



Audit de qualité

Application web *TO DO LIST* par ToDo & CO



Version : 0.1

Auteur : Bastien Vacherand

Date de la dernière mise à jour : 16 avril 2019

Sommaire

1.	Cadre du projet	3
1.1.	Contexte.....	3
1.2.	Mes objectifs	3
2.	Code initial.....	3
2.1.	Données techniques	3
2.2.	Accès à l'application	3
2.3.	Profiling.....	4
2.4.	Code Review.....	4
2.5.	Dépréciations	5
2.6.	Version de PHP et Symfony.....	6
2.7.	Sécurité.....	6
2.8.	Analyse personnelle	7
3.	Améliorations.....	7
3.1.	Sécurité	7
3.2.	Authentification	9
3.3.	Les entités	10
3.4.	Les contrôleurs	10
3.5.	Pages d'erreurs.....	11
3.6.	Php Docs	11
3.7.	Dépréciations	12
4.	Performances.....	12
5.	Tests	13
5.1.	Tests unitaires	13
5.2.	Tests fonctionnels.....	13

1. Cadre du projet

1.1. Contexte

La startup **To Do & Co** vient de réussir une levée de fond afin de développer son application *TO DO LIST* permettant de gérer ses tâches quotidiennes. Un *Minimum Viable Product* a déjà été réalisé grâce au Framework PHP Symfony.

1.2. Mes objectifs

La mission qui m'a été confiée consiste en :

- L'amélioration de la qualité du code
- L'amélioration des performances de l'application
- L'implémentation de nouvelles fonctionnalités
- La mise en place de tests automatisés

Afin de réduire la dette technique de l'application, je m'appuierais sur plusieurs outils de diagnostics qui offrent des métriques objectives sur la qualité et les performances.

2. Code initial

2.1. Données techniques

Environnement de travail :

PHP	7.2.4
Symfony	3.1
Doctrine/ORM	2.5
Swiftmailer	2.3
Bootstrap	3.3.7

2.2. Accès à l'application

Après avoir lancé l'application sur le serveur local, la page d'accueil s'affiche correctement. Cependant les pages utilisant le validator lève cette exception :



Cette erreur vient de l'incompatibilité de ma version de PHP avec des composants de Symfony trop anciens. Je les mets donc à jour grâce à la commande « `composer update` » pour pouvoir continuer l'audit initial.

2.3. Profiling

Afin de tester les performances de l'application, j'ai mis en place des profilings grâce à blackfire.io

Profil page d'accueil : <https://blackfire.io/profiles/64434812-81b4-4872-82ba-3b2bbb4febcd/graph>

Profil page des tâches : <https://blackfire.io/profiles/38a936fe-6ad2-4820-8288-4a0653e42377/graph>

La page d'accueil est générée en 304 ms et consomme 9.46 Mb de mémoire.

En analysant le call graph, on s'aperçoit que la classe `IncludeFile` de l'autoload représente 68 % du temps de réponse et qu'elle est appelée 140 fois.

On pourrait considérablement réduire ce cout en dumpant l'autoloader de composer grâce à la commande « `composer dump-autoload --optimize` ».

La page des taches est générée en en 309 ms et consomme 9.56 Mb de mémoire.

2.4. Code Review

Le code initial a été testé avec Codacy. Les fichiers purement Symfony ne sont pas couverts par l'analyse.

Une seule erreur a été reportée : un paramètre non utilisé dans le `SecurityController`.

src/AppBundle/Controller/SecurityController.php

Avoid unused parameters such as '\$request'.

Time to fix: 5 minutes

```
11  /**
12   * @Route("/login", name="login")
13   */
14  public function loginAction(Request $request)
15  {
16      $authenticationUtils = $this->get('security.authentication_utils');
```

Why is this an issue?

Avoid passing parameters to methods or constructors and then not using those parameters.

Les PSR sont respectés (analyse avec phpcodesniffer).

