

# Premiers pas avec Symfony 6

Le but de ce chapitre est de créer une nouvelle page affichant juste un message de bienvenue. Il faut donc :

- prévoir l'url qui permettra d'afficher cette page
- associer cette url à un contrôleur et une méthode de ce contrôleur
- implémenter le contrôleur et sa méthode
- créer la vue contenant les informations à afficher.

Il est nécessaire d'installer un composant supplémentaire avant de commencer :

```
composer require symfony/apache-pack
```

## Création et configuration de la base de données :

1) Sous mariadb, créer la base de données nommée webstudent

2) Configurer la connexion à la bdd dans le fichier .env

```
DATABASE_URL="mysql://root:@127.0.0.1:3307/webstudent?serverVersion=mariadb-10.6.5"
```

## Création d'un contrôleur :

Le contrôleur contient la logique applicative. Il utilise la couche modèle pour exploiter les données, effectue éventuellement des traitements et redirige vers une vue. Les contrôleurs sont situés dans le dossier ...\\src\\Controller et ont pour nom xxxController.

Créer le contrôleur EtudiantController avec la commande ci-dessous dans la console (à la racine du projet)

```
>php bin/console make:controller
```

Donner le nom EtudiantController. Vérifier la présence du nouveau fichier dans src/Controller. Symfony crée une vue par défaut en même temps dans le dossier templates/etudiant, vue nommée index.html.twig.


→ Fichiers créés : src/controller/EtudiantController.php + templates/etudiant/index.html.twig

## Tests :

Tester dans le navigateur <http://localhost/webstudent23/public/etudiant>

Note : Si erreur page non trouvée : Vérifier la présence du fichier .htaccess dans le dossier public. Si inexistant, copier ce fichier depuis le nas

## Analyse :

- 1) Analyser le fichier EtudiantController 
- 2) Analyser la vue templates/etudiant/index.html.twig

## Fichier de routes:

Il est possible de configurer les routes dans le fichier config/routes.yaml. Cela permet de centraliser toutes les routes dans un seul fichier plutôt que via les annotations au dessus de chaque méthode dans chaque contrôleur. Le fichier .../config/routes.yaml permet donc de définir un nom de route et de l'associer à une url et à un contrôleur et sa méthode. Précédemment l'url saisie était: <http://localhost/webstudent/public/etudiant>.

Dans le fichier de routes config/routes.yaml, chaque route est définie par un ensemble de 3 lignes :

- le nom de la route (ici app\_etudiant). Vous pouvez mettre ce que vous voulez mais ce nom doit être unique pour tout votre fichier.
- le path = la partie de l'url à saisir dans le navigateur derrière <http://localhost/webstudent/public> (ici /etudiant)
- le contrôleur et la méthode associée à la route (ici EtudiantController, méthode index).

a)mettre en commentaires la ligne 11 (//#[Route...])

b)configurer la route dans le fichier config/routes.yaml

```
controllers:
  resource:
    path: ../src/Controller/
    namespace: App\Controller
    type: attribute

app_etudiant:
  path: /etudiant
  controller: App\Controller\EtudiantController::index
```

c)Tester dans le navigateur

<http://localhost/webstudent23/public/etudiant>

## Activité

Le but de cette première activité est de créer une page d'accueil affichant « Bonjour les SIO – Quelles notes obtiendrez vous au BTS en 2024 ? » L'année sera un paramètre nommé pAnnee qui sera passé à la vue.

1) Dans le fichier de routes, créer une route dont l'url à saisir dans le navigateur sera <http://localhost/webstudent23/public/etudiant/accueil>. Cette route redirigera vers la méthode nommée accueil de EtudiantController

2) modifier le contrôleur EtudiantController pour ajouter la nouvelle méthode accueil.

```
namespace App\Controller;

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

class EtudiantController extends AbstractController
{
    /**
     * @Route("/etudiant", name="etudiant")
     */
    public function index()
    {
        //#[Route('/etudiant', name: 'app_etudiant')]
        public function index(): Response
        {
            return $this->render('etudiant/index.html.twig', [
                'controller_name' => 'EtudiantController',
            ]);
        }

        public function accueil(): Response
        {
            $annee = '2024';
            return $this->render('etudiant/accueil.html.twig', ['pAnnee' =>
$annee,
            ]);
        }
    }
}
```

La méthode accueil de ce contrôleur appelée par la route précédente, initialise une variable \$annee et renvoie vers une vue nommée accueil.html.twig en passant en paramètre la variable \$pAnnee.

## Création de la vue

Les vues avec symfony sont générées avec le moteur de templates Twig. Il s'agit de simples pages contenant essentiellement du Html et dans lesquelles des morceaux de code php peuvent être exécutés. Les vues doivent avoir l'extension .html.twig et se situer dans le dossier ../templates. du projet. Voici la vue renvoyée par le contrôleur ci-dessus et permettant d'afficher un simple message. Ce fichier nommé accueil.html.twig est créé dans le dossier ../templates/etudiant.

Fichier templates/etudiant/accueil.html.twig

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

{% block title %}Hello EtudiantController!{% endblock %}

{% block body %}
<style>
    .example-wrapper { margin: 1em auto; max-width: 800px; width: 95%; font:
18px/1.5 sans-serif; }
    .example-wrapper code { background: #F5F5F5; padding: 2px 6px; }
</style>

<div class="example-wrapper">

    Hello les SIO<br>
    Quelles notes obtiendrez vous au BTS en {{pAnnee}} ?
</div>
{% endblock %}
```

## Tests

- Saisir dans le navigateur l'url : <http://localhost/webstudent/public/etudiant/accueil>.
- La réponse attendue est :



From:  
<https://wiki.sio.bts/> - **WIKI SIO : DEPUIS 2017**

Permanent link:  
<https://wiki.sio.bts/doku.php?id=debutsymfony&rev=1699278289>

Last update: **2023/11/06 13:44**

