**AUDIT - TODO & CO**

**Qualité du code**

Après la soumission du projet à l'outil de référence qu'est le SensioLabs Insights, il en ressort que certaines modifications restent encore à effectuer, pour que le projet respecte intégralement les bonnes pratiques en vigueur.

* Le fichier bootstrap.js  
  Il conserve un commentaire "TODO" or cela ne respecte pas les bonnes pratiques.  
  La solution serait d'appelé les Frameworks Twitter Bootstrap et jQuery par le biais d'un service CDN.
* La classe déprécié "Symfony\Component\EventDispatcher\ContainerAwareEventDispatcher" est utilisé par le service "event\_dispatcher " alors que Symfony 3.3 préconise d'utiliser "Symfony\Component\EventDispatcher\EventDispatcher" depuis sa dernière mise à jour.  
  Le code développé dans ce projet n'utilise à aucun moment cette classe ou ce service, j'en déduis que le problème sera résolu par la prochaine mise à jour de Symfony.
* Le "favicon" par défaut de Symfony est utilisé  
  Ceci ne représente en rien un problème technique, hormis le fait qu'il serait préférable d'ajouter un favicon à l'image du projet pour améliorer l'expérience des utilisateurs.

Une fois ces modifications réalisé le SensioLabs Insights octroiera la note maximale sur le projet en terme de bonnes pratiques.

**Performance :**

Le projet a été soumis à l'outil Google Insight pour calculer la performance de l'application sur les mobiles et les ordinateurs.   
L'application obtient actuellement une note global correcte, 79/100 pour les mobiles et 89/100 pour les ordinateurs.

Ces résultats peuvent être encore améliorer grâce au préconisations de l'outils :

* À la compression des images.
* L'utilisation des en-têtes "Expires" pour la mise en cache du navigateur.
* Regroupement et minification des Assets.

Les regroupements peuvent se faire avec les Frameworks Bootstrap et jQuery, il est conseillé d'avoir un unique fichier de "grosse" taille plutôt qu'une multitude de "petit" fichier.

Le composant Assetic-Bundle de Symfony répond parfaitement à ces besoins. Il permet le regroupement et la minification de vos assets, mais il s'occupe également de la compression de vos images.  
  
Ce composant augmenterait la dette technique mais serait un véritable gain d'optimisation pour la performance de l'application.

**Dette technique :**

En l'état les seules dettes techniques révélés par l'analyse intégrale du projet Todo & Co sont les Frameworks utilisés en l'occurrence il s'agit de Symfony, Twitter Bootstrap et de jQuery.

La version d'origine de Symfony pour ce projet est la 3.1, à l'heure de l'audit la version courante de Symfony est la 3.3.  
La mise à jour ayant été effectué il en ressort que certaine fonction soit déprécié est doivent être remplacé pour anticiper les prochaines mise à jour du Framework. Ces modifications doivent suivre les recommandations de la documentation de Symfony, voici une liste non exhaustive des modifications déjà effectué :

* Modification de la gestion des routes logout et login\_check  
  Les routes étaient gérées dans le contrôleur ce qui se traduisait par des méthodes "vides", elles sont dorénavant dans le fichier routing.yml
* Déplacement des formulaires dans Form/Type  
  Les bonnes pratiques de Symfony veulent que les formulaires soit placé dans ces dossiers.
* Modification du Secret et du nom de Cookie  
  Pour des questions évidentes de sécurité le secret et le nom des cookies ont été modifié.
* Définissions des méthodes HTTP   
  Chaque routes du projet doivent être accessible via une méthodes.
* Ajout de $form->isSubmitted () avant $form->isValid() pour la vérification de soumission des formulaires
* Le choix de n'avoir utilisé aucun bundle externe jusqu’à présent pour ce projet n'ajoute aucune dette technique supplémentaire.

Il reste une petite partie de l'application encore non couverte par les tests automatisés, il peut être important d'obtenir une couverture de code de 100% pour anticiper une dette technique éventuelle sur ces portions de code.

Lors de n'importe quelles futures mises à jour veillez à ce que la dette technique ne s'accumule pas pour conserver un projet propre, lisible et compréhensible par n'importe quel développeur qui reprendrait le projet. Pour cela il est conseillé d'utilisé la technique du Test-Driven Development pour toutes les futures évolutions de votre projet. Soumettez également le code à l'outils de Sensiolabs Insight pour vérifier que les bonnes pratiques en vigueur soient respectées au maximum.