
RAPPORT D'AUDIT DE QUALITE DE CODE ET DE PERFORMANCE DE L'APPLICATION TODOLIST

SOMMAIRE

<u>PRESENTATION DU PROJET</u>	<u>3</u>
1.CONTEXTE.....	3
2.DESCRPTION DU BESOIN	3
3.PLATEFORME TECHNIQUE	3
<u>AUDIT DE QUALITE DE CODE</u>	<u>4</u>
1. L'ANALYSE DU PROJET INITIAL PAR SENSIOLABSINSIGHT	4
2. L'ANALYSE DU PROJET ACTUEL PAR SENSIOLABSINSIGHT	5
3. L'ANALYSE DU PROJET ACTUEL PAR CODACY	7
4.SOLUTIONS FOURNIES.....	7
ERREURS CRITIQUE :	7
ERREURS MAJEURES :	7
ERREURS MINEURES :	8
ERREURS INFOS :	8
5.MODIFICATION DU CODE PHP	8
<u>AUDIT DE PERFORMANCE DE L'API.....</u>	<u>9</u>
1.PROFILAGE VIA BLACKFIRE DU PROJET ACTUEL	9
2. PROFILAGE VIA BLACKFIRE DU PROJET ACTUEL AVEC AMELIORATION	10

PRESENTATION DU PROJET

1.CONTEXTE

Todolist est une application permettant de gérer ses tâches quotidiennes.

L'entreprise vient tout juste d'être montée, et l'application a dû être développée à toute vitesse pour permettre de montrer à de potentiels investisseurs que le concept est viable (on parle de Minimum Viable Product ou MVP).

Le choix du développeur précédent a été d'utiliser le framework PHP Symfony.

Mon rôle ici est donc d'améliorer la qualité de l'application.

2.DESCRPTION DU BESOIN

Le projet est donc de faire les tâches suivantes :

- L'implémentation de nouvelles fonctionnalités ;
- La correction de quelques anomalies ;
- Et l'implémentation de tests automatisés.

Il m'est également demandé d'analyser le projet grâce à des outils vous permettant d'avoir une vision d'ensemble de la qualité du code et des différents axes de performance de l'application.

3.PLATEFORME TECHNIQUE

Framework Symfony :

Version du projet initial : 3.1.*

Version après correction : 3.4.*

Environnement de développement :

Version Wamp 3.0.6 - Windows 64bit

Version Apache 2.4.23

Version de PHP 7.0.10



Version de MySQL : 5.7.14


AUDIT DE QUALITE DE CODE

1. L'ANALYSE DU PROJET INITIAL PAR SENSIOLABSINSIGHT


SensioLabsInsight


DashboardDocumentationPricingMy accountBlog

 tuyetrinhvt / mytodolist 



 Edit project

Using the Free plan, you are limited to one analysis.

 Please upgrade to remove this limit.



Analyzed 4 minutes ago by tuyetrinhvt, duration: a few seconds

 **tuyetrinhvt / mytodolist #1** 

1 Critical3 Major5 Minor5 Info


3.5 days

to get the Platinum Medal

Critical security alerts

- Projects must not depend on dependencies with known security issues

To see this analysis:


 Upgrade to get the Full Report

Details

Source:
[git@github.com:tuyetrinhvt/myt...](https://github.com/tuyetrinhvt/mytodolist)
Project type: Symfony 3/4 web project
Default branch: master

