

TD Symfony

Étape 4 : Structurer les vues

Dans le TD précédent nous avons créé un template twig : /test/name.html.twig dans lequel nous avons simplement mis une balise h1. Cette page ne respecte pas les standards W3C, elle n'est pas valide !

<https://www.pierre-giraud.com/html-css-apprendre-coder-cours/structure-minimale-page/>

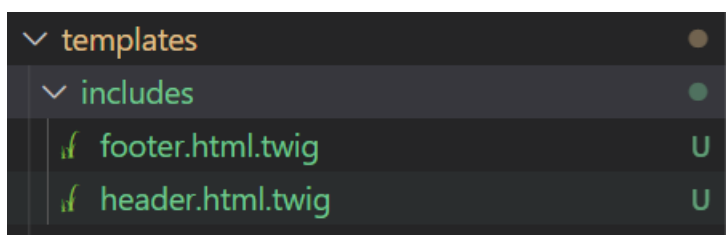
<https://www.alsacreations.com/article/lire/1374-html5-structure-globale-document.html>

Lorsque nous avons créé notre projet symfony, un template base.html.twig à été créé, ce fichier va servir de « squelette » de base à toutes les pages de notre application. Chaque template que nous allons créer devra être intégré dans le bloc body de cette page.

Commençons par modifier ce template base en déplaçant le bloc javascripts pour le mettre en bas de page html juste avant la fermeture de body.

```
templates > if base.html.twig
8      {% block stylesheets %}
9          #{encore_entry_link_tags('app')}#}
10     {% endblock %}
11
12     </head>
13     <body>
14         {% block body %}{% endblock %}
15
16         {% block javascripts %}
17             #{encore_entry_script_tags('app')}#}
18         {% endblock %}
19     </body>
20 </html>
```

Nous allons structurer notre page avec un header et un footer. Créons un sous dossier includes dans template et ajoutons nos 2 fichiers twig. Nous marquons juste «HEADER» et «FOOTER» dans ces fichiers pour l'instant (afin de visualiser quelque chose dans le navigateur)



Puis nous allons les inclure dans notre template de base.

```

templates > if base.html.twig
13 | <body>
14 |
15 |     {% block header %}{% include "includes/header.html.twig" %}{% endblock %}
16 |
17 |     {% block body %}{% endblock %}
18 |
19 |     {% block footer %}{% include "includes/footer.html.twig" %}{% endblock %}
20 |
21 |     {% block javascripts %}
22 |         {{ encore_entry_script_tags('app') }}
23 |     {% endblock %}
24 | </body>
25 | </html>

```

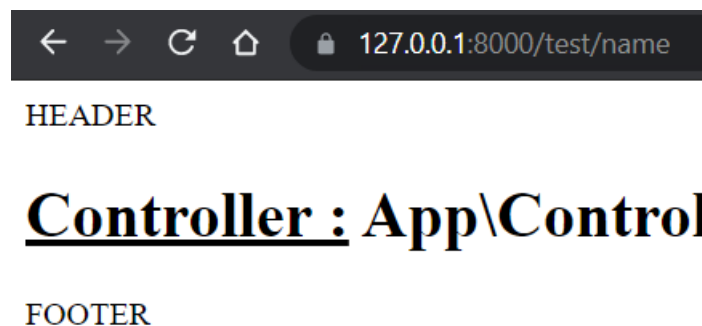
Il nous reste à modifier le template de notre route /test/name en indiquant quelle est la page de base pour ce template (L1 :extends – ci-dessous) En encadrant le contenu de ce template avec le bloc body, ce contenu viendra s'insérer dans le bloc body de la page de base.

```

templates > test > if name.html.twig
1 | {% extends 'base.html.twig' %}
2 |
3 | {% block body %}
4 | <h1>
5 |     <u>Controller :</u> {{ className }} - <u>Action Called :</u> {{ actionName }}
6 | </h1>
7 | {% endblock %}

```

Résultat :



