



# Formation PHP 7 Symfony 4.3

Hajer ALAYA

[Hajer.adahmeni@gmail.com](mailto:Hajer.adahmeni@gmail.com)

Du 15/07/2019 Au 19/07/2019



# Plan

- Les Contrôleurs et la gestion des URL
  - Les contrôleurs
  - Les routes
- La Vue et le moteur de Template Twig

## Création des contrôleurs

- Le contrôleur est la couche qui va exécuter les traitements liés à notre application et transmettre les résultats à la vue pour que ces derniers puissent être affichés à l'utilisateur.
- Les contrôleurs sont des classes qui vont contenir des fonctions qui seront appelées en fonction d'une URL. Ces dernières renverront un objet de type `Symfony\Component\HttpFoundation\Response` qui est l'abstraction d'une réponse HTTP dans Symfony.

## Création des contrôleurs

- Ajouter une classe dans le dossier src/Controller nommé DefaultController

```
<?php
namespace App\Controller;
use Symfony\Component\HttpFoundation\Response;
class DefaultController
{
    public function index(): Response
    {
        return new Response('Accueil JobRapid - Hello');
    }
}
```

## Création des contrôleurs

- Pour tester la classe `DefaultController.php`, il faut configurer la route au niveau du fichier `config/routes.yaml`.
- Ajouter alors le code suivant :

```
# config/routes.yaml  
index:  
  path: /  
  defaults: { _controller: 'App\Controller\DefaultController::index' }
```

- Démarrer votre serveur web et lancer <http://localhost:8000/>:

```
php -S 127.0.0.1:8000 -t public
```

## Création des contrôleurs

- Symfony utilise un composant [HttpFoundation](#) qui fournit une couche objet permettant de travailler avec le protocole HTTP. : L'objet Response et l'objet Request

```
<?php // src/Controller/DefaultController.php
namespace App\Controller;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
class DefaultController
{
    public function index(Request $request): Response
    {
        return new Response('Accueil Jobeet - Hello '.$request->get('name', 'World'));
        //la méthode Request::get pour récupérer un paramètre de la requête
    }
}
```

- *AbstractController met à disposition une méthode **render** qui récupérera automatiquement une instance de Twig et créera notre objet Response en un appel de méthode*
- *Simplifie le format de passage des paramètres à twig*

```
<?php // src/Controller/DefaultController.php
namespace App\Controller;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;

class DefaultController extends AbstractController
{
    public function index(Request $request): Response
    {
        return $this->render('home.html.twig', ['name' => $request->get('name', 'World')]);
    }
}
```

- Démarrer votre serveur web et lancer <http://localhost:8000/>:

```
php -S 127.0.0.1:8000 -t public
```



## La vue avec Twig

- La couche vue renvoie et met en forme les informations qui seront affichées à l'utilisateur.
- Twig est principalement utilisé. On active la dépendance via :

```
composer require symfony/twig-bundle
```

- Après avoir installer Twig on va créer et afficher notre premier Template `home.html.twig` :  

```
#templates/home.html.twig  
Hello {{ name }}
```
- Modifier la classe `DefaultController.php`

## La vue avec Twig

- Ajouter classe JobController.php
- Ajouter la Template index.html.twig
- Ajoutons la dépendance des « asset »

```
composer require symfony/asset
```

- Configurer la route au niveau du fichier config/routes.yaml.

```
# config/routes.yaml
```

```
index:
```

```
  path: /
```

```
  defaults: { _controller: 'App\Controller\JobController::index' }
```

- Démarrer votre serveur web et lancer <http://localhost:8000/>:

```
//gestion des offres
<?php // src/Controller/JobController.php
namespace App\Controller;
use App\Entity\Job;
use Doctrine\ORM\EntityManagerInterface;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
class JobController extends AbstractController
{
    public function index(EntityManagerInterface $em)
    {
        $jobs = $em->getRepository(Job::class)->findAll();

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

- Ajoutons la vue qui va afficher nos résultats

```
<!-- templates/job/index.html.twig -->
<!DOCTYPE html>
<html> <head>
  <meta charset="UTF-8" />
  <title>JobRapid - Liste des jobs</title>
  <link rel="icon" type="image/x-icon" href="{{ asset('favicon.ico') }}" />
</head>
<body> <ul>
  {% for job in jobs %}
    <li>{{ job.position }}</li>
  {% endfor %}
</ul>
</body>
</html>
```

## La vue avec Twig

- Ouvrir la page **base.html.twig** : il s'agit du Template globale de notre projet
- Twig fonctionne sur le principe **d'héritage**.
- Twig utilise un système de **blocs** (block) pour définir des templates qui contiendront des éléments qui pourront ensuite être surchargés dans d'autres templates.
  - Modifier alors les pages index.html.twig et base.html.twig comme suite
- Démarrer votre serveur web et lancer <http://localhost:8000/>:

## La vue avec Twig

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

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

```
<ul>
```

```
    {% for job in jobs %}
```

```
        <li>{{ job.position }}</li>
```

```
    {% endfor %}
```

```
</ul>
```

```
{% endblock %}
```

```
<!-- templates/job/index.html.twig -->
```

```
<!-- templates/base.html.twig -->
```

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

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

