



AUDIT DE QUALITÉ

Application web TO DO LIST

Par *ToDo & CO*

Version : 0.1

Auteur : Mirko VENTURI

Date de la dernière mise à jour : 25 juillet 2020

Sommaire

1. Cadre du projet.....	3
1.1. Contexte	3
1.2. 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.....	7
3. Améliorations.....	7
3.1. Sécurité	7
3.2. Authentification.....	9
3.3. Les entités.....	10
3.4. Les contrôleurs	10
3.5. PHP Docs	12
3.6. Dépréciations.....	12
4. Performances.....	13
5. Tests	13
5.1 Tests unitaires	13
5.2 Tests fonctionnels	14

1. Cadre du projet

1.1. Contexte

Le cœur de métier de la startup **ToDo & Co** est une application permettant de gérer ses tâches quotidiennes. L'entreprise vient tout juste d'être montée. Elle a réussi à lever des fonds pour assurer son développement et celui de l'application web pour son premier client. Un *Minimum Viable Product* (MVP) de l'application a été réalisé grâce au Framework PHP Symfony.

1.2. Objectifs

La mission qui m'a été confiée consiste en :

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

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

2. Code initial

2.1. Données techniques

Environnement de travail :

PHP	7.3.12
Symfony	3.1.10
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* lèvent cette exception :



Cette erreur vient de l'incompatibilité de la version de PHP utilisée en environnement de développement avec des composants Symfony trop anciens. Une mise à jour avec la commande *composer update* a été réalisée afin de pouvoir continuer l'audit initial.

2.3. Profiling

Afin de tester les performances de l'application, une batterie de profilings a été mise en place avec l'outil *blackfire.io*

Profil page d'accueil : <https://blackfire.io/profiles/f1e61b9c-3202-4e5b-a938-f605be4e93cc/graph>

Profil page des tâches : <https://blackfire.io/profiles/23c5cf45-b2e9-426d-abc8-ad045b583d6d/graph>

La page d'accueil est générée en 390 ms et consomme 23.7 Mb de mémoire.

En analysant le call graph, on s'aperçoit que la méthode `file_exists` de la classe `Autoload/ClassLoader` représente 72.5 % du temps de réponse et qu'elle est appelée 257 fois.

On pourrait considérablement réduire ce coût en dumpant l'autoloader de composer grâce à la commande *composer dump-autoload --optimize*.

La page des tâches est générée en 200 ms et est appelée 767 fois.

2.4. Code Review

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

Une seule erreur a été reportée : un paramètre non utilisé dans le Controller *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

Log Messages

Info. & Errors 1 Deprecations 11 Debug 31 Silenced Errors 0

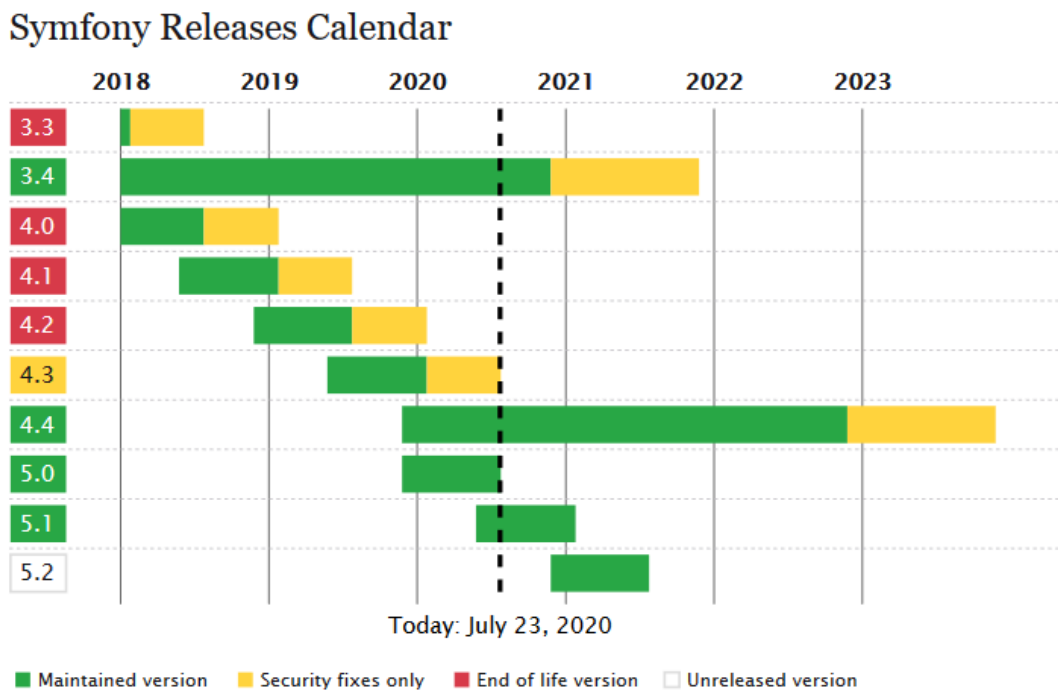
Time	Channel	Message
14.03.18	php	The Doctrine\ORM\Proxy\ProxyFactory class extends Doctrine\Common\Proxy\AbstractProxyFactory that is deprecated. The Doctrine\Common\Proxy component is deprecated, please use ocramius/proxy-manager instead. Show stack trace
14.03.18	php	Using the "Twig_Loader_Filesystem"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

