

10/5/2022

CRUD és un entorn que facilita la interacció amb la BD. **No permès!!**.

Amb el setup que vaig fer ahir al vespre:

Al directori de la carpeta del projecte; ***php bin/console doctrine:database:create*** per crear la base de dades. Prèviament he editat el fitxer ***.env*** a la mateixa carpeta i he afegit la línia ***DATABASE_URL="mysql://root@127.0.0.1:3306/bd_sy?serverVersion=5.7&charset=utf8mb4"*** per indicar la BBDD.

Al navegador, buscant ***localhost/phpmyadmin*** i posant com a usuari ***root*** apareix la BD creada.

De volta al cmd, ***php bin/console make:user*** crea una taula a la BD pels usuaris (posant només ***make*** mostra una ajuda amb les diferents opcions). Crea dos arxius:

- ***Src/Entity/User.php***: L'entitat. Reflexa la taula que tenim al ***phpmyadmin***. Entitat és el nom que dona symfony a aquest reflex de la BBDD.
- ***Src/Repository/UserRepository.php***: El repositori. Aquí es donaran les ordres per demanar les dades.

Actualitza:

- ***Config/packages/security.yaml***: Arxiu de seguretat.

DOCUMENTACIÓ SYMFONY: <https://symfony.com/doc/current/index.html>

Al cmd, ***php bin/console doctrine:schema:update --force*** actualitza les dades de la BD (indica que s'ha executat una ***query***). Si recarrego la BD (F5) ja em surt la taula ***user*** amb els camps. Puc consultar l'estructura (l'estructura reflexa allò indicat al fitxer ***src/entity/user.php***).

Checkpoint: Tinc el servidor creat i una base de dades amb una taula per afegir usuaris. Treballaré una mica amb aquesta taula i configuració per recordar conceptes i veure possibles millores i errors. Més tard aplicaré aquesta feina a la pràctica, on hauré de treballar amb una taula pels proveïdors.

==Treballant amb la taula ***user***==

Php bin/console make:entity. Actualitza una entitat creada. Indico el nom ***User*** i el nom de la columna o propietat a agregar, en aquest cas ***nom***.

== Error amb l'AnnotationDriver ==

Error: ***Class Doctrine\Persistence\Mapping\Driver\AnnotationDriver does not exist***. Possible problema amb la versió de Doctrine. Tinc la versió "2.7.1-DEV" (fitxer ***vendor/doctrine/orm/lib/Doctrine/ORM/Version.php***).

composer require symfony/orm-pack. Instala el suport per doctrine, però ja he utilitzat doctrine per crear la BBDD no?

composer require --dev symfony/maker-bundle genera codi extra i pot ajudar, però no serveix per la versió 4 de symfony (requerida per la practica).

Tot i haver instal·lat orm-pack segueix sortint l'error.

Tinc al *composer.json*, dins de *require*, "doctrine/orm": "^2.12" Canvio a "^2.7" i executo **composer update** (si posava 2.* donava problemes).

Tenia:

```
"doctrine/annotations": "^1.0",  
    "doctrine/doctrine-bundle": "^2.6",  
    "doctrine/doctrine-migrations-bundle": "^3.2",  
    "doctrine/orm": "^2.12",
```

Segueix sortint el mateix problema.

Moc el fitxer *AnnotationDriver.php* de
D:\wamp64\www\Practica_Symfony\vendor\doctrine\orm\lib\Doctrine\ORM\Mapping\Driver
a
D:\wamp64\www\Practica_Symfony\vendor\doctrine\persistence\src\Persistence\Mapping\Driver.

Problema: Compile Error: Cannot declare class
doctrine\ORM\Mapping\Driver\AnnotationDriver, because the name is already in use.

Executo **composer require doctrine/annotations**. Ara la versió de annotations és la 1.13.

Segueix fallant el make:entity.

Al doctrinehelper.php, substitueixo (comento) la línia

```
use Doctrine\Persistence\Mapping\Driver\AnnotationDriver;
```

per

```
use Doctrine\ORM\Mapping\Driver\AnnotationDriver;
```

Ara funciona el make:entity. Al doctrinehelper buscava el fitxer del driver a un path on no hi era. D'aquesta forma indico un de diferent, on si està el fitxer.

