

ESMT

DAR 2019/2020

Mini projet Symfony & Bootstrap

Gestion de produits

Nom : Mbaye

Prénom : Ndeye Dior

1.Réalisation :

Ce projet est réalisé avec Symfony et bootstrap et porte sur la gestion de produits

2.Fonctionnalités :

Les fonctionnalités sont les suivants :

- Voir la liste de tous les produits
- Voir les détails d'un produit
- Modifier les informations d'un produit
- Supprimer un produit
- Créer un nouveau produit

Pour les fonctionnalités supplémentaires nous avons :

- Pour chaque produit afficher une photo

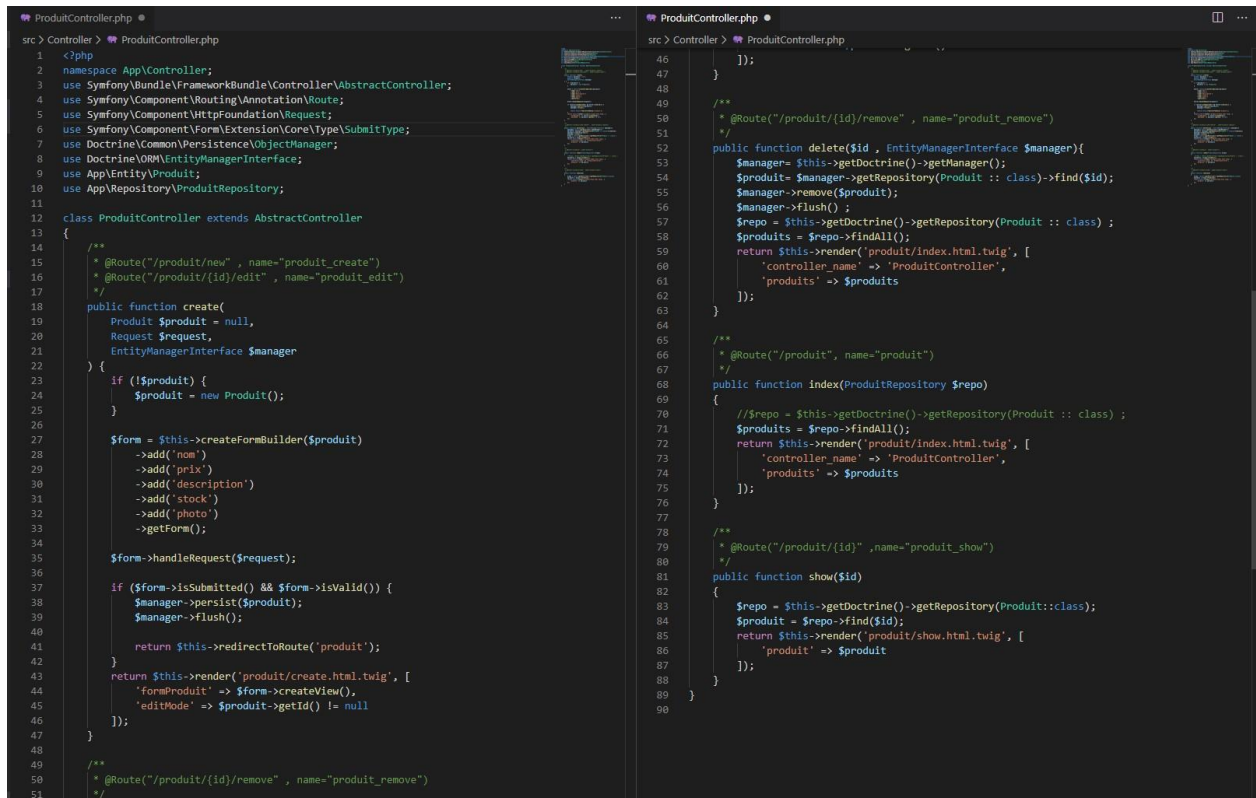
3.Structuration du code :

```
> assets
> bin
> config
> node_modules
> public
> src
✓ templates
  ✓ produit
    ✓ create.html.twig
    ✓ index.html.twig
    ✓ menu.html.twig
    ✓ show.html.twig
    ✓ base.html.twig
  > tests
  > translations
  > var
  > vendor
  ⚙ .env
  ≡ .env.test
  ⬠ .gitignore
  {} composer.json
  {} composer.lock
  {} package-lock.json
  ≡ phpunit.xml.dist
  ≡ symfony.lock
```