2.5 Dépréciations

Le profiler de Symfony nous informe que le code contient 9 dépréciations :

Log Messages

Info. & Errors 1 Deprecations 9 Debug 31 Silenced Errors 0

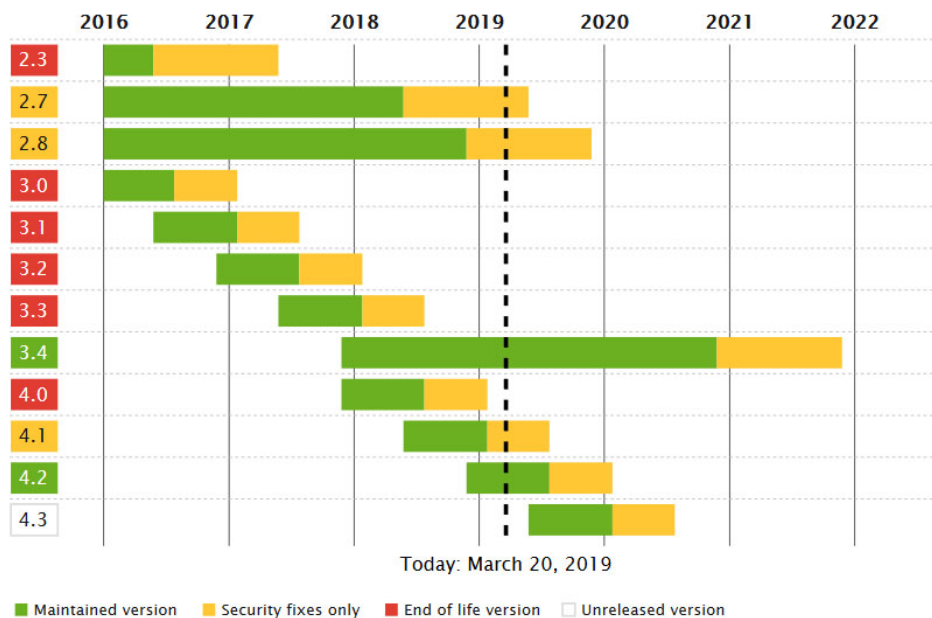
Time	Channel	Message
14:38:19	php	The Doctrine\ORM\Proxy\ProxyFactory class extends Doctrine\Common\Proxy\AbstractProxyFactory that is deprecated The Doctrine\Common\Proxy component is deprecated, please use ocradius/proxy-manager instead. Show stack trace
14:38:19	php	Using the "Twig_Loader_Filesystem" class is deprecated since Twig version 2.7, use "Twig\Loader\FilesystemLoader" instead. Show stack trace
14:38:19	php	Using the "Twig_Extension_Profiler" class is deprecated since Twig version 2.7, use "Twig\Extension\ProfilerExtension" instead. Show stack trace
14:38:19	php	Using the "Twig_Profiler_Profile" class is deprecated since Twig version 2.7, use "Twig\Profiler\Profile" instead. Show stack trace
14:38:19	php	Using the "Twig_BaseNodeVisitor" class is deprecated since Twig version 2.7, use "Twig\NodeVisitor\AbstractNodeVisitor" instead. Show stack trace
14:38:19	php	Using the "Twig_Node" class is deprecated since Twig version 2.7, use "Twig\Node\Node" instead. Show stack trace
14:38:19	php	Using the "Twig_Extension_Debug" class is deprecated since Twig version 2.7, use "Twig\Extension\DebugExtension" instead. Show stack trace
14:38:19	php	Using the "Twig_Extension_InitRuntimeInterface" class is deprecated since Twig version 2.7, use "Twig\Extension\InitRuntimeInterface" instead. Show stack trace
14:38:19	php	The Symfony\Bridge\Twig\Extension\FormExtension class implements Twig\Extension\InitRuntimeInterface that is deprecated since Twig 2.7, to be removed in 3.0 Show stack trace

Ces dépréciations sont liés au fait que doctrine et twig ont été mis à jour mais Symfony est toujours sur une version qui n'est plus maintenue depuis 2017. Une mise à jour de Symfony devrait régler ces problèmes.

2.6 Version de PHP et Symfony

La version de Symfony utilisée est la 3.1.

D'après la road map de Sensiolab elle n'est plus maintenue :



D'après ce graphique, la version la plus appropriée est la 3.4 car elle sera maintenue jusqu'à fin 2020.

