



## Gérer vos authentifications.

. Login

. Créer de nouveau utilisateurs.

. Assigner des rôles aux utilisateurs.

. Autorisation des fonctionnalités.

-----

## RAPPEL :

L'authentification sous symfony se fait grâce au système de sécurité intégré au Framework.

Il faut pour cela lors de la création initiale du projet indiquer vouloir la version –full et non le microservice.

## Powershell cmd :

\$ symfony new <projectName> --full **pour un projet standard**

*Documentation officielle*

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

Sinon il faudra télécharger le bundle de security ultérieurement

A screenshot of a terminal window with a dark background. The window has standard OS window controls (minimize, maximize, close) in the top right corner. The command entered in the terminal is: \$ composer require symfony/security-bundle

voir les instructions détaillées sur la documentation officielle.

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

**Il y à plusieurs façons de se connecter** dans notre cas on souhaite se connecter

Via un formulaire de login :

<https://symfony.com/doc/current/security.html#form-login>

## Comment se logger ?

Uri : `/login`

Routename : `app_login`

Controller : `namespace App\Controller SecurityController ::class`

---

Explication

---

## Le formulaire de login.



Please sign in

Email

Password

Sign in

View : `src > templates > security > login.html.twig`

Input email            `value="{{ last_username }}"`

Input password

Input\_csrf\_token    `value="{{ csrf_token('authenticate') }}"`

---

Explication

Voir CSRF Protection in Login Forms

<https://symfony.com/doc/current/security.html#csrf-protection-in-login-forms>

## Comment créer un nouvel utilisateur dans la bdd ?

Il nous faut une classe User qui pourra selon le principe du PPO

Instancier de nouvel objet à la demande new User().

Le Framework Symfony nous dispense d'une commande Maker très utile.

Symfony **make :auth**

Cette instruction va vous permettre de générer l'entité User

Celui-ci implémente les services **UserInterface** ainsi que **PasswordAuthenticatedUserInterface**

Cela va nous permettre de gagner du temps sur la saisie des imports, namespace ect..

Après il faudra modifier et compléter cette classe à votre convenance mais elle respectera les contraintes liées au processus d'authentification mis en place par Symfony.

### RAPPEL :

Ce Maker doit être si vous avez suivi toutes les instructions disponibles

Dans votre projet si cela n'est pas le cas je vous conseille de revoir depuis le

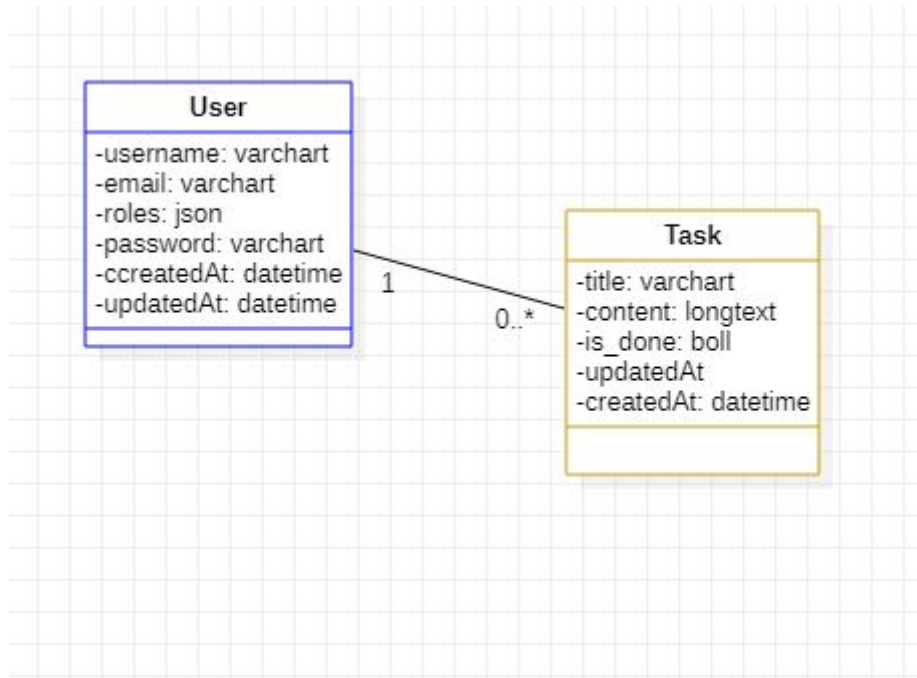
Début de l'installation sinon il faudra le rajouter manuellement

**Composer req maker**

## L' entité User

Class (constraints)

Table



Referer vous à la class User pour decouvrir toute les contraintes mises en place dans

Les annotations des attributs.

Src > Entity > User.php

```
#[ORM\Column(type: 'string', length: 25, unique:true)]
#[Assert\NotBlank(message:'Vous devez saisir un nom d\'utilisateur.')]
#[Assert\Length(min: 3, max: 20,
    minMessage:'Your first name must be at least {{ limit }}.',
    maxMessage:'Your first name cannot be longer than {{ limit }}.',
)]
private $username;
```

Ce projet requiert php 8.0.2 au minimum

```
"require": {"php": ">=8.0.2",
```

Vous allez retrouver toute les instructions pour installer le projet github directement dans votre machine locale en respectant toute les versions requie pour faire tourner ce projet.



Composer.json & composer.lock

## Rappel :

Composer.json

<https://getcomposer.org/doc/01-basic-usage.md#composer-json-project-setup>

Composer.lock (Installation des dépendances)

<https://getcomposer.org/doc/01-basic-usage.md#installing-dependencies>