



Travaux Pratiques : Enoncés

TP_ReviewPhoto_Partie_02

Objectif : Consolidation sur le framework Symfony 6
Création de la base de données des photos

On va créer la base de données « review_photo_sf6 » dans mySql avec une première table : celle des photos.

Réalisations :

1 : Modifier le fichier « .env » du projet pour paramétrer la base :

Afin de dire à Doctrine qui est l'ORM utilisé par Symfony pour dialoguer avec la base de données, quelle est la base et quel utilisateur utiliser pour la connexion, il faut modifier le paramètre DATABASE_URL du fichier « .env » se trouvant à la racine du projet.

`DATABASE_URL="mysql://root@127.0.0.1:3306/review_photo_sf6?serverVersion=8& charset=utf8mb4"`

2 : Créer la base en ligne de commande, entrer :

symfony console doctrine:database:create
ou
php bin/console doctrine:database:create

Ceci ne fait que de créer la base dans mySql, il n'y a pas de table.

3 : Vérifier par phpMyAdmin que la base est bien créée

4 : Création de la table « photo »

Attention : lancer le serveur MySql sinon les commandes ci-dessous vont échouer.

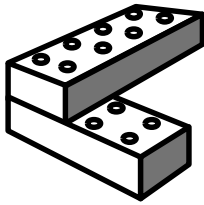
Le fait d'avoir créé l'entité « Photo » lors de la partie_01, des annotations ont été placées dans la classe. On va pouvoir demander à doctrine de créer la table.

- Entrer les commandes :

symfony console make:migration ou php bin/console make:migration

- Ceci prépare le travail, sous forme d'instructions SQL, des modifications à apporter dans la base
- Vérifier dans le dossier « Migrations » l'instruction sql se trouvant dans la méthode « up() » du fichier qui vient d'être créé

symfony console doctrine:migrations:migrate
ou php bin/console doctrine:migrations:migrate



Travaux Pratiques : Enoncés TP_ReviewPhoto_Partie_02

- Ceci applique les modifications
- Vérifier que la table « photo » a bien été créée

5 : Créer 1 ou 2 enregistrements dans cette table par phpMyAdmin

6 : Modification de la méthode « list() » du contrôleur « PhotoController »

Afin d'afficher réellement les informations des photos présentes en base, il faut modifier la méthode « list() » du contrôleur « PhotoController » :

- Injecter en paramètre de la méthode le « repository » correspondant à notre entité « Photo »

```
public function list(PhotoRepository $photoRepository) {...}
```

normalement VSC ajoute automatiquement le
« use App\Repository\PhotoRepository » en tête du fichier.

- Supprimer le contenu de la méthode en conservant l'instruction « return »
- Affecter à la variable « \$photos » le retour de la méthode « findAll() » de l'objet « \$PhotoRepository »

```
$photos = $photoRepository->findAll();
```

7 : Rafraichissez votre navigateur

Normalement les photos de la base de données doivent être affichées.

8 : création de jeux d'essais : utilisation des « Fixtures »

Nous allons demander à Doctrine de créer des jeux d'essai, 5 ou 6 photos.
Pour cela nous allons utiliser la notion de « Fixtures » mais il nous faut installer le package de cette nouvelle classe qui n'est pas chargée par défaut lors de la création du projet.

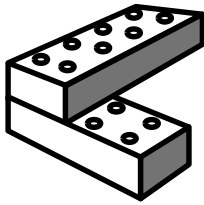
- Installation de la librairie php des fixtures pour le développement
 - Entrer la commande :

```
composer require orm-fixtures --dev      attention c'est deux -
```

- Une fois installée, entrer la commande :

```
symfony console make:fixtures
```

Entrer le nom de la classe des fixtures : PhotoFixtures



Travaux Pratiques : Enoncés

TP_ReviewPhoto_Partie_02

- **Modification de la classe ainsi créée :**

Nous allons faire une boucle pour créer 5 photos, chaque boucle créer une nouvelle instance de « Photo », fait appel aux « setteurs » pour valoriser les propriétés et persiste cette photo. A la sortie de la boucle, reste à demander à doctrine de persister ces photos en base grâce au manager.

Remplacez le code par le code suivant :

(n'oubliez pas d'ajouter le « use App\Entity\Photo » en tête du fichier car cette méthode instancie la classe « Photo »)

```
public function load(ObjectManager $manager)
{
    for ($i = 1; $i <= 5; $i++){

        $photo = new Photo();
        $photo->setTitle('Photo numéro '.$i);
        $photo->setPostAt(
            (new \DateTimeImmutable())->add(\DateInterval::createFromDateString('-'.
            '$i.' week')));
        $manager->persist($photo);
    }

    $manager->flush();    // demande à doctrine de créer l'ordre sql INSERT
}
```

- **Exécution des fixtures, entrer la commande :**

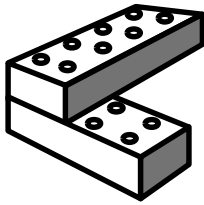
```
symfony console doctrine:fixtures:load
```

7 : Démarrer le serveur web en ligne de commande par si vous l'avez arrêté :

```
symfony server:start
```

8 : ouvrir votre navigateur et entrer l'url :

http://localhost:8000



Travaux Pratiques : Enoncés

TP_ReviewPhoto_Partie_02