Je préconise également d'utiliser la version de 7 de PHP, ce qui permet d'accroître les performances et donne accès aux dernières fonctionnalités (sources : phpbenchmarks.com).

2.7 Sécurité

La sécurité de l'application est insuffisante.

Un utilisateur qui a créé un compte peut modifier n'importe quel compte, sans même être authentifié.

Il est nécessaire de mettre en place un système de rôles et de vérifier les droits pour chaque action sensible.

Les formulaires ne sont pas protégés contre la faille CSRF. Il convient d'activer l'option dans les formulaires.

2.8 Analyse personnelle

La librairie Swiftmail est installée alors qu'elle n'est pas utilisée. Il convient de l'enlever du composer.json.

Le DefaultController ne sert qu'à afficher la page d'accueil. Je propose de transférer cette fonctionnalité dans un autre controller.

Cliquer sur « afficher la liste des taches à faire » affiche toutes les taches.

Cliquer sur « consulter la liste des taches terminer » n'affiche rien. Il faudrait replacer les taches dans la catégorie à laquelle elles appartiennent.

Le code de manière générale n'est pas documenté. Je préconise de décrire les classes et les méthodes avec des annotations phpdoc.

Les contrôleurs étendent la classe Controller, ce qui est déprécié dans les dernières versions de Symfony. On utilisera plutôt la classe AbstractController disponible depuis la version 3.3.

La méthode LoginAction utilise l'AuthenticationUtils dans le container, ce qui est également déprécié. On utilisera plutôt l'injection de dépendance. D'une manière générale, on remplacera l'utilisation du container par l'injection de dépendance.

3. Améliorations

3.1. Sécurité

L'accès aux parties sensibles du site a été restreint aux utilisateurs authentifiés.

Seules les pages de login et d'inscription sont accessibles de manière anonyme. Un système d'autorisation a été mis en place avec un ROLE_USER et un ROLE_ADMIN que l'on peut choisir à la création du compte.

L'administration des comptes est restreinte aux utilisateurs possédant le rôle ROLE_ADMIN.

```

1. // app/config/security
2.
3. role_hierarchy:
4.     ROLE_ADMIN:     ROLE_USER
5.
6. access_control:
7.     - { path: ^/login, roles: IS_AUTHENTICATED_ANONYMOUSLY }
8.     - { path: ^/users/create, roles: IS_AUTHENTICATED_ANONYMOUSLY }
9.     - { path: ^/users, roles: ROLE_ADMIN }
10.    - { path: ^/, roles: [ROLE_ADMIN, ROLE_USER]}

```

Afin de compléter le système d'autorisation, des voters ont été mis en place. Par exemple, un utilisateur ne peut modifier une tâche que s'il en est l'auteur :

```

1. // src/AppBundle/Security/Voter/TaskVoter.php
2.
3. protected function voteOnAttribute($attribute, $task, TokenInterface $token)
4. {
5.     $user = $token->getUser();
6.     // if the user is anonymous, do not grant access
7.     if (!$user instanceof UserInterface) {
8.         return false;
9.     }
10.
11.    switch ($attribute) {
12.        case 'EDIT':
13.            $role = $user->getRoles();
14.            if ($user == $task->getUser()) {
15.                return true;
16.            }
17.
18.        default:
19.            return false;
20.    }
21.
22.    return false;
23. }
24.
25. // src/AppBundle/Controller/TaskController.php
26.
27. public function editAction(Task $task, Request $request, EntityManagerInterface
28. $em)
29. {
30.     $this->denyAccessUnlessGranted('EDIT', $task);
31.
32.     // ...
33. }

```

Les tâches dont l'auteur est anonyme ne peuvent être éditées ou supprimées que par un utilisateur avec un ROLE_ADMIN :

```

1. // src/AppBundle/Security/Voter/TaskVoter.php
2.
3. if ($user->isAdmin() && $task->getUser()->isAnon()) {
4.     return true;
5. }
6. break;
7. case 'DELETE':
8.     $role = $user->getRoles();
9.     if ($user == $task->getUser()) {
10.        return true;
11.    }
12.    if ($user->isAdmin() && $task->getUser()->isAnon()) {
13.        return true;
14.    }
15. }

