

## Workshop : Intégration Template

### Objectif

L'objectif de ce workshop est d'intégrer un Template dans un projet Symfony.

### I- Pourquoi utiliser un Template ?

Un site web est constitué de plusieurs pages. Toutes contiennent des contenus spécifiques, mais certains éléments demeurent identiques pour l'ensemble des pages pour permettre aux visiteurs de garder des repères tout au long de la navigation. Le Template permet de garder cette cohérence en répétant les éléments fixes d'une page à l'autre. Il y a de nombreux avantages pour utiliser des Templates, parmi lesquels, on peut citer :

- Structuration optimale du contenu.
- Multimodalité des interfaces
- Variété des bibliothèques
- Homogénéité des pages
- Réutilisation de contenu entre différentes pages
- Maintenance plus facile

### II- Requis pour intégrer un « Template »

L'intégration d'un Template permet de construire un Template de base (Template mère) qui contient tous les éléments communs des pages d'une application Web définis comme des blocs. Les blocs représentent les parties qui se ressemblent (exemple un bloc pour le contenu des différentes pages). En effet, le Template mère contient des blocs qui peuvent être redéfinis dans chaque Template enfant. Un Template enfant peut hériter la mise en page de base (Template mère).

Dans notre cas, nous allons utiliser le fichier « **base.html.twig** » (Figure1) comme Template mère dont les autres fichiers (Template enfant) vont hériter. Ce fichier de base contient la barre de navigation de notre site, un bloc pour le contenu de nos pages ainsi que le pied de la page.

Template de base  
gabarit general

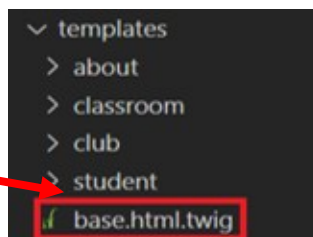


Figure 1: Emplacement du fichier « base.html.twig »

### III- Intégration du Template dans un projet Symfony

Le Template en annexe contient quatre répertoires (css, img, js et librairies) ainsi qu'un certain nombre de fichiers comme le montre la figure suivante.

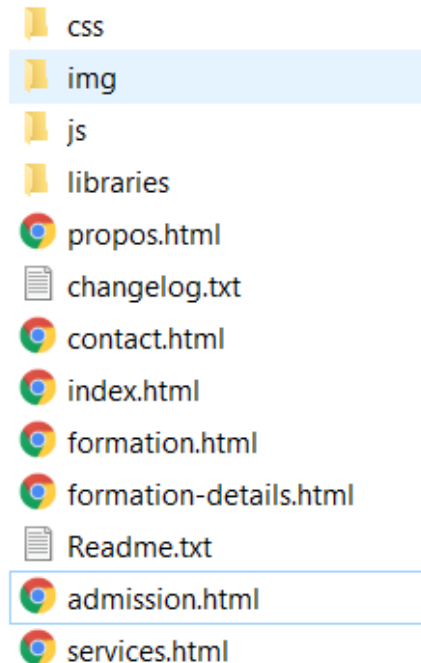


Figure 2: Différents répertoires du Template

Afin d'intégrer le Template dans notre projet, nous avons besoin de :

1. Copier les dossiers «css», «img», «js» et «librairies » sous le répertoire «assets».
2. Copier le contenu du fichier «index.html» dans le fichier « base.html.twig »
3. Dans le fichier «base.html.twig», ajouter les Blocs suivants:

- {% block title %}{% endblock %}

```
<title>{% block title %}Esprit SF {% endblock %}</title>
```

- {% block body %}{% endblock %}

```
<section id="hero" class="d-flex align-items-center">
  {% block body %}
    <div class="container d-flex flex-column align-items-center">
      <h1>Esprit</h1>
      <a href="about.html" class="btn-about">About Me</a>
    </div>
  {% endblock %}
</section><!-- End Hero -->
```

- {% block css%}{% endblock %}

```
{% block css %}
<!-- Libraries CSS Files -->
<link href="{{asset('libraries/bootstrap/css/bootstrap.min
<link href="{{asset('libraries/icofont/icofont.min.css')}}"
<link href="{{asset('libraries/owl.carousel/assets/owl.car
<link href="{{asset('libraries/boxicons/css/boxicons.min.c
<link href="{{asset('libraries/venobox/venobox.css')}}" re
<link href="{{asset('libraries/aos/aos.css')}}" rel="style
<!-- Template Main CSS File -->
<link href="{{asset('css/style.css')}}" rel="stylesheet">
{% endblock %}
```

- {% block js %}{% endblock %}

```
{% block js %}
<!-- Libraries JS Files -->
<script src="{{asset('libraries/jquery/jquery.min.js')}}"
<script src="{{asset('libraries/bootstrap/js/bootstrap
<script src="{{asset('libraries/jquery.easing/jquery
<script src="{{asset('libraries/php-email-form/valid
<script src="{{asset('libraries/waypoints/jquery.way
<script src="{{asset('libraries/counterup/counterup
<script src="{{asset('libraries/owl.carousel/owl.ca
<script src="{{asset('libraries/isotope-layout/isoto
<script src="{{asset('libraries/venobox/venobox.min
<script src="{{asset('libraries/aos/aos.js')}}"></sc
<!-- Template Main JS File -->
<script src="{{asset('js/main.js')}}"></script>
{% endblock %}
```

Pour faire appel à nos images ou à nos fichiers CSS et JS on utilise {{asset("")}}.

Exemple:

```

```

**Remarque :** Ces blocs vont être redéfinis dans les différentes vues (Templates fils).

4. Pour faire l'héritage, il suffit d'ajouter {% extends 'base.html.twig' %} dans le Template enfant (exemple: student/list.html.twig). Le fichier fils peut redéfinir le(s) bloc(s) selon son besoin en appelant le nom du bloc à redéfinir et qui est déjà

déclaré au niveau du Template mère et en mettant dedans le nouveau contenu comme le montre la figure suivante :

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

{% block body %}
<h1> Liste des classes </h1>
<table border ="1" >
<tr>
  <th scope="row">ID</th>
  <th scope="row">Nom</th>
  <th scope="row">Editer</th>
</tr>
  {% for classroom in classrooms %}
  <tr>
    <td> {{ classroom.id }} </td>
    <td> {{ classroom.name }} </td>
    <td> <a class="btn btn-primary" href="{{ path('updateClassroom', {'id':
      <a class="btn btn-primary" href="{{ path('deleteClassroom', {'id':
      <a class="btn btn-outline-danger" href="{{ path('classroomStudents'
    </td>
  </tr>
  {% endfor %}
</table>
<a href="{{ path('addClassroom') }}"> Ajouter Classe</a>
</div>
{% endblock %}
```

## IV- Personnalisation du Template

Développer le Template de base « **base.html.twig** » en définissant la partie «**nav**» comme suit :

```
<nav class="nav-menu d-none d-lg-block">

  <ul>
    <li class="active"><a href="">Esprit</a></li>
    <li><a href="{{ path('about') }}">About</a></li>
    <li><a href="{{ path('student') }}">Student</a></li>
    <li><a href="{{ path('classroom') }}">Classroom</a></li>
    <li><a href="{{ path('club') }}">Club</a></li>
    <li><a href="">Contact</a></li>
  </ul>

</nav><!-- .nav-menu -->
```