4. Présentation du code :

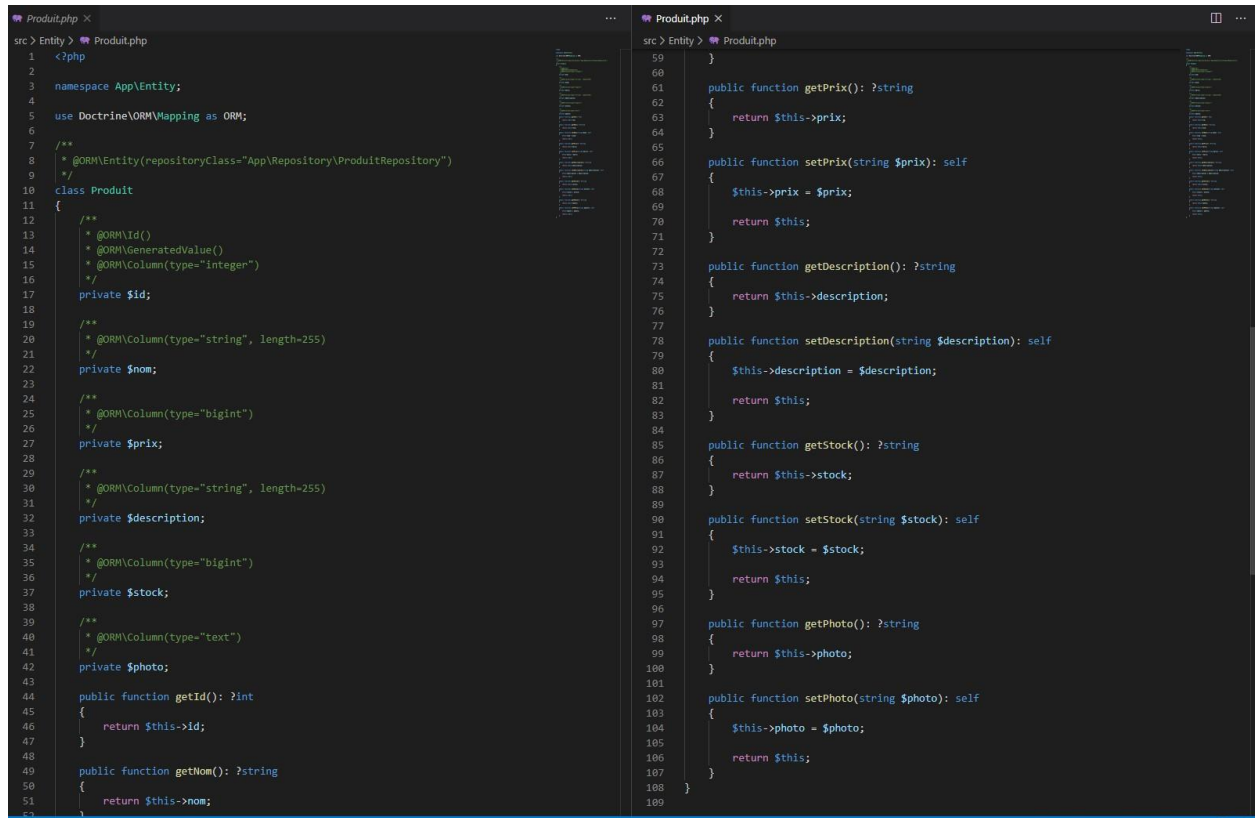
Dans le controller on trouve tous les fonctions dont on a besoin pour la réalisation de notre application :

