

## DOCUMENT EXPLICATIF SYMFONY ECF 4 :

### VIDEO 1 :

1. Installation PHP sur l'ordinateur
2. Lien pour téléchargement : [PHP: Hypertext Preprocessor \(https://www.php.net/\)](https://www.php.net/)
3. Installation Symfony sur l'ordinateur (les démarches sont décrites sur le lien : [Compositeur\(getcomposer.org\)\)](https://getcomposer.org/)):
  - Téléchargez et exécutez [Composer-Setup.exe](#) - il installera la dernière version de Composer chaque fois qu'elle sera exécutée.
  - Pour installer rapidement Composer dans le répertoire courant, exécutez le script suivant dans votre terminal (VSC)

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"  
php -r "if (hash_file('sha384', 'composer-setup.php') ===  
'e21205b207c3ff031906575712edab6f13eb0b361f2085f1f1237b7126d785e826a450292b6cf1d6  
4d92e6563bbde02') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-  
setup.php'); } echo PHP_EOL;"  
php composer-setup.php
```

**Voici le message du Terminal qui indique la bonne installation :**

```
Installer verified  
All settings correct for using Composer  
Downloading...  
  
Composer (version 2.5.8) successfully installed to: C:\Users\Utilisateur\Desktop\Formation\Symfony\PROJET-SYMFONY\composer.phar  
Use it: php composer.phar  
  
PS C:\Users\Utilisateur\Desktop\Formation\Symfony\PROJET-SYMFONY> []
```

4. Vérifier version PHP (8.0.26)

> **php -v**

*Remarque :*

- *Pour installer Symfony 6, il faut au moins la version 8.0.2 de PHP.*
- *Si la commande **php** n'est pas reconnue, il faut vérifier le PATH dans les variables d'environnement et modifier ou ajouter le chemin vers le dossier de l'exécutable souhaité. Ex : C:\wamp64\bin\php\php8.0.26*

Explications sur la page suivante :



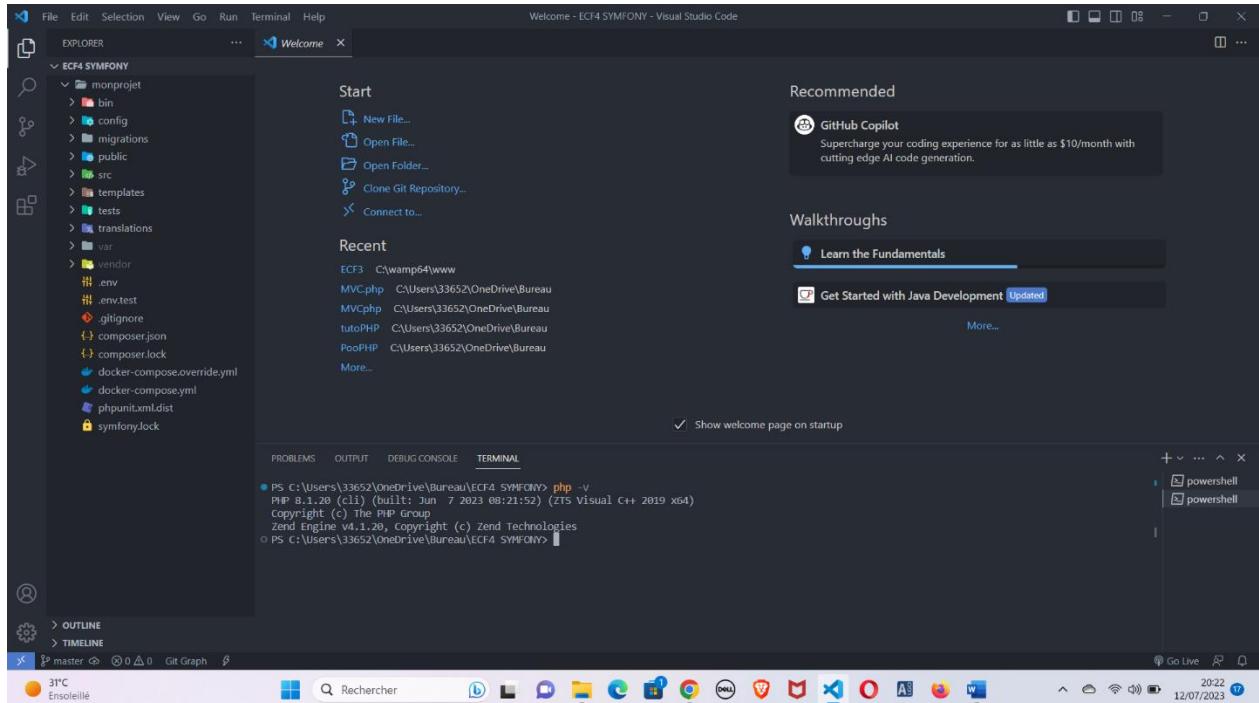
Si la commande `php` n'est pas reconnue, cela signifie généralement que le chemin vers l'exécutable PHP n'est pas correctement configuré dans les variables d'environnement.

Pour modifier ou ajouter le chemin vers le dossier de l'exécutable PHP dans les variables d'environnement, vous pouvez suivre les étapes suivantes :

1. Cliquez avec le bouton droit sur "Ce PC" (ou "Ordinateur") sur votre bureau ou dans le menu Démarrer, puis sélectionnez "Propriétés".
2. Dans la fenêtre des propriétés système, cliquez sur "Paramètres système avancés" dans le volet de gauche.
3. Dans la fenêtre "Propriétés système", cliquez sur le bouton "Variables d'environnement".
4. Dans la section "Variables système", recherchez la variable nommée "Path" (ou "PATH") et sélectionnez-la, puis cliquez sur le bouton "Modifier".
5. Dans la fenêtre "Modifier les variables système", vous verrez une liste de chemins séparés par des points-virgules (;). C'est ici que vous pouvez ajouter ou modifier le chemin vers l'exécutable PHP.
6. Si vous avez déjà le chemin vers le dossier de l'exécutable PHP, cliquez sur "Nouveau" pour ajouter un nouveau chemin et collez le chemin complet vers le dossier de l'exécutable PHP.
7. Si vous ne connaissez pas le chemin exact vers l'exécutable PHP, vous pouvez le rechercher dans le dossier d'installation de PHP. Par exemple, si vous utilisez WampServer, le chemin par défaut pourrait être `C:\wamp64\bin\php\php8.0.26`. Assurez-vous d'ajouter le chemin correct correspondant à votre installation.
8. Une fois que vous avez ajouté ou modifié le chemin, cliquez sur les boutons "OK" pour fermer toutes les fenêtres ouvertes.
9. Redémarrez votre terminal ou votre invite de commandes pour que les modifications des variables d'environnement prennent effet.

Après avoir suivi ces étapes, vous devriez pouvoir exécuter la commande `php -v` avec succès.

Voici le message qui doit s'afficher :



Installons le scoop sur l'ordinateur :

Il faut ouvrir Windows PowerShell qui est installé par défaut sur l'ordinateur

Voici le message qui doit s'afficher dans le **Windows PowerShell** :

```
Modification de la stratégie d'exécution
La stratégie d'exécution permet de vous prémunir contre les scripts que vous jugez non fiables. En modifiant la
stratégie d'exécution, vous vous exposez aux risques de sécurité décrits dans la rubrique d'aide
about_Execution_Policies à l'adresse https://go.microsoft.com/fwlink/?LinkId=135170. Voulez-vous modifier la stratégie
d'exécution ?
[O] Oui [T] Oui pour tout [N] Non [U] Non pour tout [S] Suspendre [?] Aide (la valeur par défaut est « N ») : O
Initializing...
Downloading ...
Extracting...
Creating shim...
Adding ~\scoop\shims to your path.
Scoop was installed successfully!
Type 'scoop help' for instructions.
PS C:\Users\Utilisateur>
```

### Pour installer Symfony

1. Copiez la commande **scoop install symfony-cli** sans les caractères ">".
2. Dans la fenêtre PowerShell, collez la commande en vous assurant qu'il n'y a pas de caractères supplémentaires au début de la ligne.
3. Appuyez sur "Entrée" pour exécuter la commande.

```
scoop install symfony-cli
```

Voici le message qui doit apparaître :

```
PS C:\Users\Utilisateur> scoop install symfony-cli
>>
Installing 'symfony-cli' (5.5.6) [64bit] from main bucket
symfony-cli_windows_amd64.zip (5,2 MB) [=====] 100%
Checking hash of symfony-cli_windows_amd64.zip ... ok.
Extracting symfony-cli_windows_amd64.zip ... done.
Linking ~\scoop\apps\symfony-cli\current => ~\scoop\apps\symfony-cli\5.5.6
Creating shim for 'symfony'.
'symfony-cli' (5.5.6) was installed successfully!
```

Vérifions si Symfony est bien installé avec la commande :

```
symfony check:requirements
```

Voici le message qui indique la bonne installation de Symfony :

```
Symfony Requirements Checker
~~~~~
> PHP is using the following php.ini file:
C:\wamp64\bin\php\php7.4.33\php.ini

> Checking Symfony requirements:
.....W.....
[OK]
Your system is ready to run Symfony projects

Optional recommendations to improve your setup
~~~~~

* realpath_cache_size should be at least 5M in php.ini
  > Setting "realpath_cache_size" to e.g. "5242880" or "5M" in
  > php.ini* may improve performance on Windows significantly in some
  > cases.

Note The command console can use a different php.ini file
~~~~~ than the one used by your web server.
Please check that both the console and the web server
are using the same PHP version and configuration.

PS C:\Users\Utilisateur>
```

## VIDEO 2:

### Présentation et configuration du projet (Symfony 6) :

Le projet consiste à faire un site e-commerce en utilisant le framework symfony du coup en utilisant l'invite de commande pour son installation.

Le site contient une base de donnée schématisée avec le site dB diagramme.

La base de donnée contient une entité :

**utilisateurs** qui contient ( l'id, email, mot de passe, nom, prénom, adresse, code postal, ville et date de création).

