***Compte rendu***

***Mardi 17 mars 2020***

***NIVEAU 2 (FRAMEWORK) :***

***Module 3bis : Symfony4***

1 - Découverte de Symfony 4.

2- Développer avec Symfony 4.

3- Utilisation de Doctrine sous Symfony 4.

***Pratique*** :

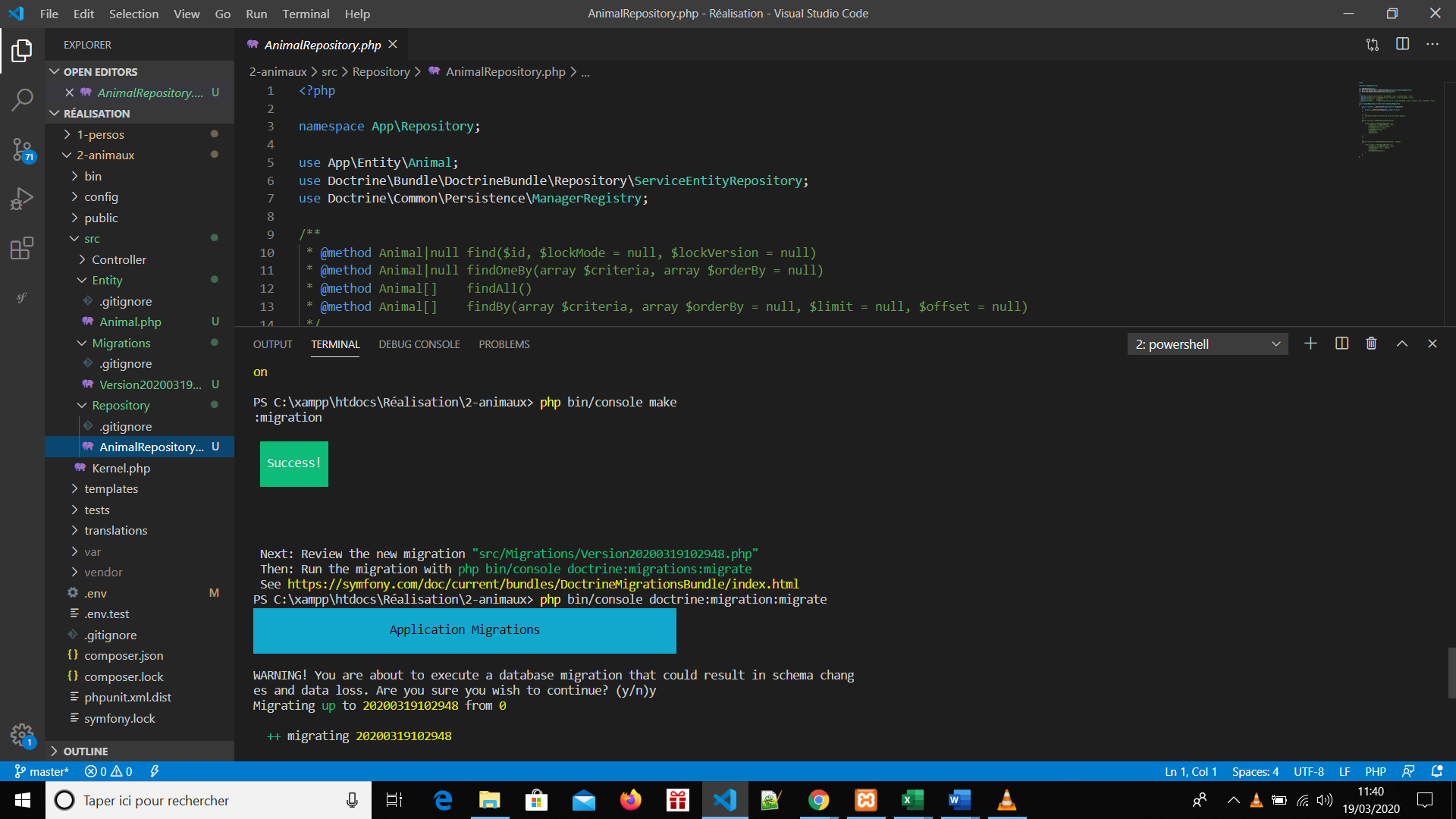
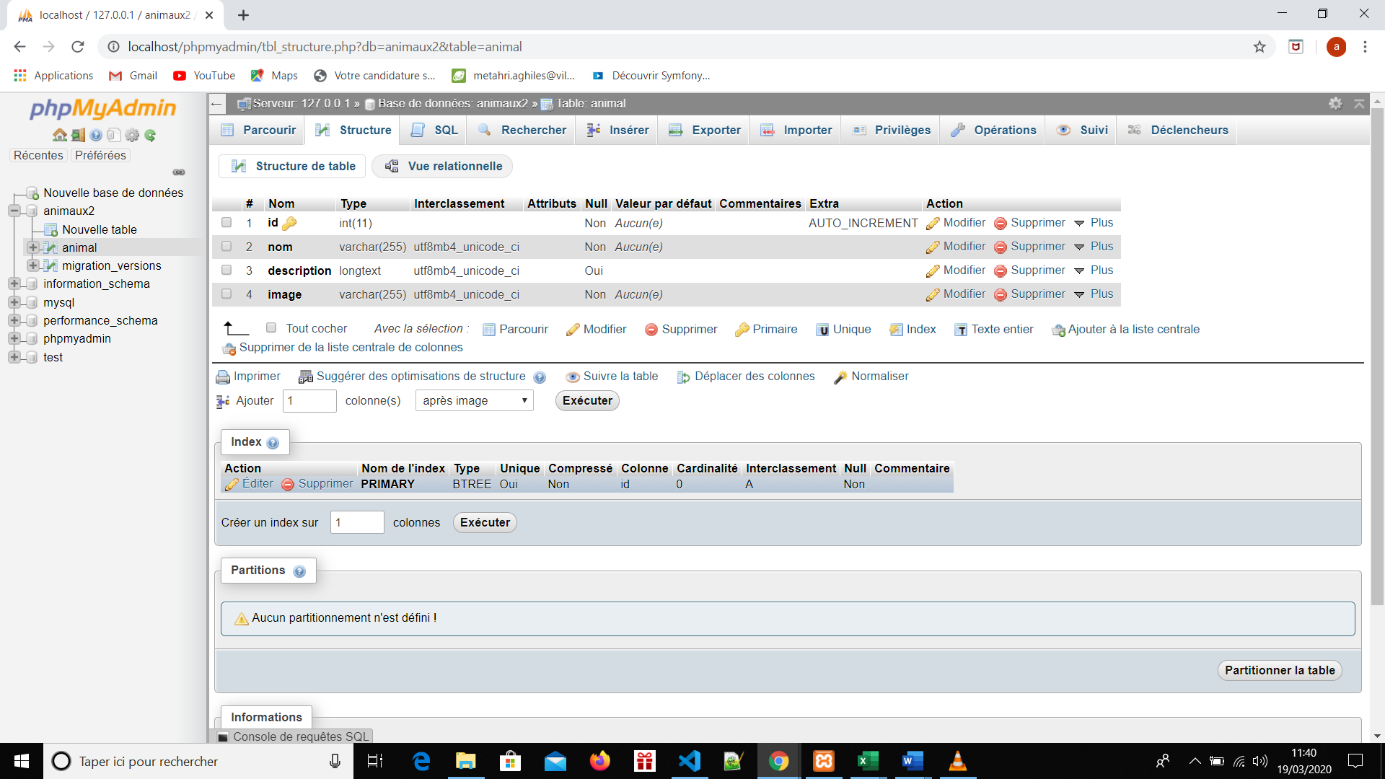
Réaliser une application en utilisant Symfony4 et l’utilisation d’une base de données.

