

Résumé « Day 1 »

Symfony 3.x est un framework avec des composants, je peux l'installer avec **composer** en utilisant :

- ▶ wget <https://getcomposer.org/download/1.6.2/composer.phar>
- ▶ php composer.phar create-project symfony/framework-standard-edition my_cv "3.4.*"

Par la suite, nous avons intégré un template css/js dans Symfony, pour cela :

1. Copier les répertoires « js », « css », « img » et « fonts » dans un répertoire que je nomme « template » dans le répertoire web de Symfony. Cela va me permettre d'accéder à ces ressources directement en utilisant http://<mon_site_web>/web/template/css/style.css
2. Simplifier mon contrôleur pour ne renvoyer qu'un tableau vide :

```
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Template;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Symfony\Component\HttpFoundation\Request;

class DefaultController extends Controller
{
    /**
     * @Route("/", name="homepage")
     * @Template()
     */
    public function indexAction(Request $request)
    {
        return array();
    }
}
```

Symfony suit le modèle MVC <https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-contr%C3%B4leur> pour plus d'informations sur les contrôleurs. Comme dans le contrôleur, j'ai rajouté l'annotation « @Template », je n'oublie pas de l'inclure avec « use ». <http://symfony.com/doc/current/bundles/SensioFrameworkExtraBundle/annotations/view.html>

3. Créer un fichier qui sera ma vue associée à ce contrôleur : dans src\AppBundle\Resources\views\Default\index.html.twig

« Default » correspond au nom du contrôleur et le nom « index » du fichier correspond au nom de l'action dans le contrôleur.

4. Dans ce fichier, je vais utiliser le langage twig <https://twig.symfony.com/> pour générer du code html :

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

{% block body %}

    <Mon code HTML>

{% endblock body %}
```

5. Mon fichier « **base.html.twig** » me permet d'éviter de dupliquer du code et de le centraliser dans le répertoire `app/Resources/views`

```
base.html.twig
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8" />
5     <title>{% block title %}Welcome!{% endblock %}</title>
6     {% block stylesheets %}{% endblock %}
7     <link rel="icon" type="image/x-icon" href="{{ asset('favicon.ico') }}" />
8     <!-- favicon -->
9     <link href="favicon.png" rel="icon">
10    <!-- web-fonts -->
11    <link href="https://fonts.googleapis.com/css?family=Hind:300,400,500,600,700" rel="stylesheet">
12    <!-- font-awesome -->
13    <link href="{{ asset('template/css/font-awesome.min.css') }}" rel="stylesheet">
14    <!-- Bootstrap -->
15    <link href="{{ asset('template/css/bootstrap.min.css') }}" rel="stylesheet">
16    <!-- owl carousel -->
17    <link href="{{ asset('template/css/owl.carousel.css') }}" rel="stylesheet">
18    <!-- Style CSS -->
19    <link href="{{ asset('template/css/style.css') }}" rel="stylesheet">
20  </head>
21  <body>
22    {% block body %}{% endblock %}
23    {% block javascripts %}
24      <!-- jquery -->
25      <script src="{{ asset('template/js/jquery-2.1.4.min.js') }}"></script>
26      <!-- Bootstrap -->
27      <script src="{{ asset('template/js/bootstrap.min.js') }}"></script>
28      <!-- owl carousel -->
29      <script src="{{ asset('template/js/owl.carousel.min.js') }}"></script>
30      <!-- theme script -->
31      <script src="{{ asset('template/js/scripts.js') }}"></script>
32    {% endblock %}
33  </body>
34 </html>
```

Je dispose ainsi de 4 « blocks » : « **title** », « **stylesheets** », « **body** », « **javascripts** » que je peux réécrire à ma guise dans d'autres pages en faisant un « **extends** » twig et en spécifiant le « **block** » que je souhaite réécrire comme dans le fichier `index.html.twig`

6. Enfin, pour que les fichiers « js », « css » et « img » fonctionnent dans mes pages je dois indiquer le répertoire « `web/template` » créé précédemment dans les liens html « `src` » qui les utilisent en utilisant la directive « `asset` ».

https://symfony.com/doc/3.4/best_practices/web-assets.html

BONUS

Je peux modifier mon contrôleur :

```
class DefaultController extends Controller
{
    /**
     * @Route("/test/{name}/{firstname}", name="homepage")
     * @Template()
     */
    public function indexAction($name, $firstname)
    {
        return array('name' => $name, 'firstname' => $firstname);
    }
}
```

Et utiliser les variables du contrôleur dans mon template twig de la façon suivante :

```
<div class="name-wrapper">
  <h1 class="name">{{ firstname }} {{ name }}</h1>
  <span>Freelance Web Designer</span>
</div>
```