**Les commandes** : qui contient (l'id, la référence de la commande, l'id de l'utilisateur et la date de la commande).

**Les details** : qui vont faire intervenir un produit ou plusieurs avec la quantité et le prix.

**Les categories** : qui contiennent un lien avec la catégorie parents et pour chaque produit un ou plusieurs images.

#### **1- Configuration du projet :**

Avant de commencer, nous devons nous rendre sur le terminal et vérifier la version php avec la commande : **php -v** et appuyer sur entrer et on aura le message suivant (par exemple) :

```
PS C:\Users\Utilisateur\Desktop\Formation\Symfony\PROJET-SYMFONY> php -v
PHP 8.2.7 (cli) (built: Jun 7 2023 10:25:24) (NTS Visual C++ 2019 x64)
Copyright (c) The PHP Group
Zend Engine v4.2.7, Copyright (c) Zend Technologies
```

La version demandée peut également être vérifiée sur le composer.json qui est php 8.0 au minimum.

Comme on peut vérifier également la version symfony par la commande suivante : **Symfony -v** et on aura un résultat comme suite :

```
PS C:\Users\33652\OneDrive\Bureau\ECF4 SYMFONY> symfony -v
Symfony CLI version 5.5.6 (c) 2021-2023 Fabien Potencier standwithukraine support Ukraine (2023-05-16T13:26:30Z - stable)
Symfony CLI helps developers manage projects, from local code to remote infrastructure
```

## 2- Création d'un nouveau projet Symfony :

Avec la commande : « symfony new » on peut à présent créer un nouveau projet, qui inclura composer. Exemple :

```
PS C:\Users\r dame\OneDrive\Documents\afpa\back_end\ECF4\test ECF4_1> symfony new nouveau_projet
* Creating a new Symfony project with Composer
  (running C:\ProgramData\ComposerSetup\bin\composer.phar create-project symfony/skeleton C:\Users\r dame\OneDrive\Documents\afpa\back_end\ECF4\test ECF4_1\nouveau_projet --no-interaction)

* Setting up the project under Git version control
  (running git init C:\Users\r dame\OneDrive\Documents\afpa\back_end\ECF4\test ECF4_1\nouveau_projet)

[OK] Your project is now ready in C:\Users\r dame\OneDrive\Documents\afpa\back_end\ECF4\test ECF4_1\nouveau_projet
```

Mais ce n'est pas souvent suffisant pour créer une architecture Symfony complète. Avec cette ligne de commande, on aura le minimum. L'idéal pour créer et initialiser une architecture Symfony complète est de créer un projet avec la ligne de commande Symfony new –webapp.

```
PS C:\Users\r dame\OneDrive\Documents\afpa\back_end\ECF4\test ECF4_1> symfony new --webapp reda
* Creating a new Symfony project with Composer
  (running C:\ProgramData\ComposerSetup\bin\composer.phar create-project symfony/skeleton C:\Users\r dame\OneDrive\Documents\afpa\back_end\ECF4\test ECF4_1\reda --no-interaction)

* Setting up the project under Git version control
  (running git init C:\Users\r dame\OneDrive\Documents\afpa\back_end\ECF4\test ECF4_1\reda)
```

## 3- Lancement du serveur Symfony :

Pour lancer le serveur symfony, il faut utiliser la ligne de commande « symfony serve »

```

Welcome - ECF4 SYMFONY - Visual Studio Code

EXPLORER
  ECF4 SYMFONY
    monprojet
      bin
      config
      migrations
      public
      src
      templates
      tests
      translations
      var
      vendor
        .env
        .env.test
        .gitignore
        composer.json
        composer.lock
        docker-compose.override.yml
        docker-compose.yml
        phunit.xml.dist
        symfony.lock

Start
  New File...
  Open File...
  Open Folder...
  Clone Git Repository...
  Connect to...

Recent
  ECF4 C:\wamp64\www

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\j33652\OneDrive\Bureau\ECF4 SYMFONY> symfony serve
[WARNING] run "symfony.exe server:ca:install" first if you want to run the web server with TLS support, or use "--p12" or "--no-tls" to avoid this warning

Following Web Server log file (C:\Users\j33652\symfony5\log\ae6988198463ab634ba890d3c573bedbbf67a4.log)
[WARNING] The local web server is optimized for local development and MUST never be used in a production setup.

[OK] Web server listening
http://127.0.0.1:8000

[Web Server] Jul 14 13:13:43 [WARN] HTTP unable to detect the front controller, disabling the PHP server error="Passthru script \"index.php\" does not exist under C:\\\\Users\\\\j33652\\\\OneDrive\\\\Bureau\\\\ECF4 SYMFONY\\\\: CreateFile C:\\\\Users\\\\j33652\\\\OneDrive\\\\Bureau\\\\ECF4 SYMFONY\\\\index.php: The system cannot find the file specified."

```

( Ps : Le warning signifie que pour l'instant, nous n'avons pas la connexion sécurisée mise en place sur le serveur)

#### 4- Installation d'un certificat de sécurité sur le projet en cours

Avec la ligne de commande « Symfony server : ca install » vous pouvez installer un certificat de sécurité sur le projet en cours

```

Welcome - .env.local - ECF4 SYMFONY - Visual Studio Code

EXPLORER
  ECF4 SYMFONY
    monprojet
      .env.local
        # In all environments, the following files are loaded if they exist,
        # the latter taking precedence over the former:
        #
        #   .env
        #     contains default values for the environment variables needed by the app
        #   .env.Local
        #     uncommitted file with local overrides
        #   .env.$APP_ENV
        #     committed environment-specific defaults
        #   .env.$APP_ENV.Local
        #     uncommitted environment-specific overrides
        #
        # Real environment variables will overwrite env files
        #
        # DO NOT DEFINE PRODUCTION SETTINGS HERE
        # https://symfony.com/doc/current/install/recommended/docker.html#ssl

        Command "server" does not exist.
        Did you mean one of these?
          serve
          server:ca:install
          server:ca:uninstall
          server:list
          server:log
          server:prod
          server:restart
          server:stop
          server:stop

PS C:\Users\j33652\OneDrive\Bureau\ECF4 SYMFONY> server:ca:install
server:ca:install : Le terme 'server:ca:install' n'existe pas. Veuillez vérifier l'orthographe du nom, ou si un chemin d'accès existe, vérifiez que le chemin d'accès est correct et réessayez.
Au caractère Ligne1 : 1
+ server:ca:install
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (server:ca:install:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

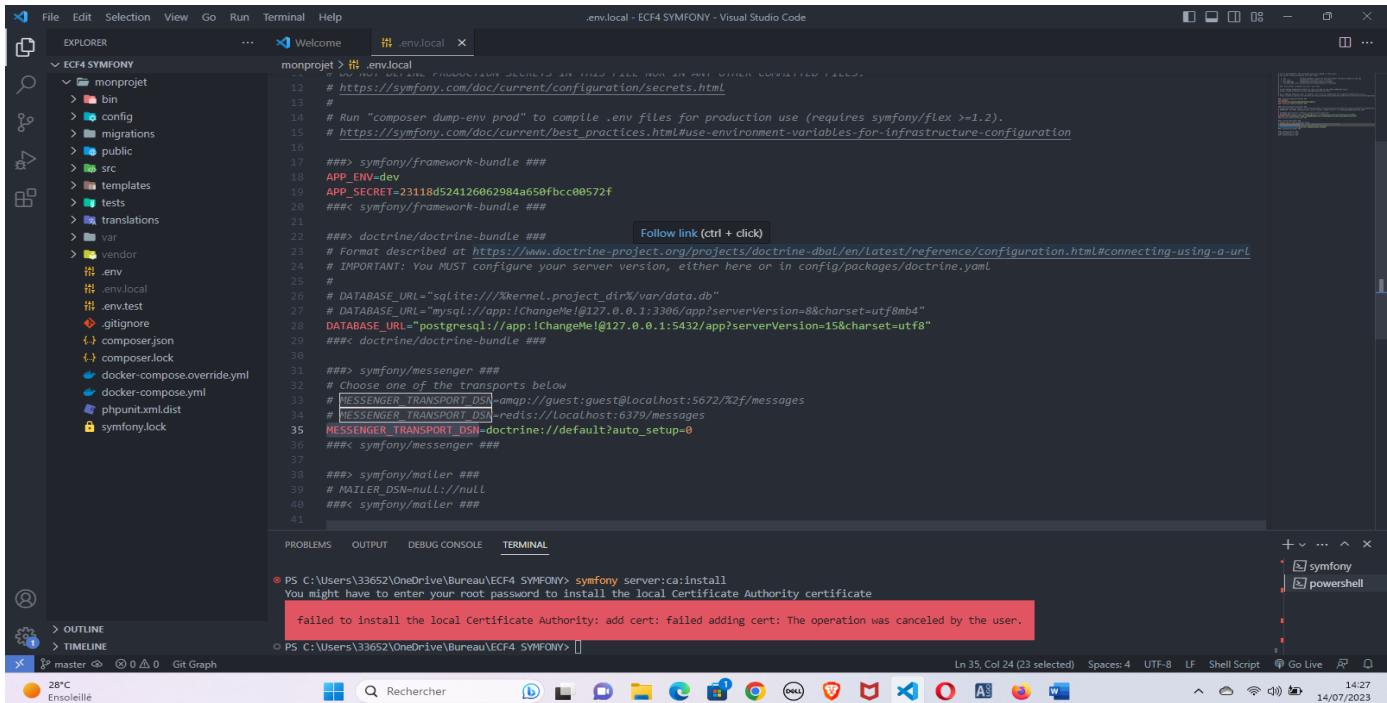
PS C:\Users\j33652\OneDrive\Bureau\ECF4 SYMFONY> symfony server:ca:install
You might have to enter your root password to install the local Certificate Authority certificate

```

Un certificat de sécurité est parfois obligatoire pour pouvoir tester ses projets web dans de bonnes conditions.