← → ↺ 🏠 🔒 127.0.0.1:8000/test/name

HEADER

Controller : App\Control

FOOTER

Le bloc body correspond au contenu de la vue à insérer dans la page de base. Pour plus de clarté, nous renommons ce bloc en « content » dans le template de base et donc dans notre template /test/name.base.html

```

templates > if base.html.twig
14 |
15 |     {% block header %}{% include "inc
16 |
17 |     {% block content %}{% endblock %}
18 |
19 |     {% block footer %}{% include "inc
20 |

```

```

templates > test > if name.html.twig
1 | {% extends 'base.html.twig' %}
2 |
3 | {% block content %}
4 | <h1>

```

Nous allons maintenant inclure Bootstrap dans notre projet. La façon la plus simple et intuitive de le faire est de récupérer les liens cdn du css et du js sur le site de Bootstrap (V 5.1) et de les lier à notre page de base.

<https://getbootstrap.com/docs/5.1/getting-started/introduction/>

Nous plaçons le lien css dans le bloc stylesheets et le script du bundle js dans le bloc javascripts de notre page de base.

```

templates > if base.html.twig
8 | {% block stylesheets %}
9 |   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css" rel
10 |   #{encore_entry_link_tags('app')}>
11 | {% endblock %}

```

```

templates > if base.html.twig
22 | {% block javascripts %}
23 |   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.bundle.min.js" int
24 |   #{encore_entry_script_tags('app')}>
25 | {% endblock %}
26 | </body>

```

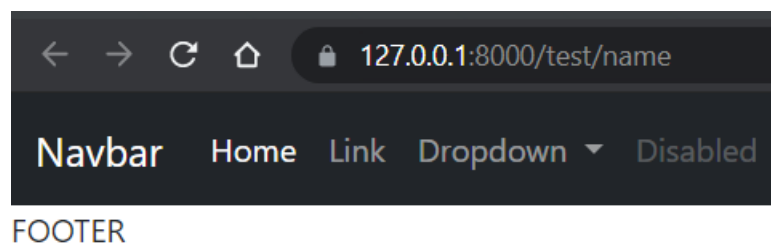
Nous allons utiliser une navbar Bootstrap (Récupérez le code sur le site Bootstrap) que nous allons customiser.

```

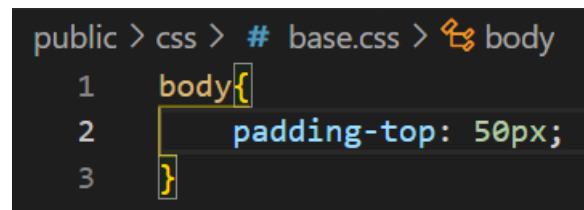
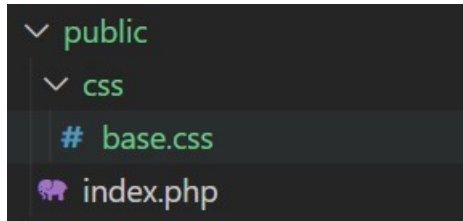
templates > includes > if header.html.twig
1 | <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
2 |   <div class="container-fluid">
3 |     <a class="navbar-brand" href="#">Navbar</a>

```

Testons avant de continuer ...

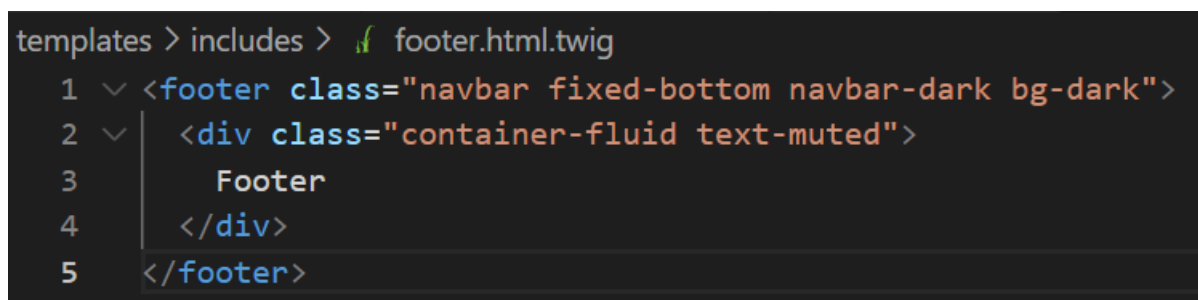


Le contenu de notre page /test/name est masqué par la navbar ! Nous allons remédier à cela en ajoutant un padding-top à notre balise body. Nous créons donc notre fichier css dans public/css (nous l'appelons base.css) nous ajoutons notre propriété css sur la balise body et nous faisons le lien entre base.html.twig et notre fichier css.