Le profiler de Symfony nous informe que le code contient 9 dépréciations : Ces dépréciations sont liées 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 3.1.10.

D'après le *release* de *Sensiolabs*, cette version du framework 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.

L'utilisation de la version 7 de PHP est préconisée pour cette application. PHP 7 permet d'accroître les performances et de donner accès aux dernières fonctionnalités (source : phpbenchmarks.com).

2.7 Sécurité

Le niveau de sécurité de l'application est insuffisant.

Un utilisateur qui a créé un compte peut modifier n'importe quel autre 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 cette option dans les formulaires.

2.8 Analyse

Le *DefaultController* ne sert qu'à afficher la page d'accueil. Le transfert de cette fonctionnalité dans un autre *controller* est préconisé.

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

Cliquer sur « consulter la liste des tâches terminées » n'affiche rien. Il faudrait replacer les tâches dans leurs catégories.

De manière générale, le code n'est pas documenté. Une description des classes et méthodes avec des annotations *phpdoc* est préconisée.

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 du framework.

Dans la classe *TaskType*, il manque les labels pour les champs *title* et *content* : les labels de ces champs apparaissent en anglais.

3. Améliorations

3.1. Sécurité

L'accès aux parties sensibles du site a été restreint aux utilisateurs authentifiés. Seules la page de login est accessible 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.

Pour renforcer le niveau de sécurité de l'application, seulement les utilisateurs ayant un `ROLE_ADMIN` peuvent créer d'autres comptes utilisateur.

Sur le même principe, l'administration des comptes (modification, suppression) est restreinte aux utilisateurs possédant un `ROLE_ADMIN`.

```

25 // app/config/security.yml
26     access_control:
27         - { path: ^/login, roles: IS_AUTHENTICATED_ANONYMOUSLY }
28         - { path: ^/users/create, roles: IS_AUTHENTICATED_ANONYMOUSLY }
29         - { path: ^/users, roles: 'ROLE_ADMIN' }
30         - { path: ^/, roles: ['ROLE_ADMIN', 'ROLE_USER'] }
31

```

Afin de compléter le système d'autorisation, un *UserVoter* a été mis en place. Seulement l'utilisateur ayant un `ROLE_ADMIN` peut gérer le CRUD des utilisateurs :

```

// src/AppBundle/Security/UserVoter.php
26 protected function voteOnAttribute($attribute, $subject, TokenInterface $token)
27 {
28     $user = $token->getUser();
29
30     if (!$user instanceof UserInterface) {
31         // the user must be logged in; if not, deny access
32         return false;
33     }
34
35     $user = $subject;
36
37     if ($attribute === 'ADD' || $attribute === 'GET' || $attribute === 'EDIT' || $attribute === 'REMOVE') {
38         if ($user->isAdmin()) {
39             return true;
40         }
41         return false;
42     }
43     // @codeCoverageIgnoreStart
44 }
45 // @codeCoverageIgnoreEnd