Stats

Lines of code: 749

 **SensioLabsInsight**
No medal yet 2018-02-19

The screenshot displays the SensioLabsInsight interface. At the top, there's a navigation bar with links like 'Dashboard', 'Documentation', 'Pricing', 'My account', and 'Blog'. The main header shows the project name 'tuyettrinhvt / mytodolist #1' and a status 'Analyzed 3 minutes ago by tuyettrinhvt, duration: a few seconds'. A sidebar on the left provides project details: '3.5 days to get the Platinum Medal', a search bar, and sections for 'Severity' (1 Critical), 'Category' (5 Security), 'Developer' (1 Collective), 'Stats' (Lines of code: 749, Nb of violations: 14), and 'Last commit' (Merge pull request #1 from tuyettrinhvt/create-project by TuyettrinhVO 3 days ago). The main content area features a yellow banner prompting an upgrade to a paid plan for full report access. Below this, a section titled 'Projects must not depend on dependencies with known security issues' lists four detected security issues in the 'symfony/symfony' package (version 3.1.6.0):
 1) CVE-2017-16653: CSRF protection does not use different tokens for HTTP and HTTPS
 2) CVE-2017-16654: Intl bundle readers breaking out of paths
 3) CVE-2017-16790: Ensure that submitted data are uploaded files
 4) CVE-2017-16652: Open redirect vulnerability on security handlers
 The interface also includes buttons for 'Comment', 'Ignore', 'Open issue', and 'Permalink'.

La légende des types d'erreurs :


Critique : indique un point risquant de mettre en péril de bon fonctionnement de l'application.

Major : indique un point qui risque de créer des bugs, ou des points de sécurité moins graves que ceux marqués en critique.


Minor : est des points moins importants tels que problèmes de style ou de lisibilité.


Info : est des points d'informations sur les bonnes procédures en vigueur ou conseils pouvant améliorer le contenu.

2. L'ANALYSE DU PROJET ACTUEL PAR SENSIOLABSINSIGHT

 **tuyettrinhvo / mytodolist #1**

Analyzed 2 minutes ago by tuyettrinhvo, duration: a few seconds


 **Edit project**



43/100


6.5 hours
to get the Platinum Medal


Stats
Lines of code: 450
Nb of violations: 10

Last commit
edit config.yml by
tuyettrinhvo 2 minutes
ago. **master**


 **SensioLabsInsight**
Gold Medal 2018-03-03


Upgrade your plan to get full report access
Using the Free plan, you are limited to one analysis. To unleash the full power of **SensioLabsInsight**, you have to upgrade to a paid plan.
 **Upgrade to get the Full Report**

 **tuyettrinhvo / mytodolist**


 **Edit project**


SensioLabsInsight really shines when each commit of your project is analyzed.
[Enable automatic analysis](#)

Using the Free plan, you are limited to one analysis.
 **Please upgrade to remove this limit.**



43/100

Analyzed 2 minutes ago by tuyettrinhvo, duration: a few seconds

 **tuyettrinhvo / mytodolist #1**

 10 Info

6.5 hours to get the Platinum Medal

To see this analysis:
 **Upgrade to get the Full Report**

Details


Source:
git@github.com:tuyettrinhvo/myt...