**Objectifs :**

* Utiliser le système de BD de Symfony.
* Utiliser ORM (Object-Relational Mapping) : DOCTRINE permettant de faire le lien entre les classes et la base de données.

**Réalisation :**

* Lancer la création de la BD : php bin/console doctrine :database :create
* Créer la première entité : php bin/console make :entity.
* Créer un fichier de migration.



* Insertion des données (Fixtures) : remplir la table :

Installer le module : composer require orm-fixtures—dev

Créer une fixture pour notre Entity Animal :php bin/console make :fixtures AnimalFixtures

<?php

namespace App\DataFixtures;

use App\Entity\Animal;

use Doctrine\Bundle\FixturesBundle\Fixture;

use Doctrine\Common\Persistence\ObjectManager;

class AnimalFixtures extends Fixture

{

    public function load(ObjectManager $manager)

    {

        $a1 = new Animal();

        $a1->setNom("Chien")

           ->setDescription("Un animal domestique")

           ->setImage("chien.png")

           ;

           $manager->persist($a1);

        // $product = new Product();

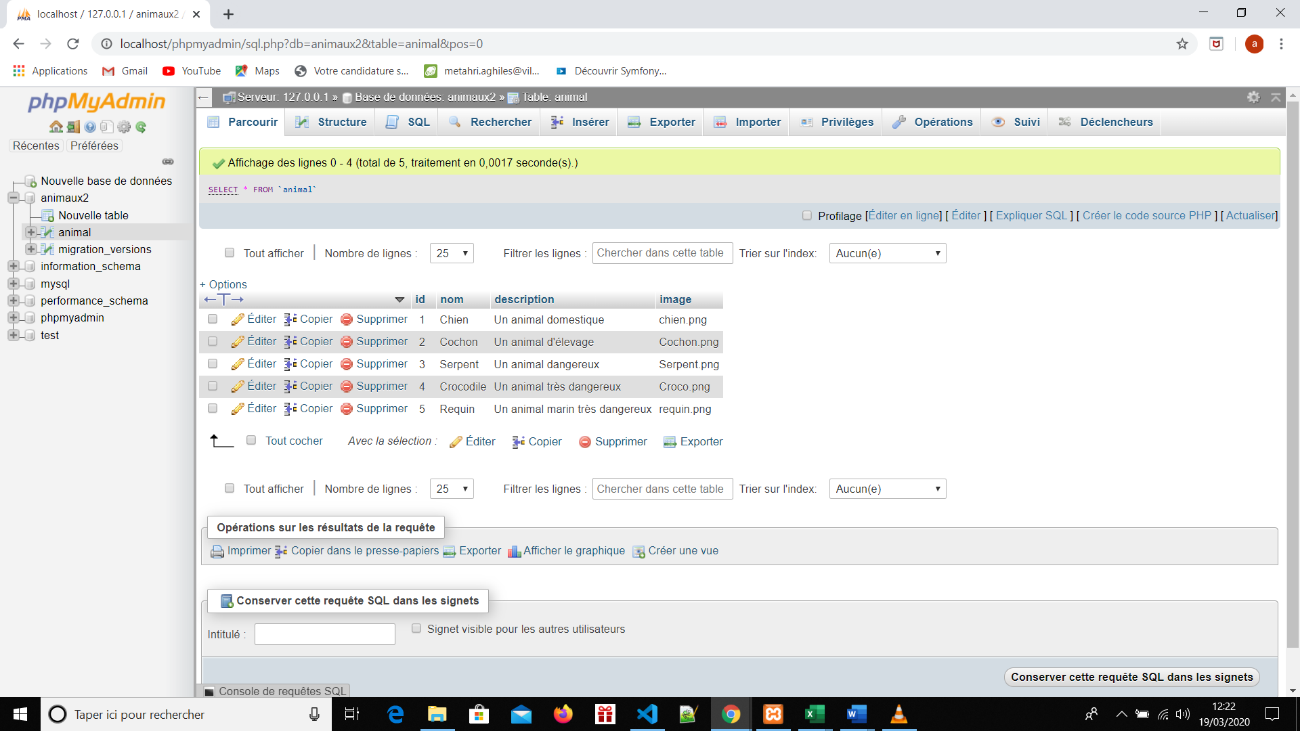
        // $manager->persist($product);

        $manager->flush();