```

Concernant les tâches dont l'auteur est anonyme, la méthode *deleteTaskAction()* du *TaskController* a été implémentée pour que les tâches en question ne puissent être supprimées que par un utilisateur avec `ROLE_ADMIN` :

```

// src/AppBundle/Controller/TaskController.php
if ($user->isAdmin() && $task->getUser() === null) {
    // @codeCoverageIgnoreStart
    $em->remove($task);
    $em->flush();

    $this->addFlash('success', 'La tâche a bien été supprimée.');
```


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

```
// src/AppBundle/Form/UserType.php

41     /**
42      * @param OptionsResolver $resolver
43      */
44     public function configureOptions(OptionsResolver $resolver)
45     {
46         $resolver->setDefaults([
47             'data_class'      => User::class,
48             'csrf_protection' => true,
49         ]);
50     }
```

3.2 Authentification

Le fichier *security.yml* a été vérifié et implémenté. L'encodeur utilisé est *bcrypt*, le provider est la classe *User*.

```
// app/config/security.yml

security:
    encoders:
        AppBundle\Entity\User: bcrypt

    providers:
        doctrine:
            entity:
                class: AppBundle\User
                property: username

    firewalls:
        dev:
            pattern: ^/(_(profiler|wdt)|css|images|js)/
            security: false

    main:
        anonymous: ~
        pattern: ^/
        form_login:
            login_path: login
            check_path: login_check
            always_use_default_target_path: true
            default_target_path: /
        logout: ~
```

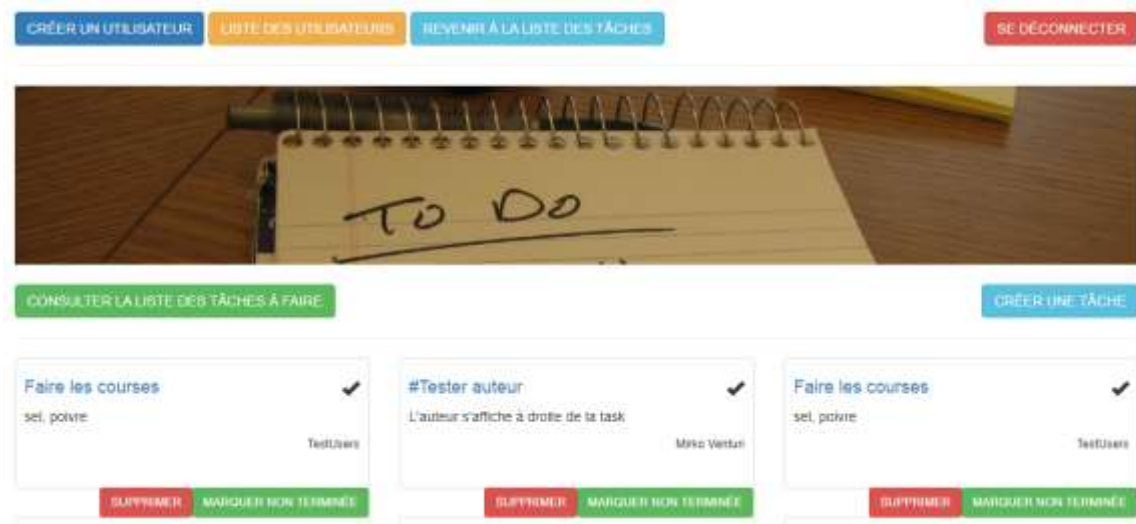
3.3 Les entités

Une relation a été créée entre la tâche et son utilisateur.

```
// src/AppBundle/Entity/Task.php
```

```
38
39
40     /**
41      * @ORM\ManyToOne(targetEntity="AppBundle\Entity\User", inversedBy="tasks")
42      * @ORM\JoinColumn(nullable=true)
43      */
44     private $user;
```

Dans les *views*, le nom de l'auteur est désormais affiché sur les tâches :