#### 5- Le fichier « .env » contient un certain nombre de variables d'environnement qui permettent de créer notre environnement de développement qu'on va configurer en cliquant sur « enregistrer sous » ce fichier et le nommer .env.local afin de garder nos propres paramètres.

Les variables à voir sur ce fichier sont :



```
# https://symfony.com/doc/current/configuration/secrets.html
# Run "composer dump-env prod" to compile .env files for production use (requires symfony/flex >=1.2).
# https://symfony.com/doc/current/best_practices.html#use-environment-variables-for-infrastructure-configuration

##> symfony/framework-bundle ##
APP_ENV=dev
APP_SECRET=23118d524126062984a650fbcc00572f
##< symfony/framework-bundle ##

##> doctrine/doctrine-bundle ##
# Format described at https://www.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/configuration.html#connecting-using-a-url
# IMPORTANT: You MUST configure your server version, either here or in config/packages/doctrine.yaml
#
# DATABASE_URL="sqlite:///kernel.project_dir/var/data.db"
# DATABASE_URL="mysql://app:!ChangeMe@127.0.0.1:3306/app?serverVersion=8&charset=utf8mb4"
# DATABASE_URL="postgresql://app:!ChangeMe@127.0.0.1:5432/app?serverVersion=15&charset=utf8"
# DATABASE_URL="redis://localhost:6379/messages"
MESSANGER_TRANSPORT_DSN=doctrine://default:auto_setup
##< doctrine/doctrine-bundle ##

##> symfony/messenger ##
# Choose one of the transports below
# MESSANGER_TRANSPORT_DSN=amqp://guest:guest@localhost:5672/%2f/messages
# MESSANGER_TRANSPORT_DSN=redis://localhost:6379/messages
MESSANGER_TRANSPORT_DSN=doctrine://default:auto_setup
##< symfony/messenger ##

##> symfony/mailer ##
# MAILER_DSN=null://null
##< symfony/mailer ##
```

- **APP\_ENV= DEV** qui est en **DEV** car on est en mode développement et qui peut passer en **PROD** lorsqu'on souhaite passer en mode **PROD**.
- **APP-SECRET** : Elle est utilisée pour générer les jetons pour la validation des formulaires. (Cette clé doit être modifiée lorsqu'on passe en mode PROD).
- **MESSENGER \_TRANSPORT\_DSN** pour la partie notifications et envoie de mails en fil d'attente ainsi que la configuration des différents services.
- **La configuration de la base de données** : Par défaut elle configure avec **Postgresql** et pour changer par exemple à **mysql** on met la ligne de **postgresql** en commentaire (désactiver). Puis remplacer les mots users, password et db\_name par le nom d'utilisateur, le mot de passe lui-même ainsi que le nom de votre BDD. (Les valeurs de chaque case).
- **MAILER DSN** : Si nous voulons envoyer des mails il nous faudra un intercepteur de mails ou un server distant. L'envoie se fait par le biais de localhost en utilisant mailHog.

**Le mailHog n'a pas fonctionné pour moi... !!!**

### VIDEO 3:

#### 1- Création de la base de donnée :

Se mettre sur notre projet dans le terminal avec la commande suivante: **cd monprojet** puis créer la base de donnée avec la commande : **symfony console doctrine:database:create**

```

monprojet > env.local
17 ##> symfony/framework-bundle ##
18 APP_ENV=dev
19 APP_SECRET=23118d524126062984a650fbcc00572f
20 ##< symfony/framework-bundle ##
21
22 ##> doctrine/doctrine-bundle ##
23 # Formatt described at https://www.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/configuration.html#connecting-using-a-url
24 # IMPORTANT: You MUST configure your server version, either here or in config/packages/doctrine.yaml
25 #
26 # DATABASE_URL="sqlite:///kernel.project_dir/var/data.db"
27 # DATABASE_URL="mysql://root@127.0.0.1:3306/maBDD Symfony?serverVersion=8&charset=utf8mb4"
28 # DATABASE_URL="postgresql://app:ChangeMe@127.0.0.1:5432/app?serverVersion=15&charset=utf8"
29 ##< doctrine/doctrine-bundle ##
30
31 ##> symfony/messenger ##
32 # Choose one of the transports below
33 # MESSENGER_TRANSPORT_DSN=amp://guest:guest@localhost:5672/%f/messages
34 # MESSENGER_TRANSPORT_DSN=redis://localhost:6379/messages
35 MESSENGER_TRANSPORT_DSN=doctrine://default?auto_setup=0
36 ##< symfony/messenger ##
37
38 ##> symfony/mailer ##
39 MAILER_DSN=smt://localhost:1025
40 ##< symfony/mailer ##

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\33652\OneDrive\Bureau\ECF4 SYMFONY\monprojet> symfony console doctrine:database:create  
Created database "maBDD Symfony" for connection named default  
PS C:\Users\33652\OneDrive\Bureau\ECF4 SYMFONY\monprojet>

Ln 40, Col 24 Spaces: 4 UTF-8 LF Shell Script Go Live 16:41 15/07/2023

- Création de la table users : (la table qui servira pour l'authentification) avec la commande suivante : **symfony console make:user**
    - Il nous demandera le nom de la classe qu'on mettra au nom de : **Users**
    - Il nous demandera si nous voulons stocker les utilisateurs dans la base de donnée : **Yes**
    - Quel champ utiliser pour se connecter : **email**
    - Voulez-vous hacher les mots de passe : **Yes**
- Et voilà la table créée :

```

monprojet > env.local
17 ##> symfony/framework-bundle ##
18 APP_ENV=dev
19 APP_SECRET=23118d524126062984a650fbcc00572f
20 ##< symfony/framework-bundle ##
21
22 ##> doctrine/doctrine-bundle ##
23 # Formatt described at https://www.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/configuration.html#connecting-using-a-url
24 # IMPORTANT: You MUST configure your server version, either here or in config/packages/doctrine.yaml
25 #
26 # DATABASE_URL="sqlite:///kernel.project_dir/var/data.db"
27 # DATABASE_URL="mysql://root@127.0.0.1:3306/maBDD Symfony?serverVersion=8&charset=utf8mb4"
28 # DATABASE_URL="postgresql://app:ChangeMe@127.0.0.1:5432/app?serverVersion=15&charset=utf8"
29 ##< doctrine/doctrine-bundle ##
30

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

> Users
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]: >
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/Users.php
created: src/Repository/UsersRepository.php
updated: src/Entity/Users.php
updated: config/packages/security.yaml

Success!

Next Steps:
- Review your new App\Entity\Users class.
- Use make:entity to add more fields to your Users entity and then run make:migration.
- Create a way to authenticate. See https://symfony.com/doc/current/security.html#PS C:\Users\33652\OneDrive\Bureau\ECF4 SYMFONY\monprojet>
```

Ln 40, Col 24 (1 selected) Spaces: 4 UTF-8 LF Shell Script Go Live 16:57 15/07/2023

On a donc créé :

- **Users.php** dans **src/Entity** c'est dans cette entité que **Users** va implementer **UsersInterface** qui va permettre de gérer la sécurité et l'authentification.
- On a donc les annotations qui sont des attributs php8 qui vont nous permettre de

savoir également à quoi correspond chaque propriété :

**#[ORM\Id]** : ca veut dire que c'est une clé primaire.

**#[ORM\GeneratedValue]** Va faire un auto-increment.

**#[ORM\Column(type: 'integer')]** Le type est un entier.

**GetEmail** pour récupérer l'information d'un email d'un utilisateur.

**SetEmail** qui va permettre de stocker les informations.

On a également pour chaque propriété un Geter et un Seter.

On a la colonne pour les mails de type '**string**' avec **180** caractères et a l'option **unique**.

- **UsersRepository.php** dans `src/Repository`
  - **Security.yaml** dans `config/packages`: dans ce fichier on a ajouté le nom de l'entité avec laquelle on va se connecter et a modifié le provider afin d'indiquer que c'est l'entité **Users** et que c'est la propriété **email** ainsi que le **provider** qui a été ajouté dans le **firewalls**.

Si nous avons envie de modifier cette entité, nous allons procéder avec la commande suivante : **Symfony console make :entity** et avoir le résultat suivant :

Class name of the entity to create or update (e.g. OrangePizza):

>

Puis choisir : **Users** puis donner l'emble des propriétés l'id, l'email, le mot de passe...

Ajouter par exemple le **Lastname** (string) et mentionner le nombre de caractères qu'il peut contenir ; dans notre cas, nous avons mis **100**.

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder structure with ".env.local" and "security.yaml" files.
- Editor:** The ".env.local" file is open, displaying configuration for a Symfony application. It includes environment variables like APP\_ENV and APP\_SECRET, and doctrine/doctrine-bundle configurations.
- Terminal:** A terminal window is active, showing a series of prompts for creating a new entity named "OrangePizza". The user is asked for the class name, which is already present, and then adds a "Lastname" field of type string with a length of 255. The field is nullable and has a default value of "100". The user then adds a "firstname" field of type string with a length of 100.
- Status Bar:** Shows the current branch as "master", file statistics ("+0.36%", "0 △ 0"), and various system icons.

Après avoir terminé de rentrer l'ensemble des propriétés, il suffit de cliquer sur **entrer** et voici le résultat attendu :

```

File Edit Selection View Go Run Terminal Help
.env.local x security.yaml M
monprojet > .env.local
17 ##### symfony/framework-bundle #####
18 APP_ENV=dev
19 APP_SECRET=23118d524126862984a650fbcc00572f
20 ##### symfony/framework-bundle #####
21
22 ##### doctrine/doctrine-bundle #####
23 # Format described at https://www.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/configuration.html#connecting-using-a-url
24 # IMPORTANT: You MUST configure your server version, either here or in config/packages/doctrine.yaml
25 #
26 # DATABASE_URL="sqlite:///Xkernel.project_dir%/var/data.db"
27 # DATABASE_URL="mysql://root@127.0.0.1:3306/mab00Symfony?serverVersion=8&charset=utf8mb4"
28 # DATABASE_URL="postgresql://app:1ChangeMe!@127.0.0.1:5432/app?serverVersion=15&charset=utf8"
29 ##### doctrine/doctrine-bundle #####
30

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Add another property? Enter the property name (or press <return> to stop adding fields):
> created_at

Field type (enter ? to see all types) [datetime_immutable]:
>

Can this field be null in the database (nullable) (yes/no) [no]:
>

updated: src/Entity/Users.php

Add another property? Enter the property name (or press <return> to stop adding fields):
>

Success!

