

ToDo & Co – TodoList

Axes d'amélioration

Recommandation d'optimisation par blackfire :

Blackfire conseille de créer un autoloader optimisé via la commande « `composer dump-autoload -o` » ce qui va accélérer le chargement des classes.

Mise en place d'un système de cache http pour certaines routes :

Par exemple pour afficher la liste des tâches je suggère la création du cache http mais dès qu'il y a une action de type CUD il faut invalider le cache pour actualiser l'affichage.

Proposition de fonctionnalités :

Misent en place d'une deadline pour une tâche: Il est possible de mettre une échéance en place, qui mettrait en avant les dates arrivant à échéance.

Affichage de la date de création, ainsi que de l'utilisateur ayant créé une tâche : Pour le moment aucune des deux informations n'est affichée.

Ne plus afficher les tâches effectuées : Il est possible de retirer de la liste des tâches terminées, qui n'auront pas forcément d'intérêt à rester affiché, pour les laisser afficher sur une autre page qui s'occupera d'afficher seulement les tâches fini.

Affichage restreint du nombre de tâches : Pour éviter un chargement trop long de la page listant les tâches, il est possible de restreindre le nombre d'éléments et en implémentant la pagination.

Affichage des tâches individuelles : Maintenant que les tâches sont reliées aux utilisateurs, il est souhaitable de créer une page qui affichera uniquement les tâches liées à l'utilisateur connecté.

Page de gestion du compte de l'utilisateur connecté: Il faudrait ajouter la possibilité à l'utilisateur connecté de pouvoir modifier son profil et son mot de passe s'il le souhaite.

Vérification d'email : Pour le moment aucune vérification d'email n'est effectuée sur l'authenticité de l'email. Il est possible assez facilement de faire une vérification de l'email en envoyant un email qui permet de valider le compte utilisateur.

Personnalisation des pages d'erreurs : Pour le moment les pages d'erreurs sont les pages que propose Symfony. Il faudrait ajouter vos propres pages d'erreurs qui rentrent dans votre charte graphique.

Amélioration du code :

Misent en place d'un outil d'intégration : Via par exemple Travis CI, il est possible à chaque pull request de vérifier tous les tests passent correctement avant de valider les modifications en production.

Respecter les normes PHP : Via les normes PSR (<https://www.php-fig.org/psr/>), vous pourrez mettre à jour la structure du code PHP en restant à jour par rapport au normes PHP, ce qui permettra aux prochains développeurs de faciliter la lectures du code, il est possible via différents outils d'automatiser la restructuration du code au normes PHP (avec par exemple : php-cs-fixer, lien github : <https://github.com/FriendsOfPHP/PHP-CS-Fixer>).

Dépréciation :

Il est recommandé de mettre régulièrement à jour Symfony pour éviter les dépréciations du code et de certains composants.

Penser aussi à migrer le code PHP vers PHP 8.