

# Symfony

Alexzandre  
LE HELLAY

---

<b>Installation du framework.....</b>	<b>2</b>
<b>Nos premières pages.....</b>	<b>2</b>
Création d'une route.....	2
Exemple.....	3
Autre.....	3
<b>Moteur de template Twig.....</b>	<b>4</b>
Exemple.....	5
Nav Url.....	5
Exemple.....	6
Include.....	6
<b>L'ORM Doctrine.....</b>	<b>6</b>
Creation d'une entity.....	7
<b>Les formulaires.....</b>	<b>7</b>
<b>Valider les données.....</b>	<b>7</b>

# Installation du framework

```
composer create-project symfony/skeleton:"7.0.*" nom_du_projet
```

```
cd nom_du_projet
```

```
composer require webapp
```

```
y
```

Les Différent dossier

.env

Piloter les variable d'environnement (ex: choix de la base de donnée)

public

Dossier Racine

Démarrage d'un serveur

```
php -S localhost:8000 -t public
```

## Nos premières pages

```
php bin/console make:controller nom_du_controle [HomeController]
```

Ça vas crée deux chose

- src/Controle/Nom\_du\_controller.php [src/Controle/HomeController.php]
- templates/Nom\_du\_controller/index.html.twig [templates/Home/index.html.twig]

```
src/Controller/Nom_du_controller.php [src/Controller/HomeController.php]
```

Supprimer la fonction par défaut :

Ajout d'un text sur une page vide

```
Fonction index (): Response {  
    Return new Response('Bonjour les gens');  
}
```

## Création d'une route

```
config/routes.yaml
```

Ajout d'une nouvelle route

```
home:  
    path: /  
    controller: App\Controller\Nom_du_controller::index [App\Controller\HomeController::index]
```

**Ou**

```
[src/Controller/HomeController.php]
```

```
Class HomeController {  
    #[Route("/", name: "home")]  
  
    Function ...  
}
```

Connaitre tout les route

php bin/console debug:router

## Exemple

Localhost 8000/?name=john

src/Controller/HomeController.php

```
Class HomeController extends AbstractController {  
    #[Route("/", name: "home")]  
    Function index (): Response {  
        Return new Response('Bonjour ' . $_GET['name']);  
    }  
}
```

### Avancer

```
Function index (Request $request): Response {  
    dd($request);  
    Return new Response('Bonjour ' . $request->query->get('name', 'Inconnu'));  
}
```

## Autre

Php bin/controler make:controller RecipeController

src/Controller/RecipeController.php

```
Class RecipeController extends AbstractController {  
    #[Route('/recette/{slug}-{id}', name: 'recipe.show')]  
  
    Public Function index(Request $request): Response  
    {  
        dd($request->attributes->get('slug'), $request->attributes->get('id'))  
    }  
}
```

```
Class RecipeController extends AbstractController {  
    #[Route('/recette/{slug}-{id}', name: 'recipe.show', requirements: ['id' => '\d+', 'slug' => '[a-z0-9-]+' ])]  
  
    Public Function index(Request $request): Response  
    {  
        dd($request->attributes->get('slug'), $request->attributes->getInt('id'))  
    }  
}
```

```

Class RecipeController extends AbstractController {
    #[Route('/recette/{slug}-{id}', name: 'recipe.show', requirements: ['id' => '\d+', 'slug' => '[a-z0-9-]+'
    ])

    Public Function index(Request $request, string $slug, int $id): Response
    {
        dd($slug, $id);
    }
}

```

src/Controller/RecipeController.php

```

Class RecipeController extends AbstractController {

    #[Route('/recette', name: 'recipe.index')]

    Public Function index(Request $request): Response
    {
        return new Response('Recettes');
    }

    #[Route('/recette/{slug}-{id}', name: 'recipe.show', requirements: ['id' => '\d+', 'slug' => '[a-z0-9-]+'
    ])
    {
        Return new Response('Recette : ' . $slug);
        Ou
        Return new JsonResponse([
            'slug' => $slug
        ]);
        Ou
        Return $this->json([
            'slug' => $slug
        ]);
    }
}

```

## Moteur de template Twig

template/recipe/index.html.twig

```

{% extends 'base.html.twig' %} // la référence de tout les pages

{% block title %} blablabla {% endblock %}

Ou

{% block title "Toutes les recettes" %}

{% block body %}
azdazhjdajdoiajdoazjo
{% endblock %}

```

src/Controller/RecipeController.php

```

Public ...
{
    Return $this->render('recipe/index.html.twig');
}

```

```
}
```

Modification de la page template/base.html.twig pour modifier tout les autre page en ajoutant le css ou une nav bar par défaut.