Next: When you're ready, create a migration with symfony.exe console make:migration
PS C:\Users\J33652\OneDrive\Bureau\ECF4 SYMFONY\monprojet>

```

PS master\* ① ② ③ ④ Git Graph ⑤

CHF/EUR -0,39% Rechercher

Ln 40 Col 24 (1 selected) Spaces: 4 UTF-8 LF Shell Script Go Live 11:58 16/07/2023

On peut vérifier dans entité/Users.php si les informations sont bien mises à jour et on a donc cela :

```

File Edit Selection View Go Run Terminal Help
EXPLORER .env.Local Users.php U
monprojet > src > Entity > Users.php > ...
private array $roles = [];

/**
 * @var string The hashed password
 */
#[ORM\Column]
private ?string $password = null;

#[ORM\Column(length: 100)]
private ?string $lastname = null;

#[ORM\Column(length: 100)]
private ?string $firstname = null;

#[ORM\Column(length: 255)]
private ?string $address = null;

#[ORM\Column(length: 5)]
private ?string $zipcode = null;

#[ORM\Column(length: 150)]
private ?string $city = null;

#[ORM\Column]
private ?DateTimeImmutable $created_at = null;

public function getId(): ?int
{
    return $this->id;
}

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Success!

Next: When you're ready, create a migration with symfony.exe console make:migration
PS C:\Users\J33652\OneDrive\Bureau\ECF4 SYMFONY\monprojet>

```

PS master\* ① ② ③ ④ Git Graph ⑤

27°C Ensoleillé Rechercher

Ln 1 Col 1 Spaces: 4 UTF-8 LF PHP Go Live 12:04 16/07/2023

Comme on a mis qu'il y a une valeur par défaut dans la base de donnée, on va à **DateTimelImmutable** et ajouter les options (**je n'ai pas pu la modifier**)

La nous avons fini avec **Users**, on passera aux **catégories** :

- Name : **(string)** **(100 caractères)**, ne peut pas être **nul**.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The file "Users.php" is open, showing PHP code for an Entity named "User". It includes annotations like `@var string` and `#ORM\Column`. A terminal window is visible at the bottom of the editor area.
- Terminal:** The terminal shows the command-line interaction for generating the entity. It asks for property names ("name", "password") and their lengths (255 and 100). It also asks if the "password" field can be null in the database (the answer is no).
- Side Bar:** Shows a sidebar with multiple "powershell" entries and one "symfony m..." entry.
- Bottom Bar:** Shows system status (30°C, Ensoleillé), taskbar icons (Windows, Search, Task View, File Explorer, etc.), and system tray icons (Dell, battery, signal, etc.).

La on va avoir la notion de parent (parent-id); on va créer une propriété qui s'appelle « **parent** » qui va avoir « **relation** » comme type de propriété.

Quelle classe doit être reliée à la catégorie : **La catégorie elle-même**.

On aura le résultat suivant :

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The file "Users.php" is open, showing PHP code for an Entity named "User". It includes a method `getUserIdentifier()` returning the email address.
- Terminal:** The terminal shows the command-line interaction for defining a relationship. It asks for the type of relationship and provides options: ManyToOne, OneToMany, ManyToMany, and OneToOne. The user selects "ManyToOne".
- Side Bar:** Shows a sidebar with multiple "powershell" entries and one "symfony m..." entry.
- Bottom Bar:** Shows system status (30°C, Ensoleillé), taskbar icons (Windows, Search, Task View, File Explorer, etc.), and system tray icons (Dell, battery, signal, etc.).

On va avoir un assistant qui va nous indiquer ce qu'on a comme type de relation.

On lit la phrase de la **ManyToOne** comme suite :

- Chaque « **Catégorie** » est en relation avec un « **parent** ».
- Chaque « **parent** » peut avoir plusieurs « **catégories** » à l'intérieur.

Dans l'assistant, on tape : **ManyToOne**

- Est-ce que le « **parent** » peut être nul « **Yes** »
- Est-ce que je veux ajouter une nouvelle propriété pour que je puisse accéder aux catégories d'un parent. « **Yes** »
- Comment s'appelle la propriété qui va être créée dans la classe ? « **Catégories** ».

Après avoir fini, on tape sur entrer et on a le résultat suivant :

The screenshot shows the Visual Studio Code interface. In the top right, there's a terminal window with the following text:

```
Do you want to add a new property to Categories so that you can access/update Categories objects from it - e.g. $categories->getCategories()? (yes/no) [yes]:>
A new property will also be added to the Categories class so that you can access the related Categories objects from it.
New field name inside Categories [categories]:>
updated: src/Entity/Categories.php
Add another property? Enter the property name (or press <return> to stop adding fields):>
```

Below the terminal, a green box says "Success!".

At the bottom of the screen, the taskbar shows various icons for file operations like Open, Save, and Close, along with browser and system status indicators.

Du coup dans catégories, nous avons deux propriétés vu que c'est une propriété qui se relie à elle-même :

ManyToOne dans « **parent** » : son entité cible est elle-même et elle est inversée par la propriété « **catégories** »

OneToMany dans « **catégories** » qui est reliée à « **parent** » et son Target est l'entité elle-même. Il y a un constructeur qui est créé aussi pour créer la collection de catégories, puis on a les 3 méthodes :

**getCategories()** permet de chercher les catégories « **enfant** »

**addCategory** permet d'ajouter une catégorie « **enfant** »

**removeCategory** permet de supprimer/enlever une catégorie « **enfant** »

Nous avons également le **getParent** et le **setParent** qui nous permettent de récupérer un « **parent** ».

Après avoir créé la table « **Users** » et « **Catégories** » on va pouvoir accéder par exemple aux **coupons** et pour pouvoir faire les coupons, il faudra avoir les **types « coupons types »**. On fera les coupons types dans le terminal :

Faire la commande suivante : **symfony console make:entity**, on va l'appeler **Coupons Types**

The screenshot shows the Visual Studio Code interface. In the top right, there's a terminal window with the following text:

```
updated: src/Entity/CouponsTypes.php
Add another property? Enter the property name (or press <return> to stop adding fields):>
```

Below the terminal, a green box says "Success!".

At the bottom of the screen, the taskbar shows various icons for file operations like Open, Save, and Close, along with browser and system status indicators.

Comme cela nous avons notre table ainsi que notre entité.

Nous allons créer de la même manière ; les **Coupons, description, discount, max\_usage** : on va

mettre **is\_valid** pour qu'il le met automatiquement en boolean., **validity**, **created\_at**.  
Ensuite, il nous reste à faire la relation avec la commande suivante :

Nom de la relation : **coupons\_types**

Type : **relation**

Reliée à **CouponsTypes**

**Nous aurons le résultat suivant :**

The screenshot shows the Visual Studio Code interface with the Categories.php file open. The code defines a Categories entity with a many-to-many relationship to CouponsTypes. The IDE has provided several annotations in the margin:

- OneToMany: Each Coupons can relate to (can have) many CouponsTypes objects.  
Each CouponsTypes relates to (has) one Coupons.
- ManyToMany: Each Coupons can relate to (can have) many CouponsTypes objects.  
Each CouponsTypes can also relate to (can also have) many Coupons objects.
- OneToOne: Each Coupons relates to (has) exactly one CouponsTypes.  
Each CouponsTypes also relates to (has) exactly one Coupons.

At the bottom, there is a question: "Relation type? [ManyToOne, OneToMany, ManyToMany, OneToOne]:".

Si on met ManyToOne

- Est-ce qu'on peut ne pas avoir de type de coupons? No
- Est-ce qu'on peut aller chercher les coupons d'un type particulier ? Yes
- Est-ce qu'on veut supprimer automatiquement les coupons orphelins ?(En supprimant un type, on supprime les coupons correspondants) Yes

On aura ce résultat :

The screenshot shows the Visual Studio Code interface with the Categories.php file open. The code defines a Categories entity with a many-to-one relationship to CouponsTypes. The IDE has provided annotations and a success message:

A Coupons is "orphaned" when it is removed from its related CouponsTypes.  
e.g. \$couponsTypes->removeCoupons(\$coupons)

NOTE: If a Coupons may \*change\* from one CouponsTypes to another, answer "no".

Do you want to automatically delete orphaned App\Entity\coupons objects (orphanRemoval)? (yes/no) [no]: yes

updated: src/Entity/Coupons.php  
updated: src/Entity/CouponsTypes.php

Add another property? Enter the property name (or press <return> to stop adding fields):

>

Success!

Next: When you're ready, create a migration with symfony.exe console make:migration

PS C:\Users\33652\OneDrive\Bureau\ECF4 SYMFONY\monprojet>

Remarque : le code (varchar) dans la table coupons doit être unique, on ne peut pas utiliser plusieurs fois le même code.

On doit donc ajouter la propriété **unique :true**

### Créer l'entité Products :

Se mettre sur notre projet avec : **cd monprojet** ensuite : **symfony console make:entity**

On l'appellera : **Products**

Name, string, 255, ne peut pas être nul,

Description : type : text, ne peut pas être nul

Price : entier, ne peut pas être nul

Stock : entier, ne peut pas être nul

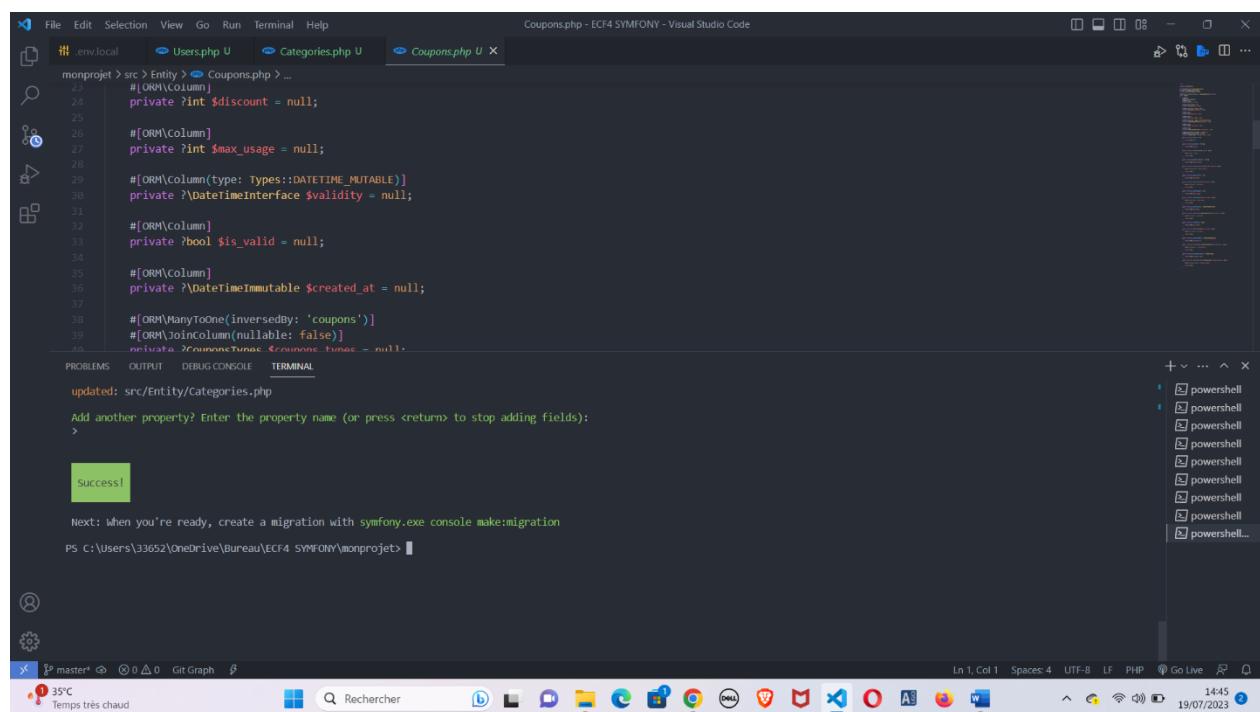
Created\_at : type : datetime, ne peut pas être nul

On va faire la relation : catégories : relation (string) reliée à catégories

ManyToOne : les catégories ne sont pas autorisées nul.

- Est-ce que la propriété peut y accéder : oui
- Si je supprime une catégorie, cela supprimera les produits en question : si le produit peut changer de catégorie, préférence mettre : non.

On a donc ce résultat :



