

Chapitre 2 - Ajouter des pages

Introduction.....	1
Création de la page « Contact ».....	1
Ajouter un contrôleur.....	1
Création du template.....	2
Intégration de la nouvelle page dans le menu.....	3
Exercices	5

Introduction

Dans le chapitre précédent, nous avons construit la page d'accueil du site. Nous allons maintenant créer quelques pages que nous retrouvons habituellement sur tous les sites ou presque. Voici ces pages :

- contact
- à propos
- mentions légales

Création de la page « Contact »

Ajouter un contrôleur

Je rappelle que nous sommes dans une architecture MVC (Modèle Vue Contrôleur). Par conséquent, nous allons dans le fichier « src/Controller/BaseController.php » et pour nous éviter de tout retaper, nous allons faire un copier-coller de notre bloc existant pour le mettre en dessous comme ceci :

Dans « src/Controller/BaseController.php »

```
<?php

namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class BaseController extends AbstractController
{
    #[Route('/', name: 'app_accueil')]
    public function index(): Response
    {
        return $this->render('base/index.html.twig', [

        ]);
    }
}
```

```
#[Route('/', name: 'app_accueil')]
public function index(): Response
{
    return $this->render('base/index.html.twig', [
    ]);
}
```

Puis, nous réalisons les étapes suivantes :

1. Modification du nom de la route et l'url pour accéder à la page contact.
2. Modification du nom de la fonction pour la renommer en « contact ».
3. Modification du template à appeler en copiant « template/base/index.html.twig » dans le même répertoire et en le nommant « contact.html.twig ».

Voici le résultat :

```
<?php

namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class BaseController extends AbstractController
{
    #[Route('/', name: 'app_accueil')]
    public function index(): Response
    {
        return $this->render('base/index.html.twig', [

        ]);
    }

    #[Route('/contact', name: 'app_contact')]
    public function contact(): Response
    {
        return $this->render('base/contact.html.twig', [

        ]);
    }
}
```

Création du template

Dans le fichier « templates/base/contact.html.twig », nous redéfinissons les éléments du template afin de personnaliser notre nouvelle page.

```
{% extends 'base.html.twig' %}

{% block title %}{{parent()}}Contact{% endblock %}

{% block body %}

<h1 class="text-center text-primary mt-4 pt-4 display-1 fw-bold"> CONTACT</h1>

{% endblock %}
```

Commentaire : pour l’instant, nous personnalisons uniquement le titre de la fenêtre du navigateur et le titre de la page.

Vous pouvez visualiser le résultat en tapant l’URL suivante « /contact ».

Exemple :

```
http://s3-4400.nuage-peda.fr/share/contact
```

Intégration de la nouvelle page dans le menu

Nous allons maintenant modifier notre menu afin de nous rendre sur la page « contact » plus facilement.

Dans « templates/base.html.twig », modifier votre « navbar » comme ceci :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>{% block title %}Share - {% endblock %}</title>
    {% block stylesheets %}
    <link href="https://bootswatch.com/5/lumen/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.1/font/bootstrap-
icons.css">
    {% endblock %}
  </head>
  <body>
<nav class="navbar navbar-expand-lg bg-light" data-bs-theme="light">
  <div class="container-fluid">
    <a class="navbar-brand" href="{{ path('app_accueil') }}">Share</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarColor03" aria-controls="navbarColor03" aria-expanded="false" aria-
label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
```

```

</button>
<div class="collapse navbar-collapse" id="navbarColor03">
  <ul class="navbar-nav me-auto">
    <li class="nav-item">
      <a class="nav-link active" href="{{path('app_accueil')}}">Accueil
      <span class="visually-hidden">(current)</span>
    </a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="{{path('app_contact')}}">Contact</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Pricing</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">About</a>
    </li>
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" data-bs-toggle="dropdown" href="#" role="button"
aria-haspopup="true" aria-expanded="false">Dropdown</a>
      <div class="dropdown-menu">
        <a class="dropdown-item" href="#">Action</a>
        <a class="dropdown-item" href="#">Another action</a>
        <a class="dropdown-item" href="#">Something else here</a>
        <div class="dropdown-divider"></div>
        <a class="dropdown-item" href="#">Separated link</a>
      </div>
    </li>
  </ul>
  <form class="d-flex">
    <input class="form-control me-sm-2" type="search" placeholder="Search">
    <button class="btn btn-secondary my-2 my-sm-0" type="submit">Search</button>
  </form>
</div>
</div>
</nav>
{% block body %}Share{% endblock %}
{% block javascripts %}
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Q
yq46cDfL" crossorigin="anonymous"></script>
{% endblock %}
</body>
</html>

```

Voici le rendu du menu modifié :

Share	Accueil	Contact	Pricing	About	Dropdown ▼	<input type="text" value="Search"/>	SEARCH
-------	---------	---------	---------	-------	------------	-------------------------------------	--------

CONTACT

Exercices

En vous appuyant sur ce chapitre, réalisez les pages suivantes :

- à propos
- mentions légales

Informations complémentaires pour réaliser ce travail :

Les fonctions, le nom des routes ainsi que les URL ne peuvent pas contenir certains caractères spéciaux (accents, &, %, ...) et des espaces.