Project type: Symfony 3/4 web project

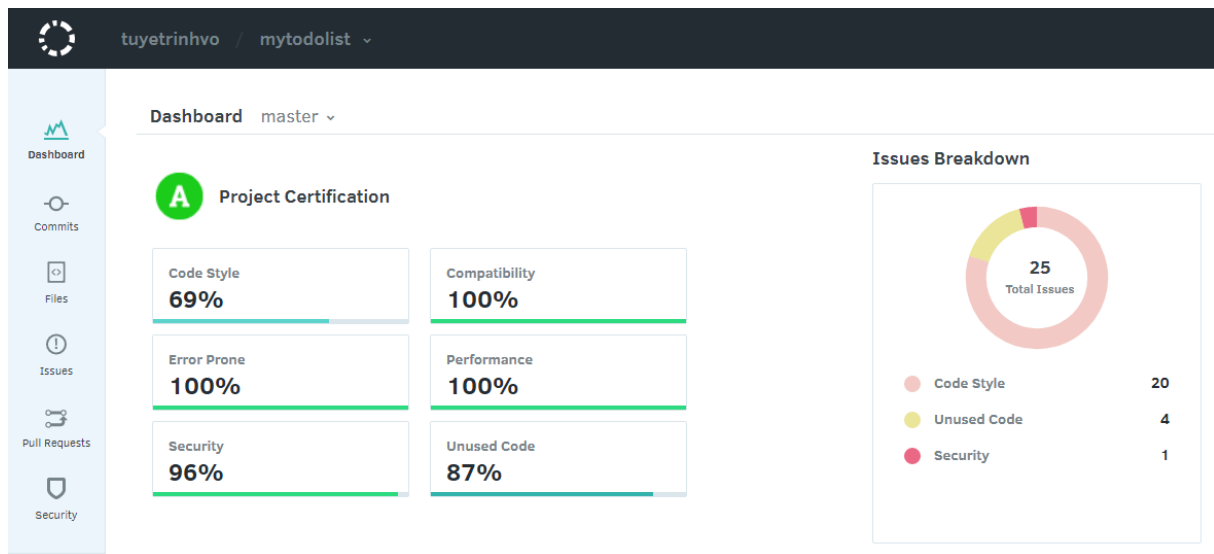
Default branch: master

Stats

Lines of code: 450

 **SensioLabsInsight**
Gold Medal 2018-03-03

3. L'ANALYSE DU PROJET ACTUEL PAR CODACY



4.SOLUTIONS FOURNIES

Erreurs critiques :

La version **3.1.*** n'est plus maintenue par la société Sensiolabs, créatrice du Framework Symfony depuis 2017. J'ai donc mise à jour la version Symfony à **3.4.***

Dans le fichier `parameters.yml.dist`, j'ai corrigé la série de caractères du paramètre secret.

Erreurs majeures :

J'ai mis en commentaire les méthodes `loginCheck()` et `loginOut()` du Security Controller. J'ai mis les routes dans `app/config/routing.yml`

Dans le fichier `composer.json` : J'ai ajouté le champ « description »

Dans le fichier `config.yml` : J'ai déclaré la version serveur de mysql utilisé, la 5.7.14 et corriger le paramètre `framework\router\resource` : `"%kernel.root_dir%/app/config/routing.yml"`

Dans le dossier Web, j'ai supprimé le fichier `config.php`

Erreurs mineures :

J'ai créé le dossier AppBundle/Form/Type et déplacé les fichiers TaskType.php et UserType.php dedans.

Dans le fichier parameters.yml.dist, j'ai corrigé la série de caractères du paramètre secret.

Dans le dossier Web/app.php, j'ai relevé les lignes HttpCache (qui ont été mis en commentaire)

Dans le fichier config.yml : J'ai déclaré le nom du cookie de session.

Dans le fichier base.html.twig, j'ai inclus liens CDN pour les fichiers bootstrap.min.js, bootstrap.min.css, jquery.min.js.

Erreurs infos :

J'ai personnalisé les pages d'erreurs en créant les fichiers par exemple error.html.twig dans dossier app/Resources/TwigBundle/views/Exceptions.

J'ai changé les favicons dans le dossier Web.

J'ai ajouté les méthodes http GET, POST correspondantes aux routes dans les _Controller.php

Dans le fichier composer.json : J'ai ajouté le namespace « AppBundle\ » du champ « autoload »

J'ai mis à jour le fichier composer.json avant de versionner sur Git.

5.MODIFICATION DU CODE PHP

Sur les méthodes de création ou de modification par formulaire : j'ai corrigé Form::isValid() en Form::isSubmitted() && Form::isValid()

J'ai changé \$em en \$entityManager.

AUDIT DE PERFORMANCE DE L'API

1.PROFILAGE VIA BLACKFIRE DU PROJET ACTUEL

Le test de performance est le processus permettant de déterminer la vitesse ou l'efficacité de l'application.

Blackfire Profiler, développé par la société Sensiolabs, est un outil qui instrumentalise les applications PHP pour collecter des données sur les ressources serveur consommées telles que la mémoire, le temps CPU et les opérations d'I/O.

Blackfire peut être utilisé à n'importe quelle étape du cycle de vie de l'application : pendant le développement, le test, la mise en scène et la production, pour profiler, tester, déboguer et optimiser ses performances.

