Authentification

Introduction

Le système d'authentification se trouve dans le fichier "security.yaml" et est basé sur l'entité "User".
La documentation officielle est disponible ici : https://symfony.com/doc/current/security.html

Providers

En plus de la class "User", il y a également besoin d'un "User provider" : une classe qui peut aider avec le rechargement des données de l'utilisateur à partir de la session et certaines fonctionnalités facultatives, comme "remember me".

Exemple :

```yaml
# config/packages/security.yaml
encoders:
  App\Entity\User:
    algorithm: auto
```

Password Hasher

Utilise le hachage de mot de passe natif, qui sélectionne et migre automatiquement le meilleur algorithme de hachage possible (qui est actuellement "bcrypt").

Exemple :

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

Firewall

La partie FireWall est la section la plus importante. Le firewall est le système d'authentification : le firewall définit quelles parties de votre application sont sécurisées et comment vos utilisateurs pourront s'authentifier (par exemple, formulaire de connexion, jeton API, etc.).

Exemple :

```yaml
# config/packages/security.yaml
firewalls:

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

    main:
        lazy: true
        provider: app_user_provider
        pattern: ^/
        form_login:
            login_path: login
            check_path: login
            csrf_parameter: _csrf_token
            csrf_token_id: authenticate
        logout:
            path: logout
            target: homepage

    secured_area:
        form_login:
            login_path: login
            check_path: login
            enable_csrf: true
```

Role Hierarchy

Au lieu d'attribuer plusieurs rôles à chaque utilisateur, on peut définir des règles d'héritage des rôles en créant une hiérarchie des rôles.

Exemple :

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

Access Control

Le contrôle d'accès affine l'autorisation nécessaire pour accéder à certaines routes, par exemple certaines routes peuvent être rendues accessibles à n'importe quel utilisateur ou uniquement aux utilisateurs administrateurs.

Exemple :

```yaml
# config/packages/security.yaml
access_control:
    - { path: ^/tasks, roles: ROLE_USER }
```