**Audit de qualité du code & performance de l’application**

Table des matières

[II. Contexte 2](#_Toc157437142)

[III. Migration de la version de Symfony 2](#_Toc157437143)

[IV. Anomalies… 2](#_Toc157437144)

[a) …à corriger 2](#_Toc157437145)

[b) …découvertes et corrigées 2](#_Toc157437146)

[V. Fonctionnalités ajoutées 2](#_Toc157437147)

[VI. Tests (unitaires et fonctionnels) 2](#_Toc157437148)

[VII. Qualité du code 2](#_Toc157437149)

[VIII. Performance 2](#_Toc157437150)

# Contexte

Mon rôle pour ToDo & Co est d’améliorer la qualité de l’application et de définir les contours de la contribution de futurs développeurs.

# Migration de la version de Symfony

# Anomalies…

## …à corriger

A la demande initiale, les anomalies repérées et à corriger sont :

* Une tâche doit être rattachée à un utilisateur à sa sauvegarde. (Les tâches déjà présentes doivent être rattachées à un utilisateur « anonyme ».
* Choisir un rôle pour un utilisateur à sa création. A la modification d’un utilisateur (par un admin) il doit être possible de changer le rôle d’un utilisateur.

## …découvertes et corrigées

Outre les anomalies précédentes corrigées, j’ai également mis en place la création de l’utilisateur par lui-même directement.

L’utilisateur connecté peut modifier/supprimer ses propres tâches uniquement. Les tâches anonymisées peuvent, elles, uniquement être supprimées par un administrateur de l’application.

L’utilisateur connecté pouvait consulter sur un même dashboard l’ensemble des tâches ouvertes et fermées simultanément. Comme dit précédemment, il n’a désormais accès qu’à ses propres tâches. De plus, il peut consulter séparément ses tâches ouvertes et celles terminées (et passer de l’un à l’autre le statut d’une de ses tâches) pour plus de visibilité.

# Fonctionnalités ajoutées

De manière à assurer l’ensemble de ces fonctionnalités de manière pérenne, un système de sécurité et d’authentification a été mis en place (cf. le document de documentation technique).

Il est donc impossible pour un utilisateur non connecté de réaliser certaines actions, impossible également pour un utilisateur d’interagir avec les tâches d’autres utilisateurs, de modifier le rôle d’un utilisateur sans être administrateur.

# Tests (unitaires et fonctionnels)

Une batterie de tests unitaires et fonctionnels a été implémentée de manière à garantir le bon fonctionnement du projet. Ces tests ont été mis en place avec PHP/Unit et XDebug. Lors de l’implémentation d’une nouvelle fonctionnalité il est également demandé au développeur d’implémenter les tests correspondants. Il est également demandé d’avoir une couverture de code d’au moins 70%.

# Qualité du code

# Performance