```



```
15.         break;
```

La protection contre la faille CSRF a été activée pour les formulaires :

```
1. // src/AppBundle/Form/UserType.php
2.
3.     public function configureOptions(OptionsResolver $resolver)
4.     {
5.         $resolver->setDefaults([
6.             'data_class'      => User::class,
7.             'csrf_protection' => true,
8.         ]);
9.     }
```

3.2 Authentification

Un authenticator personnalisé a été mis en place pour le formulaire de login :

```
1. // app/config/security.yml
2.     guard:
3.         authenticators:
4.             - AppBundle\Security\LoginFormAuthenticator
```

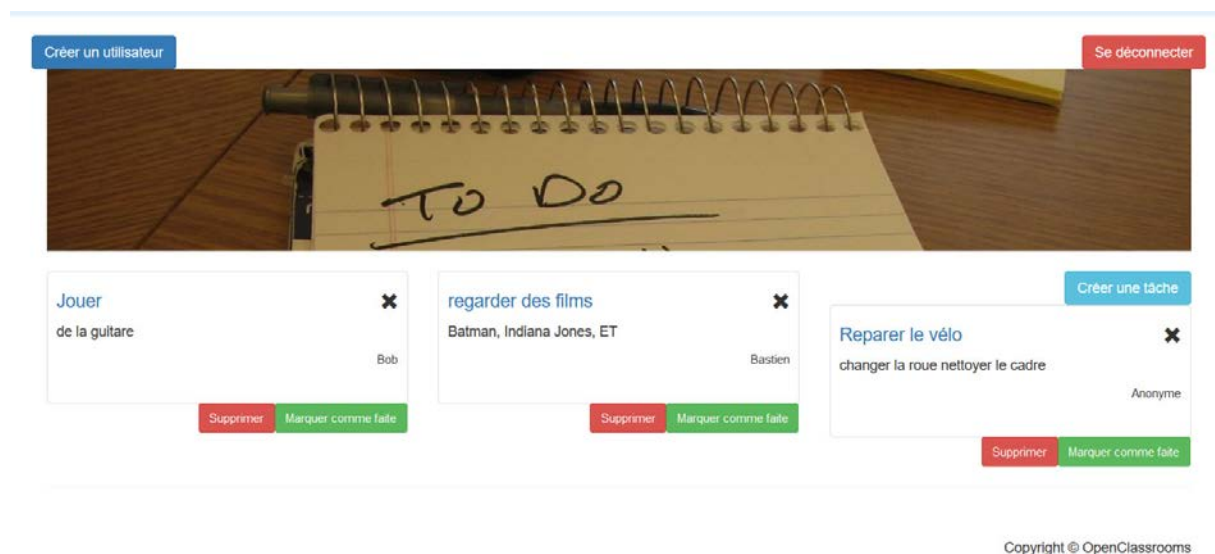
Cela permet d'avoir des messages d'erreurs plus précis en cas de problème d'authentification :

```
1. // src/AppBundle/Security/LoginFormAuthenticator.php
2.
3.     public function getUser($credentials, UserProviderInterface $userProvider)
4.     {
5.         $token = new CsrfToken('authenticate', $credentials['csrf_token']);
6.         if (!$this->csrfTokenManager->isTokenValid($token)) {
7.             throw new InvalidCsrfTokenException();
8.         }
9.
10.        $user = $this->entityManager->getRepository(User::class)-
>findOneBy(['username' => $credentials[
11.            'username'
12.        ]]);
13.
14.        if (!$user) {
15.            // fail authentication with a custom error
16.            throw new CustomUserMessageAuthenticationException('Cet utilisateur n\'e
st pas enregistré.');
```

3.3 Les entités

Une relation a été créée entre la tâche et son utilisateur. Le nom de l'auteur est désormais affiché sur les tâches :

```
1. // src/AppBundle/Entity/Task.php
2.
3. /**
4.  * @var User
5.  *
6.  * @ORM\ManyToOne(targetEntity="AppBundle\Entity\User", inversedBy="tasks")
7.  * @ORM\JoinColumn(nullable=true)
8.  */
9. private $user;
```



