



**TEK-UP Ecole Supérieure Privée Technologie &
Ingénierie**

Ateliers Framework (**Symfony**)

Jasser Jammeli
jasserjammeli98@gmail.com

A.U. 2025-2026

Atelier 6

**Symfony
Security**

Créer un utilisateur :

php bin/console make:user

Une entité **User** sera créée. Un utilisateur est identifié par : **id, email, roles, password et name.**

Cette entité implémente l'interface **UserInterface**.


```
C:\xampp\php\php.exe C:/xampp/htdocs/SymfonyPr/bin/console make:user

The name of the security user class (e.g. User) [User]:
>

Do you want to store user data in the database (via Doctrine)? (yes/no) [yes]:
>yes
yes

Enter a property name that will be the unique "display" name for the user (e.g.
email, username, uuid) [email]:
>email
email

Will this app need to hash/check user passwords? Choose No if passwords are not
needed or will be checked/hashed by some other system (e.g. a single sign-on se
rver).

Does this app need to hash/check user passwords? (yes/no) [yes]:
>yes
```


Créer un authentificateur d'utilisateur :

php bin/console make:auth

```
C:\xampp\php\php.exe C:/xampp/htdocs/SymfonyPr/bin/console make:auth

What style of authentication do you want? [Empty authenticator]:
  [0] Empty authenticator
  [1] Login form authenticator
>1
1

The class name of the authenticator to create (e.g. AppCustomAuthenticator):
>UserAuthenticator
UserAuthenticator

Choose a name for the controller class (e.g. SecurityController) [SecurityController]:
>

Do you want to generate a '/logout' URL? (yes/no) [yes]:
>yes
yes
```



```
Do you want to support remember me? (yes/no) [yes]:
```

```
>yes
```

```
yes
```

```
How should remember me be activated? [Activate when the user checks a box]:
```

```
[0] Activate when the user checks a box
```

```
[1] Always activate remember me
```

```
>1
```

```
1
```


UserAuthenticator: OnAuthenticationSuccess

```
public function onAuthenticationSuccess(Request $request, TokenInterface $token, string $firewallName):  
{  
    if ($targetPath = $this->getTargetPath($request->getSession(), $firewallName)) {  
        return new RedirectResponse($targetPath);  
    }  
  
    $user = $token->getUser();  
    if(in_array(needle: "ROLE_CLIENT", $user->getRoles(), strict: true)){  
        return new RedirectResponse($this->urlGenerator->generate( name: "app_voiture"));  
    }  
  
    if(in_array(needle: "ROLE_AGENT", $user->getRoles(), strict: true)){  
        return new RedirectResponse($this->urlGenerator->generate( name: "app_client_index"));  
    }  
    return new RedirectResponse($this->urlGenerator->generate( name: "app_location_new"));  
}
```

Get the
connected user

**Rediriger l'utilisateur
en fonction de son
rôle :**

Si l'utilisateur a le rôle
« **ROLE_CLIENT** », il
sera redirigé vers la
liste des voitures. (la
route de cette page est
« **app_voiture** »)

php bin/console make:registration

```
C:\xampp\php\php.exe C:/xampp/htdocs/SymfonyPr/bin/console make:registration
Creating a registration form for App\Entity\User

Do you want to add a #[UniqueEntity] validation attribute to your User class to
make sure duplicate accounts aren't created? (yes/no) [yes]:
>yes
yes

Do you want to send an email to verify the user's email address after registrat
ion? (yes/no) [yes]:
>no
no

Do you want to automatically authenticate the user after registration? (yes/no)
[yes]:
>yes
yes
```


RegistrationForm: ajouter un nouveau champ “**roles**”

```
$builder
    ->add( child: 'email')

    ->add( child: 'roles', type: ChoiceType::class, [
        'choices' => [
            'Client' => 'ROLE_CLIENT',
            'Agent'  => 'ROLE_AGENT',
        ],
        'expanded' => true,
        'multiple' => false,
        'label' => 'Choisissez un rôle',
    ]);
```


RegistrationForm: ajouter un transformeur pour convertir une chaîne de caractères (le rôle sélectionné depuis le **ChoiceType**) **en un tableau** (le champ **roles**, qui est un tableau).

```
// TRANSFORMER
$builder->get('roles')
    ->addModelTransformer(new CallbackTransformer(
        function ($rolesArray): ?string {
            return $rolesArray[0] ?? null;
        },
        function ($roleString): array {
            return [$roleString];
        }
    ));
```


Security.yaml

Nous pouvons contrôler l'accès aux différentes pages en ajoutant un **contrôle d'accès** pour chaque page.

Par exemple, pour le chemin **/addVoiture**, seuls les utilisateurs ayant le rôle « **ROLE_AGENT** » peuvent y accéder.

```
# Easy way to control access for large sections of your site
# Note: Only the *first* access control that matches will be used
access_control:
    - { path: ^/addVoiture, roles: ROLE_AGENT }
    # - { path: ^/profile, roles: ROLE_USER }
```