- Create() : qui permet de sauvegarder un produits dans la base de données ou de le modifié.
- Delete() : qui permet de supprimer un produit dans la base de donnée
- Index() : qui permet de récupérer tous les pays disponible dans la base de donnée
- Show(\$id) : qui parmet de récupérer un produit grâce à l'id fourni.



```
1 <?php
2 namespace App\Controller;
3 use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
4 use Symfony\Component\Routing\Annotation\Route;
5 use Symfony\Component\HttpFoundation\Request;
6 use Symfony\Component\Form\Extension\Core\Type\SubmitType;
7 use Doctrine\Common\Persistence\ObjectManager;
8 use Doctrine\ORM\EntityManagerInterface;
9 use App\Entity\Produit;
10 use App\Repository\ProduitRepository;
11
12 class ProductController extends AbstractController
13 {
14     /**
15      * @Route("/produit/new", name="produit_create")
16      * @Route("/produit/{id}/edit", name="produit_edit")
17      */
18     public function create(
19         Produit $produit = null,
20         Request $request,
21         EntityManagerInterface $manager
22     ) {
23         if (!$produit) {
24             $produit = new Produit();
25         }
26
27         $form = $this->createFormBuilder($produit)
28             ->add('nom')
29             ->add('prix')
30             ->add('description')
31             ->add('stock')
32             ->add('photo')
33             ->getForm();
34
35         $form->handleRequest($request);
36
37         if ($form->isSubmitted() && $form->isValid()) {
38             $manager->persist($produit);
39             $manager->flush();
40
41             return $this->redirectToRoute('produit');
42         }
43         return $this->render('produit/create.html.twig', [
44             'formProduit' => $form->createView(),
45             'editMode' => $produit->getId() != null
46         ]);
47     }
48
49     /**
50      * @Route("/produit/{id}/remove", name="produit_remove")
51      */
52
53     public function delete($id, EntityManagerInterface $manager){
54         $manager= $this->getDoctrine()->getManager();
55         $produit= $manager->getRepository(Produit::class)->find($id);
56         $manager->remove($produit);
57         $manager->flush();
58         $repo = $this->getDoctrine()->getRepository(Produit::class);
59         $produits = $repo->findAll();
60         return $this->render('produit/index.html.twig', [
61             'controller_name' => 'ProductController',
62             'produits' => $produits
63         ]);
64     }
65
66     /**
67      * @Route("/produit", name="produit")
68      */
69     public function index(ProduitRepository $repo)
70     {
71         // $repo = $this->getDoctrine()->getRepository(Produit::class);
72         $produits = $repo->findAll();
73         return $this->render('produit/index.html.twig', [
74             'controller_name' => 'ProductController',
75             'produits' => $produits
76         ]);
77     }
78
79     /**
80      * @Route("/produit/{id}", name="produit_show")
81      */
82     public function show($id)
83     {
84         $repo = $this->getDoctrine()->getRepository(Produit::class);
85         $produit = $repo->find($id);
86         return $this->render('produit/show.html.twig', [
87             'produit' => $produit
88         ]);
89     }
90 }
```

On a un entité Produit ou on a la classe Produit avec ces attributs : id – nom – prix – description – photo, ce qui va nous permettre aussi de créer la base de données.



```
src > Entity > Produit.php
1 <?php
2
3 namespace App\Entity;
4
5 use Doctrine\ORM\Mapping as ORM;
6
7 /**
8  * @ORM\Entity(repositoryClass="App\Repository\ProduitRepository")
9  */
10 class Produit
11 {
12     /**
13      * @ORM\Id()
14      * @ORM\GeneratedValue()
15      * @ORM\Column(type="integer")
16      */
17     private $id;
18
19     /**
20      * @ORM\Column(type="string", length=255)
21      */
22     private $nom;
23
24     /**
25      * @ORM\Column(type="bigint")
26      */
27     private $prix;
28
29     /**
30      * @ORM\Column(type="string", length=255)
31      */
32     private $description;
33
34     /**
35      * @ORM\Column(type="bigint")
36      */
37     private $stock;
38
39     /**
40      * @ORM\Column(type="text")
41      */
42     private $photo;
43
44     public function getId(): ?int
45     {
46         return $this->id;
47     }
48
49     public function getNom(): ?string
50     {
51         return $this->nom;
52     }
53
54     public function getPrix(): ?string
55     {
56         return $this->prix;
57     }
58
59     public function setPrix(string $prix): self
60     {
61         $this->prix = $prix;
62         return $this;
63     }
64
65     public function getDescription(): ?string
66     {
67         return $this->description;
68     }
69
70     public function setDescription(string $description): self
71     {
72         $this->description = $description;
73         return $this;
74     }
75
76     public function getStock(): ?string
77     {
78         return $this->stock;
79     }
80
81     public function setStock(string $stock): self
82     {
83         $this->stock = $stock;
84         return $this;
85     }
86
87     public function getPhoto(): ?string
88     {
89         return $this->photo;
90     }
91
92     public function setPhoto(string $photo): self
93     {
94         $this->photo = $photo;
95         return $this;
96     }
97
98 }
```

On a un dossier templates qui permet de gérer l’affichage de notre application :