Ci-dessous le tableau récapitulatif des requêtes testés sur l'environnement de production, donc le cache Symfony et httpCache sont activés :

URI	Time	I/O Wait	CPU Time	Memory	Network	SQL Queries
/login	266 ms	218 ms	48.1 ms	8.24 MB	0 B	0 µs / 0 rq
/	362 ms	299 ms	62.9 ms	11.9 MB	753 B	347 µs / 1 rq
/tasks/create	386 ms	318 ms	68.3 ms	14.2 MB	753 B	515 µs / 1 rq
/tasks/1/edit	388 ms	320 ms	67.7 ms	14.1 MB	1.35 kB	842 µs / 2 rq
/tasks	328 ms	276 ms	52 ms	11.3 MB	1.91 kB	1.52 ms / 2 rq
/users/create	386 ms	318 ms	68.3 ms	14.2 MB	753 B	515 µs / 1 rq
/users/1/edit	384 ms	316 ms	68 ms	14.2 MB	1.32 kB	779 µs / 2 rq
/users	308 ms	258 ms	49.7 ms	11.2 MB	2.46 kB	733 µs / 2 rq

2. PROFILAGE VIA BLACKFIRE DU PROJET ACTUEL AVEC AMELIORATION

La performance de l'application après les mises à jour est raisonnable, en plus l'application est testée en local, et sous Windows.

Nous ne pouvons pas vraiment analyser la performance d'une application sans le server réel.

Cependant, j'ai suivi les recommandations du Blackfire, toujours testé sur l'environnement de production avec le cache Symfony et httpCache sont activés, j'ai obtenu ensuite les résultats ci-dessous :

Après avoir mis en place le cache Doctrine orm

L'utilisation d'annotations pour le mappage a un coût : Doctrine doit transformer cette configuration en code PHP standard exécuté par l'application. Dans les applications réelles avec beaucoup d'entités complexes, ce processus de conversion a un impact sévère sur les performances. C'est pourquoi nous devons mettre en cache l'analyse des annotations Doctrine en production.

URI	Time	I/O Wait	CPU Time	Memory	Network	SQL Queries
/login	128 ms	95.2 ms	32.6 ms	7.54 MB	0 B	0 µs / 0 rq
/	173 ms	133 ms	39.4 ms	10.5 MB	753 B	341 µs / 1 rq
/tasks/create	223 ms	168 ms	55.3 ms	14 MB	753 B	344 µs / 1 rq
/tasks/4/edit	225 ms	169 ms	56.9 ms	14 MB	1.41 kB	620 µs / 2 rq
/tasks	181 ms	136 ms	44.7 ms	11.2 MB	3.47 kB	1.19 ms / 3 rq
/users/create	226 ms	167 ms	59.2 ms	14.2 MB	753 B	310 µs / 1 rq
/users/1/edit	229 ms	168 ms	61 ms	14.2 MB	1.32 kB	687 µs / 2 rq
/users	175 ms	133 ms	41.7 ms	11 MB	2.08 kB	626 µs / 2 rq

Après avoir activé la cache d'opcode PHP (Zend OPcache) en plus :

Chaque fois que PHP charge une classe, Composer doit rechercher le fichier correspondant sur le système de fichiers, ce qui est un processus lent.

Parce que le classmap peut être énorme, il est fortement recommandé d'avoir un cache d'opcode PHP installé (comme Zend OPcache).

URI	Time	I/O Wait	CPU Time	Memory	Network	SQL Queries
/login	33.7 ms	17.6 ms	16.1 ms	1.59 MB	0 B	0 µs / 0 rq
/	50.4 ms	29.7 ms	20.6 ms	2.29 MB	753 B	317 µs / 1 rq
/tasks/create	61.5 ms	33.6 ms	27.9 ms	3.14 MB	753 B	333 µs / 1 rq
/tasks/4/edit	61.6 ms	32.5 ms	29 ms	3.17 MB	1.41 kB	646 µs / 2 rq
/tasks	52.1 ms	30.3 ms	21.8 ms	2.39 MB	3.47 kB	898 µs / 3 rq
/users/create	64.4 ms	31.8 ms	32.6 ms	3.33 MB	753 B	352 µs / 1 rq
/users/1/edit	69.1 ms	35.9 ms	33.1 ms	3.35 MB	1.32 kB	637 µs / 2 rq
/users	47.4 ms	27.7 ms	19.7 ms	2.31 MB	2.08 kB	619 µs / 2 rq