== Error solucionat pel que sembla ==

Amb lo de entity actualitzo la entitat user. Afegeixo els següents camps:

- name: string, 255, no pot ser null.
- img: string, 255, **si** pot ser null. Un user pot no tenir foto.
- active: boolean, pot ser null.
- surname: string, 255, pot ser null.

S'ha actualitzat la informació del fitxer *Src/Entity/User.php* però no a la BBDD. Falta executar **php bin/console doctrine:schema:update --force** per actualitzar la BBDD amb la informació de la entitat.

Afegeixo un usuari manualment al phpmyadmin, a inserir i omplint els camps. S'executa
INSERT INTO `user` (`id`, `email`, `roles`, `password`, `name`, `img`, `active`, `surname`) VALUES
(NULL, 'user1@mail.com', '[\"ROLE_USER\", \"ROLE_ADMIN\"]', '1234', 'Joel', 'default.jpg', '1',
'Chaves');

He afegit tres usuaris.

== Controladors ==

Permeten tenir una pàgina diferent segons la URL. Per exemple, una per usuari i una altra per l'inici.

Creat amb php bin/console make:controller. Crea src/Controller/UserController.php

i templates/user/index.html.twig (pagina on anirà el codi HTML).

Accedeixo amb el navegador amb *localhost/Practica_Symfony/public/user*.

== Errors, herència de plantilles a Twig i sistema de blocs a twig ==

Base.html.twig és la plantilla de plantilles. Admet codi HTML. Cada sector ha d'estar dins d'un block, i cada block te un nom. Si hi ha dos blocs amb mateix nom, un al pare i un al fill, fa cas al del pare.

Si faig l'herència de base, només surt el text si ho poso dins d'un bloc body. Si te un altre nom no.

== Repàs i afegir un camp a la BBDD ==

Cadena:

BBDD (Reflexa la informació de la entitat) -> Entitat (Defineix la estructura) -> Controlador (controla rutes) -> Template (Disseny de la pagina).

S'hauria d'afegir i fer migració.

== Llibreries en symfony, mètodes predefinits i mètodes per tractar dades ==

Die() talla l'execució del programa.

Problema amb l'objectManager. Diu q no existeix el servei. Amb ManagerRegistry ho puc fer bé.

ManagerRegistry és més idoni pels casos on hi pot haver més d'un EntityManager o ObjectManager. ManagerRegistry sap quins EntityManagers hi ha disponibles i en troba el correcte (<https://matthiasnoback.nl/2014/05/inject-the-manager-registry-instead-of-the-entity-manager/>).

= Dump i for a twig. Mostra de dades de la BD. =

Dump ho mostra tot. Es pot mostrar camp per camp de forma semblant a com queda en un document de text.

= Afegir un formulari =

Php bin/console make:form per crear un formulari. Crea la carpeta form amb el fitxer **UserType.php**.

Elimino camps de imatge, rol i actiu. Els decidiré jo, no l'usuari.

Havia posat la contrasenya com a que no podia ser null. Ho canvio afegint un ? abans de String al mètode getPassword.

= Importar llibreries per validar formularis i enviar dades =

Assegurarem dades correctes i evitarem codi maliciós amb les validacions.

== bootstrap i CSS. Apartat visual. Més endavant amb la feina interna feta. ==

= Veure les dades del formulari i afegir les dades a la BD =

Fet, ja puc afegir nous usuaris. Ara bé, quan envio les dades els camps del formulari no es buiden, només el de la contrasenya.

Seguint la docu del symfony he afegit per editar i eliminar usuaris.

Checkpoint: Tinc ja una primera aproximació de l'aplicació!!! Enlloc de proveïdors treballo amb usuaris, amb camps lleugerament diferents als requerits pels proveïdors.