3.4 Les contrôleurs

Grace à l'utilisation de PHP 7, le Type Hinting a été mis en place dans les contrôleurs :

```
1. // src/AppBundle/Controller/TaskController.php
2.
3. public function toggleTaskAction(Task $task, EntityManagerInterface $em)
4. {
5.     $task->toggle(!$task->isDone());
6.     $em->flush();
7.
8.     $this->addFlash('success', sprintf('La tâche %s a bien été marquée comme faite.', $task->getTitle()));
9.
10.    return $this->redirectToRoute('task_todo_list');
11. }
```

Les contrôleurs étendent désormais la classe AbstractController :

```

1. // src/AppBundle/Controller/TaskController.php
2.
3. class TaskController extends AbstractController
4. {
5.
6.     /**
7.      * @Route("/", name="homepage")
8.      *
9.      * @return Response
10.     */
11.     public function indexAction()
12.     {
13.         $response = $this->render('task/index.html.twig');
14.         $response->setSharedMaxAge(200);
15.         $response->headers->addCacheControlDirective('must-revalidate', true);
16.
17.         return $response;
18.     }

```

Le DefaultController a été supprimé et la méthode indexAction() transférée dans le TaskController.

3.5 Pages d'erreurs

Des pages d'erreurs ont été créées pour les status code 403, 404 et 500 :



3.6 Php Docs

Des annotations PHP Docs ont été ajoutées à toutes les classes pour faciliter le codage des personnes travaillant sur le projet. Ils pourront bénéficier de l'affichage automatique des paramètres et retours de méthode :

```

1. // src/AppBundle/Security/Voter/TaskVoter.php
2.
3. /**
4.  * Class TaskVoter.
5.  */
6. class TaskVoter extends Voter
7. {

```

```

8.      /**
9.      * @param $attribute
10.     * @param $subject
11.     *
12.     * @return boolean
13.     */
14.     protected function supports($attribute, $subject)
15.     {
16.         return in_array($attribute, ['DELETE', 'EDIT'])
17.             && $subject instanceof \AppBundle\Entity\Task;
18.     }

```

3.7 Dépréciations

Les nombreuses dépréciations rapportées par le profiler ont toutes été corrigées :

The screenshot shows the Symfony Profiler interface. The top bar displays the URL `http://127.0.0.1:8000/tasktodo` and technical details like `Method: GET`, `HTTP Status: 200`, `IP: 127.0.0.1`, `Profiled on: Thu, 11 Apr 2019 18:25:15 +0000`, and `Token: 5dfc63`. The left sidebar contains navigation links for 'Request / Response', 'Performance', 'Validator', 'Forms', 'Exception', and 'Logs'. The main area is titled 'Log Messages' and shows a summary of log categories: 'Info. & Errors' (1), 'Deprecations' (0), 'Debug' (40), 'PHP Notices' (0), and 'Container' (604). Below this, a message states: 'Log messages generated by using features marked as deprecated. There are no log messages about deprecated features.'

4. Performances

Profil page d'accueil : <https://blackfire.io/profiles/db33e68a-1217-4bc9-b552-8bfbe2b60a83/graph>

Profil page des tâches : <https://blackfire.io/profiles/9a227b08-5c78-45fa-b540-b7a7010d67b2/graph>

	Avant modifications		Après modifications	
<u>Routes</u>	<u>Temps</u>	<u>Mémoire</u>	<u>Temps</u>	<u>Mémoire</u>
/login	304 ms	9.46 Mb	167 ms	11.57 Mb
/taskstodo	309 ms	9.56 MB	193 ms	14.2 Mb



















Comme on peut le constater, le temps de génération de la page s'est bien réduit mais au détriment de la mémoire qui a légèrement augmentée. Cependant il est préférable de privilégier la vitesse à la mémoire car les serveurs en possèdent généralement une grande quantité. L'optimisation de l'autoloader explique dans une large mesure ce gain de rapidité.

5. Tests

5.1 Tests unitaires

Les tests unitaires ont été mis en place avec Php Unit et couvrent 100% du code. Les contrôleurs ne sont pas couverts car les tests fonctionnels sont réalisés avec Behat.