## Exemple

Creation d'un fichier dans template/recipe [show.html.twig]

Modification de Controller/RecipeController

```
#[Route('/recette/{slug}-{id}', name: 'recipe.show', requirements: ['id' => '\d+', 'slug' => '[a-z0-9-]+'  
)]  
{  
    Return $this->render('recipe/show.html.twig', [  
        'slug' => $slug  
        'id' => $id  
  
        'person' => [  
            'firstname' => 'John',  
            'lastname' => 'Doe'  
        ]  
    ]);  
}
```

Modification de la page show.html.twig

```
Ajout  
{% block title "Recette : " ~ slug %}  
  
{% block body %}  
    {{ slug }} {{ person.firstname ~ person.lastname}}  
{% endblock%}
```

Les information sont échapper ex: <strong> sera écrit en tout lettre et pas considérer comme une balise  
OU  
Ajouter le filtre raw

Ajout de filtre  
<https://twig.symfony.com/doc/3.x/>

```
{{ slug | upper }} //majuscule
```

## Nav Url

On a le choix entre url ou path (préférence sur path)

```
<a class="navbar-brand" href="/">Accueil </a> // nop
```

```
<a class="navbar-brand" href="{{ path('home') }}">Accueil </a> //home est le name de ma route
```

Note : 13:01

## Exemple

recipe/index.html.twig

```
{% block body %}
    <ul>
        <li>
            <a href="{{ path('recipe.show', {id: 32, slug: 'pate-bolognaise'}) }}">pâtes bolo </a>
        </li>
    </ul>
{% endblock %}
```

base.html.twig

```
<ul>
    <li>
        <a class="nav-link {{ app.current_route == 'home' . 'active' : '' }}" href="{{
            path('hom') }}">Accueil</a>
    </li>
    <li>
        <a class="nav-link {{ app.current_route starts with 'recipe.' ? 'active' : '' }}" href="{{
            path('recipe.index') }}">Recettes</a>
    </li>
</ul>
```

## Include

## L'ORM Doctrine

Base de donnée gérer au niveau .env

Téléchargement de adminer qu'on va aller le placer dans le dossier public et le nommer (adminer.php)

<https://www.adminer.org/en/#download>

### Downloads

- [Adminer 4.8.1](#) (.php, 465 kB), [English only](#) (.php, 310 kB)
- [Adminer 4.8.1 for MySQL](#) (.php, 356 kB), [English only](#) (.php, 208 kB)
- [Source codes](#) (.zip, 785 kB), [Current development version](#)
- 💰 Donate: [Paypal](#), [Patreon](#), [Revolut](#)
- Latest stable version (use e.g. by wget): [https://www.adminer.org/latest\[-mysql\]\[-en\].php](https://www.adminer.org/latest[-mysql][-en].php)
- [Change log](#), [blog](#)
- User contributed packages: [Debian package](#), [Arch Linux package](#), [Wordpress plugin](#), [Drupal module](#), [Docker](#), [Dockette](#), [Joomla extension](#), [Moodle plugin](#), [TYPO3 extension](#), [CMS Made Simple Module](#), [Laravel](#), [Laravel](#), [Laravel](#), [AMPPS](#), [Electron](#), [Jaxon](#)
- Adminer is also bundled with [Nette Framework](#) (which this site runs on).
- [Older versions](#)

Puis allez sur localhost:8000/adminer.php

#### Exemple

Système : MySQL

Serveur : 127.0.0.1

Username : root

Password : password

Database : main

Vérification de la version :

```
SELECT VERSION();
```

Changer dans .env

```
DATABASE_URL = "mysql://root:password@127.0.0.1:3306/main?serverVersion=8.0.31..."
```

## Création d'une entité

## Les formulaires

## Valider les données

