# Configuration du projet concours étape par étape

# Installation initiale

Création d’un projet v5.2 en mode full.

# Ajout des dépendances pour la barre de debug

composer require symfony/apache-pack

# Librairies utilisées

Bootstrap 4.6.0 :

css\bootstrap.min.css

css\bootstrap.min.css.map

js\ bootstrap.min.js

js\ bootstrap.min.js.map

Librairie core JQuery 3.6.0:

jquery-3.6.0.min.js

Jinplace :

js\jinplace-1.2.1.min.js

Pour les upload (jQuery File Upload Plugin):

css\jquery.fileupload.css

js\jquery.fileupload.js

js\jquery.iframe-transport.js

js\jquery.ui.widget.js

Pour les datepicker :

css\jquery-ui.min.css

js\jquery-ui.min.js

Pour les modal boxes (jquery confirm v3.3.4):

css\jquery-confirm.min.css

js\jquery-confirm.min.js

# Configuration de la sécurité

# Ajout de la classe user

composer require symfony/security-bundle

C:\wamp64\www\concours>php bin/console make:user

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

> concours\_user

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

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

> licence

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 server).

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

created: src/Entity/ConcoursUser.php

created: src/Repository/ConcoursUserRepository.php

updated: src/Entity/ConcoursUser.php

updated: config/packages/security.yaml

Success!

Next Steps:

- Review your new App\Entity\ConcoursUser class.

- Use make:entity to add more fields to your ConcoursUser entity and then run make:migration.

- Create a way to authenticate! See https://symfony.com/doc/current/security.html

Modification de entity\ConcoursUser :

Ajout de :

use Symfony\Bridge\Doctrine\Validator\Constraints\UniqueEntity;

\* @UniqueEntity(fields={"licence"}, message="Il y a déjà un utilisateur avec cette licence")

Ajouter les champs supplémentaires.

## Ajout d’un formulaire d’enregistrement pour les admins

Pré-requis :  
composer require symfonycasts/verify-email-bundle

composer require symfony/swiftmailer-bundle

Installation :

C:\wamp64\www\concours>php bin/console make:registration

Creating a registration form for App\Entity\ConcoursUser

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

By default, users are required to be authenticated when they click the verification link that is emailed to them.

This prevents the user from registering on their laptop, then clicking the link on their phone, without

having to log in. To allow multi device email verification, we can embed a user id in the verification link.

Would you like to include the user id in the verification link to allow anonymous email verification? (yes/no) [no]:

> yes

What email address will be used to send registration confirmations? e.g. mailer@your-domain.com:

> postmaster@arc-cholet.com

What "name" should be associated with that email address? e.g. "Acme Mail Bot":

> TA Cholet

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

>

! [NOTE] No Guard authenticators found - so your user won't be automatically authenticated after registering.

What route should the user be redirected to after registration?:

[0 ] \_preview\_error

[1 ] \_wdt

[2 ] \_profiler\_home

[3 ] \_profiler\_search

[4 ] \_profiler\_search\_bar

[5 ] \_profiler\_phpinfo

[6 ] \_profiler\_search\_results

[7 ] \_profiler\_open\_file

[8 ] \_profiler

[9 ] \_profiler\_router

[10] \_profiler\_exception

[11] \_profiler\_exception\_css

[12] accueil

> 12

updated: src/Entity/ConcoursUser.php

created: src/Security/EmailVerifier.php

created: templates/registration/confirmation\_email.html.twig

created: src/Form/RegistrationFormType.php

created: src/Controller/RegistrationController.php

created: templates/registration/register.html.twig

Success!

Next:

1) In RegistrationController::verifyUserEmail():

\* Customize the last redirectToRoute() after a successful email verification.

\* Make sure you're rendering success flash messages or change the $this->addFlash() line.

2) Review and customize the form, controller, and templates as needed.

3) Run "php bin/console make:migration" to generate a migration for the newly added ConcoursUser::isVerified property.

Then open your browser, go to "/register" and enjoy your new form!