```
# .env.local  Users.php U  Categories.php U  Coupons.php U
monprojet > src > Entity > Coupons.php > ...
23     #[ORM\Column]
24     private ?int $discount = null;
25
26     #[ORM\Column]
27     private ?int $max_usage = null;
28
29     #[ORM\Column(type: Types::DATETIME_MUTABLE)]
30     private ?\DateTimeInterface $validity = null;
31
32     #[ORM\Column]
33     private ?bool $is_valid = null;
34
35     #[ORM\Column]
36     private ?\DateTimeImmutable $created_at = null;
37
38     #[ORM\ManyToOne(inversedBy: 'coupons')]
39     #[ORM\JoinColumn(nullable: false)]
40     private ?\App\Entity\Categories $category; //<-- here
41
42     /**
43      * @return mixed
44     */
45     public function getDiscount(): ?int
46     {
47         return $this->discount;
48     }
49
50     public function getMaxUsage(): ?int
51     {
52         return $this->max_usage;
53     }
54
55     public function getValidity(): ?\DateTimeInterface
56     {
57         return $this->validity;
58     }
59
60     public function getIsValid(): ?bool
61     {
62         return $this->is_valid;
63     }
64
65     public function getCreatedAt(): ?\DateTimeImmutable
66     {
67         return $this->created_at;
68     }
69
70     public function getCategory(): ?\App\Entity\Categories
71     {
72         return $this->category;
73     }
74
75     public function setCategory(?\App\Entity\Categories $category): self
76     {
77         $this->category = $category;
78
79         return $this;
80     }
81
82     public function __toString(): string
83     {
84         return $this->getCode();
85     }
86
87     public function getCode(): string
88     {
89         return $this->getCode();
90     }
91
92     public function __construct()
93     {
94         $this->category = null;
95     }
96
97     public function __wakeup()
98     {
99         $this->category = null;
100    }
101
102    /**
103     * @return mixed
104     */
105    public function __clone()
106    {
107        $this->category = null;
108    }
109
110    public function __sleep()
111    {
112        $this->category = null;
113    }
114
115    public function __get(string $name)
116    {
117        if ($name === 'category') {
118            return $this->category;
119        }
120
121        return null;
122    }
123
124    public function __set(string $name, mixed $value)
125    {
126        if ($name === 'category') {
127            $this->category = $value;
128        }
129
130        return null;
131    }
132
133    public function __isset(string $name)
134    {
135        if ($name === 'category') {
136            return true;
137        }
138
139        return false;
140    }
141
142    public function __unset(string $name)
143    {
144        if ($name === 'category') {
145            $this->category = null;
146        }
147
148        return null;
149    }
150
151    public function __call(string $name, array $arguments)
152    {
153        if ($name === 'category') {
154            return $this->category;
155        }
156
157        return null;
158    }
159
160    public function __invoke()
161    {
162        return null;
163    }
164
165    public function __toString(): string
166    {
167        return $this->getCode();
168    }
169
170    public function __wakeup()
171    {
172        $this->category = null;
173    }
174
175    public function __sleep()
176    {
177        $this->category = null;
178    }
179
180    public function __get(string $name)
181    {
182        if ($name === 'category') {
183            return $this->category;
184        }
185
186        return null;
187    }
188
189    public function __set(string $name, mixed $value)
190    {
191        if ($name === 'category') {
192            $this->category = $value;
193        }
194
195        return null;
196    }
197
198    public function __isset(string $name)
199    {
200        if ($name === 'category') {
201            return true;
202        }
203
204        return false;
205    }
206
207    public function __unset(string $name)
208    {
209        if ($name === 'category') {
210            $this->category = null;
211        }
212
213        return null;
214    }
215
216    public function __call(string $name, array $arguments)
217    {
218        if ($name === 'category') {
219            return $this->category;
220        }
221
222        return null;
223    }
224
225    public function __invoke()
226    {
227        return null;
228    }
229
230    public function __toString(): string
231    {
232        return $this->getCode();
233    }
234
235    public function __wakeup()
236    {
237        $this->category = null;
238    }
239
240    public function __sleep()
241    {
242        $this->category = null;
243    }
244
245    public function __get(string $name)
246    {
247        if ($name === 'category') {
248            return $this->category;
249        }
250
251        return null;
252    }
253
254    public function __set(string $name, mixed $value)
255    {
256        if ($name === 'category') {
257            $this->category = $value;
258        }
259
260        return null;
261    }
262
263    public function __isset(string $name)
264    {
265        if ($name === 'category') {
266            return true;
267        }
268
269        return false;
270    }
271
272    public function __unset(string $name)
273    {
274        if ($name === 'category') {
275            $this->category = null;
276        }
277
278        return null;
279    }
280
281    public function __call(string $name, array $arguments)
282    {
283        if ($name === 'category') {
284            return $this->category;
285        }
286
287        return null;
288    }
289
290    public function __invoke()
291    {
292        return null;
293    }
294
295    public function __toString(): string
296    {
297        return $this->getCode();
298    }
299
300    public function __wakeup()
301    {
302        $this->category = null;
303    }
304
305    public function __sleep()
306    {
307        $this->category = null;
308    }
309
310    public function __get(string $name)
311    {
312        if ($name === 'category') {
313            return $this->category;
314        }
315
316        return null;
317    }
318
319    public function __set(string $name, mixed $value)
320    {
321        if ($name === 'category') {
322            $this->category = $value;
323        }
324
325        return null;
326    }
327
328    public function __isset(string $name)
329    {
330        if ($name === 'category') {
331            return true;
332        }
333
334        return false;
335    }
336
337    public function __unset(string $name)
338    {
339        if ($name === 'category') {
340            $this->category = null;
341        }
342
343        return null;
344    }
345
346    public function __call(string $name, array $arguments)
347    {
348        if ($name === 'category') {
349            return $this->category;
350        }
351
352        return null;
353    }
354
355    public function __invoke()
356    {
357        return null;
358    }
359
360    public function __toString(): string
361    {
362        return $this->getCode();
363    }
364
365    public function __wakeup()
366    {
367        $this->category = null;
368    }
369
370    public function __sleep()
371    {
372        $this->category = null;
373    }
374
375    public function __get(string $name)
376    {
377        if ($name === 'category') {
378            return $this->category;
379        }
380
381        return null;
382    }
383
384    public function __set(string $name, mixed $value)
385    {
386        if ($name === 'category') {
387            $this->category = $value;
388        }
389
390        return null;
391    }
392
393    public function __isset(string $name)
394    {
395        if ($name === 'category') {
396            return true;
397        }
398
399        return false;
400    }
401
402    public function __unset(string $name)
403    {
404        if ($name === 'category') {
405            $this->category = null;
406        }
407
408        return null;
409    }
410
411    public function __call(string $name, array $arguments)
412    {
413        if ($name === 'category') {
414            return $this->category;
415        }
416
417        return null;
418    }
419
420    public function __invoke()
421    {
422        return null;
423    }
424
425    public function __toString(): string
426    {
427        return $this->getCode();
428    }
429
430    public function __wakeup()
431    {
432        $this->category = null;
433    }
434
435    public function __sleep()
436    {
437        $this->category = null;
438    }
439
440    public function __get(string $name)
441    {
442        if ($name === 'category') {
443            return $this->category;
444        }
445
446        return null;
447    }
448
449    public function __set(string $name, mixed $value)
450    {
451        if ($name === 'category') {
452            $this->category = $value;
453        }
454
455        return null;
456    }
457
458    public function __isset(string $name)
459    {
460        if ($name === 'category') {
461            return true;
462        }
463
464        return false;
465    }
466
467    public function __unset(string $name)
468    {
469        if ($name === 'category') {
470            $this->category = null;
471        }
472
473        return null;
474    }
475
476    public function __call(string $name, array $arguments)
477    {
478        if ($name === 'category') {
479            return $this->category;
480        }
481
482        return null;
483    }
484
485    public function __invoke()
486    {
487        return null;
488    }
489
490    public function __toString(): string
491    {
492        return $this->getCode();
493    }
494
495    public function __wakeup()
496    {
497        $this->category = null;
498    }
499
500    public function __sleep()
501    {
502        $this->category = null;
503    }
504
505    public function __get(string $name)
506    {
507        if ($name === 'category') {
508            return $this->category;
509        }
510
511        return null;
512    }
513
514    public function __set(string $name, mixed $value)
515    {
516        if ($name === 'category') {
517            $this->category = $value;
518        }
519
520        return null;
521    }
522
523    public function __isset(string $name)
524    {
525        if ($name === 'category') {
526            return true;
527        }
528
529        return false;
530    }
531
532    public function __unset(string $name)
533    {
534        if ($name === 'category') {
535            $this->category = null;
536        }
537
538        return null;
539    }
540
541    public function __call(string $name, array $arguments)
542    {
543        if ($name === 'category') {
544            return $this->category;
545        }
546
547        return null;
548    }
549
550    public function __invoke()
551    {
552        return null;
553    }
554
555    public function __toString(): string
556    {
557        return $this->getCode();
558    }
559
560    public function __wakeup()
561    {
562        $this->category = null;
563    }
564
565    public function __sleep()
566    {
567        $this->category = null;
568    }
569
570    public function __get(string $name)
571    {
572        if ($name === 'category') {
573            return $this->category;
574        }
575
576        return null;
577    }
578
579    public function __set(string $name, mixed $value)
580    {
581        if ($name === 'category') {
582            $this->category = $value;
583        }
584
585        return null;
586    }
587
588    public function __isset(string $name)
589    {
590        if ($name === 'category') {
591            return true;
592        }
593
594        return false;
595    }
596
597    public function __unset(string $name)
598    {
599        if ($name === 'category') {
600            $this->category = null;
601        }
602
603        return null;
604    }
605
606    public function __call(string $name, array $arguments)
607    {
608        if ($name === 'category') {
609            return $this->category;
610        }
611
612        return null;
613    }
614
615    public function __invoke()
616    {
617        return null;
618    }
619
620    public function __toString(): string
621    {
622        return $this->getCode();
623    }
624
625    public function __wakeup()
626    {
627        $this->category = null;
628    }
629
630    public function __sleep()
631    {
632        $this->category = null;
633    }
634
635    public function __get(string $name)
636    {
637        if ($name === 'category') {
638            return $this->category;
639        }
640
641        return null;
642    }
643
644    public function __set(string $name, mixed $value)
645    {
646        if ($name === 'category') {
647            $this->category = $value;
648        }
649
650        return null;
651    }
652
653    public function __isset(string $name)
654    {
655        if ($name === 'category') {
656            return true;
657       }
658
659        return false;
660    }
661
662    public function __unset(string $name)
663    {
664        if ($name === 'category') {
665            $this->category = null;
666       }
667
668        return null;
669    }
670
671    public function __call(string $name, array $arguments)
672    {
673        if ($name === 'category') {
674            return $this->category;
675       }
676
677        return null;
678    }
679
680    public function __invoke()
681    {
682        return null;
683    }
684
685    public function __toString(): string
686    {
687        return $this->getCode();
688    }
689
690    public function __wakeup()
691    {
692        $this->category = null;
693    }
694
695    public function __sleep()
696    {
697        $this->category = null;
