Humanbooster

TD Symfony

Jour 2

Novembre 2021

Cindy PONCIN

Création du projet

```
symfony new lanimalerie -full
```

symfony new nomDuProjet -full

> Création d'un projet d'application web, site web

symfony new nomDuProjet

> Création d'un projet d'API, micro service

Démarrer l'application

```
cd lanimalerie
symfony server:start
```

Pour accéder au site : utiliser l'adresse indiquée dans le cadre vert après avoir démarré votre serveur.

exemple https://127.0.0.1:8000

Pour arrêter le serveur : ctrl + c

Connexion à la base de données

```
#.env
DATABASE_URL="mysql://db_user:db_password@127.0.0.1:3306/db_name
?serverVersion=5.7"
```

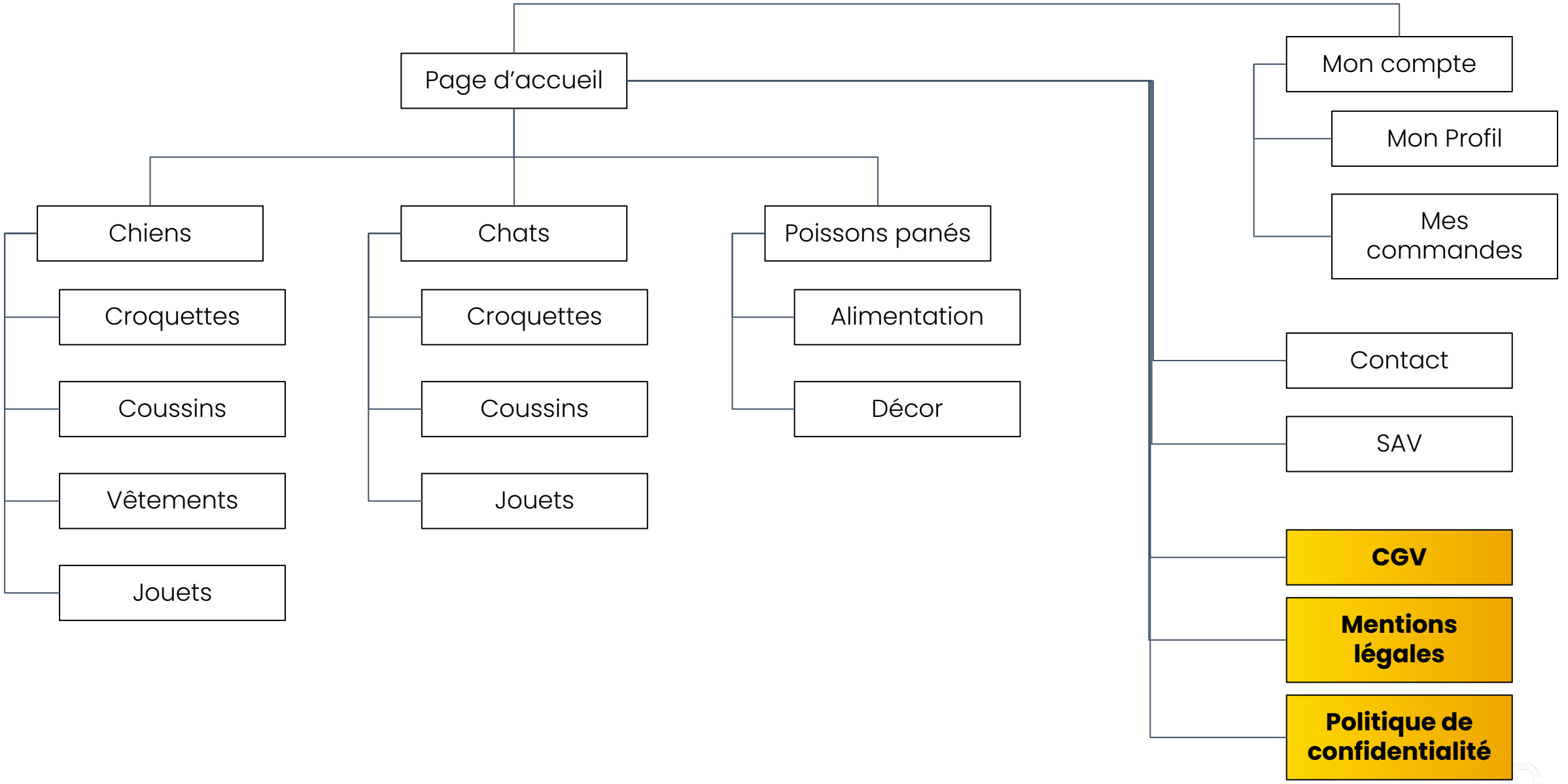
Créer depuis phpmyadmin une base de données *lanimalerie*

