



Guide général sur l'authentification de l'application

1. Entité User	3
2. Authentification d'un utilisateur	3
3. Stockage des utilisateurs (user)	4
4. Contraintes sur les champs des utilisateurs (user)	4
5. Accès aux pages d'administration	4
6. Ajouter des droits d'administrateur à un utilisateur	6

1. Entité User

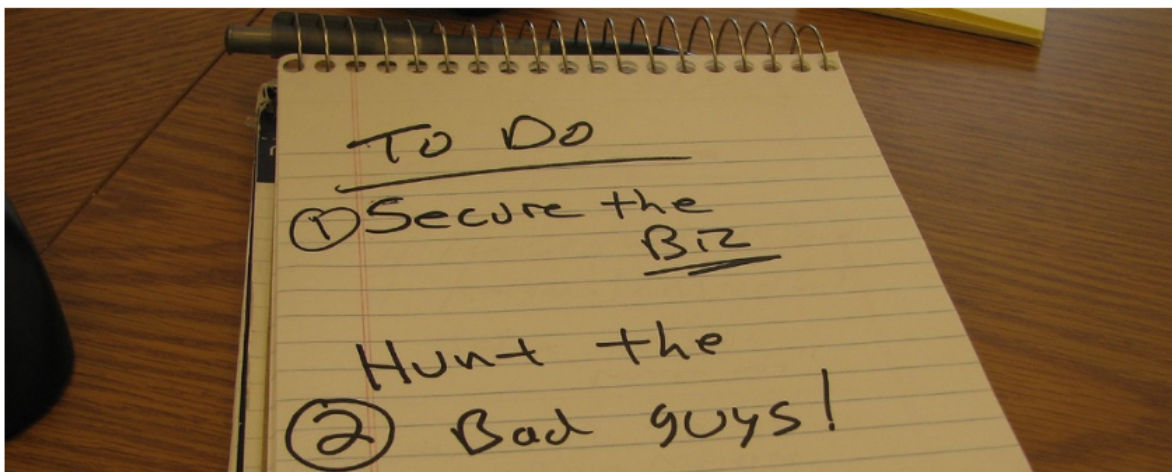
Un utilisateur est représenté par l'entité User de l'application. L'utilisateur est identifié avec son username. Ces paramètres sont configurés dans le fichier config/packages/security.yaml :

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

2. Authentification d'un utilisateur

Pour s'authentifier, un utilisateur doit se rendre sur la page /login et remplir les champs "Nom d'utilisateur" et "Mot de passe" du formulaire.

To Do List app



Nom d'utilisateur : Mot de passe :

3. Stockage des utilisateurs (user)

Les différents utilisateurs enregistrés sur l'application sont sauvegardés dans la base de données dans la table user. Chaque entrée dans cette table est composée de :

- l'ID de l'utilisateur (généré automatiquement lors de sa création)
- un username unique
- un mot de passe chiffré
- un email unique
- un tableau de rôles permettant de gérer les droits d'accès de l'utilisateur

Exemple :

id	username	password	email	roles
1	yoann	\$2y\$13\$fdxcZGs48zHbS3...	test@gmail.com	["ROLE_ADMIN"]
7	test5	\$2y\$13\$hdbkNqGULuaPIX...	test5@test.com	["ROLE_USER"]

4. Contraintes sur les champs des utilisateurs (user)

La contrainte d'unicité sur le nom d'utilisateur et sur l'email est configurée dans l'entité User de l'application ("UniqueEntity")

```
// src/Entity/User.php
#[ORM\Entity(repositoryClass: UserRepository::class)]
#[ORM\Table(name: '`user`')]
#[UniqueEntity("username", message: "Ce nom d'utilisateur est déjà utilisé.")]
#[UniqueEntity("email", message: "Cet email est déjà utilisé.")]
class User implements UserInterface, PasswordAuthenticatedUserInterface
{
```

5. Accès aux pages d'administration

L'accès aux pages d'administration est géré avec les RÔLES des utilisateurs.

Exemple :

```
// src/Controller/UserController.php
$hasAccess = $this->isGranted('ROLE_ADMIN');
if (!$hasAccess) {
    $this->addFlash('error', 'Accès non autorisé.');
```

Pour les autres pages et fonctionnalités, leurs accès sont gérés avec les Voters de symfony.
Exemple :

```
// src/Controller/TaskController.php
#[Route('/tasks/create', name: 'task_create')]
public function create(ManagerRegistry $doctrine, Request $request): Response
{
    $task = new Task();

    // can the user create a task ?
    $hasAccess = $this->isGranted('create', $task);
    if (!$hasAccess) {
        $this->addFlash('error', 'Accès non autorisé.');
```

```
}
    $this->denyAccessUnlessGranted('create', $task);
}

// src/Security/TaskVoter.php
protected function voteOnAttribute(string $attribute, mixed $subject, TokenInterface $token): bool
{
    $user = $token->getUser();

    if (!$user instanceof User) {
        // the user must be logged in; if not, deny access
        // this condition filter if the user can toggle a task
        return false;
    }

    // ROLE_ADMIN can do anything
    if (in_array('ROLE_ADMIN', $user->getRoles())) {
        return true;
    }

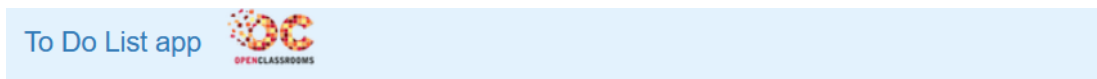
    // you know $subject is a Task object, thanks to `supports()`
    /** @var Task $task */
    $task = $subject;

    switch ($attribute) {
        case self::EDIT:
            return $this->canEdit($task, $user);
        case self::DELETE:
            return $this->canDelete($task, $user);
        case self::TOGGLE:
        case self::CREATE:
            return true; // filtered above with "if (!$user instanceof User)"
    }

    throw new \LogicException('This code should not be reached!');
}
```

6. Ajouter des droits d'administrateur à un utilisateur

Pour ajouter les droits d'administrateur à un utilisateur, il faut se rendre sur la page /users. Seuls les administrateurs de l'application ont accès à cette page.



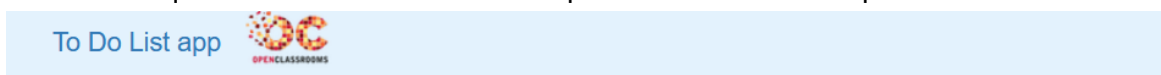
Créer un utilisateur

Se déconnecter

Liste des utilisateurs

#	Nom d'utilisateur	Adresse d'utilisateur	Rôles	Actions
1	yoann	yoann.leonard@gmail.com	ROLE_ADMIN ROLE_USER	<button>Edit</button>
2	test	yoannleonard@gmail.com	ROLE_USER	<button>Edit</button>
3	test2	test@test.com	ROLE_USER	<button>Edit</button>
4	test3	test3@test.com	ROLE_USER	<button>Edit</button>

Il suffit de cliquer sur le bouton "Edit" correspondant à l'utilisateur que l'on souhaite modifier.



Créer un utilisateur

Se déconnecter

Modifier test

Nom d'utilisateur

test

Adresse email

yoannleonard@gmail.com

Roles

☐ ROLE_ADMIN

☒ ROLE_USER

Modifier

Pour ajouter les droits d'administrateur à un utilisateur, il faut que la case "ROLE_ADMIN" soit cochée.