erreur 404 personnalisée.	Tirez parti de vos erreurs en fournissant une "belle" page d'erreur 404 à vos visiteurs.

Security :

Attaque XSS	Il manque une politique de sécurité sur la provenance de vos ressources (trouvé par « Dareboost »). Présence de protection via validator dans les entités.	Les attaques XSS sont un type d'attaque dans laquelle des données malveillantes sont malicieusement ajoutées aux sites Web.
Clickjacking	la page est exposée à des attaques du type "clickjacking"	Ce type d'attaque consiste à intégrer votre page sur un site malveillant via des balises <frame> ou <iframe>. Ainsi il est possible de faire croire à un internaute qu'il est sur votre propre page. L'internaute peu averti sera en confiance et sera potentiellement amené à saisir des informations que le site malveillant sera à même d'intercepter.

d) La qualité par Codacy :

Pour permettre d'avoir un niveau de qualité du code, j'ai utilisé Codacy, analyste de code automatisé. Codacy donne la note de « C » à l'application après les modifications demandées. Codacy trouve 193 problèmes sur le site web dont une grande partie touche le style de code. Certain de celle-ci sont juste de l'information ou des éléments liées à Symfony.



e) Visuellement :

Visuellement le site est loin d'être finie, les éléments ne sont pas optimisé dans l'espace affiché par le navigateur. De plus des élements sont présent alors qu'il devait êre caché si l'utilisateur n'est pas connecté voir même administrateur.

III) Rappel sur les axes d'améliorations :

De mon point de vue, il y a plusieurs améliorations à faire avant de commencé a ajouté d'autre fonctionnalités. Au niveau de la qualité du code. Suivant Codacy et les bonnes pratiques à adopter comme les Bests Pratices et le principe SOLID. Celle-ci permettra d'avoir un code de qualités connus et suivie par tous. L'amélioration des performances passe par deux choses. La mise en place d'un système de cache et des configurations requises pour éviter la refabrication du container de service et l'optimisation de l'autoloader. Sur un troisième axe, il faudrait améliorer la partie visuelle du site.

Liens utiles:

Site web: http://todoandco.michael-gt.fr/

Github: https://github.com/michaelgtfr/todoList

Codacy: https://app.codacy.com/manual/michaelgtfr/todoList/dashboard

Woorank: https://www.woorank.com/fr

Blackfire: https://blackfire.io/