

SYMFONY

SYMFONY



Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMFONY

Table des matières

Liste des compétences du module :	6
1 Présentation du Framework :	7
1.1 Présentation Symfony :	7
2 Installation et prérequis du Framework :	7
2.1 Prérequis :	7
2.2 Installation de scoop :	7
2.3 Installation de CLI :	8
2.4 Installation de composer :	8
2.5 Exemple création de projets en ligne de commande :	8
2.6 Utilisation du serveur interne de Symfony :	9
3 Structure des projets :	11
3.1 Présentation de la structure des projets :	11
4 Routing avec Symfony :	12
4.1 Création de route avec routes.yaml :	12
4.2 Création de route avec les annotations :	14
4.3 Création de route avec maker-bundle :	15
4.4 Exercice :	18
5 Les templates Twig :	18
5.1 Block de code HTML :	18
5.2 Tester son interface :	19
5.3 Intégrer des assets (Bootstrap) à un projet :	20
5.4 Liaison fichier Bootstrap à base.html.twig :	20
5.5 Exercice :	20
6 ORM et Doctrine :	21
6.1 Création d'une base de données avec Symfony :	21

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMFONY

6.2 Modifier les entités (classes) et ajout des contraintes :	24
6.3 Configuration de l'accès à la base de données :	26
6.4 Effectuer une Migration :	27
6.5 Exercice :	28
7 ORM-Fixtures et Faker :	30
7.1 Installation de l'outil ORM-Fixtures :	30
7.2 Création de la classe AppFixtures.php :	30
7.3 Installation de la librairie Faker :	31
7.4 Intégration de Faker :	32
7.5 Exercice :	33
7.6 Sauvegarder des données en BDD :	34
8 Création d'une API (Micro-services):	35
8.1 MCD MLD du projet task :	35
8.2 Définitions :	36
8.3 Structure de l'API :	36
8.4 Création des controllers :	36
8.5 Créer les entités :	37
8.6 Configurer l'accès à la base de données :	38
8.7 Créer la base de données :	38
8.8 Paramétrage et migration :	39
8.9 Exercice :	40
9 Création d'une API suite (outils et Méthodes) :	40
9.1 Installation de Postman :	40
9.2 Edition des controllers :	41
9.3 Exercice :	42
9.4 Encodage en Json :	43

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMFONY

9.5 Sériàlisation :	44
9.6 Exercice :	48
10 API POST ajouter des enregistrements en BDD :	48
10.1 Ajout de la méthode create en POST :	48
10.2 Exercice :	53
11 Création d'une web-App :	53
11.1 Création d'un projet Web-App :	53
11.2 Configuration et intégration des dépendances du projet :	53
11.3 Ajouter une structure de données (entités, classe) :	54
11.4 Créer et migrer la base de données :	54
11.5 Exercice :	55
11.6 Ajoutez des données de tests :	56
11.7 Exercice :	56
12 Construire une page dynamique :	56
12.1 Génération d'un controller :	56
12.2 Modification de la méthode index :	57
12.2 Modification de l'interface twig :	57
12.3 Bloc FOR twig :	58
12.4 Exercice :	59
12.5 Intégration d'attribut et filtrage :	59
12.6 Afficher un seul enregistrement dans une interface :	60
12.7 Liaison bouton vers une route :	62
12.8 Exercice :	62
13 Création d'un panel Administrateur :	63
13.1 installation et création du Dashboard :	63
13.2 Configuration intégration des entités (CRUD) :	64

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMFONY

13.3 Exercice :	65
13.4 Configuration intégration des entités (CRUD) (suite) :	65
14 Créer un formulaire avec Symfony :	67
14.1 Générer un formulaire :	67
14.2 Configurer un formulaire ajouter des attributs non mappés :	68
14.3 intégrer le formulaire à un controller :	69
14.4 Ajouter le formulaire dans la vue :	69
14.5 Ajouter le champ submit :	70
14.5 Récupérer les données :	70
14.6 Sauvegarder les données en BDD :	71

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMFONY

Liste des compétences du module :

Etre capable de créer des pages Web Dynamique,

Etre capable de mettre en place un système d'API,

Etre capable de connecter une application serveur à une base de données côté Back-end,

Etre capable de gérer des requêtes HTTP d'interaction côté Back-end.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMFONY

1 Présentation du Framework :

1.1 Présentation Symfony :

Symfony est un ensemble de composants **PHP**, un **Framework** d'application Web, une philosophie et une communauté.

