Documentation technique

Introduction

L'authentification est une fonctionnalité essentielle pour notre application. Elle permet de sécuriser l'accès à l'application et de gérer les droits d'accès des utilisateurs.

Packages utilisées

Pour l'authentification, nous avons utilisé la librairie `*symfony/security-bundle*`. Cette librairie permet de gérer l'authentification et les droits d'accès des utilisateurs.

Fonctionnement

Authentification

L'authentification est gérée par le fichier `*security.yaml*` situé dans le dossier `*config/packages*`. Ce fichier contient la configuration de l'authentification.

```yaml
security:
    # https://symfony.com/doc/current/security.html#registering-the-user-hashing-passwords
    password_hashers:
        Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface: 'auto'
    # https://symfony.com/doc/current/security.html#loading-the-user-the-user-provider
    providers:
        # used to reload user from session & other features (e.g. switch_user)
        app_user_provider:
            entity:
                class: App\Entity\User
                property: username
    firewalls:
        dev:
            pattern: ^/(_(profiler|wdt)|css|images|js)/
            security: false
        main:
            lazy: true
            provider: app_user_provider

            form_login:
                login_path: login
                check_path: login
                always_use_default_target_path:  true
                default_target_path:  /

            logout:
                path: logout

            # activate different ways to authenticate
            # https://symfony.com/doc/current/security.html#the-firewall

            # https://symfony.com/doc/current/security/impersonating_user.html
            # switch_user: true

    # Easy way to control access for large sections of your site
    # Note: Only the *first* access control that matches will be used
    access_control:
        # - { path: ^/admin, roles: ROLE_ADMIN }
        # - { path: ^/profile, roles: ROLE_USER }
        - { path: ^/login, roles: PUBLIC_ACCESS }
        - { path: ^/users, roles: ROLE_ADMIN }
        - { path: ^/, roles: ROLE_USER }
```

Ce fichier contient plusieurs sections :

- `*password_hashers*` : Cette section permet de définir le type de hachage utilisé pour les mots de passe des utilisateurs. Nous avons choisi d'utiliser le hachage `auto` qui permet de choisir le type de hashage en fonction de la configuration du serveur.

- `*providers*` : Cette section permet de définir le fournisseur d'utilisateurs. Nous avons choisi d'utiliser le fournisseur `entity` qui permet de récupérer les utilisateurs depuis la base de données.

- `*firewalls*` : Cette section permet de définir les pares-feux de l'application. Nous avons choisi d'utiliser deux pare-feu : `dev` et `main`. Le pare-feu `dev` est utilisé pour les environnements de développement. Le pare-feu `main` est utilisé pour les environnements de production. Le pare-feu `main` est le pare-feu principal de l'application. Il permet de gérer l'authentification des utilisateurs.

- `*access_control*` : Cette section permet de définir les contrôles d'accès de l'application. Nous avons choisi de définir plusieurs contrôles d'accès. Le premier contrôle d'accès permet d'accéder à la page de connexion. Le deuxième contrôle d'accès permet d'accéder à la page de gestion des utilisateurs. Le troisième contrôle d'accès permet d'accéder à toutes les autres pages de l'application.

Gestion des droit d’accès

La gestion des droits d'accès est gérée par l'annotation `*#[IsGranted]*` située dans les contrôleurs. Cette annotation permet de définir les droits d'accès nécessaires pour accéder à une fonctionnalité.
Les permissions peuvent être également définis dans le fichier `*security.yaml*` comme vu précédemment dans la section `*access_control*`

```php
#[Route('/users', name: 'users')]
#[IsGranted('ROLE_ADMIN')]
public function index(UserRepository $userRepository): Response
{
    // ...
}
```

Comment modifier ou ajouter utilisateur ?

Pour modifier la création d'un utilisateur, il faut modifier les fonctions `*createAction*` ou `*editAction*` situées dans le contrôleur `*UserController.php*` disponible dans le dossier `*src/Controller*`.

Celles-ci contiennent un formulaire qui permet de créer un utilisateur.

Vous pouvez modifier ce formulaire en modifiant le fichier `*UserType.php*` situé dans le dossier `*src/Form*`.

Comment s'opère l'authentification ?

L'authentification est gérée par le fichier `*SecurityController.php*` situé dans le dossier `*src/Controller*`.

Ce fichier contient deux fonctions :

- `*loginAction*` : Cette fonction permet d'afficher la page de connexion.
- `*logoutCheck*` : Cette fonction permet de déconnecter l'utilisateur.

Où sont stockés les utilisateurs ?

Les utilisateurs sont stockés dans la base de données. Pour accéder à la base de données, il faut utiliser le fichier `*User.php*` situé dans le dossier `*src/Entity*`.

Ce fichier contient les informations relatives aux utilisateurs.