



TEK-UP Ecole Supérieure Privée Technologie &
Ingénierie

Ateliers Framework (**Symfony 6**)

Wisssem ELJAOUED
wisssem.eljaoued@ensi-uma.tn

A.U. 2023-2024

Atelier 6

Symfony Security

Create a user:

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

A **User** entity will be created. A User is identified by : **id, email, roles, password and a name**. This entity implements a **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
```

Create a user authenticator:

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

Redirect the user
based on its Role:
- If it is a
« ROLE_CLIENT », it
will be redirected to the
list of cars. (the route if
this page is
« 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: add a new field “roles”

```
    ->add( child: 'email')
    ->add( child: "roles", type: ChoiceType::class,
    [
        "choices"=> [
            "ROLE_CLIENT"=>"ROLE_CLIENT",
            "ROLE_AGENT"=>"ROLE_AGENT"
        ]
    ])
}
```

RegistrationForm: add a transformer to convert a string (the selected role from the choicetype) to an array (the field roles which is an array).

```
$builder->get('roles')
    ->addModelTransformer(new CallbackTransformer(
        function ($tagsAsArray): string {
            // transform the array to a string
            return implode(separator: ', ', $tagsAsArray);
        },
        function ($tagsAsString): array {
            // transform the string back to an array
            return explode(separator: ', ', $tagsAsString);
        }
    ));
;
```

Security.yaml

We can control the access to different pages by adding an access control to each page. For example, for the path `/addVoiture`, only users they have `"ROLE_AGENT"` can access to it.

```
# 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 }
```