Un **Framework** permet aux développeurs de gagner du temps en réutilisant des modules génériques pour se concentrer sur d'autres domaines. Il va nous aider à développer mieux et plus vite !

Un **Framework** vous apporte la certitude que l'on développe une application parfaitement conforme aux règles métier, structurée, maintenable et évolutive.

C'est un ensemble de composants découplés et réutilisables sur lesquels sont construites les applications PHP : **Drupal**, **Prestashop** et **Laravel**.

2 Installation et prérequis du Framework :

2.1 Prérequis :

Windows 10,

Terminal (PowerShell, Gitbash),

Composer,

Scoop (gestionnaire de package),

Editeur de code Vscodé,

Xampp (Wamp, Mamp), apache, php et MySQL.

Navigateur web (Chrome, Edge, Firefox etc....)

2.2 Installation de scoop :

-Ouvrir le terminal PowerShell et saisir la commande suivante :

```
Set-ExecutionPolicy RemoteSigned  
irm https://get.scoop.sh | iex (new-object net.webclient).downloadstring('https://get.scoop.sh')
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMFONY

NB : si vous avez une erreur **SSL TLS** exécutez la commande suivante :

```
[Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12
Set-ExecutionPolicy RemoteSigned -scope CurrentUser
iwr -useb get.scoop.sh | iex
```

Lien issues : <https://github.com/ScoopInstaller/Scoop/issues/4002>

2.3 Installation de CLI :

Toujours depuis PowerShell saisir la commande ci-dessous :

```
scoop install symfony-cli
```

NB : avec Symfony nous n'avons pas à utiliser le serveur local (**Xampp, wamp**), il dispose de son propre serveur intégré.

2.4 Installation de composer :

Pour installer composer téléchargez le package **Composer-Setup.exe** à l'adresse ci-dessous :

<https://getcomposer.org/download/>

2.5 Exemple création de projets en ligne de commande :

2.5.1 Création d'un micro service en CLI:

Depuis PowerShell saisir la commande suivante :

```
symfony new micro_service
```

```
PS C:\> cd .\symphony\
PS C:\symphony> symfony new micro_service2
* Creating a new Symfony project with Composer
(running C:\ProgramData\ComposerSetup\bin\composer.phar create-project symfony/skeleton C:\symphony\micro_service2 --no-interaction)

* Setting up the project under Git version control
(running git init C:\symphony\micro_service2)

[OK] Your project is now ready in C:\symphony\micro_service2
```

2.5.2 Création d'une Web App en CLI :

Depuis PowerShell saisir la commande suivante :

```
symfony new website --webapp
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMFONY

```
PS C:\symphony> symfony new website --webapp
* Creating a new Symfony project with Composer
(running C:\ProgramData\ComposerSetup\bin\composer.phar create-project symfony/skeleton C:\symphony\website --no-interaction)

* Setting up the project under Git version control
(running git init C:\symphony\website)

(running C:\ProgramData\ComposerSetup\bin\composer.phar require webapp --no-interaction)

[OK] Your project is now ready in C:\symphony\website
```

2.5.3 Création d'un micro service avec composer :

Depuis PowerShell ou Gitbash saisir la commande suivante :

```
composer create-project symfony/skeleton microservice_composer
```

2.5.4 Création d'une Webapp avec composer :

```
composer create-project symfony/website-skeleton website_composer
```

NB : à la question docker taper **n** puis **entrée**.

2.6 Utilisation du serveur interne de Symfony :

Pour lancer un projet avec le serveur interne de Symfony :

Ouvrir un terminal et se positionner dans le dossier du site à exécuter (par ex website),

Saisir la commande suivante :

