

DOCUMENTATION

Utilisateurs

1/ Création de l'entité User

Commande utilisée: `php/bin console make:user`

- Choix du nom
- Stockage dans database
- Ajout des propriétés
- Hasher le mot de passe
- Création des fichiers correspondants:
 - src/Entity/Users.php
 - src/Repository/UsersRepository.php
 - config/packages/security.yaml(updated)

L'entité pourra être modifié par la commande :

`php bin/console make:entity [son nom]`

2/ Création de l'authentification

Il faut un formulaire de connexion.

Commande utilisée: `php/bin console make:auth`

- Comment souhaitez-vous vous authentifier?
Login form authenticator
- Le nom de la classe à créer

LoginAuthenticator

- Choisir le nom du controller

SecurityController

- Souhaitez-vous générer un fichier logout (déconnexion)?

Yes

- Création des fichiers correspondants :

src/Security/LoginAuthenticator.php

src/Controller/SecurityController.php

Templates/security/login.html.twig

Config/packages/security.yaml (update).

L'authentification

1/ le fichier de configuration **security.yaml**

- Rassemble chaque partie nécessaire à l'authentification
- Il décrit les règles d'authentifications et d'autorisations
- Les informations sont utilisées par la classe User

La partie security indique le système de password-hasher (mode du mot de passe)

La partie provider indique quelle entité est utilisée pour la gestion des utilisateurs

La partie Access control indique les chemins et accès

Le but de la sécurité est d'interdire l'accès à une ressource à laquelle il n'aurait pas droit

La configuration de la sécurité se fait en utilisant des données en mémoire et en stockant les données dans une database.

Il pourra y être utilisé : des annotations et des règles dans le fichier security.yaml.

=> Security.yaml

- Une partie utilisateur
- Une partie authentication (formulaire)
- La gestion des rôles

La commande : php bin/console make:auth =>

- Modifie security.yaml
- Ajoute un controller
- Ajoute une classe qui hérite de AbstractForLoginAuthenticator
- Ajoute un template

Nous avons un formulaire de login sécurisé et une route.

Le controller , ajoutons un use de l'encoderInterface pour crypter le mot de passe saisi lors de l'inscription.

Le formulaire pour un champ de password caché.

Ensuite nous pouvons modifier notre template pour ajouter l'input de verification du mot de passe.