Retouches dans RegistrationFormType.php :

Suppression de agreeTerms

Ajout des rôles

Retouches dans register.html.twig : ajouter les forms\_error et les champs manquants

Dans RegistrationController : ajout d’une ligne avec un return sur la page d’accueil pour invalider l’enregistrement.

Penser à mettre à jour la base de données

## Ajout d’un formulaire de connexion

C:\wamp64\www\concours>php bin/console make:auth

What style of authentication do you want? [Empty authenticator]:

[0] Empty authenticator

[1] Login form authenticator

> 1

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

> LoginFormAuthenticator

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

> LoginFormAuthenticator

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

>

created: src/Security/LoginFormAuthenticator.php

updated: config/packages/security.yaml

created: src/Controller/LoginFormAuthenticatorController.php

created: templates/security/login.html.twig

Success!

Next:

- Customize your new authenticator.

- Finish the redirect "TODO" in the App\Security\LoginFormAuthenticator::onAuthenticationSuccess() method.

- Review & adapt the login template: templates/security/login.html.twig.

## Ajout d’un formulaire « mot de passe oublié »

Pré-requis :

composer require symfonycasts/reset-password-bundle

Installation:

php bin/console make:reset-password

C:\wamp64\www\concours>php bin/console make:reset-password

Let's make a password reset feature!

====================================

Implementing reset password for App\Entity\ConcoursUser

- ResetPasswordController -

---------------------------

A named route is used for redirecting after a successful reset. Even a route that does not exist yet can be used here.

What route should users be redirected to after their password has been successfully reset? [app\_home]:

> accueil

- Email -

---------

These are used to generate the email code. Don't worry, you can change them in the code later!

What email address will be used to send reset confirmations? e.g. mailer@your-domain.com:

> postmaster@arc-cholet.com

What "name" should be associated with that email address? e.g. "Acme Mail Bot":

> TA Cholet

created: src/Controller/ResetPasswordController.php

created: src/Entity/ResetPasswordRequest.php

updated: src/Entity/ResetPasswordRequest.php

created: src/Repository/ResetPasswordRequestRepository.php

updated: src/Repository/ResetPasswordRequestRepository.php

updated: config/packages/reset\_password.yaml

created: src/Form/ResetPasswordRequestFormType.php

created: src/Form/ChangePasswordFormType.php

created: templates/reset\_password/check\_email.html.twig

created: templates/reset\_password/email.html.twig

created: templates/reset\_password/request.html.twig

created: templates/reset\_password/reset.html.twig

Success!

Next:

1) Run "php bin/console make:migration" to generate a migration for the new "App\Entity\ResetPasswordRequest" entity.

2) Review forms in "src/Form" to customize validation and labels.

3) Review and customize the templates in `templates/reset\_password`.

4) Make sure your MAILER\_DSN env var has the correct settings.

5) Create a "forgot your password link" to the app\_forgot\_password\_request route on your login form.

Then open your browser, go to "/reset-password" and enjoy!

Dans src\Controller\Form\ ResetPasswordRequestFormType.php, changer la première ligne du builder pour mettre le champ licence :

->add('licence', EmailType::class, [

Dans C:\wamp64\www\concours\src\Controller\ResetPasswordController.php :

private function processSendingPasswordResetEmail(string $emailFormData, MailerInterface $mailer): RedirectResponse

{

$user = $this->getDoctrine()->getRepository(ConcoursUser::class)->findOneBy([

'email' => $emailFormData,

]);

Remplacer email par licence

## Divers

Ajouter :  
access\_denied\_url: /

Dans config\packages\security.yaml, section « security »

Activer les traductions en français dans config\packages\translation.yaml

default\_locale: fr

Activer CSRF :

# config/packages/framework.yaml

framework:

# ...

csrf\_protection: true

Ajouter l’option se rappeler de moi et configurer la cible de logout :

# config/packages/security.yaml

Firewalls :

remember\_me:

secret: '%kernel.secret%'

lifetime: 604800 # 1 week in seconds

path: /

logout:

path: app\_logout

# where to redirect after logout

target: accueil