698    }
699
700    public function __get(string $name)
701    {
702        if ($name === 'category') {
703            return $this->category;
704       }
705
706        return null;
707    }
708
709    public function __set(string $name, mixed $value)
710    {
711        if ($name === 'category') {
712            $this->category = $value;
713       }
714
715        return null;
716    }
717
718    public function __isset(string $name)
719    {
720        if ($name === 'category') {
721            return true;
722       }
723
724        return false;
725    }
726
727    public function __unset(string $name)
728    {
729        if ($name === 'category') {
730            $this->category = null;
731       }
732
733        return null;
734    }
735
736    public function __call(string $name, array $arguments)
737    {
738        if ($name === 'category') {
739            return $this->category;
740       }
741
742        return null;
743    }
744
745    public function __invoke()
746    {
747        return null;
748    }
749
750    public function __toString(): string
751    {
752        return $this->getCode();
753    }
754
755    public function __wakeup()
756    {
757        $this->category = null;
758    }
759
760    public function __sleep()
761    {
762        $this->category = null;
763    }
764
765    public function __get(string $name)
766    {
767        if ($name === 'category') {
768            return $this->category;
769       }
770
771        return null;
772    }
773
774    public function __set(string $name, mixed $value)
775    {
776        if ($name === 'category') {
777            $this->category = $value;
778       }
779
780        return null;
781    }
782
783    public function __isset(string $name)
784    {
785        if ($name === 'category') {
786            return true;
787       }
788
789        return false;
790    }
791
792    public function __unset(string $name)
793    {
794        if ($name === 'category') {
795            $this->category = null;
796       }
797
798        return null;
799    }
800
801    public function __call(string $name, array $arguments)
802    {
803        if ($name === 'category') {
804            return $this->category;
805       }
806
807        return null;
808    }
809
810    public function __invoke()
811    {
812        return null;
813    }
814
815    public function __toString(): string
816    {
817        return $this->getCode();
818    }
819
820    public function __wakeup()
821    {
822        $this->category = null;
823    }
824
825    public function __sleep()
826    {
827        $this->category = null;
828    }
829
830    public function __get(string $name)
831    {
832        if ($name === 'category') {
833            return $this->category;
834       }
835
836        return null;
837    }
838
839    public function __set(string $name, mixed $value)
840    {
841        if ($name === 'category') {
842            $this->category = $value;
843       }
844
845        return null;
846    }
847
848    public function __isset(string $name)
849    {
850        if ($name === 'category') {
851            return true;
852       }
853
854        return false;
855    }
856
857    public function __unset(string $name)
858    {
859        if ($name === 'category') {
860            $this->category = null;
861       }
862
863        return null;
864    }
865
866    public function __call(string $name, array $arguments)
867    {
868        if ($name === 'category') {
869            return $this->category;
870       }
871
872        return null;
873    }
874
875    public function __invoke()
876    {
877        return null;
878    }
879
880    public function __toString(): string
881    {
882        return $this->getCode();
883    }
884
885    public function __wakeup()
886    {
887        $this->category = null;
888    }
889
890    public function __sleep()
891    {
892        $this->category = null;
893    }
894
895    public function __get(string $name)
896    {
897        if ($name === 'category') {
898            return $this->category;
899       }
900
901        return null;
902    }
903
904    public function __set(string $name, mixed $value)
905    {
906        if ($name === 'category') {
907            $this->category = $value;
908       }
909
910        return null;
911    }
912
913    public function __isset(string $name)
914    {
915        if ($name === 'category') {
916            return true;
917       }
918
919        return false;
920    }
921
922    public function __unset(string $name)
923    {
924        if ($name === 'category') {
925            $this->category = null;
926       }
927
928        return null;
929    }
930
931    public function __call(string $name, array $arguments)
932    {
933        if ($name === 'category') {
934            return $this->category;
935       }
936
937        return null;
938    }
939
940    public function __invoke()
941    {
942        return null;
943    }
944
945    public function __toString(): string
946    {
947        return $this->getCode();
948    }
949
950    public function __wakeup()
951    {
952        $this->category = null;
953    }
954
955    public function __sleep()
956    {
957        $this->category = null;
958    }
959
960    public function __get(string $name)
961    {
962        if ($name === 'category') {
963            return $this->category;
964       }
965
966        return null;
967    }
968
969    public function __set(string $name, mixed $value)
970    {
971        if ($name === 'category') {
972            $this->category = $value;
973       }
974
975        return null;
976    }
977
978    public function __isset(string $name)
979    {
980        if ($name === 'category') {
981            return true;
982       }
983
984        return false;
985    }
986
987    public function __unset(string $name)
988    {
989        if ($name === 'category') {
990            $this->category = null;
991       }
992
993        return null;
994    }
995
996    public function __call(string $name, array $arguments)
997    {
998        if ($name === 'category') {
999            return $this->category;
1000       }
1001
1002      return null;
1003  }
1004
1005  public function __invoke()
1006  {
1007      return null;
1008  }
1009
1010  public function __toString(): string
1011  {
1012      return $this->getCode();
1013  }
1014
1015  public function __wakeup()
1016  {
1017      $this->category = null;
1018  }
1019
1020  public function __sleep()
1021  {
1022      $this->category = null;
1023  }
1024
1025  public function __get(string $name)
1026  {
1027      if ($name === 'category') {
1028          return $this->category;
1029     }
1030
1031      return null;
1032  }
1033
1034  public function __set(string $name, mixed $value)
1035  {
1036      if ($name === 'category') {
1037          $this->category = $value;
1038     }
1039
1040      return null;
1041  }
1042
1043  public function __isset(string $name)
1044  {
1045      if ($name === 'category') {
1046          return true;
1047     }
1048
1049      return false;
1050  }
1051
1052  public function __unset(string $name)
1053  {
1054      if ($name === 'category') {
1055          $this->category = null;
1056     }
1057
1058      return null;
1059  }
1060
1061  public function __call(string $name, array $arguments)
1062  {
1063      if ($name === 'category') {
1064          return $this->category;
1065     }
1066
1067      return null;
1068  }
1069
1070  public function __invoke()
1071  {
1072      return null;
1073  }
1074
1075  public function __toString(): string
1076  {
1077      return $this->getCode();
1078  }
1079
1080  public function __wakeup()
1081  {
1082      $this->category = null;
1083  }
1084
1085  public function __sleep()
1086  {
1087      $this->category = null;
1088  }
1089
1090  public function __get(string $name)
1091  {
1092      if ($name === 'category') {
1093          return $this->category;
1094     }
1095
1096      return null;
1097  }
1098
1099  public function __set(string $name, mixed $value)
1100  {
1101      if ($name === 'category') {
1102          $this->category = $value;
1103     }
1104
1105      return null;
1106  }
1107
1108  public function __isset(string $name)
1109  {
1110      if ($name === 'category') {
1111          return true;
1112     }
1113
1114      return false;
1115  }
1116
1117  public function __unset(string $name)
1118  {
1119      if ($name === 'category') {
1120          $this->category = null;
1121     }
1122
1123      return null;
1124  }
1125
1126  public function __call(string $name, array $arguments)
1127  {
1128      if ($name === 'category') {
1129          return $this->category;
1130     }
1131
1132      return null;
1133  }
1134
1135  public function __invoke()
1136  {
1137      return null;
1138  }
1139
1140  public function __toString(): string
1141  {
1142      return $this->getCode();
1143  }
1144
1145  public function __wakeup()
1146  {
1147      $this->category = null;
1148  }
1149
1150  public function __sleep()
1151  {
1152      $this->category = null;
1153  }
1154
1155  public function __get(string $name)
1156  {
1157      if ($name === 'category') {
1158          return $this->category;
1159     }
1160
1161      return null;
1162  }
1163
1164  public function __set(string $name, mixed $value)
1165  {
1166      if ($name === 'category') {
1167          $this->category = $value;
1168     }
1169
1170      return null;
1171  }
1172
1173  public function __isset(string $name)
1174  {
1175      if ($name === 'category') {
1176          return true;
1177     }
1178
1179      return false;
1180  }
1181
1182  public function __unset(string $name)
1183  {
1184      if ($name === 'category') {
1185          $this->category = null;
1186     }
1187
1188      return null;
1189  }
1190
1191  public function __call(string $name, array $arguments)
1192  {
1193      if ($name === 'category') {
1194          return $this->category;
1195     }
1196
1197      return null;
1198  }
1199
1200  public function __invoke()
1201  {
1202      return null;
1203  }
1204
1205  public function __toString(): string
1206  {
1207      return $this->getCode();
1208  }
1209
1210  public function __wakeup()
1211  {
1212      $this->category = null;
1213  }
1214
1215  public function __sleep()
1216  {
1217      $this->category = null;
1218  }
1219
1220  public function __get(string $name)
1221  {
1222      if ($name === 'category') {
1223          return $this->category;
1224     }
1225
1226      return null;
1227  }
1228
1229  public function __set(string $name, mixed $value)
1230  {
1231      if ($name === 'category') {
1232          $this->category = $value;
1233     }
1234
1235      return null;
1236  }
1237
1238  public function __isset(string $name)
1239  {
1240      if ($name === 'category') {
1241          return true;
1242     }
1243
1244      return false;
1245  }
1246
1247  public function __unset(string $name)
1248  {
1249      if ($name === 'category') {
1250          $this->category = null;
1251     }
1252
1253      return null;
1254  }
1255
1256  public function __call(string $name, array $arguments)
1257  {
1258      if ($name === 'category') {
1259          return $this->category;
1260     }
1261
1262      return null;
1263  }
1264
1265  public function __invoke()
1266  {
1267      return null;
1268  }
1269
1270  public function __toString(): string
1271  {
1272      return $this->getCode();
1273  }
1274
1275  public function __wakeup()
1276  {
1277      $this->category = null;
1278  }
1279
1280  public function __sleep()
1281  {
1282      $this->category = null;
1283  }
1284
1285  public function __get(string $name)
1286  {
1287      if ($name === 'category') {
1288          return $this->category;
1289     }
1290
1291      return null;
1292  }
1293
1294  public function __set(string $name, mixed $value)
1295  {
1296      if ($name === 'category') {
1297          $this->category = $value;
1298     }
1299
1300      return null;
1301  }
1302
1303  public function __isset(string $name)
1304  {
1305      if ($name === 'category') {
1306          return true;
1307     }
1308
1309      return false;
1310  }
1311
1312  public function __unset(string $name)
1313  {
1314      if ($name === 'category') {
1315          $this->category = null;
1316     }
1317
1318      return null;
1319  }
1320
1321  public function __call(string $name, array $arguments)
1322  {
1323      if ($name === 'category') {
1324          return $this->category;
1325     }
1326
1327      return null;
1328  }
1329
1330  public function __invoke()
1331  {
1332      return null;
1333  }
1334
1335  public function __toString(): string
1336  {
1337      return $this->getCode();
1338  }
1339
1340  public function __wakeup()
1341  {
1342      $this->category = null;
1343  }
1344
1345  public function __sleep()
1346  {
1347      $this->category = null;
1348  }
1349
1350  public function __get(string $name)
1351  {
1352      if ($name === 'category') {
1353          return $this->category;
1354     }
1355
1356      return null;
1357  }
1358
1359  public function __set(string $name, mixed $value)
1360  {
1361      if ($name === 'category') {
1362          $this->category = $value;
1363     }
1364
1365      return null;
1366  }
1367
1368  public function __isset(string $name)
1369  {
1370      if ($name === 'category') {
1371          return true;
1372     }
1373
1374      return false;
1375  }
1376
1377  public function __unset(string $name)
1378  {
1379      if ($name === 'category
```

## Créer l'entité Images:

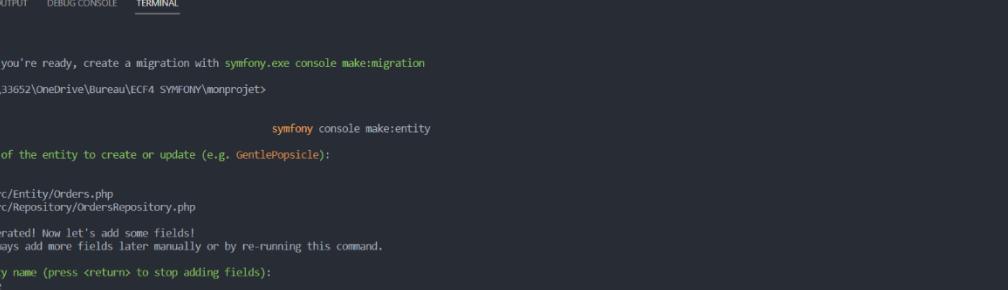
Reprendre les mêmes étapes que l'entité précédente

- Est-ce qu'elle peut etre nul : non
  - Je peux aller chercher les images d'un produit : oui
  - Nom : images
  - Est-ce qu'une image peut changer de produit : non
  - Est-ce que si je supprime le produit je supprime les images : oui