    }

}

Lancer la création des données : > php bin/console doctrine:fixtures:load

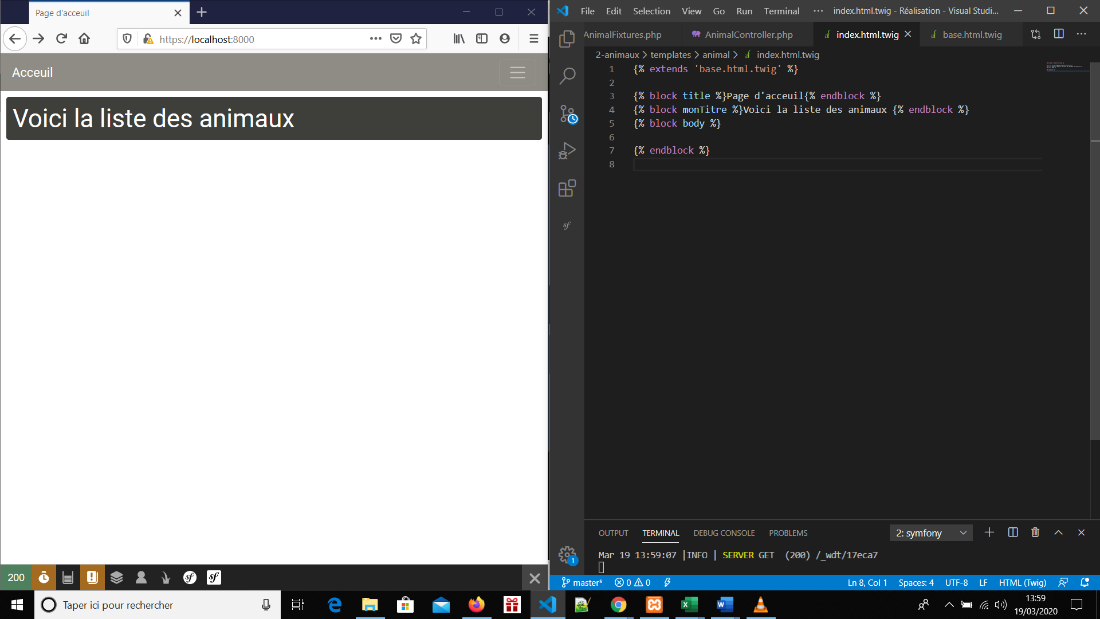


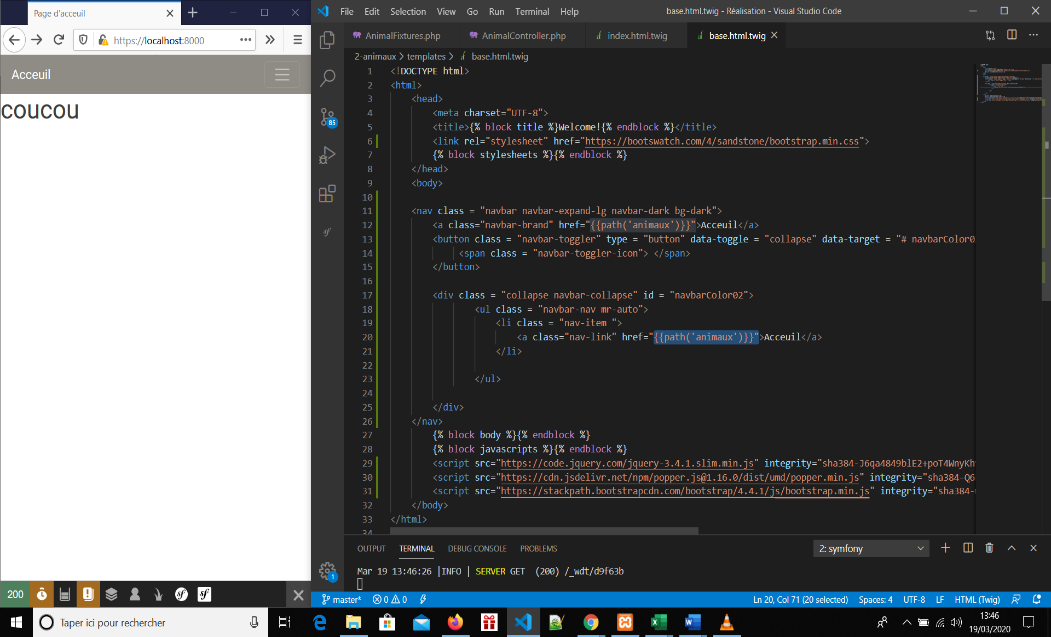
* Créer la page d’accueil listant les animaux

Créer le contrôleur : php bin/console make :controller AnimalController

Réalisez la vue correspondante et lancer le serveur : symfony server : start.

