

Guide sur l'Authentification

La configuration de la sécurité pour l'authentification est gérée dans le fichier **config/packages/security.yaml**. Pour des informations plus détaillées sur ce fichier, vous pouvez consulter la documentation officielle de Symfony.

1. L'entité User

L'utilisateur correspond à l'entité User, qui se situe dans `App\Entity\User`. Cette classe doit implémenter les interfaces `UserInterface` et `PasswordAuthenticatedUserInterface`, ainsi que leurs différentes méthodes définies dans celle-ci.

```
# App/Entity/User.php
namespace App\Entity;

#[ORM\Table('user')]
class User implements UserInterface, PasswordAuthenticatedUserInterface
{
```

2. Providers

Un provider indique où se trouvent les informations nécessaires pour l'identification de l'utilisateur. Dans notre cas, on utilise l'ORM Doctrine pour récupérer l'utilisateur via l'entité `User`, en utilisant la propriété `username`.

```
# config/packages/security.yaml
providers:
    doctrine:
        entity:
            class: App\Entity\User
            property: username
```

3. Encoders

Un encodeur de mot de passe définit l'algorithme utilisé pour hasher les mots de passe. Nous utilisons l'algorithme `auto` pour hacher les mots de passe

depuis l'interface `PasswordAuthenticatedUserInterface` implémenté dans l'entité `User` :

```
# config/packages/security.yaml
password_hashers:
    Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface:
        App\Entity\User: "auto"
```

4. Firewalls

Le pare-feu va permettre d'autoriser l'accès à certaines routes, en fonction des droits de l'utilisateur.

- **Configuration dev :**

Ce pare-feu gère l'autorisation d'accès à des routes spécifiques sans nécessiter d'authentification, qui s'appliquent aux outils de développements comme le **profilier de Symfony** ou des **ressources statiques** (fichiers css, JavaScript ou les images). La sécurité est désactivée pour ces routes, ce qui les rend accessibles à tous les utilisateurs.

- **Configuration main :**

Celui-ci gère la sécurité pour le reste de l'application. L'authentification est gérée par un authenticateur personnalisé `App\Security\LoginFormAuthenticator`, qui est également défini comme **point d'entrée** pour rediriger les utilisateurs non authentifiés vers le formulaire de connexion.

```
# config/packages/security.yaml
firewalls:
    dev:
        pattern: ^/(_(profiler|wdt)|css|images|js)/
        security: false

    main:
        pattern: ^/
        lazy: true
        custom_authenticators:
            - App\Security\LoginFormAuthenticator
        logout:
```

```
path: logout
entry_point: App\Security\LoginFormAuthenticator
```

5. Acces Control

Le contrôle d'accès permet de restreindre l'accès à certaines routes de l'application en fonction des rôles des utilisateurs.

- **Route `/login` :**

L'accès à cette route est autorisé à tous les utilisateurs, y compris ceux non authentifiés, grâce au rôle `PUBLIC_ACCESS`.

- **Route `/users` :**

Seuls les utilisateurs ayant le rôle `ROLE_ADMIN` peuvent accéder à cette route.

- **Route `/` (racine) :**

L'accès à toutes les autres routes de l'application est limité aux utilisateurs authentifiés ayant le rôle `ROLE_USER`.

```
# config/packages/security.yaml
access_control:
    - { path: ^/login, roles: PUBLIC_ACCESS }
    - { path: ^/users, roles: ROLE_ADMIN }
    - { path: ^/, roles: ROLE_USER }
```

6. Role Hierarchy

Définit une hiérarchie entre les rôles :

- `ROLE_ADMIN` hérite de `ROLE_USER`.

Un administrateur dispose ainsi des droits d'un utilisateur standard, en plus de ses permissions spécifiques.

```
# config/packages/security.yaml
role_hierarchy:
    ROLE_ADMIN: ROLE_USER
```

7. Stockage des Utilisateurs

Les utilisateurs enregistrés sont stockés dans une base de données gérées par l'ORM Doctrine.

Dans la base de donnée, on y retrouve la table `user` contenant les différentes propriétés de l'entité `User`, telles que :

- L'identifiant unique (`id`)
- Le nom d'utilisateur (`username`)
- l'adresse email (`email`)
- Le mot de passe chiffré (`password`)
- Le rôle de l'utilisateur (`roles`), par défaut "ROLE_USER"

Cette documentation expliquant l'implémentation de l'authentification sous Symfony 6.4 se destine aux prochains développeurs qui rejoindront l'équipe dans un futur proche. Pour toute question, n'hésitez pas à consulter la documentation officielle de Dymfony ou Doctrine ou à demander de l'aide à un membre de l'équipe.