- Crear un nou usuari (*localhost/Practica_Symfony/*): Mostro un formulari on es poden omplir els camps.
- Editar un usuari (*localhost/Practica_Symfony/user/edit/id*): Mostro el mateix formulari que per crear-ho, i les dades introduïdes s'actualitzen.
- Eliminar un usuari (*localhost/Practica_Symfony/user/remove/id*): S'elimina correctament i mostra l'html.
- Visualitzar usuari (*localhost/Practica_Symfony/read*): Es mostra informació dels usuaris a la BD.

A partir d'aquí, en ordre de prioritat:

1. Adaptar l'aplicació als proveïdors, enlloc de tractar amb usuaris.
2. Mostrar millor la informació dels proveïdors de la BD. Ara mateix mostro diversos camps per veure exemples.
3. Buidar els camps del formulari quan es crea un usuari per poder introduir-ne un de nou sense haver de recarregar la pàgina.
4. Buidar els camps del formulari quan s'edita un usuari per poder tornar a modificar-ho sense haver de recarregar la pàgina (no estic segur, realment la edició es faria només un cop a excepció de si hi ha un error).
5. Afegir bootstrap i CSS perquè les pàgines quedin millor visualment.
6. Adaptar l'aplicació a Docker.

Crearé primer que tot la taula de proveïdors. Haurà de tenir els camps:

- Nom: string. Pot ser null.
- Correu: string. No pot ser null.
- Telefon: string. Pot null.
- Tipus de proveïdor: string. No pot null. Hauria de controlar que fos un d'aquests?
Repasar BBDD i MySQL.
 - Hotel
 - Pista
 - Complement
- Actius o no: Boolean. Null no.
- Data d'alta: datetime. Null no.
- Data de la darrera edició: datetime. Null no.

Ja he creat la nova entitat i afegit la taula. A partir d'aquí seria repetir el procés que fet pels usuaris adaptat als proveïdors, cuidant més la part de mostrar la informació i posar títols adequats.

Demà segueixo.

11/5/2022

Començaré a fer de nou el procés, ara per l'entitat proveïdor.

Fitxers de twig creats per cada pàgina (read, add, edit i delete).

Vaig a pel controlador. Fet. Crearé ara al controlador funcions per cada pàgina i modificare els twig.

Idea: Puc fer de la pantalla d'inici una guia amb els *path* per cada pàgina, i un missatge de benvinguda. (fet)

Al formulari pel telèfon poso teltype tot i que realment no ho validarà, ja que en funció del país la validació seria diferent (<https://symfony.com/doc/current/reference/forms/types/tel.html>).

Idea: Redirigir a la pantalla inicial quan hagi acabat d'afegir o editar usuari. (fet)

Checkpoint: Ja tinc un primer prototip amb els proveïdors!!!! La idea ara seria mirar lo del bootstrap i afegir CSS perquè quedi millor visualment. Voldria també mirar d'afegir botons.

Ara puc indicar quin proveïdor vull editar, sense haver d'escriure-ho a la url perquè s'edita o elimines directament com feia abans (era més insegur). Ho he aplicat també a l'hora d'eliminar proveïdor. Idealment s'hauria de controlar l'accés ja que un usuari podria accedir igualment fent la url per error i acabar canviant o eliminant coses.

Em poso amb lo del CSS.

Afegits línies del CSS i Bundle al base.html.twig. Ja es veu el canvi al navegador. Ara form_themes: ['bootstrap_4_layout.html.twig'] (5 no funciona) a config/packages/twig.yaml per afegir el tema dels formularis.

Afegint lo del container al base.html.twig creo un marge a esquerra i dreta.

Checkpoint: He afegit una barra de navegació des d'on es pot accedir a les diferents funcionalitats (deixo igualment els enllaços a la pantalla principal) de forma molt ràpida. Visualment tot queda més centrat i els formularis es veuen molt millor. També incorpore a la barra de navegació el logo de l'empresa amb el nom (que redirigeix a la web de l'empresa), i un logo sense el nom com a icona de la pestanya.

Em poso amb la part de valoració i README.

Eliminaré les parts de lo de usuari que vaig fer com a prova i ho actualitzo tot al git.