- Create.html.twig : Permet d’afficher le formulaire de création d’un produit

```

templates > produit > ✓ create.html.twig
1  {% extends 'base.html.twig' %}
2  {% form_theme formProduit 'bootstrap_4_layout.html.twig' %}
3  {% block body %}
4      <nav class="navbar navbar-expand-lg" style="background-color: #0d3283;color:white">
5          <a class="navbar-brand" href="#" style="color:white">Gestion de produits</a>
6          <button class="navbar-toggler"
7              type="button"
8              data-toggle="collapse"
9              data-target="#navbarSupportedContent"
10             aria-controls="navbarSupportedContent"
11             aria-expanded="false"
12             aria-label="Toggle navigation">
13              <span class="navbar-toggler-icon"></span>
14          </button>
15
16      <div class="collapse navbar-collapse" id="navbarSupportedContent" >
17          <ul class="navbar-nav mr-auto">
18              <li class="nav-item" style="color:white">
19                  <a class="nav-link" href="{{ path('produit') }}" style="color:white">
20                      Liste des produits
21                  </a>
22              </li>
23              <li class="nav-item" style="color:white">
24                  <a class="nav-link" href="{{ path('produit_create') }}" style="color:white">
25                      Ajouter un produit
26                  </a>
27              </li>
28          </ul>
29      </div>
30  </nav>
31  <div class="container" style="margin-top:20px">
32      <div class="row" style="justify-content:center">
33          <h3 align="center" style="border-bottom:4px solid #0d3283;padding-bottom:10px">Ajouter un produits</h3>
34      </div>
35      <div class="row" style="justify-content:center">
36          <div class="col-md-6">
37              {{ form_start(formProduit) }}
38              {{ form_widget(formProduit) }}
39              <button type="submit" class="btn btn-success">
40                  {% if editMode %}
41                      Enregistrer
42                  {% else %}
43                      Ajouter le produit
44                  {% endif %}
45              </button>
46              {{ form_end(formProduit) }}
47          </div>
48      </div>
49  </div>
50  {% endblock %}
51

```

- Index.html.twig : Permet d'afficher la liste totale des produits



- `Base.html.twig` : Elle permet de regrouper tous les codes dupliquer dans les autres fichiers.


```

templates > ./ base.html.twig
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="UTF-8" />
5      <title>
6        Produits
7      </title>
8      {% block head_css %}
9        <style>
10         hr.style5 { background-color: #fff; border-top: 2px dashed #8c8b8b; }
11        </style>
12        <link rel="stylesheet"
13          href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
14          integrity="sha384-Vkoo8x4CGs03+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
15          crossorigin="anonymous" />
16      {% endblock %}
17    </head>
18    <body>
19      {% block menu %}
20
21      {% endblock %}
22      {% block body %}
23
24      {% endblock %}
25      {% block head_js %}
26        <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
27          integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
28          crossorigin="anonymous"></script>
29        <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
30          integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
31          crossorigin="anonymous"></script>
32        <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
33          integrity="sha384-wfSDF2E50Y2D1uUdJ003uMBJnJjuUD4Ih7YwaYd1i1qfktj0Uod8GCExl30g8ifwB6"
34          crossorigin="anonymous"></script>
35      {% endblock %}
36    </body>
37  </html>
38


```

5.Présentation des interfaces :

La Liste de tous les produits :

Gestion de produits
Liste des produits
Ajouter un produit


La liste des produits disponibles



Fruits 250 F CFA

Nous souhaitons vous offrir la meilleure expérience possible en adaptant les annonces et le contenu que vous voyez sur notre site.


[Editer](#)
[Supprimer](#)



Légumes 250 F CFA

Nous souhaitons vous offrir la meilleure expérience possible en adaptant les annonces et le contenu que vous voyez sur notre site.


[Editer](#)
[Supprimer](#)



Poissons 250 F CFA

Nous souhaitons vous offrir la meilleure expérience possible en adaptant les annonces et le contenu que vous voyez sur notre site.


[Editer](#)
[Supprimer](#)



Volailles 250 F CFA

Nous souhaitons vous offrir la meilleure expérience possible en adaptant les annonces et le contenu que vous voyez sur notre site.

[Editer](#)
[Supprimer](#)



Viandes 250 F CFA

Nous souhaitons vous offrir la meilleure expérience possible en adaptant les annonces et le contenu que vous voyez sur notre site.

[Editer](#)
[Supprimer](#)

200
@ produit_remove
444 ms
2.0 MB
2
22 in 8.29 ms
anon.
16 ms
5 in 5.57 ms
Server
5.0.4

Nous avons la page d'accueil de notre application web, on peut voir tous les produits disponibles à savoir le nom, le prix, la description et la photo. Pour chaque produit on peut l'éditer ou le supprimer.

Ajouter un nouveau produit :

[Gestion de produits](#) [Liste des produits](#) [Ajouter un produit](#)

Ajouter un produits

Nom

Prix

Description

Stock

Photo

Ajouter le produit

Pour ajouter un produit, on saisit son nom, le prix, la description, la quantité et le lien de la photo. Puis après sauvegarde, on est directement redirigé au niveau de la page d'accueil.