Guide de l'authentification

L'utilisateur est représenté par la classe App\Entity\User. Une contrainte d'unicité est appliquée à l'attribut email afin de ne pas avoir de doublon.

```
# src/App/Entity/User.php
/**
 * @ORM\Entity(repositoryClass=UserRepository::class)
 * @ORM\Table("user")
 * @UniqueEntity("email")
 */
class User implements UserInterface
{
```

La sécurité de l'application est configurée dans le fichier config/packages/security.yaml. Les utilisateurs sont enregistrés en BDD par doctrine. L'utilisateur est authentifié par l'attribut username :

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

Un pare-feu est désigné afin d'empêcher un utilisateur non authentifié d'accéder à certaines parties du site. Pour s'authentifier, on utilise un formulaire accessible à la route login :

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

  main:
    anonymous: ~
    pattern: ^/
    form_login:
      login_path: login
      check_path: login_check
      always_use_default_target_path: true
      default_target_path: /
    logout: ~
```

On permet à l'utilisateur anonyme d'accéder à cette route grâce au paramètre access_control :

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

L’accès au formulaire de création d'un utilisateur est également laissé libre. Par contre, l’accès à la gestion des utilisateurs (route qui commence par /admin/) est réservée aux membres possédant le rôle : ROLE_ADMIN.

Les utilisateurs avec le rôle "ROLE_ADMIN" hérite aussi du rôle utilisateur "ROLE_USER" :

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