

Authentication Guide

La sécurité concernant l'authentification est configurée dans le fichier

`config/packages/security.yaml`

Vous trouverez plus d'informations concernant ce fichier et ses différentes parties dans la [documentation officielle de Symfony](#).

L'entité User

Avant toute chose, il est nécessaire d'avoir défini une entité qui représente l'utilisateur connecté. Cette classe doit implémenter l'interface `UserInterface` et donc implémenter les différentes méthodes définies dans celle-ci.

Cette classe a déjà été implémentée et se situe dans le fichier

`src/Entity/User.php`.

Les Providers

Un provider va nous permettre d'indiquer où se situent les informations que l'on souhaite utiliser pour authentifier l'utilisateur, pour notre cas, on indique qu'on récupérera les utilisateurs via Doctrine grâce à l'entité `User` dont la propriété `email` sera utilisée pour s'authentifier sur le site.

```
providers:
    # used to reload user from session & other features (e.g. switch user)
    app_user_provider:
        entity:
            class: App\Entity\User
            property: email
```

Les encoders

Un encoder va simplement nous permettre de déterminer quel est l'algorithme que l'on souhaite utiliser lors de l'encodage d'une certaine information dans une certaine entité.

On utilisera l'algorithme `auto` lorsque quelque chose doit être encodé dans l'entité

`App\Entity\User` via `UserPasswordEncoderInterface`,

```
security:
  encoders:
    App\Entity\User:
      algorithm: auto
```

Les Firewalls

Un firewall va définir comment nos utilisateurs vont être authentifiés sur certaines parties du site.

Le firewall `dev` ne concerne que le développement ainsi que le profiler et ne devra à priori pas être modifié.

Le firewall `main` englobe tout le site à partir de la racine, l'accès y est autorisé en anonyme c-à-d sans être authentifié, on y indique que c'est le provider "doctrine" qui sera utilisé. Afin de s'authentifier, on définit un formulaire de connexion via `form_login`: où sont indiqués le nom des routes correspondant à ce formulaire, la route de vérification du login ainsi que la route vers laquelle l'utilisateur devra être redirigé par défaut après son authentification.

```
firewalls:
  dev:
    pattern: ^/(_(profiler|wdt)|css|images|js)/
    security: false
  main:
    anonymous: true
    lazy: true
    provider: app_user_provider
    form_login:
      login_path: security_login
      check_path: security_login
      username_parameter: login[email]
      password_parameter: login[password]
    logout:
      path: security_logout
```

Les Access Control

Un access_control va définir les limitations d'accès à certaines parties du site.

Dans notre cas, on indique que :

- L'url /login est accessible sans authentification.
- L'url /admin n'est accessible qu'en étant authentifié avec un utilisateur ayant le rôle "ROLE_ADMIN".
- Tout le reste du site n'est accessible qu'aux utilisateurs authentifiés c-à-d ayant le rôle "ROLE_USER".

```
access_control:  
- { path: ^/admin, roles: ROLE_ADMIN }  
- { path: ^/, roles: ROLE_USER }  
- { path: ^/login, roles: IS_AUTHENTICATED_ANONYMOUSLY }
```