Tutoriel Symfony

Composer = libraire, gestionnaire de dependances de PHP.

sert a telecharger des frameworks / librairies.

npm = gestionnaire de dependances de JavaScript.

Framework outil permettant dutiliser une techno (CSS, HTML, JS, PHP, ...) dune autre maniere. Cest un tableau de paquet

Telecharger Symfony :

https://symfony.com/download

aller dans Binaries et cliquer sur 386.

Le telecharger sur le disque :c -> dans un dossier tools

aller dans variables denvironnement -> Path -> new -> tools puis tout valider

Ouvrir un terminal et taper juste symfony, sil ny a pas de message derreur cest bon

Creer un projet Symfony :

Aller dans le dossier ou lon veut travailler avec cd :

symfony new *nom-projet* webapp

(le premier on la appele cours_symfony)

Analyse des dossiers/fichiers du projet :

.env :

DATABASE_URL : tous les commenter sauf celui qui utilise mysql

changer app:!ChangeMe! en nomUtilisateur:motDePasse (pour nous juste root car pas de mot de passe)

Tutoriel Symfony

@127.0.0.1:3306 = adresse bdd

app = nom de la base de donnees (nous on a mis cours_symfony)

Pour creer une base de donnees a partir de la DATABASE_URL modifiee avant :

Dans un terminal, entrez la commande suivante :

symfony console doctrine:database:create

ou en raccourci : symfony console d:d:c

NE JAMAIS MODIFIER LA STRUCTURE DE LA BDD SUR PHPMYADMIN

Pour creer ou modifier les classes ici appelees entity :

symfony console make:entity

valider le [no]

Entrer le nom de lentity : (ici) Livre

Ensuite on entre les proprietes :

Ne jamais entrer lid car il le fait tout seul

name -> valider string -> valider 255 -> valider no -> valider

Verifier dans le dossier Entity si il y a lentite creee

Si on sest trompe en creant un attribut dune entity, il suffit de la supprimer dans la page concernee ainsi que ses get et set.

symfony console make:migration => creer un fichier preparant la creation des tables (une migration)

symfony console doctrine:migration:migrate

ou le raccourci symfony console d:m:m

Pour visualiser le site il faut aller sur lindex.php dans le dossier public avec la commande :

php -S localhost:8080 -t public

ou symfony serve

Un controller : est une classe quon cree nous-meme qui va contenir plusieurs methodes.

Chaque methode correspond a une URL particuliere.

Symfony console make:controller

Debug router : symfony console debug:router -> permet dobtenir la liste du nom des routes creees dans le projet en cours

Connexion utilisateur :

Ne pas creer lentity user avec make:entity mais avec symfony console make:user

Creer notre user

The name of the security user class : par defaut

Do you want to store data in the database : yes (par defaut)

Enter a property name that will be the unique display name for the user : comme on veut

Does this app need to hash/check user passwords : yes (par defaut)

Faire les migrations vers la bdd

Pour ajouter des proprietes a user : symfony console make:entity

Gerer lauthentification

symfony console make:auth

0 : authentification a distance; 1 : formulaire de connexion

The class name of the authenticator to create : AppCustomAuthenticator

Choose a name for the controller class : par defaut

Do you want to generate a '/logout' URL? : yes (defaut)

Do you want to support remember me? : comme on veut

Dans le fichier src\Security\AppCustomAuthenticator.php :

Commenter la ligne 53

Decommenter la ligne 52

Changer (sur cette meme ligne) some_route en le nom de la route ou lon veut aller (ex: app_home)

Formulaire de connexion :

symfony console make:registration-form

Do you want to add a #[UniqueEntity] validation attribute to your User class to make sure duplicate accounts aren't created? : yes (defaut)

Do you want to send an email to verify the user's email address after registration? : pour linstant no

Do you want to automatically authenticate the user after registration? yes defaut

Do you want to generate PHPUnit tests? [Experimental] no

Au final ca creer une entity user, trois routes (pour aller se connecter, se deconnecter et senregistrer)

Formulaire pour ajouter/modifier une entite :

symfony console make:form

Nommer : nomEntityType

Aller dans le fichier c nouvellement cree

Modifier le $builder dans la premiere fonction

Aller dans le controller qui modifie lentite

Creer une variable form qui creer un formulaire lie a notre entite vide a partir de nomEntityType

Le passer en parametre pour le twig avec createView()

Creer un filtre Twig :

symfony console make:twig-extension

Choisir nom de lextension

ca cree 2 sous-dossiers dans le dossier twig

Faire la fonction (ou le filtre) dans le fichier Runtime

La connecter au twig avec le fichier Extension

Creer une commande symfony :

symfony console make:command

la nommer (sans espaces) en commencant par app:

exemple app:hello, app:product-create...

Creation dun dossier Command dans src contenant notre fichier

On peut lappeler avec symfony console app:...

Modifier notre commande

Ne jamais toucher au __consturct()

Methode configure() permet dajouter des arguments et des options

pour les arguments, sutilise comme suit : app:hello world

Utiliser Bootstrap :

aller dans le fichier twig.yaml

Ajouter la ligne suivante dans twig:

form_themes: ["bootstrap_5_layout.html.twig"]

php bin/console importmap:require bootstrap

Reserver des routes a des roles :

Aller dans security.yaml

decommenter les lignes 42 et 43

car sinon les gens peuvent entrer manuellement lurl et acceder aux parties admin ou aux profils des autres users

Faire les tests :

Installer le pack de test de Symfony avec :

composer require dev symfony/test-pack

Creer un test avec :

symfony console make:test

Celui qui nous interesse le plus cest le WebTestCase

La base de donnees de test est definie dans le .env.test, il faut, comme pour un .env normal, ajouter le DATABASE_URL.

L'interet est de ne pas alterer la bdd de production ou de dev avec les tests.

Mise en place dun environnement de dev :

Creation DB de test :

php bin/console d:d:c --env=test

Application des migrations en test :

php bin/console d:m:m --env=test

Application de la friture en test :

php bin/console hautelook:fixtures:load --purge-with-truncate -n --env=test

Transferer des donnees vers js :

Plusieurs methodes :

Dans le twig avec json_encode() :

Attention : cette methode peut ne pas marcher avec les entity (circular reference)

Creer une div vide avec un data attribute (ex pour un animal: data-anima={{animal.json_encode()}}

La recuperer dans le js (querySelector)

Dans le controller avec Serializer et des groups

en haut du controller :

use Symfony\Component\Serializer\SerializerInterface;

Dans lentity principale, faire un constructor :

```
 public function __construct(
        private SerializerInterface $serializer
    ) {}
```

dans les entity (plusieurs possibles si liees) :

#[Groups(['nom_choisi'])]

private ?int $id = null; (attribut voulu)

dans la route :

$json = $this->serializer->serialize(

la donnee a convertir,

json, (le format)

[

groups => [les groupes],

]);

2.3 Dans une div comme pour methode 1