Modifier les valeurs suivantes avec vos données de connexion

- db_user
- db_password
- db_name

Créer uniquement la base : ne pas créer de table

Arborescence du site



Création de la page CGV | Création du controller

```php
<?php
// src/Controller/ContentController.php
namespace App\Controller;

use Symfony\Component\HttpFoundation\Response;

class ContentController
{
    public function cgv(): Response
    {
        return new Response(
            '<html><body><h1>CGV</h1></body></html>'
        );
    }
}
```

Création de la page CGV | Création de la route

```
# config/routes.yaml

app_cgv:
    path: /cgv/
    controller: App\Controller\ContentController::cgv
```

Tester l'accès à votre page

> https://127.0.0.1:8000/cgv

Création de la page CGV | Création du template twig

Prérequis
Installation du package twig

```
composer require twig
```

Création de la page CGV | Modification du controller

```php
<?php
// src/Controller/ContentController.php
namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;

class ContentController extends AbstractController
{
    public function cgv(): Response
    {
        return $this->render('content/cgv.html.twig', [
            'title' => 'CGV',
        ]);
    }
}
```

namespace App \ Controller

> Permet de séparer les constantes, les fonctions et les classes dans différents espaces afin d'éviter tout conflit

> Permet d’avoir plusieurs constantes, fonctions ou classes du même nom dans la même application

Symfony \ Component \ HttpFoundation \ Response

> Permet d'utiliser des requêtes HTTP

Symfony \ Bundle \ FrameworkBundle \ Controller \ AbstractControl ler

> Permet d'utiliser la méthode render

Création de la page CGV | Création du template

```
{# templates/content/cgv.html.twig #}
<h1>L'Animalerie | {{ title }}</h1>
```

Création de la page CGV | Ajout d'une description

Modification du controller

```
public function cgv(): Response
    {
        return $this->render('content/cgv.html.twig', [
            'title' => 'CGV',
            'description' => 'Contenu de la page',

        ]);
    }
```

Modification du template

```
{# templates/content/cgv.html.twig #}
<h1>L'Animalerie | {{ title }}</h1>
<p>{{description}}</p>
```

Création des pages politique de confidentialité et mentions légales

En utilisant les étapes précédentes créer les pages mentions légales et politiques de confidentialités

Création de la page CGV | Passage de paramètre

Modification du controller

```
class ContentController extends AbstractController
{
    public function cgv(string $title): Response
    {
        return $this->render('content/cgv.html.twig', [
            'title' => $title,
            'description' => 'contenu de la page'
        ]);
    }
}
```

Création de la page CGV | Passage de paramètre

Modification du fichier routes.yaml

```
app_cgv:
    path: /cgv/{title}
    controller: App\Controller\ContentController::cgv
    defaults:
        title: 'CGV'
```

Modifications des pages politique de confidentialité et mentions légales

Modifier les pages Politiques de confidentialité et mentions légales pour passer le titre de la page en paramètres

Création du footer

Créer un footer et intégrer un lien vers la page CGV

```
<a href="{{ path('app_cgv') }}" title="CGV">CGV</a>
```

Création du footer

Créer un footer qui intégrera des liens vers les pages

- mentions légales
- politique de confidentialité
- CGV

CSS / JS / Images statiques

Prérequis
Installation du package asset

```
composer require symfony/asset
```

> Permet l'utilisation de la fonction asset()
> Permet de faire un lien vers les différentes ressources

CSS / JS / Images statiques

```
{# public/images/logo.png #}
<img src="{{ asset('images/logo.png') }}" alt="Symfony!"/>

{# public/css/style.css #}
<link href="{{ asset('css/style.css') }}" rel="stylesheet"/>

{# public/js/script.js #}
<script src="{{ asset('js/script.js') }}"></script>
```

CSS / JS / Images statiques

Intégrer à votre projet des css, du js et un logo

CSS / JS / Images statiques

Créer un template de base incluant

- un header avec un logo
- le footer avec le menu
- un block main contiendra le contenu de chaque template déjà créé