C:\todoandco\src\AppBundle / (Dashboard)

	Code Coverage							
	Lines			Functions and Methods			Classes and Traits	
Total		100.00%	104 / 104		100.00%	38 / 38		100.00% 6 / 6
Controller		n/a	0 / 0		n/a	0 / 0		n/a 0 / 0
Entity		100.00%	54 / 54		100.00%	30 / 30		100.00% 2 / 2
Form		100.00%	16 / 16		100.00%	4 / 4		100.00% 2 / 2
Security		100.00%	34 / 34		100.00%	4 / 4		100.00% 2 / 2
AppBundle.php		n/a	0 / 0		n/a	0 / 0		n/a 0 / 0

Legend

Low: 0% to 50% Medium: 50% to 90% High: 90% to 100%

Generated by php-code-coverage 5.3.2 using PHP 7.2.4 with Xdebug 2.6.0 and PHPUnit 6.5.14 at Sun Apr 14 17:51:34 UTC 2019.

5.2 Tests fonctionnels

L'application est testée fonctionnellement à l'aide de Behat, un framework de BDD pour PHP. Les tests sont basés sur des scénarios écrits en langage Gherkin, qui est compréhensible par des personnes non développeurs. Les tests utilisent l'extension Mink qui communique avec un serveur Selenium en local.

Les scénarios couvrent l'intégralité des fonctionnalités de l'application avec un dénouement de réussite et un autre d'échec :

```
1. @register
2. Scenario: A anonymous user register to the website
3.   Given I'm on "/users/create" page
4.   Then the page should contain "Créer un utilisateur"
5.   Then I enter "Marco" in the "Nom d'utilisateur" field
6.   Then I enter "Mysecretecode" in the "Mot de passe" field
7.   Then I enter "Mysecretecode" in the "Tapez le mot de passe à nouveau" field
8.   Then I enter "marco@mysite.com" in the "Adresse email" field
9.   When I click on button "Ajouter"
10.  Then the page should contain "Superbe ! L'utilisateur a bien été ajouté."
11.
12. @register_fail
13. Scenario: A anonymous user tries to register to the website and miss confirmation password
14.   Given I'm on "/users/create" page
15.   Then the page should contain "Créer un utilisateur"
16.   Then I enter "Marco" in the "Nom d'utilisateur" field
17.   Then I enter "Mysecretecode" in the "Mot de passe" field
18.   Then I enter "MysecretecodeJJJ" in the "Tapez le mot de passe à nouveau" field
19.   Then I enter "marco@mysite.com" in the "Adresse email" field
20.   When I click on button "Ajouter"
21.   Then the page should contain "Les deux mots de passe doivent correspondre."
22.
23. @login
24. Scenario: A anonymous user login to the website
25.   Given I'm on "/login" page
26.   Then I enter "Marco" in the "Nom d'utilisateur :" field
27.   Then I enter "Mysecretecode" in the "Mot de passe :" field
28.   When I click on button "Se connecter"
29.   Then the page should contain "Bienvenue sur Todo List"
30.
31. @login_fail
32. Scenario: A anonymous user tries to login to the website and enter a wrong password
33.   Given I'm on "/login" page
34.   Then I enter "Marco" in the "Nom d'utilisateur :" field
35.   Then I enter "MyWrongPassword" in the "Mot de passe :" field
36.   When I click on button "Se connecter"
37.   Then the page should contain "Mot de passe incorrect !"
```

```
@users_edit_fail
Scenario: An authenticated admin user edits a user but enters different passwords
  Given I'm logged with ADMIN role
  Given I'm on "/users" page
To ○
  When I click on link "Modifier"
  Then I enter "Mysecretecode" in the "Mot de passe" field
  Then I enter "MysecretecodeLLL" in the "Tapez le mot de passe à nouveau" field
  When I click on button "Modifier"
  Then the page should contain "Les deux mots de passe doivent correspondre"
in ○
19 scenarios (19 passed)
93 steps (93 passed)
2m14.28s (14.63Mb)
PS C:\todoandco> █
```

