

Authentication

Sommaire

1 Security.....	1
2 Providers.....	1
3 Authenticator.....	2
4 Acces Control.....	2
5 Entité User.....	3

1 Security

Dans le fichier « security.yaml », la configuration de toute la partie sécurité de l'authentification est faite.

Pour activer le nouveau système de sécurité de symfony (à partir de symfony 5.1) :

```
1 | # config/packages/security.yaml
2 | security:
3 |     enable_authenticator_manager: true
```

<https://symfony.com/blog/new-in-symfony-5-1-updated-security-system>

Pour définir le hashage (encodage) du mot de passe :

```
5 | password_hashers:
6 |     Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface: 'auto'
7 |     App\Entity\User:
8 |         algorithm: auto
```

<https://symfony.com/doc/current/security.html>

Attention, l'entité User demandé à la ligne 7 doit respecter les règles définit à la partie 5 : **Entité User**.

2 Providers

Le providers charge l'utilisateur. L'entité User (voir partie 5) est demandé, le champs property définit le champs utilisé dans l'entité User, ici « username » :

```
10 | # https://symfony.com/doc/current/security.html#loading-the-user-the-user-provider
11 | providers:
12 |     # used to reload user from session & other features (e.g. switch_user)
13 |     app_user_provider:
14 |         entity:
15 |             class: App\Entity\User
16 |             property: username
```

https://symfony.com/doc/current/security/user_providers.html

3 Authenticator

La classe Authenticator permet l'authentification, elle hérite de la classe « AbstractLoginFormAuthenticator » de Symfony, cette classe se chargera du traitement des données entrées par l'utilisateur voulant s'authentifier.

```
21     main:
22         lazy: true
23         provider: app_user_provider
24         custom_authenticator: App\Security\Authenticator
```

https://symfony.com/doc/current/security/custom_authenticator.html

4 Access Control

L'access control impose des autorisations pour charger une route, « path » étant la route, « roles », l'autorisation nécessaire.

```
38     access_control:
39         # - { path: ^/admin, roles: ROLE_ADMIN }
40         # - { path: ^/profile, roles: ROLE_USER }
41         - { path: ^/login, roles: IS_AUTHENTICATED_ANONYMOUSLY }
42         - { path: ^/user, roles: ROLE_ADMIN }
43         - { path: ^/, roles: ROLE_USER }
44
```

https://symfony.com/doc/current/security/access_control.html

5 Entité User

L'entité User, utilisé pour l'authentification doit implémenter les interfaces de Symfony « `UserInterface` » et « `PasswordAuthenticatedUserInterface` », les champs « `id` », « `username` », « `roles` » ainsi que « `password` » sont nécessaires. Les annotations présentes ci-dessous permettent l'enregistrement en base de données via l'orm doctrine :

```
1  <?php
2  // src/Entity/User.php
3  namespace App\Entity;
11
12  /**
13   * @ORM\Entity(repositoryClass=UserRepository::class)
14   */
15  class User implements UserInterface, PasswordAuthenticatedUserInterface
16  {
17      /**
18       * @ORM\Id
19       * @ORM\GeneratedValue
20       * @ORM\Column(type="integer")
21       */
22      private $id;
23
24      /**
25       * @ORM\Column(type="string", length=180, unique=true, nullable=true)
26       */
27      private $username;
28
29      /**
30       * @ORM\Column(type="json")
31       */
32      private $roles = [];
33
34      /**
35       * @var string The hashed password
36       * @ORM\Column(type="string")
37       */
38      private $password;
```

<https://symfony.com/doc/current/security.html#the-user>