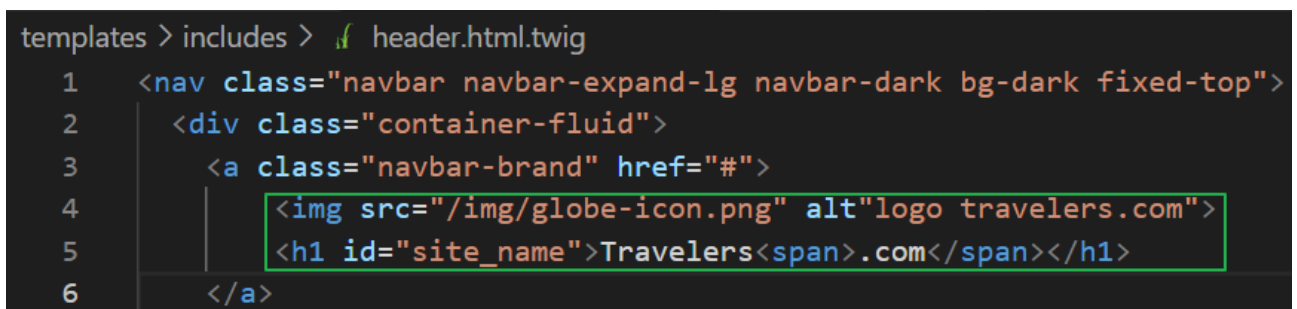
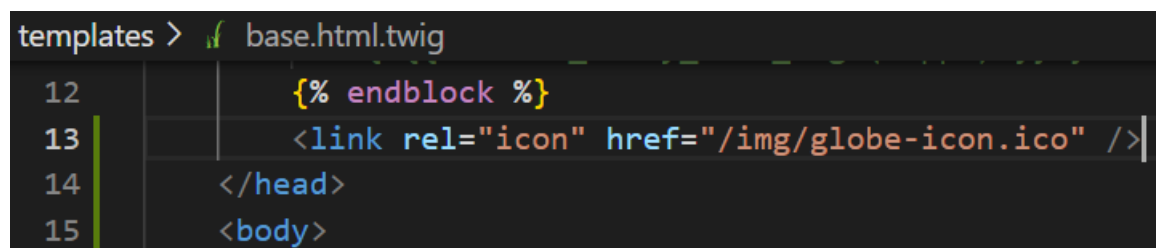


Le contenu de notre vue apparaît maintenant sous la navbar Bootstrap.

Ajoutons un footer :



Ajoutons un favicon et un logo (emplacement des fichiers : public/img)



Un peu de css ...

```
public > css > # base.css > ...
1  body{
2    |   padding-top: 62px;
3  }
4
5  @font-face {
6    |   font-family: 'Cassandra';
7    |   src: url('../fonts/cassandra.ttf') format('truetype');
8  }
9  .navbar.fixed-top{
10   |   font-family: 'Cassandra';
11 }
12 .navbar.fixed-top input{
13   |   font-family: var(--bs-body-font-family);
14 }
15
```

```
15
16 ✓ .navbar-brand img{
17   |   max-height: 48px;
18   |   margin: -6px 0px;
19 }
20 ✓ #site_name{
21   |   display: inline;
22   |   font-size: 20px;
23 }
24 ✓ #site_name span{
25   |   font-size: 16px;
26   |   position: relative;
27   |   top: 16px;
28   |   right: 16px;
29 }
30
```

Et voilà le résultat :