```
File Edit Selection View Go Run Terminal Help Coupons.php - ECF4 SYMFONY - Visual Studio Code .env.local Users.php U Categories.php U Coupons.php U monprojet > src > Entity > Coupons.php > ...  
24  
25     #[ORM\Column]  
26     private ?int $discount = null;  
27  
28     #[ORM\Column]  
29     private ?int $max_usage = null;  
30  
31     #[ORM\Column(type: Types::DATETIME_MUTABLE)]  
32     private ?\DateTimeInterface $validity = null;  
33  
34     #[ORM\Column]  
35     private ?bool $is_valid = null;  
36  
37     #[ORM\ManyToMany(inversedBy: 'coupons')]  
38     #[ORM\JoinColumn(nullable: false)]  
39     private ?\Doctrine\Common\Collections\ArrayCollection $coupons = null;  
40  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL  
Do you want to automatically delete orphaned App\Entity\Images objects (orphanRemoval)? (yes/no) [no]:  
> yes  
updated: src/Entity/Images.php  
updated: src/Entity/Products.php  
Add another property? Enter the property name (or press <return> to stop adding fields):  
>  
  
Success!  
  
Next: When you're ready, create a migration with symfony.exe console make:migration  
PS C:\Users\33652\OneDrive\Bureau\ECF4 SYMFONY\monprojet>
```

## Créer l'entité Orders:

Reprendre les mêmes étapes que l'entité précédente



File Edit Selection View Go Run Terminal Help Coupons.php - ECF4 SYMFONY - Visual Studio Code

.env.local Users.php U Categories.php U Coupons.php X

monprojet > src > Entity > Coupons.php > Coupons

24 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Next: when you're ready, create a migration with `symfony console make:migration`

PS C:\Users\33652\OneDrive\Bureau\ECF4 SYMFONY\monprojet>

```
symfony console make:entity

Class name of the entity to create or update (e.g. GentlePopsicle):
> Orders

created: src/Entity/Orders.php
created: src/Repository/OrdersRepository.php

Entity generated! Now let's add some fields!
You can always add more fields later manually or by re-running this command.

