

Documentation technique Ô' Tako

Sommaire :

- I. Deployment
- II. Organisation du code
- III. Organisation de la base de données
- IV. Organisation du web service

Deployment :

Récupération du projet :

```
git clone https://github.com/charles66820/PPE3.git
```

ou décompresser l'archive.

Avec Docker :

prérequis :

- Une connexion internet
- Docker
- Docker-compose

Ouvrez un terminal et taper les commande suivant :

```
cd PPE3
docker-compose up
#CTRL + C
docker-compose start
docker-compose exec web /bin/bash
```

En mode test :

prérequis :

- Une connexion internet
- Un serveur SGBD mysql 5.7.

Avec apache :

prérequis :

- Une connexion internet
- Un serveur SGBD mysql 5.7.
- Apache2
- PHP 7.2
- Composer

Ajouter un virtual host :

```
<VirtualHost *:80>
    ServerName localhost
    ServerAdmin charles@magicorp.fr
    DocumentRoot /var/www/html/public
    <Directory /var/www/html/public>
        AllowOverride All
        Order Allow,Deny
        Allow from All
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Dans tous les cas :

Dans un terminal rendez-vous dans le dossier du projet :

```
cd PPE3
```

Installation des dépendances :

```
composer install
```

Dans le fichier ".env" modifier la connexion à base de données (db pour Docker sinon 127.0.0.1):

```
# dans le fichier .env

#important
TRUSTED_HOSTS=localhost

DATABASE_URL="mysql://root:123456@db:3306/PPE3"

MAILER_URL=smtp://localhost

# paypal
PAYPALCLIENT=
PAYPALSECRET=

ACCESSTOKEN=rainbow
```

En mode test et avec apache on crée la base de données :

```
php bin/console doctrine:database:create
```

Puis effectuer la migration du schéma de la base de données :

```
php bin/console doctrine:migrations:migrate
```

Avec docker lancer le server avec la commande suivant :

```
docker-compose start
```

En mode test lancer le server avec la commande suivant :

```
php bin/console server:start
```

Les fixtures :

Charger les fixtures de test dans la base de données :

```
php -d memory_limit=1G bin/console doctrine:fixtures:load
--group=AppFixtures
```

Commande à faire avant de faire les autre fixture :

```
cd tests/
unzip ppe2Bdd.zip
mv dbData.json ../src/DataFixtures/
mv imgs/* ../public/img/products/
rmdir imgs/
cd ..
```

Charger les fixtures de de donnée dans la base de données :

```
php -d memory_limit=1G bin/console doctrine:fixtures:load  
--group=DataFixtures
```

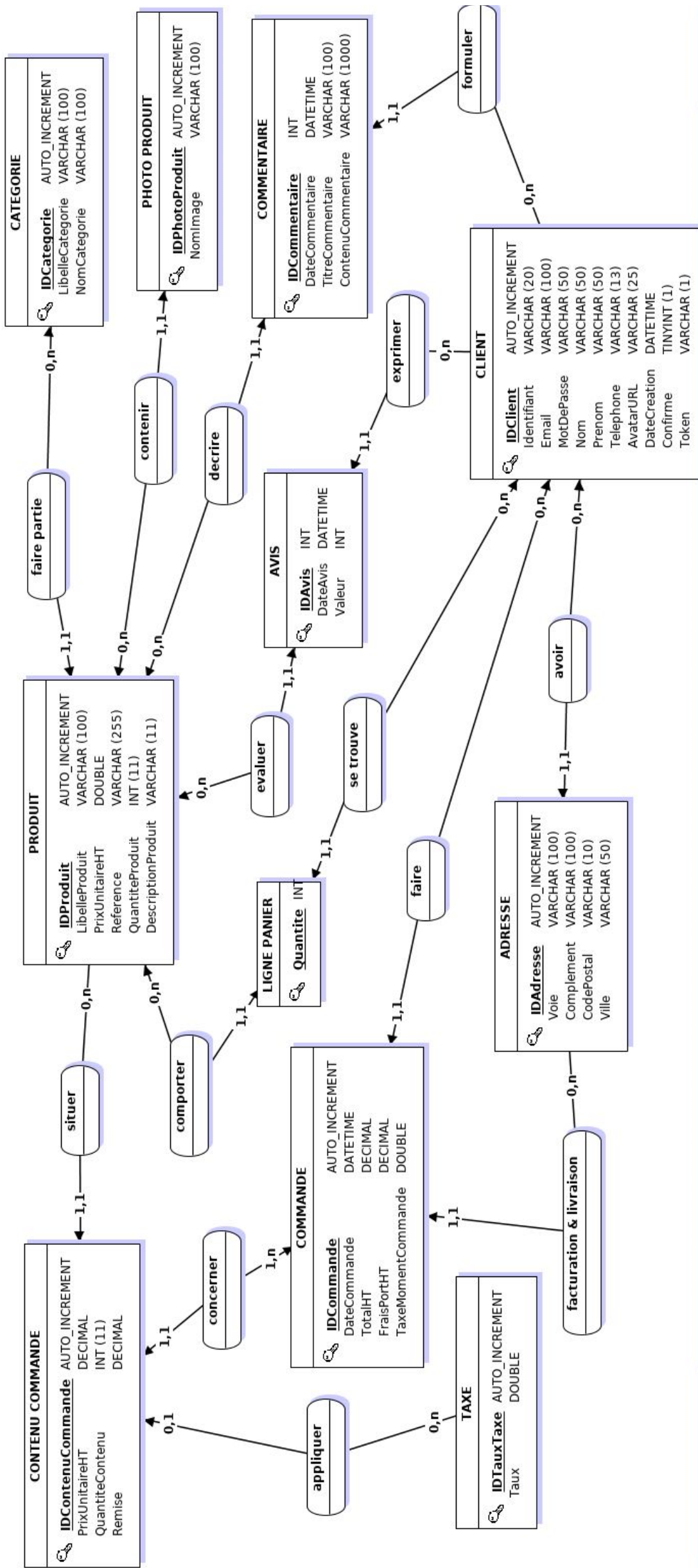
Charger toute les fixtures dans la base de données :

```
php -d memory_limit=1G bin/console doctrine:fixtures:load
```

Organisation du code :

Respect de l'organisation de symfony 4.

Organisation de la base de données :



Organisation du web service:

Le web service respect les bonne pratique comme le fait qu'il soit intégralement en anglais.

les route du web services sont :

Host : api.ppe3.magicorp.fr

route	Method	description
/products	GET	Récupère tous les produits
/products/{id}	GET	Fiche du produit
/products/{id}/comments	GET	Récupère tous les commentaire d'un produit
/catalog	GET	Récupère toute les catégories contenant des produit produits
/categories	GET	Récupère toute les catégories
/customer/{identfient}	GET	Info d'un client grâce à l'identifiant
/customer/{id}/addresses	GET	Récupère les adresses d'un client
/customer/{id}/orders	GET	Récupère les commandes du client
/orders/{id}	GET	Récupère le détail d'une commandes
Barre code		
/products/barcode/{code}	GET	Fiche du produit avec le code barres
/products/{id}/quantity	PATCH	Modifie la quantité