3.4 Les contrôleurs

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

```
// src/AppBundle/Controller/UserController.php
```

```
72     /**
73      * @param Request $request
74      * @param EntityManagerInterface $em
75      *
76      * @Route("/users/{id}/edit", name="user_edit")
77      */
78     public function editAction(User $user, Request $request, EntityManagerInterface $em = null)
79     {
80         $this->denyAccessUnlessGranted('EDIT', $this->getUser());
81
82         $form = $this->createForm(UserType::class, $user);
83
84         $form->handleRequest($request);
85
86         if ($form->isSubmitted() && $form->isValid()) {
87             $em = $this->getDoctrine()->getManager();
88
89             // ...
```

Les contrôleurs étendent désormais la classe *AbstractController* :

```
// src/AppBundle/Controller/UserController.php

12 /**
13  * Class UserController
14  */
15 class UserController extends AbstractController
16 {
17     /**
18      * @Route("/users", name="user_list")
19      *
20      * @return Response
21      */
22     public function listAction()
23     {
24         $this->denyAccessUnlessGranted('GET', $this->getUser());
25
26         $response = $this->render('user/list.html.twig', ['users' => $this->getDc
27
28         $response->setSharedMaxAge(200);
29
30         $response->headers->addCacheControlDirective('must-revalidate', true);
31
32         return $response;
33     }
}
```

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

```
// src/AppBundle/Controller/TaskController.php

17 class TaskController extends AbstractController
18 {
19     // @codeCoverageIgnoreStart
20     /**
21      * @Route("/", name="homepage")
22      *
23      * @return Response
24      */
25     public function indexAction()
26     {
27         $response = $this->render('task/index.html.twig');
28         $response->setSharedMaxAge(200);
29         $response->headers->addCacheControlDirective('must-revalidate', true);
30
31         return $response;
32     }
33     // @codeCoverageIgnoreEnd
}
```

3.5 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 :

```
// src/AppBundle/Controller/SecurityController.php
```

```
10 /**
11  *
12  * Class SecurityController
13  */
14 class SecurityController extends AbstractController
15 {
16     /**
17      * @param AuthenticationUtils $authenticationUtils
18      * @Route("/login", name="login")
19      *
20      * @return Response
21      */
22     public function loginAction()
23     {
```

3.6 Dépréciations

La version de base de Symfony a été mise à jour : 3.1.10 => 3.4.43.

Dans le cas où une nouvelle mise à jour vers la version 4 de Symfony est envisagée, quelques dépréciations seront à corriger. Néanmoins, la plupart des dépréciations (ex. classe *Kernel.php*) disparaîtront automatiquement.

Log Messages		
Info & Errors 1 Deprecations 9 Debug 26 PHP Notices 3 Container 384		
Log messages generated by using features marked as deprecated.		
Time	Channel	Message
2016-11-17 10:00:17	php	User Deprecated: Symfony\Component\HttpFoundation\Kernel::loadClassCache() is deprecated since Symfony 3.3, to be removed in 4.0. Show context Show trace
2016-11-17 10:00:17	php	User Deprecated: Symfony\Component\HttpFoundation\Kernel::loadClassCache() is deprecated since Symfony 3.3, to be removed in 4.0. Show context Show trace
2016-11-17 10:00:17	php	User Deprecated: Creating Doctrine\ORM\Mapping\UnderscoreNamingStrategy without making it number aware is deprecated and will be removed in Doctrine ORM 3.0. Show context Show trace
2016-11-17 10:00:18	-	Duplicate key "web_profiler" detected whilst parsing YAML. Short handling of duplicate mapping keys in YAML is deprecated since Symfony 3.2 and will throw Symfony\Component\Yaml\Exception\ParseException in 4.0 in C:\wamp64\www\PS-ToDoList\app\config\config_dev.yml in "C:\wamp64\www\PS-ToDoList\app\config\config_dev.yml" on line 52. Show context Show trace
2016-11-17 10:00:18	-	The "framework.trusted_proxies" configuration key has been deprecated in Symfony 3.3. Use the Request::setTrustedProxies() method in your front controller instead. Show context Show trace
2016-11-17 10:00:18	-	Not setting "logout_on_user_change" to true on firewall "main" is deprecated as of 3.4, it will always be true in 4.0. Show context Show trace
2016-11-17 10:00:19	- (8 times)	Autowiring types are deprecated since Symfony 3.3 and will be removed in 4.0. Use aliases instead for "Psr\Log\LoggerInterface". Show context Show trace
2016-11-17 10:00:19	-	Symfony\Component\HttpFoundation\Kernel::DependencyInjection\Extension::addClassesToCompile() is deprecated since Symfony 3.3, to be removed in 4.0. Show context Show trace