New property name (press <return> to stop adding fields):
> reference

Field type (enter ? to see all types) [string]:
>

Field length [255]:
> 20

Can this field be null in the database (nullable) (yes/no) [no]:
>

updated: src/Entity/orders.php

Add another property? Enter the property name (or press <return> to stop adding fields):
> created_at

Field type (enter ? to see all types) [datetime_immutable]:
>

Can this field be null in the database (nullable) (yes/no) [no]:
>
```

Ln 37, Col 18 (97 selected) Spaces: 4 UTF-8 LF PHP Go Live

36°C Ensoleillé

Rechercher

15:54 19/07/2023

Coupons.php - EC14 SYMFONY - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

> relation

What class should this entity be related to?:  
> Coupons

What type of relationship is this?

Type	Description
ManyToOne	Each Orders relates to (has) one Coupons. Each Coupons can relate to (can have) many Orders objects.
OneToMany	Each Orders can relate to (can have) many Coupons objects. Each Coupons relates to (has) one Orders.
ManyToMany	Each Orders can relate to (can have) many Coupons objects. Each Coupons can also relate to (can also have) many Orders objects.
OneToOne	Each Orders relates to (has) exactly one Coupons. Each Coupons also relates to (has) exactly one Orders.

Relation type? [ManyToOne, OneToMany, ManyToMany, OneToOne]:  
> ManyToOne

Is the Orders.coupons property allowed to be null (nullable)? (yes/no) [yes]:  
>

Coupons.php - EC14 SYMFONY - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

> relation

What class should this entity be related to?:  
> Users

What type of relationship is this?

Type	Description
ManyToOne	Each Orders relates to (has) one Users. Each Users can relate to (can have) many Orders objects.
OneToMany	Each Orders can relate to (can have) many Users objects. Each Users relates to (has) one Orders.
ManyToMany	Each Orders can relate to (can have) many Users objects. Each Users can also relate to (can also have) many Orders objects.
OneToOne	Each Orders relates to (has) exactly one Users. Each Users also relates to (has) exactly one Orders.

Relation type? [ManyToOne, OneToMany, ManyToMany, OneToOne]:  
> ManyToOne

Is the Orders.users property allowed to be null (nullable)? (yes/no) [yes]:  
> no

Do you want to add a new property to Users so that you can access/update Orders objects from it - e.g. \$users->getOrders()? (yes/no) [yes]:  
>

## Créer l'entité Orders\_details:

```
File Edit Selection View Go Run Terminal Help
Orders.php - ECF4 SYMFONY - Visual Studio Code
.env.local Users.php U Categories.php U Orders.php U Coupons.php U Products.php U
monprojet > src > Entity > Orders.php > Orders
1 <?php
2
3 namespace App\Entity;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\33652\OneDrive\Bureau\ECF4 SYMFONY\monprojet> symfony console make:entity
Class name of the entity to create or update (e.g. GrumpyPizza):
> OrdersDetails
created: src/Entity/OrdersDetails.php
created: src/Repository/OrdersDetailRepository.php
Entity generated! Now let's add some fields!
You can always add more fields later manually or by re-running this command.
New property name (press <return> to stop adding fields):
> quantity
Field type (enter ? to see all types) [string]:
> integer
Can this field be null in the database (nullable) (yes/no) [no]:
>
updated: src/Entity/OrdersDetails.php
> price
Field type (enter ? to see all types) [string]:
> integer
Can this field be null in the database (nullable) (yes/no) [no]:
>
updated: src/Entity/OrdersDetails.php
Add another property? Enter the property name (or press <return> to stop adding fields):
> 
Ln 16, Col 44 Spaces: 4 UTF-8 LF PHP Go Live
35°C Ensoleillé Rechercher DELL Mozilla Firefox A3 W 17:41 19/07/2023
```

On ajoute les deux clés primaire qui seront : **orders\_id** et **products\_id**

```
File Edit Selection View Go Run Terminal Help
OrdersDetails.php - ECF4 SYMFONY - Visual Studio Code
.env.local Users.php U Categories.php U Orders.php U OrdersDetails.php U Coupons.php U Products.php U
monprojet > src > Entity > OrdersDetails.php > OrdersDetails > setQuantity
2
3 #[ORM\Column]
4 private ?int $price;
5
6 #[ORM\ManyToOne(inversedBy: "ordersdetails")]
7
8
9 OrdersDetails objects from it - e.g. $products->getOrdersDetails()?
(yes/no) [yes]:
acls:
>
Do you want to activate orphanRemoval on your relationship?
A OrdersDetails is "orphaned" when it is removed from its related Products.
e.g. $products->removeOrdersDetails($ordersDetails)
NOTE: If a OrdersDetails may *change* from one Products to another, answer "no".
Do you want to automatically delete orphaned App\Entity\OrdersDetails objects (orphanRemoval)?
(yes/no) [no]:
updated: src/Entity/OrdersDetails.php
updated: src/Entity/Products.php
Add another property? Enter the property name (or press <return> to stop adding fields):
>
Next: When you're ready, create a migration with symfony.exe console make:migration
PS C:\Users\33652\OneDrive\Bureau\ECF4 SYMFONY\monprojet> 

Ln 37, Col 22 Spaces: 4 UTF-8 LF PHP Go Live
35°C Ensoleillé Rechercher DELL Mozilla Firefox A3 W 17:56 19/07/2023
```

Faire la migration :

Taper la commande : **symfony console make:migration** et une fois réussi on met la commande : **symfony.exe console doctrine:migrations:migrate** et on aura le résultat suivant :

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "ECF4 SYMFONY".
- Terminal:**

```
[ERROR] Migration cancelled!
PS C:\Users\33652\OneDrive\Bureau\ECF4 SYMFONY\monprojet> #[ORM\Id]
PS C:\Users\33652\OneDrive\Bureau\ECF4 SYMFONY\monprojet> symfony console make:migration
created: migrations/Version20230720141529.php
success!
```
- Status Bar:** Shows the current file is "OrdersDetails.php - ECF4 SYMFONY - Visual Studio Code". Other files listed include Categories.php, Orders.php, OrdersDetails.php, Entity U, Coupons.php, Images.php, and Products.php.
- Bottom Bar:** Shows the date and time as 20/07/2023, 16:17.

La base de donnée est bien affichée sur mon phpmyadmin :

The screenshot shows the phpMyAdmin interface with the following details:

- Left Panel:** Shows the database structure of "mabddsymfony" with tables: categories, coupons, coupons\_types, doctrine\_migration\_versions, images, messenger\_messages, orders, orders\_details, products, and users.
- Right Panel:** Shows the list of tables in the "mabddsymfony" database, each with actions like Parcourir, Structure, Rechercher, Insérer, Vider, and Supprimer.
- Bottom Panel:** Shows the SQL query console with the text "Console de requêtes SQL de données".
- Bottom Bar:** Shows the date and time as 20/07/2023, 16:31.

## Vidéo 4 : Mise en place des templates et des assets CSS et JS :

Démarrer le server avec la commande : [symfony serve -d](#)

The screenshot shows the Visual Studio Code interface. The top navigation bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and tabs for various files like Users.php, OrdersDetails.php, Categories.php, Orders.php, OrdersDetails.php, Entity, composer.json, Coupons.php, and Products.php. The left sidebar has sections for Explorer, Search, Problems, Output, Debug Console, and Terminal. The main area displays the contents of the composer.json file:

```

{
    "name": "mabddsymfony",
    "version": "0.0.1",
    "description": "A simple Symfony project for learning.",
    "require": {
        "php": ">=7.4.0",
        "symfony/framework-bundle": "6.0.*",
        "symfony/validator": "6.0.*",
        "symfony/web-link": "6.0.*",
        "symfony/yaml": "6.0.*",
        "twig/extras-bundle": "2.12|^3.0",
        "twig/twig": "2.12|^3.0"
    },
    "extra": {
        "branch": "main"
    }
}

```

The terminal window at the bottom shows the following command execution:

```

PS C:\Users\j33652\OneDrive\Bureau\ECF4 SYMFONY\monprojet> symfony console doctrine:migrations:migrate
WARNING! You are about to execute a migration in database "mabddsymfony" that could result in schema changes and data loss. Are you sure you wish to continue? (yes/no) [yes]:
>

[notice] Migrating up to DoctrineMigrations\Version20230720141529
[notice] Finished in 1501.1ms, used 22M memory, 1 migrations executed, 17 sql queries
PS C:\Users\j33652\OneDrive\Bureau\ECF4 SYMFONY\monprojet> fwi-i-search:
PS C:\Users\j33652\OneDrive\Bureau\ECF4 SYMFONY\monprojet> symfony serve -d
[WARNING] The local web server is optimized for local development and MUST
never be used in a production setup.

[OK] web server listening
The web server is using PHP CGI 8.0.28
https://127.0.0.1:8000

```

Below the terminal, a message says "Stream the logs via symfony.exe server:log". The status bar at the bottom right shows "17:27 20/07/2023".

Par la suite créer un contrôleur dans notre projet dans le dossier **src/controller** et générer le fichier **main** dans le dossier **Templates** et qui contient un fichier **index**.

The screenshot shows the Visual Studio Code interface again. The top navigation bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and tabs for various files like MainController.php, Categories.php, Orders.php, OrdersDetails.php, Entity, Coupons.php, and Products.php. The left sidebar has sections for Explorer, Search, Problems, Output, Debug Console, and Terminal. The main area shows the code for MainController.php:

```

namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class MainController extends AbstractController
{
    #[Route('/main', name: 'app_main')]
    public function index(): Response
    {
        return $this->render('main/index.html.twig');
    }
}

```

The terminal window at the bottom shows the command "symfony console make:controller MainController" being run, followed by "Success!" and a note to open the new controller class.

The status bar at the bottom right shows "19:02 20/07/2023".

**Maincontroller** contient par défaut les controllers dont symfony étendent une classe abstraite qui s'appelle **abstractController** qui va nous permettre d'avoir un certain nombre de services qui sera implémenté par défaut comme les services sécurité, gestion twig... qui seront automatiquement injectés. C'est à abstractController de tout faire...

On utilise aussi les class **Response**; et **Route**; qui va nous permettre de gérer la partie des annotations (attributs) et qui nous montre la route (l'url) pour accéder au site.

Le fichier **twig** est appelé par notre **controller** et affiche le navigateur.

Ce fichier contient le DOCTYPE, une balise html, une balise head et body.

Dans le fichier **base.html.twig** nous allons définir la structure de base html de notre projet

- Vider le bloc **stylesheets** car c'est une entrée de **webpack encore**.
- Vider le bloc **javascript** pour la même raison

Chaque bloc va être ouvert par `{% block stylesheets %}` dans le cas de `stylesheets` et sera fermé par `{% endblock %}`

Cette ligne `{% extends 'base.html.twig' %}`

Indique que mon fichier index est un fichier qui va utiliser les blocs et étendre **base.html.twig** (le prendre en entier et à chaque fois qu'il trouve un bloc du même nom, il remplace le contenu par ce qu'on lui a demandé).