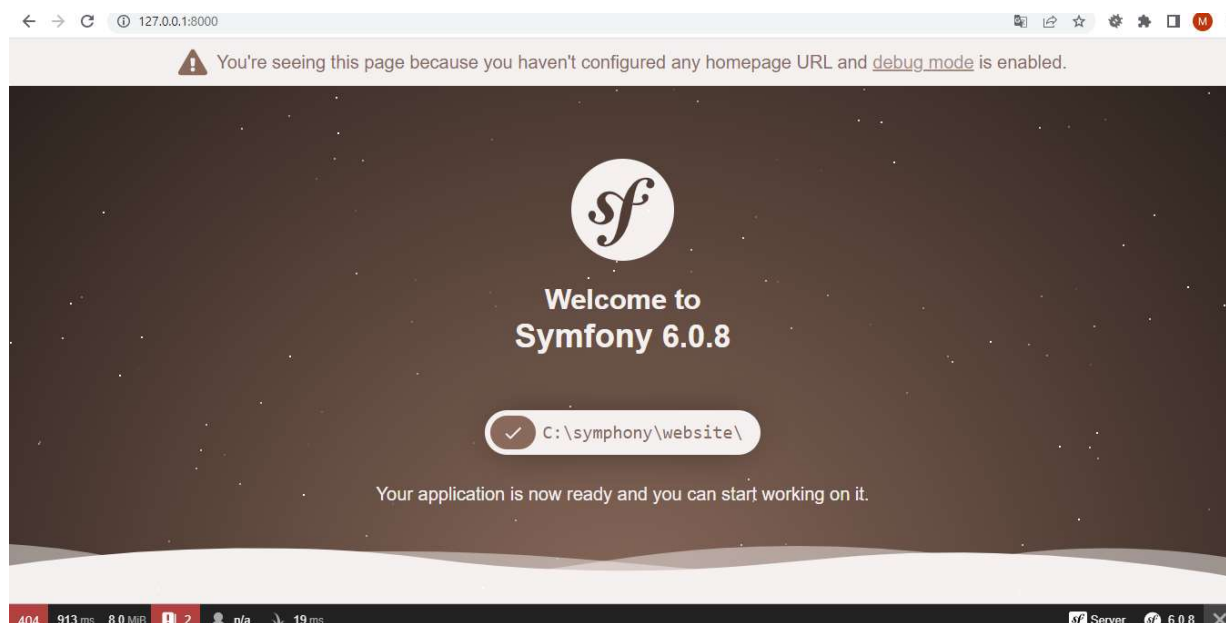
```
symfony server:start -d
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMFONY

Le site sera accessible dans le navigateur web à l'adresse suivante :

<http://127.0.0.1:8000>



Pour arrêter le serveur saisir la commande suivante :

`symfony server:stop`

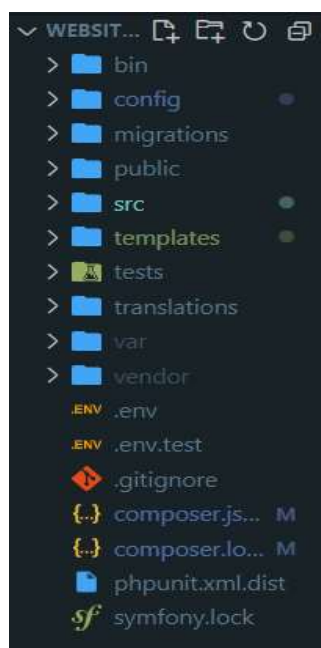
Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMFONY

3 Structure des projets :

3.1 Présentation de la structure des projets :

Les projets symfony sont structurés comme ci-dessous :



Dossier **bin** contient 2 fichiers :

Console (exécutable pour manager le projet),

Phpunit (exécutable pour le testing de l'application).

Dossier **config** (contient les informations de configuration du projet),

Dossier **migrations** (contient le code SQL du projet),

Dossier **public** (contient les fichiers et dossiers accessible au public),

Dossier **src** (contient le code source du projet, fichiers et dossier non accessible au public),

Dossier **templates** (contient le templates HTML des pages),

Dossier **test** (contient les tests du projet),

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMFONY

Dossier **translations** (contient les différentes traductions (langues)du projet),

Dossier **var** (contient les variables et les fichiers temporaires),

Dossier **vendor** (contient les dépendances du projet),

Fichier **.env** (contient les variables d'environnement).

Fichier **.env.test** (contient les variables des tests),

Fichier **.gitignore** (fichier git pour exclure les fichiers et dossier),

Fichier **composer.json** (fichier qui contient les dépendances du projet),

Fichier **composer.lock** (fichier qui contient les dépendances du projet),

Fichier **phpunit.xml.dist** (fichier de configuration des tests du projet),

Fichier **symfony.lock** (fichier de configuration des dépendances installées sur le projet).

4 Routing avec Symfony

Un des principes de base de Symfony, va être de rattacher nos **méthodes** (qui sont contenues dans nos classes **controller**) à des **routes** (à la différence de ce que l'on a vu précédemment avec PHP où nous rattachions les **routes** à des **controllers**).

Exemple :

Création d'une fonction qui va afficher bonjour dans une page web, elle sera accessible depuis la **route /bonjour**.

4.1 Création de route avec **routes.yaml**

Création d'un nouveau controller dans le répertoire **src/controller**

Intégrer le code ci-dessous :

```
<?php
```

```
namespace App\Controller;
```

```
use Symfony\Component\HttpFoundation\Response;
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMFONY

```
class HelloCtrl{
```

```
    public function sayHello():Response{
```

```
        return new Response("Bonjour");
```

```
    }
```

```
    public function sayHelloUtil($name):Response{
```

```
        return new Response("Bonjour ".$name);
```

```
    }
```

```
}
```

```
?>
```

Paramétrage de la route dans le fichier **routes.yaml**.

