**DS SYMFONY**

1/ Environnement du travail :

Xampp

Vscode

Composer

Symfony CLI

2/ Les commandes utilisées dans le projet :

Symfony new *Projet-DS* –full

Cd Projet\_DS

symfony server:start

php bin/console doctrine:database:create Vente

composer require annotations

composer require twig

composer require symfony/orm-pack

composer require –dev symfony/maker-bundle

php bin/console make:entity Voiture (marque, couleur, prix)

php bin/console make:migration

php bin/console doctrine:migrations:migrate

php bin/console make:controller CarController

composer require –dev orm-fixtures

php bin/console make:fixtures

php bin/console doctrine :fixtures :load

3/ Partie code :

1. .env :
2. *###> symfony/framework-bundle ###*
3. APP\_ENV=dev
4. APP\_SECRET=d51e78725c18dc5da5d72736a3b7d8df
5. *###< symfony/framework-bundle ###*
6. *###> doctrine/doctrine-bundle ###*
7. *# Format described at https://www.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/configuration.html#connecting-using-a-url*
8. *# IMPORTANT: You MUST configure your server version, either here or in config/packages/doctrine.yaml*
9. *#*
10. *# DATABASE\_URL="sqlite:///%kernel.project\_dir%/var/data.db"*
11. DATABASE\_URL="mysql://root:@127.0.0.1:3306/vente?serverVersion=8&charset=utf8mb4"
12. *# DATABASE\_URL="postgresql://app:!ChangeMe!@127.0.0.1:5432/app?serverVersion=14&charset=utf8"*
13. *###< doctrine/doctrine-bundle ###*

1. Controller :

* Fonction Index et use :
* <?php
* namespace App\Controller;
* use App\Form\CarType;
* use App\Entity\Voiture;
* use Doctrine\ORM\EntityManager;
* use App\Repository\CarRepository;
* use App\Repository\VoitureRepository;
* use Doctrine\ORM\EntityManagerInterface;
* use Symfony\Component\HttpFoundation\Request;
* use Symfony\Component\HttpFoundation\Response;
* use Symfony\Component\Routing\Annotation\Route;
* use Symfony\Component\HttpFoundation\JsonResponse;
* use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
* use Doctrine\Persistence\ManagerRegistry as PersistenceManagerRegistry;
* class CarController *extends* AbstractController
* {
* */\*\**
* *\* @Route("/", name="car")*
* *\*/*
* *public* function index(): Response {
* return $this->render("home.html.twig");
* }
* Function allCar :
* */\*\**
* *\* @Route("/allCar", name="allCar")*
* *\*/*
* *public* function allCar(VoitureRepository $rep): Response {
* $cars = $rep->findAll();
* return $this->render(
* "allCar.html.twig",
* ["cars"=>$cars]
* );
* }
* Function detail :
* */\*\**
* *\* @Route("/detail/{id}", name="details")*
* *\*/*
* *public* function oneCar (VoitureRepository $rep, $id) : Response {
* $oneCar = $rep->find($id);
* return $this->render("detail.html.twig", ["onecar"=>$oneCar]);
* }
* Function addCar :

*/\*\**

*\* @Route("/add", name="addCar")*

*\*/*

*public* function addCar(Request $req , PersistenceManagerRegistry $doctrine){

         $car = new Voiture();

         $form = $this->createForm(CarType::class, $car);

         $form->handleRequest($req);

         if ($form->isSubmitted() && $form->isValid()) {

            $car = $form->getData();

             $entityManger = $doctrine->getManager();

             $entityManger->persist($car);

             $entityManger->flush();

            $this->addFlash('notice','Une nouvelle voiture a été ajoutée avec succès ...');

            return $this->redirectToRoute('allCar');

         }

         return $this->render('addCar.html.twig', [

             'form' => $form->createView()

         ]);

    }

* Function update :
* */\*\**
* *\* @Route("/update/{id}", name="update")*
* *\*/*
* *public* function update(Request $req , $id, EntityManagerInterface $em, PersistenceManagerRegistry $doctrine){
* $rep = $em->getRepository("App\Entity\Voiture");
* $car = $rep->find($id);
* $form = $this->createForm(CarType::class, $car);
* $form->handleRequest($req);
* if ($form->isSubmitted() && $form->isValid()) {
* $em = $doctrine->getManager();
* $em->persist($car);
* $em->flush();
* $this->addFlash('notice', 'Une voiture a été modifiée avec succès ...');
* return $this->redirectToRoute('allCar');
* }
* return $this->render('updateCar.html.twig', [
* 'form' => $form->createView()
* ]);
* }
* Function delete :
* */\*\**
* *\* @Route("/delete/{id}", name="delete")*
* *\*/*
* *public* function delete($id, EntityManagerInterface $em, PersistenceManagerRegistry $doctrine){
* $rep = $em->getRepository("App\Entity\Voiture");
* $data = $rep->find($id);
* $em = $doctrine->getManager();
* $em->remove($data);
* $em->flush();
* $this->addFlash('notice','Une voiture a été supprimée avec succès ...');
* return $this->redirectToRoute('allCar');
* }

1. Entity :
2. <?php
3. namespace App\Entity;
4. use App\Repository\VoitureRepository;
5. use Doctrine\ORM\Mapping as ORM;
6. #[ORM\Entity(repositoryClass: VoitureRepository::class)]
7. class Voiture
8. {
9. #[ORM\Id]
10. #[ORM\GeneratedValue]
11. #[ORM\Column]
12. *private* ?int $id = null;
13. #[ORM\Column(length: 255)]
14. *private* ?string $marque = null;
15. #[ORM\Column(length: 255)]
16. *private* ?string $couleur = null;
17. #[ORM\Column(length: 255)]
18. *private* ?string $prix = null;
19. Fixtures :
20. <?php
21. namespace App\DataFixtures;
22. use Doctrine\Bundle\FixturesBundle\Fixture;
23. use Doctrine\Persistence\ObjectManager;
24. use App\Entity\Voiture;
25. class CarFixture *extends* Fixture
26. {
27. *public* function load(ObjectManager $manager): void
28. {
29. for ($i = 1; $i <= 10; $i++) {
30. $car = new Voiture();
31. $car->setMarque("La masque " . $i);
32. $car->setCouleur("Le couleur " . $i);
33. $car->setPrix(" 100.000 " . $i);
34. $manager->persist($car);
35. }
36. $manager->flush();
37. }
38. }
39. Form :

<?php

namespace App\Form;

use App\Entity\Voiture;

use Symfony\Component\Form\AbstractType;

use Symfony\Component\Form\FormBuilderInterface;

use Symfony\Component\OptionsResolver\OptionsResolver;

class CarType *extends* AbstractType

{

*public* function buildForm(FormBuilderInterface $builder, array $options): void

    {

        $builder

            ->add('marque')

            ->add('couleur')

            ->add('prix')

        ;

    }

*public* function configureOptions(OptionsResolver $resolver): void

    {

        $resolver->setDefaults([

            'data\_class' => Voiture::class,

        ]);

    }

}

1. Templates :

* base.html.twig :
* home.html.twig :
* allCar.html.twig :
* addCar.html.twig :
* detail.html.twig :
* update.html.twig :