

Fichier Création et Alimentation BDD

1. Les Migrations:

J'ai utilisé les commandes `make:entity` pour créer les entités, et la commande `make:migrations`, pour créer les migrations vers la base de données, se basant sur les valeurs des entités.

Voici un exemple complet de fichier migrations, demandé dans l'énoncé, concernant les horaires :

```
1  <?php
2
3  declare(strict_types=1);
4
5  namespace DoctrineMigrations;
6
7  use Doctrine\DBAL\Schema\Schema;
8  use Doctrine\Migrations\AbstractMigration;
9
10 /**
11  * Auto-generated Migration: Please modify to your needs!
12  */
13 final class Version20230914150045 extends AbstractMigration
14 {
15     public function getDescription(): string
16     {
17         return '';
18     }
19
20     public function up(Schema $schema): void
21     {
22         // this up() migration is auto-generated, please modify it to your needs
23         $this->addSql('CREATE TABLE horaires_garage (id INT AUTO_INCREMENT NOT NULL,
24 jour VARCHAR(255) NOT NULL,
25 ouv_am INT DEFAULT NULL,
26 fer_am INT DEFAULT NULL,
27 ouv_pm INT DEFAULT NULL,
28 fer_pm INT DEFAULT NULL,
29 PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4_unicode_ci` ENGINE = InnoDB');
30
31     }
32
33     public function down(Schema $schema): void
34     {
35         // this down() migration is auto-generated, please modify it to your needs
36         $this->addSql('DROP TABLE horaires_garage');
37     }
38 }
39
```

Les fichiers migrations contiennent 3 fonctions:

La première servant à récupérer les informations de la migration;

La deuxième, servant à effectuer les requetes SQL adequat pour push les informations en base de données,

La troisième, pour supprimer la table de la base de données.

Voici ci dessous les autres migrations, se basant exclusivement sur la fonction de push en base de données(Les migrations sont accessibles sur GitHub.) :

Les Services (Principaux et Secondaires) :

```
// this up() migration is auto-generated, please modify it to your needs
$this->addSql('CREATE TABLE services_principaux (id INT AUTO_INCREMENT NOT NULL,
titre VARCHAR(255) NOT NULL,
description LONGTEXT NOT NULL,
prix INT NOT NULL,
PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4_unicode_ci` ENGINE = InnoDB');

$this->addSql('CREATE TABLE services_secondaires (id INT AUTO_INCREMENT NOT NULL,
titre VARCHAR(255) NOT NULL,
PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4_unicode_ci` ENGINE = InnoDB');
```

Les Avis Utilisateurs (En attente et validés par le modérateur)

```
$this->addSql('CREATE TABLE avis_attente (id INT AUTO_INCREMENT NOT NULL,
titre VARCHAR(255) NOT NULL,
nom VARCHAR(255) NOT NULL,
commentaire LONGTEXT NOT NULL,
email VARCHAR(255) NOT NULL,
date DATE NOT NULL,
note INT NOT NULL,
PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4_unicode_ci` ENGINE = InnoDB');

$this->addSql('CREATE TABLE avis_utilisateur (id INT AUTO_INCREMENT NOT NULL,
titre VARCHAR(255) NOT NULL,
nom VARCHAR(255) NOT NULL,
commentaire LONGTEXT NOT NULL,
email VARCHAR(255) NOT NULL,
date_ajout DATE NOT NULL,
note INT NOT NULL,
affichage TINYINT(1) NOT NULL,
PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4_unicode_ci` ENGINE = InnoDB');
```

Migrations véhicules (Relation OneToOne entre Entity/Vehicules & Entity/InfosVehicules) :

```
// this up() migration is auto-generated, please modify it to your needs
$this->addSql('CREATE TABLE info_vehicules (id INT AUTO_INCREMENT NOT NULL,
id_vehicule_id INT NOT NULL,
nbportes INT NOT NULL,
nb_sieges INT NOT NULL,
abs TINYINT(1) NOT NULL,
roue_motrices INT NOT NULL,
dir_assistee TINYINT(1) NOT NULL,
esp TINYINT(1) NOT NULL,
ab_av TINYINT(1) NOT NULL,
ab_ar TINYINT(1) NOT NULL,
wifi TINYINT(1) NOT NULL,
bluetooth TINYINT(1) NOT NULL,
reg_vitesse TINYINT(1) NOT NULL,
lim_vitesse TINYINT(1) NOT NULL,
correc_traj TINYINT(1) NOT NULL,
aide_parking TINYINT(1) NOT NULL,
prop_trac VARCHAR(255) NOT NULL,
avis_vendeur LONGTEXT DEFAULT NULL,
UNIQUE INDEX UNIQ_D55131105258F8E6 (id_vehicule_id),
PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4_unicode_ci` ENGINE = InnoDB');

$this->addSql('CREATE TABLE vehicules (id INT AUTO_INCREMENT NOT NULL,
marque VARCHAR(255) NOT NULL,
modele VARCHAR(255) NOT NULL,
chevaux INT NOT NULL,
annee DATE NOT NULL,
kilometres INT NOT NULL,
type_carburant VARCHAR(255) NOT NULL,
date_ajout DATE NOT NULL,
prix INT NOT NULL,
PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4_unicode_ci` ENGINE = InnoDB');

$this->addSql('ALTER TABLE info_vehicules ADD CONSTRAINT FK_D55131105258F8E6 FOREIGN KEY (id_vehicule_id) REFERENCES vehicules (id)');
```