Ajouter le code comme ci-dessous :

HelloController:

```
path: /bonjour
```

```
controller : App\Controller\HelloCtrl::sayHello
```

HelloControllerName:

```
path: /bonjour/{toto}
```

```
controller : App\Controller\HelloCtrl::sayHelloUtil
```

Pour tester les routes veuillez saisir :

http://127.0.0.1/bonjour

Affiche *Bonjour* (dans la page)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMFONY

`http://127.0.0.1//bonjour/Adrar`

Affiche *Bonjour Adrar* (dans la page)

NB : le serveur web intégré de **symfony** doit être lancé préalablement avec la commande :

```
symfony server:start -d
```

4.2 Création de route avec les annotations :

Symfony permet de **générer automatiquement** les routes avec le système **d'annotations**.
 (Commentaire sur une fonction).

Pour pouvoir utiliser cette fonctionnalité nous allons devoir installer la **bibliothèque** avec la commande suivante (dans un **terminal**) :

```
composer require --dev annotations
```

Créer un nouveau contrôleur dans le dossier `src/controller`

```
<?php
```

```
//src/Controller/NameController.php
```

```
namespace App\Controller;
```

```
use Symfony\Component\HttpFoundation\Response;
```

```
use Symfony\Component\Routing\Annotation\Route;
```

```
class NameController
```

```
{
```

```
/**
```

```
 * @Route("/hello")
```

```
 */
```

```
public function Hello(): Response
```

```
{
```

```
return new Response(
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMFONY

```
'<html><body>Hello World</body></html>'

);

}

//fonction avec Paramètre (route/param)

/**
 * @Route("/hello/{name}")
 */

public function HelloUser($name): Response
{
    return new Response(
        '<html><body>Hello '.$name.'</body></html>'
    );
}

}

?>
```

Pour générer les routes nous utilisons la syntaxe suivante :

-@Route("/nom_route") pour une route simple,

-@Route("/nom_route/{param}") pour une route avec un paramètre.

4.3 Création de route avec **maker-bundle** :

Installer la bibliothèque **maker-bundle** avec la commande suivante (dans un terminal) :

symfony console make:controller

Nous devons saisir un nom de **controller** (exemple Accueil) :

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMFONY

```
$ symfony console make:controller

Choose a name for your controller class (e.g. TinyPopsicleController):
> Accueil|
```

La commande va générer automatiquement le controller Accueil et un template HTML :

```
created: src/Controller/AccueilController.php
created: templates/accueil/index.html.twig
```

Success!

Fichier **AccueilController** (src/controller):

```
<?php
```

```
namespace App\Controller;
```

```
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
```

```
use Symfony\Component\HttpFoundation\Response;
```

```
use Symfony\Component\Routing\Annotation\Route;
```

```
class AccueilController extends AbstractController
```

```
{
```

```
    #[Route('/accueil', name: 'app_accueil')]

```

```
    public function index(): Response

```

```
    {

```

```
        return $this->render('accueil/index.html.twig', [

```

```
            'controller_name' => 'AccueilController',

```

```
        ]);

```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMPHONY

```
}
}
```

Le fichier **AccueilController.php** contient : une méthode **index** qui va générer la **vue** avec la **fonction render** (qui possède **2 paramètres** en entrée : un **Template html twig** et un **tableau**).

Fichier **Template html twig** (**templates/accueil/index.html.twig**) :

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

{% block title %}Hello AccueilController!{% 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">

    <h1>Hello {{ controller_name }}! ✓</h1>

    This friendly message is coming from:

    <ul>

        <li>Your controller at <code><a href="{{
'C:/symphony/website_composer/src/Controller/AccueilController.php'|file_link(
0) }}">src/Controller/AccueilController.php</a></code></li>

        <li>Your template at <code><a href="{{
'C:/symphony/website_composer/templates/accueil/index.html.twig'|file_link(
0) }}">templates/accueil/index.html.twig</a></code></li>

    </ul>

</div>

{% endblock %}
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Mathieu MITHRIDATE	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMFONY

Le fichier **index.html.twig** (correspond à la **vue** dans le model **MVC**) contient le style **CSS** au sein d'une balise (**style**), les scripts **JS** (balise **script**) et le code **HTML**.

4.4 Exercice :

Dans la class **AccueilController**, en vous inspirant de la méthode **index** généré par **make-bundle**, ajoutez une nouvelle méthode **index2** qui va :

- prendre en paramètre (un nom).
- Récupérer le paramètre dans la fonction render (Vous allez devoir pour cela modifier le tableau en paramètre et ajouter à celui-ci une entrée **'test_name' => \$name**).
- Modifier le **template html** pour gérer l'affichage du nom dans la page.
- Créez une **route** pour appeler la fonction.

Le nom devra être récupéré dans la route comme ceci : **/accueil/nom**.

5 Les templates Twig :

Les templates Twig sont des fichiers qui vont contenir le code HTML et CSS de nos vues. Si nous avons généré un projet complet nous avons dans le dossier templates un fichier **base.html.twig**, celui-ci va contenir le code HTML.

Si le **package twig** n'est pas présent dans notre projet nous allons l'intégrer avec la commande suivante (dans un **terminal**) :