Utiliser un thème boostrtrap.(bootswatch.com)





* (Repository) :

Récupération des données :

Afficher les images : concaténer avec Twig {% extends 'base.html.twig' %}

{% block title %}Page d'acceuil{% endblock %}

{% block monTitre %}Voici la liste des animaux {% endblock %}

{% block body %}

    {% for animal in animaux %}

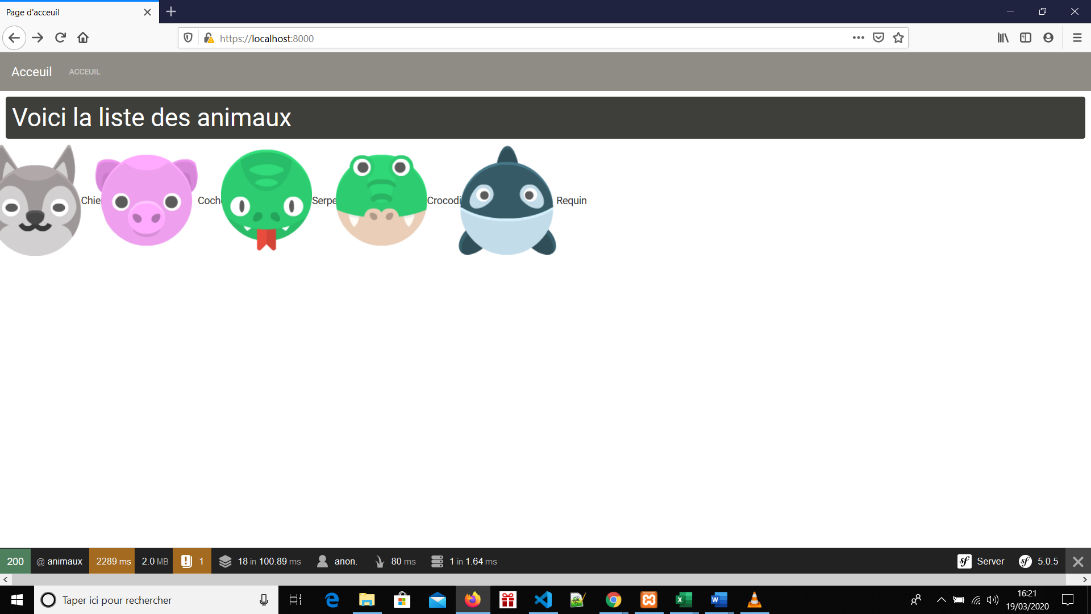
        <div class="row align-items-center">

            <img src="{{asset('images/' ~ animal.image)}}">

        {{animal.nom}}

    {% endfor %}

{% endblock %}



* Récupération des données : injection de dépendance :

class AnimalController extends AbstractController

{

    /\*\*

     \* @Route("/", name="animaux")

     \*/

    public function index()

    {

        $repository = $this->getDoctrine()->getRepository(Animal::class);

        $animaux = $repository->findAll();

        return $this->render('animal/index.html.twig', [

            "animaux" => $animaux

        ]);

    }

}

class AnimalController extends AbstractController

{    /\*\*

     \* @Route("/", name="animaux")

     \*/

    public function index(AnimalRepository $repository)

    {

        $animaux = $repository->findAll();

        return $this->render('animal/index.html.twig', [

            "animaux" => $animaux

        ]);

* Ajouter des champs :

**Poids**: correspondant au poids moyen d’un animal (integer)

**Dangereux** : un booléen indiquant si l’animal est dangereux ou non.

Vider la base de données manuellement. (Pour éviter des problèmes lors de la migration)

Lancer la migration.

Ajouter des données dans les fixtures pour créer les animaux avec ces nouvelles infrmations.

Envoyer les données dans la DataBase.