```
    <h1 class="my-4">Liste des offres</h1>
```

```
    {% for job in jobs %}
```

```
        <div class="row">
```

```
            <div class="col-md-7">
```

```
                <a href="#">
```

```
                    
```

```
                </a>
```

```
            </div>
```

```
            <div class="col-md-5">
```

```
                <h3>{{ job.position }}</h3>
```

```
                <p>{{ job.description }}</p>
```

```
                <p>Posted on {{ job.createdAt|date("m/d/Y") }}</p>
```

```
                <a class="btn btn-primary" href="#">See more</a>
```

```
            </div>
```

```
        </div>
```

```
        <hr>
```

```
    {% endfor %}
```

```
<ul class="pagination justify-content-center">
  <li class="page-item disabled">
    <a class="page-link" href="#" aria-label="Previous">
      <span aria-hidden="true">&laquo;</span>
      <span class="sr-only">Previous</span>
    </a>
  </li>
  <li class="page-item">
    <a class="page-link" href="#">1</a>
  </li>
  <li class="page-item disabled">
    <a class="page-link" href="#" aria-label="Next">
      <span aria-hidden="true">&raquo;</span>
      <span class="sr-only">Next</span>
    </a>
  </li>
</ul>
{% endblock %}
```



- Ajouter dans le projet JobRapid JobControlle
- L'EntityManager est l'objet Doctrine qui permet de manipuler nos entités
- Doctrine fournit un ensemble de méthodes par défaut permettant de récupérer nos objets persistés. Signalons entre autres les fonctions [find](#), [findBy](#), findOneByet findAll.

```
<?php // src/Controller/JobController.php
namespace App\Controller;
use App\Entity\Job;
use Doctrine\ORM\EntityManagerInterface;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;

class JobController extends AbstractController
{
    public function index(EntityManagerInterface $em)
    {
        $jobs = $em->getRepository(Job::class)->findAll();

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

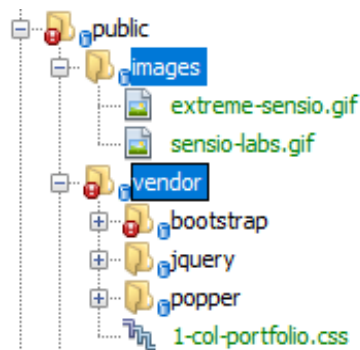
## La vue avec Twig

- Ajouter un thème de mise en forme en utilisant le framework [bootstrap](#) au travers de dossier



Bootstrap.rar

- Décompresser le dossier sous le répertoire public

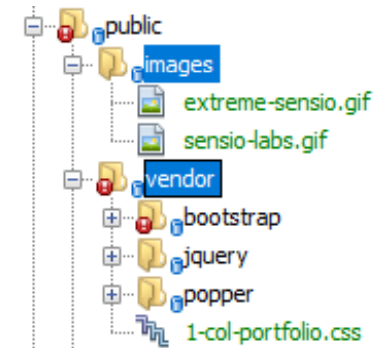


# La couche Modèle du Framework



## La vue avec Twig

- Ajouter un thème de mise en forme on utilisant le framework [bootstrap](#) au travers de dossier   
Bootstrap.rar
- Décompresser le dossier sous le répertoire public
- Modifier les pages [base.html.twig](#) et [index.html.twig](#) pour ajouter les références à la mise en forme en Bootstrap



"base.html.twig"

```
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>{% block title %}Jobeet{% endblock %}</title>
    <link rel="stylesheet" href="/vendor/bootstrap/css/bootstrap.min.css">
    <link rel="stylesheet" href="/vendor/1-col-portfolio.css">
    {% block stylesheets %}{% endblock %}
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
      <div class="container">
        <a class="navbar-brand" href="#">Jobeet</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse"
          data-target="#navbarResponsive" aria-controls="navbarResponsive"
          aria-expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
      </div>
    </nav>

    <div class="container">
      {% block body %}{% endblock %}
    </div>

    <script src="/vendor/jquery/jquery-3.2.1.slim.min.js"></script>
    <script src="/vendor/popper/popper.min.js"></script>
    <script src="/vendor/bootstrap/js/bootstrap.min.js"></script>
    {% block javascripts %}{% endblock %}
  </body>
</html>
```

"base.html.twig"

```
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>{% block title %}Jobeet{% endblock %}</title>
    <link rel="stylesheet" href="/vendor/bootstrap/css/bootstrap.min.css">
    <link rel="stylesheet" href="/vendor/1-col-portfolio.css">
    {% block stylesheets %}{% endblock %}
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
      <div class="container">
        <a class="navbar-brand" href="#">Jobeet</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse"
          data-target="#navbarResponsive" aria-controls="navbarResponsive"
          aria-expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
      </div>
    </nav>

    <div class="container">
      {% block body %}{% endblock %}
    </div>

    <script src="/vendor/jquery/jquery-3.2.1.slim.min.js"></script>
    <script src="/vendor/popper/popper.min.js"></script>
    <script src="/vendor/bootstrap/js/bootstrap.min.js"></script>
    {% block javascripts %}{% endblock %}
  </body>
</html>
```

```

{% extends "base.html.twig" %}
{% block body %}
<h1 class="my-4">Liste des offres</h1>
{% for job in jobs %}
  <div class="row">
    <div class="col-md-7">
      <a href="#">
        
      </a>
    </div>
    <div class="col-md-5">
      <h3>{{ job.position }}</h3>
      <p>{{ job.description }}</p>
      <p>Posted on {{ job.createdAt|date("m/d/Y") }}</p>
      <a class="btn btn-primary" href="#">See more</a>
    </div>
  </div>
  <hr>
{% endfor %}
<ul class="pagination justify-content-center">
  <li class="page-item disabled">
    <a class="page-link" href="#" aria-label="Previous">
      <span aria-hidden="true">&laquo;</span>
      <span class="sr-only">Previous</span>
    </a>
  </li>
  <li class="page-item">
    <a class="page-link" href="#">1</a>
  </li>
  <li class="page-item disabled">
    <a class="page-link" href="#" aria-label="Next">
      <span aria-hidden="true">&raquo;</span>
      <span class="sr-only">Next</span>
    </a>
  </li>
</ul>
{% endblock %}

```