```
composer require --dev twig
```

5.1 Block de code HTML :

Les pages HTML contiennent des blocs de code avec la syntaxe suivante :

```
{% block title %}titre de la page{% endblock %}
```

Ces blocs (encadrés par des accolades) fonctionnent comme des variables, nous allons pouvoir les modifier en fonction de nos besoins en les surchargeant (réécriture de ceux-ci).

Dans la partie précédente (**routing**) nous avons généré une page avec **make-bundle (accueil)**.

Nous allons remplacer son contenu afin de comprendre l'utilisation des blocs.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMFONY

Balise extends :

La balise **extends** sert à indiquer depuis quel fichier elle doit hériter la structure html de base dans notre cas nous allons nous servir du fichier **base.html.twig** qui se trouve à la racine de templates.

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

Balise title :

La balise title permet de redéfinir le titre de notre page (comme en HTML). Elle s'ouvre avec **block** et se referme avec **endblock**.

```
{% block title %} Accueil {% endblock %}
```

Balise body :

La balise body permet de redéfinir le corps de la page (body en HTML). Elle s'ouvre avec **block** et se referme avec **endblock**. (Elle va **surcharger** le body de **base.html.twig**)

```
{% block body %}
```

```
<h1>
```

```
Bienvenue sur La page Accueil
```

```
</h1>
```

```
{% endblock %}
```

5.2 Tester son interface :

Pour tester notre **interface** nous allons revenir dans le **controller accueil** et réécrire la méthode **index** comme ci-dessous :

```
/*Attention !!! pensez à renommer le name (app_accueil en app_accueil2) si vous avez plusieurs routes dans votre controller */
```

```
#[Route('/home', name: 'app_accueil2')]
```

```
public function index(): Response
```

```
{
```

```
return $this->render('accueil/index.html.twig');
```

```
}
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

SYMFONY

La page est accessible à l'adresse suivante :

localhost :8000/home ou **localhost :8000/accueil**

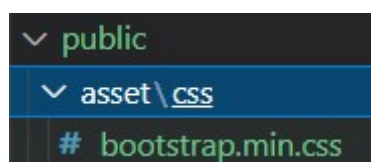
5.3 Intégrer des assets (Bootstrap) à un projet :

Pour utiliser des **thèmes Bootstrap** dans un projet nous allons les télécharger depuis le site ci-dessous :

<https://bootswatch.com/>

Nous allons créer à l'intérieur du répertoire **public** (à la **racine** du projet) un dossier **asset->css**

Nous déplacerons le fichier **bootstrap.min.css** téléchargé précédemment à l'intérieur de celui-ci :



5.4 Liaison fichier Bootstrap à base.html.twig :

Pour **lier** le fichier **style CSS** à notre page nous allons intégrer le code ci-dessous :

```
{% block stylesheets %}
```

```
<link rel="stylesheet" href="../asset/css/bootstrap.min.css">
```

```
{% endblock %}
```

NB : Le fichier **style CSS** se trouve dans le répertoire **public/asset/css/**.

5.5 Exercice :

Recréer la vue **index.html.twig** en intégrant :

-Un Menu Bootstrap, (Menu, liens (Accueil, Présentation, Prix et A propos))

-Un titre H1 avec Bienvenue sur la page d'accueil

Un titre H1 avec Bienvenue sur la page d'accueil **prénom** (version qui récupère le prénom dans la **route**)

Réaliser les 2 versions (avec et sans le prénom : 2 méthodes dans la classe **AccueilController**)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu MITHRIDATE</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		