Projet 1 - Galerie de photos aléatoire

Objectifs de l'application

- Créer une galerie de photos aléatoire

Objectifs pédagogiques

- Installer un projet Symfony
- Découvrir l'architecture de Symfony
- Utiliser le serveur de développement de Symfony
- Créer une route/Créer un contrôleur
- Découvrir Twig

Avant de commencer

Avant de commencer, nous allons vérifier que nous avons bien les pré-requis pour developper avec Symfony.

Pour cela, nous allons utiliser le "Symfony Check" :

```
symfony check:requirements
```

Installer un projet Symfony

Est-ce que vous êtes **prêts** et **prêtes** ?

Vous allez voir qu'initialiser un projet Symfony est très simple.

Il suffit de lancer la commande suivante :

```
symfony new projet1 --webapp
```

Découvrir l'architecture de Symfony

L'architecture de Symfony est très simple :

- `config/` : contient les fichiers de configuration de l'application
- `public/` : contient les fichiers statiques (css, js, images, ...)
- `src/` : contient les fichiers sources de l'application
- `templates/` : contient les fichiers Twig
- `var/` : contient les fichiers temporaires de l'application
- `vendor/` : contient les dépendances de l'application
- `bin/` : contient les fichiers exécutables de l'application

Utiliser le serveur de développement

Symfony fournit un serveur de développement qui permet de tester l'application en local.

Pour lancer :

```
symfony serve
```

Pour le lancer en arrière plan :

```
symfony serve -d
```

Créer une route

Une route est une association entre une URL et un contrôleur.

Nous allons utiliser le `MakerBundle` pour créer une route.

```
symfony console make:controller RandomImageController
```

Vous pouvez maintenant accéder à la page en tapant l'URL suivante dans votre navigateur :

- `http://127.0.0.1:8000/random/image` (le port `8000` peut être différent).

Découvrir Twig - Un peu de théorie

Twig est un moteur de template qui permet de générer du code HTML.

Il est utilisé par Symfony pour générer les pages HTML.

Twig utilise des fichiers `.twig` qui sont des fichiers HTML avec des "balises" Twig.

Découvrir Twig - Syntaxe

Twig utilise deux types de balises :

- `{{ }}` : pour afficher une variable
- `{% %}` : pour exécuter du code

Découvrir Twig - syntaxe

Twig permet également des structures de contrôle :

- `if` : pour exécuter du code en fonction d'une condition
- `for` : pour exécuter du code en boucle
- `extends` : pour étendre un template
- `block` : pour définir un bloc de contenu
- `include` : pour inclure un autre template
- `set` : pour définir une variable

Découvrir Twig - Premiers pas

La commande précédente a créé deux fichiers :

- `RandomImageController.php` dans le dossier `src/Controller/` qui est en charge de gérer la route et de générer la réponse HTTP (en "transmettant" les données à Twig)
- `index.html.twig` dans le dossier `templates/random_image`.

Le fichier Twig fait référence à un autre fichier `base.html.twig` qui se trouve dans `templates/` : c'est le template de base de notre application !

Ajoutons Bootstrap à notre projet

Une grande partie du travail va être réalisé dans le fichier `base.html.twig` que l'on va ensuite étendre dans les autres fichiers Twig.

Modifions le pour ajouter Bootstrap via le CDN :

```
<html>
    <head>
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet" crossorigin="anonymous">

        // ...

    </head>
    <body>
        {% block body %}{% endblock %}
        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js" crossorigin="anonymous"></script>
    </body>
</html>
```

Des images aléatoires ?

Pour notre projet nous allons utiliser le site Lorem Picsum 🪄

Par exemple, l'URL suivante permet de générer une image de 300x200 pixels :

```
<img src="https://picsum.photos/300/200" alt="Image aléatoire" />
```

Pour générer plusieurs images dans le même format, il suffit d'ajouter un paramètre `?random` :

```
<img src="https://picsum.photos/200/300?random=1" alt="Image aléatoire" />
```

Notre "template"

Histoire de donner un peu de "style" à notre première application, nous utiliserons le modèle `Album`

Nous éclatons le template en plusieurs fichiers que nous inclurons dans notre `base.html.twig` :

- `_header.html.twig` (Les balises `<header>` du template)
- `_footer.html.twig` (Les balises `<footer>` du template)

Et notre contenu :

- `ìndex.html.twig` (Les balises `<main>` du template)

Ajoutons nos images aléatoires

Modifions le code du fichier `templates/random_image/index.html.twig` pour y ajouter nos images aléatoires !

Nous modifions le code entre les balise `<svg></svg>` par `<img src="https://picsum.photos/300/200?random=1">` (jusqu'à 9)

A chaque fois que vous rechargez la page, vous aurez des images différentes !

Refactorisons notre code !

OK ça marche, mais c'est pas très propre !

Déplacons une partie du code dans le contrôleur `RandomImageController.php`, notaement la génération des images aléatoires.

```php
class RandomImageController extends AbstractController
{
    #[Route('/random/image', name: 'app_random_image')]
    public function index(): Response
    {
        $images = [];

        for ($i=0; $i < 9; $i++) {
            $images[] = 'https://picsum.photos/300/200?random=' . $i;
        }

        return $this->render('random_image/index.html.twig', [
            'images' => $images,
        ]);
    }
}
```

Puis, modifions le fichier `templates/random_image/index.html.twig` pour utiliser les données du contrôleur.

Pour cela, nous allons utiliser les syntaxes Twig `{{ }}` / `{% %}` et la fonction `for` :

```
// ...

{% for image in images %}
    <img src="{{ image }}" alt="Image aléatoire" />
{% endfor %}

// ...
```

Conclusions

Nous avons vu comment créer une application Symfony, comment utiliser le serveur de développement, comment créer une route et comment utiliser Twig.

Nous avons également vu comment ajouter Bootstrap à notre projet et comment générer des images aléatoires.

C'est un premier pas vers la création d'applications web avec Symfony !

Bravo ! 🎉