4. Performances

Profil page d'accueil : <https://blackfire.io/profiles/e4c91c53-afad-4064-9546-2b3047e629fd/graph>

Profil page des tâches : <https://blackfire.io/profiles/535b6b7a-582e-4fa7-9390-ff9275b17add/graph>

















	Avant modifications		Après modifications	
<u>Routes</u>	<u>Temps</u>	<u>Mémoire</u>	<u>Temps</u>	<u>Mémoire</u>
/login	390 ms	23.7 Mb	167 ms	31.7 Mb
/taskstodo	200 ms	33.1 MB	146 ms	33.3 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 PHPUnit et couvrent 99% du code.

		Code Coverage				
		Lines		Functions and Methods		
Total		99.39%	163 / 164		97.92%	47 / 48
 Controller		100.00%	75 / 75		100.00%	11 / 11
 Entity		98.28%	57 / 58		96.77%	30 / 31
 Form		100.00%	18 / 18		100.00%	4 / 4
 Repository		n/a	0 / 0		n/a	0 / 0
 Security		100.00%	13 / 13		100.00%	2 / 2
 AppBundle.php		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 7.2.18](#) with [Xdebug 2.7.2](#) and [PHPUnit 6.5.14](#) at Sat Jul 25 14:35:30 UTC 2020.

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 scénarii 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 scénarii couvrent l'intégralité des fonctionnalités de l'application avec un dénouement de réussite et un autre d'échec (ci-dessous, la batterie de tests pour l'ADMIN) :

```
1  Feature:
2      In order to manage users
3      As an authenticated admin user
4      I want to access to different pages
5
6      @users_list
7      Scenario: An authenticated admin user can access to users page
8          Given I'm logged with ROLE_ADMIN role
9          Given I'm on "/" page
10         When I click on link "Liste des utilisateurs"
11         Then the page should contain "LISTE DES UTILISATEURS"
12
13     @users_list_fail
14     Scenario: An authenticated user wants to access to users page with as a user
15         Given I'm logged with ROLE_USER role
16         Given I'm on "/users" page
17         Then the page should contain "Access Denied"
18
19     @users_edit
20     Scenario: An authenticated admin user edit a user
21         Given I'm logged with ROLE_ADMIN role
22         Given I'm on "/users" page
23         When I click on link "Modifier"
24         Then I enter "admin" in the "Mot de passe" field
25         Then I enter "admin" in the "Tapez le mot de passe à nouveau" field
26         When I click on button "Modifier"
27         Then the page should contain "Superbe ! L'utilisateur a bien été modifié"
28
29     @users_edit_fail
30     Scenario: An authenticated admin user edits a user but enters different passwords
31         Given I'm logged with ROLE_ADMIN role
32         Given I'm on "/users" page
33         When I click on link "Modifier"
34         Then I enter "Mon mot de passe" in the "Mot de passe" field
35         Then I enter "Mon mot de passeLLL" in the "Tapez le mot de passe à nouveau" field
36         When I click on button "Modifier"
37         Then the page should contain "Les deux mots de passe doivent correspondre"
```

Comme on peut le lire dans les résultats affichés ci-dessous en ligne de commande, tous les tests fonctionnels de l'application ont réussi.



```
MINGW64:/c:/wamp64/www/P8-ToDoList

  When I click on button "Supprimer"
  Then the page should contain "Supprimer"

@logout
Scenario: An authenticated user logs out

  Given I'm logged with ROLE_USER role
  Given I'm on "/tasks" page
  When I click on link "Se déconnecter"
  Then the page should contain "Nom d'utilisateur"

15 scenari (15 superati)
69 passaggi (69 superati)
1m13.41s (12.89Mb)

Utente@Mirko MINGW64 /c:/wamp64/www/P8-ToDoList (master)
$ |
```