

# Documentation technique

Samir Mehal

ToDo & Co

Openclassrooms - Projet 8

France

11 février 2024

## Table des matières

<b>1 - Résumé du projet</b>	<b>3</b>
<b>2 - Création de l'Authentification</b>	<b>3</b>
2 -.1 Entité . . . . .	3
2 -.2 Controller . . . . .	3
<b>3 - Pour modifier le projet</b>	<b>4</b>
<b>4 - Se connecter</b>	<b>5</b>

## 1 - Résumé du projet

Le projet ToDo & Co est un site web qui permet de créer et de modifier des utilisateurs ainsi que de créer et de modifier des tâches. Le site web possède un système d'authentification que nous allons voir dans ce document.

## 2 - Création de l'Authentification

L'authentification permet à un utilisateur de s'authentifier, c'est à dire d'être reconnu comme étant connecté à son propre compte, avec ses propres informations. Avant de pouvoir créer l'authentification sur Symfony, d'après la documentation officielle de Symfony, il nous faut installer le 'SecurityBundle' avec la commande 'composer require symfony/security-bundle'.

### 2 -.1 Entité

Une entité est un fichier qui permet de relier le site web à une base de données (ici MySQL). Ce fichier est composé de propriétés et de méthodes qui déterminent les données qui seront dans la table; un fichier par table.

Nous commençons par créer l'entité User qui nous permettra de définir la table User et les données qui iront dedans. Pour cela, nous utilisons la commande 'php bin/console make :user'. Une série de question nous est posée et nous devons y répondre. Le nom de la classe, ici User; si nous souhaitons stocker les données dans la base de données, ici oui; d'entrer le nom de la propriété qui serait unique, ici le username; et si notre application a besoin de hasher le mot de passe, ici la réponse est oui.

Une fois répondu, le fichier User.php apparaît dans le dossier src puis dans le dossier Entité. Le fichier User.php est déjà pré-rempli avec les informations concernant l'id, le username et le mot de passe. Le rôle est aussi présent mais nous le modifierons car au début il garanti un 'role\_user' à tout le monde, ce que nous ne souhaitons pas dans notre projet.

À présent, nous modifions notre fichier User.php en ajoutant un 'email' (ajout de la propriété et des méthodes getter et setter) et en modifiant le 'roles'.

Une fois le fichier User.php modifié, nous pouvons enregistrer les modifications dans la base de données avec la commande 'php bin/console make :migration' suivie de la commande 'php bin/console doctrine :migrations :migrate'.

Avec la commande 'php bin/console make :user' que nous avons utilisé au début, nous avons également modifié automatiquement le fichier 'security.yaml' qui se trouve dans le dossier config puis dans le dossier package.

### 2 -.2 Controller

Un contrôleur est un fichier qui permet d'échanger avec l'entité (la base de données) pour injecter ces données dans le template.

Une fois que l'entité User est créée, il nous faut créer le contrôleur qui générera le formulaire d'authentification. Pour cela, nous utilisons la commande

'php bin/console make :controller Login', ce qui crée le contrôleur 'LoginController.php' que nous renommons 'SecurityController.php'. À l'intérieur de ce fichier, nous modifions le nom de la class en 'SecurityController' et le nom de la route en 'login'. Puis, nous ajoutons l'AuthenticationUtils (en argument de la fonction et nous n'oublions pas le use qui est nécessaire pour le faire fonctionner). D'après la documentation officielle de Symfony, nous devons ajouter l'erreur lié à l'authentification ainsi que le dernier username entré par l'utilisateur :

```
$error = $authenticationUtils->getLastAuthenticationError();  
$lastUsername = $authenticationUtils->getLastUsername();
```

Il ne faut pas oublier de transmettre ces deux données au template que nous appellerons 'security/login.html.twig'. Enfin, nous pouvons générer le template 'login.html.twig' qui se trouvera dans le dossier 'security', avec le formulaire à l'intérieur de celui-ci.

Ensuite, nous allons générer le contrôleur qui nous permettra de créer et de modifier les utilisateurs avec la commande 'php bin/console make :controller User'. Puis, nous créons trois routes, une pour lister les actions, une pour créer un utilisateur et une pour modifier un utilisateur.

À présent, nous vérifions que le fichier 'security.yaml', dans le dossier packages qui se trouve être dans le dossier config, soit bien configuré. Nous vérifions que le login\_path et le check\_path soient bien identiques au nom de la route (ici login). Nous ajoutons aussi le logout avec le bon chemin (path en anglais), le nom du chemin doit être identique au nom de la route du contrôleur correspondant au logout (ici logout).

### 3 - Pour modifier le projet

L'authentification fonctionne et ne nécessite pas de modifications mais nous devons toujours vérifier que nos fichiers soient à jour en fonction de la documentation officielle de Symfony. Le fichier 'security.yaml' (dossier config-> dossier packages) ne devrait pas nécessiter de modification. Le fichier 'User.php' (dossier src -> dossier Entity) peut être modifiable si nous voulons ajouter des données à notre table user. Nous pouvons y ajouter des propriétés et des méthodes (getter et setter).

La partie authentification ne nécessite pas d'être modifié à moins d'avoir une modification de la documentation officielle. Nous ne devons donc pas modifier le contrôleur 'SecurityController.php' (dossier src -> dossier Controller). Par contre, nous pouvons modifier le contrôleur 'UserController.php' si à l'avenir nous voulons ajouter d'autres fonctionnalités.

Pour ajouter des modifications au projet, nous devons bien suivre la documentation officielle. Nous devons suivre l'architecture MVC où l'entité est le modèle qui fait le lien avec la base de données, le template est la vue qui génère la page et le contrôleur récupère les données du modèle et les envoie dans le template.

## 4 - Se connecter

Pour se connecter, nous devons utiliser l'authentification que nous avons créé. Pour cela, nous lançons symfony avec la commande terminale 'symfony server :start', puis nous nous rendons sur la page 'http ://localhost :8000/login'. Nous trouverons alors le formulaire de connexion auquel nous rentrerons l'identifiant et le mot de passe correspondant à notre compte (des fixtures sont préchargées). Si l'identifiant et si le mot de passe sont correctes alors la connexion aura réussie sinon la connexion aura échoué et il faudra recommencer.