Migrations utilisateurs :

```
// this up() migration is auto-generated, please modify it to your needs
$this->addSql('CREATE TABLE utilisateur (id INT AUTO_INCREMENT NOT NULL,
email VARCHAR(180) NOT NULL,
roles LONGTEXT NOT NULL COMMENT \'(DC2Type:json)\',
password VARCHAR(255) NOT NULL,
UNIQUE INDEX UNIQ_1D1C63B3E7927C74 (email),
PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4_unicode_ci` ENGINE = InnoDB');
```

2. Les Fixtures:

Les fixtures servent à ajouter des éléments dans la base de données, sans passer par les options graphiques, ou par les fonctions de l'application.

Elles peuvent servir de valeurs de test ou de valeurs par défaut pour l'application.

Création des Horaires :

```
1  <?php
2
3  namespace App\DataFixtures;
4
5  use Doctrine\Bundle\FixturesBundle\Fixture;
6  use Doctrine\Persistence\ObjectManager;
7  use App\Entity\HorairesGarage;
8
9  class Horaires extends Fixture
10 {
11     public function load(ObjectManager $manager): void
12     {
13
14         for ($i = 0; $i < 7; $i++) {
15             $horaire[$i] = new HorairesGarage();
16             $jours = ['Lundi', 'Mardi', 'Mercredi', 'Jeudi', 'Vendredi', 'Samedi', 'Dimanche'];
17
18             $horaire[$i]->setJour($jours[$i]);
19
20             if ($jours[$i] !== 'Samedi' && $jours[$i] !== 'Dimanche') {
21                 $horaire[$i]->setOuvAm('9');
22                 $horaire[$i]->setFerAm('12');
23                 $horaire[$i]->setOuvPm('14');
24                 $horaire[$i]->setFerPm('19');
25
26             }
27
28             $manager->persist($horaire[$i]);
29
30         }
31
32         $manager->flush();
33     }
34 }
35
```

Les Services :

```
$s = new ServicesPrincipaux();
$s->setTitre('Vidange');
$s->setDescription('Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.');
```

```
$s->setPrix('90');
$manager->persist($s);
```

```
$s1 = new ServicesPrincipaux();
$s1->setTitre('Depannage');
$s1->setDescription('Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.');
```

```
$s1->setPrix('130');
$manager->persist($s1);
```

```
$s2 = new ServicesPrincipaux();
$s2->setTitre('Changement pare-brise');
$s2->setDescription('Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Lorem ipsum dolor sit a
```

```
$s2->setPrix('110');
$manager->persist($s2);
```

```
$s3 = new ServicesPrincipaux();
$s3->setTitre('Changement Pneus');
$s3->setDescription('Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.');
```

```
$s3->setPrix('90');
$manager->persist($s3);
```

```
$manager->flush();
```

Utilisateur :

```
$user1 = new Utilisateur();
$plaintextPassword = '123456';
$user1->setEmail('testAdmin@mail.com');
$user1->setRoles(['ROLE_ADMIN']);
$hashedPassword = $this->passwordHasher->hashPassword(
    $user1,
    $plaintextPassword
);
$user1->setPassword($hashedPassword);
$manager->persist($user1);

$user2 = new Utilisateur();
$plaintextPassword = '123456';
$user2->setEmail('testVendeur@mail.com');
$user2->setRoles(['ROLE_VENDEUR']);
$hashedPassword = $this->passwordHasher->hashPassword(
    $user2,
    $plaintextPassword
);
$user2->setPassword($hashedPassword);
$manager->persist($user2);

$user3 = new Utilisateur();
$plaintextPassword = '123456';
$user3->setEmail('testModo@mail.com');
$user3->setRoles(['ROLE_MODERATEUR']);
$hashedPassword = $this->passwordHasher->hashPassword(
    $user3,
    $plaintextPassword
);
$user3->setPassword($hashedPassword);
$manager->persist($user3);
```