ToDo_Co

Guide de l'authentification

Sommaire

1. Configuration globale

La sécurité de l'application est configurée dans le fichier app/config/security.yml.

L'ensemble des fichiers qui utilisés pour l'authentification sont présentés ci-dessous :

- config/packages/security.yaml : Configuration du processus d'authentification
- src/Entity/User.php : Entité utilisateur
- src/Controller/SecurityController.php : Controleur connexion/déconnexion
- src/Security/UsersAuthenticator.php : Méthode du processus d'authentification
- templates/security/login.html.twig : Page du formulaire de connexion

2. L’entité User

L'utilisateur est représenté par la classe App\Entity\User. Cette classe implémente toutes les méthodes définies dans l’interface UserInterface.

Une contrainte d'unicité est appliquée à l'attribut email afin de ne pas avoir de doublon.

```
#[ORM\Entity(repositoryClass: UserRepository::class)]

#[UniqueEntity(fields: ['email'], message: 'There is already an account with this email')]

class User implements UserInterface, PasswordAuthenticatedUserInterface

{
```

3. Les providers:

Les utilisateurs sont enregistrés en BDD par doctrine. Doctrine charge l’entité Utilisateur à l’aide de la propriété “email” qui sert à authentifier l'utilisateur.

```
# app/config/security.yml
    providers:

        app_user_provider:

            entity:

                class: App\Entity\User

                property: email
```

4. Les encodeurs:

L’encodeur détermine quel est l’algorithme qui va être utilisé pour le hashage des mots de passe. Nous avons conservé l’encryptage automatique proposé par Symfony.

```
Password_hashers:

      Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface:

            algorithm: auto
```

5. Les firewalls

Un pare-feu est désigné afin d'empêcher un utilisateur non authentifié d'accéder à certaines parties du site. Pour s'authentifier, on utilise un formulaire accessible à la route login :

```
# app/config/security.yml
    firewalls:

        dev:

            pattern: ^/(_(profiler|wdt)|css|images|js)/

            security: false

        main:

            lazy: true

            provider: app_user_provider

            custom_authenticator: App\Security\UsersAuthenticator

            logout:

                path: app_logout
```

On permet à l'utilisateur anonyme d'accéder à cette route grâce au paramètre access_control.

```
access_control:

        # - { path: ^/admin, roles: ROLE_ADMIN }

        # - { path: ^/profile, roles: ROLE_USER }
```

L'accès au formulaire d'inscription d'un utilisateur est laissé libre.

Par contre, l'accès à la gestion des utilisateurs est réservée aux membres possédant le rôle : ROLE_ADMIN.

Cette permission est réalisée grâce à l'utilisation des voters dans les contrôleurs, l'accès aux fichiers voters: App\Security\Voter