ToDoCo

Document d’audit

Une image contenant texte, lettre

Description générée automatiquement

Table des matières

Table des matières

[Le contexte 3](#_Toc133218269)

[Migration de la version de Symfony 3](#_Toc133218270)

[Les anomalies 3](#_Toc133218271)

[Les tâches 3](#_Toc133218272)

[Le rôle Administrateur 3](#_Toc133218273)

[Bugs mineurs 4](#_Toc133218274)

[Les fonctionnalités ajoutées 4](#_Toc133218275)

[Création de compte 4](#_Toc133218276)

[Liste des utilisateurs 4](#_Toc133218277)

[Sécurisation des pages 4](#_Toc133218278)

[Gestion des tâches 4](#_Toc133218279)

[Les tests 4](#_Toc133218280)

[Qualité du code 5](#_Toc133218281)

[Optimisations 5](#_Toc133218282)

[Optimisations effectuées 5](#_Toc133218283)

[Suggestions d’optimisations 7](#_Toc133218284)

## Le contexte

Todoco est une application permettant de lister des taches à effectuer au quotidien. L’utilisateur peut créer une « todolist », où il peut rajouter des tâches, les modifier, les valider lorsqu’elles sont effectuées ou les supprimer.

Notre objectif est d’implémenter de nouvelles fonctionnalités, de corriger les anomalies existantes et de mettre en place des tests afin de veiller au bon fonctionnement de l’application.

## Migration de la version de Symfony

Lorsque nous avons commencé à travailler sur l’application, elle fonctionnait sous la version 3.1 de Symfony. Nous avons fait le choix de migrer vers la dernière version de ce framework, la 6.2, afin de pouvoir utiliser pleinement les dernières fonctionnalités.

Nous avons fait une migration classique de la version 3.1 vers la version 3.4.

Nous avons ensuite créé un nouveau projet symfony en version 6.2, et nous avons récupéré tout ce qui était encore viable de la version 3.4, tout en supprimant ce qui n’était plus d’actualité pour obtenir une application plus puissante et moins lourde.

## Les anomalies

L’application comportait plusieurs anomalies qu’il nous a fallu corriger.

### Les tâches

Les tâches créées n’étaient rattachées à aucun utilisateur, elles étaient toutes anonymes, ce qui donnait l’accès à toutes les tâches à tous les utilisateurs. Il a fallu remédier à ce problème en rattachant les tâches créées à l’utilisateur qui remplissait le formulaire. De plus chaque utilisateur pouvait supprimer n’importe quelle tâche et ce problème a aussi été résolu en ne donnant l’autorisation de suppression qu’aux utilisateurs administrateurs, et ce si la tâche est anonyme uniquement, c’est-à-dire si la valeur de l’utilisateur était nulle. Les utilisateurs peuvent bien sûr supprimer les tâches qui leur appartiennent et qu’ils sont les seuls à pouvoir consulter.

### Le rôle Administrateur

Il a donc fallu créer un rôle administrateur, rôle qu’il est possible de choisir lors de la création de compte. Ces utilisateurs administrateurs, en plus de pouvoir supprimer les tâches anonymes, peuvent aussi gérer les autres utilisateurs, et leur attribuer le rôle administrateur ou au contraire les rétrograder.

### Bugs mineurs

Il y avait aussi quelques bugs d’affichage ainsi que des liens qui ne fonctionnaient pas. Ils ont été corrigés pour le confort de l’utilisateur.

## Les fonctionnalités ajoutées

Nous avons ajouté plusieurs fonctionnalités à l’application.

### Création de compte

Il est désormais possible de créer un compte sur l’application, en ayant le choix du rôle, que ce soit administrateur ou simple utilisateur.

### Liste des utilisateurs

Il est désormais possible pour les administrateurs de voir la liste des utilisateurs, et de les promouvoir ou de les rétrograder.

### Sécurisation des pages

Les différentes pages ont été sécurisées et les pages avec la liste des tâches ne sont accessibles qu’aux utilisateurs connectés, les pages restantes accessibles aux utilisateurs non connectés sont uniquement l’accueil et les pages de connexion ou de création de compte.

### Gestion des tâches

Au niveau de la gestion des tâches, nous avons rajouté la possibilité aux administrateurs de gérer les tâches anonymes, tout en retirant cette possibilités aux simples utilisateurs.

Les tâches créées sont à présent rattachées à l’utilisateur qui les a créées, et il est le seul à pouvoir les voir.

## Les tests

Nous avons mis en place différents tests afin de vérifier le bon fonctionnement de l’application. Pour chaque fonction des différents contrôleurs, des entités ainsi que des formulaires un test a été mis en place, afin que lors des prochaines modifications dans le code de l’application nous puissions vérifier rapidement que celle-ci fonctionne encore correctement.

Pour cela nous avons utilisé phpunit, en ayant pour objectif d’obtenir au moins 95% de code coverage pour l’ensemble des fonctions de l’application. Nous avons donc mis en places des tests unitaires afin d’obtenir ce résultat. Une image contenant texte

Description générée automatiquement

## Qualité du code

Nous avons veillé pour le code à respecter le psr 12.Il définit un guide de style de codage pour PHP, afin que les développeurs puissent écrire un code clair et facilement lisible. Il spécifie des règles pour l'indentation, les espaces, les parenthèses, les accolades, les noms de variables, les commentaires et plus encore.

Nous avons ainsi obtenu une note de B sur Codacy.

Une image contenant logo

Description générée automatiquement

## Optimisations

### Optimisations effectuées

Opcache :

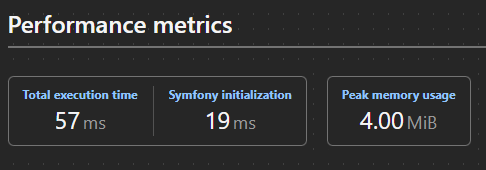
Activer opcache dans php.ini nous a permis d’optimiser considérablement l temps de chargement du site.

Avant activation :

Une image contenant texte, plein air, capture d’écran

Description générée automatiquement

Après activation :



On voit qu’après activation on obtient un gain d’environ 250ms.

Afin d’optimiser davantage le chargement du site, nous avons utilisé le query builder de symfony pour fusionner plusieurs requêtes auprès de la base de données en une seule. Cela permet de limiter la bande passante ainsi que le tps de chargement.

Sur cette page par exemple, avant optimisation nous voyons qu’il y a 3 requêtes pour un temps de chargement de 2.31ms :

Une image contenant texte, écran, capture d’écran

Description générée automatiquement

Après optimisation, le temps de chargement est plus court et il n’y a que deux requêtes :

Une image contenant texte, écran, capture d’écran, noir

Description générée automatiquement

Il y a donc un gain de temps et de performances.

### Suggestions d’optimisations

Afin d’optimiser d’avantage le site, nous suggérons la mise en place du cache, ce qui accélèrerait encore le temps de chargement des pages.