SIN6U5 : Développement web 2 (Prog. web coté serveur) Templates et Twig

Luigi Santocanale
LIF, Aix-Marseille Université

29 mars 2018

Plan

Introduction

Compilation des templates Twig

Le langage Twig

Plan

Introduction

Compilation des templates Twig

Le langage Twig

« *Templating* »

Pros :

- Séparation claire entre la logique de présentation et le reste de l'application.
- Sole responsabilité de présentation du contenu.
- Code plus propre et lisible.
- Aide lors du travail en équipe :
  - développeurs : code coté serveur (contrôleurs, modèles) ;
  - « designers » : code coté client.

Langages de templates

PHP est déjà un moteur de template . . . mais :

Templates compilés (langages de templates, moteurs de templates) :

- Twig, développé par SensioLab (Symfony), mais indépendant du framework ;
- Blade, spécifique au framework Laravel ;
- Mustache, traverse à plusieurs langages : Ruby, JavaScript, Python, PHP, Haskell, OCaml . . .
- Smarty.

Bibliothèques de templates natives en PHP

- Plates ;
- Aura.View.

Avantages des templates compilés

- concision ;
- lisibilté ;
- syntaxe orientée vues ;
- structures de contrôles simplifiées ;
- inheritance multiple ;
- échappement automatique.

Mais :

- . . . lenteur de l'évaluation du code ;
- résolue avec la mise en place de caches.

Plan

Introduction

Compilation des templates Twig

Le langage Twig

Utilisation élémentaire (hors Symfony)

Objectif : un petit script pour tester le fonctionnement de Twig.

```
composer require twig/twig:~2.0
```

```
.
|-- bin
|   `-- renderTemplate.php
|-- cache
|   `-- 3e
|       `-- 3e77703d5875a759f0a4f8be47630f5fd60f09c75f0555655ee1f7152e351f9a.php
|-- composer.json
|-- composer.lock
|-- templates
|   |-- base.html.twig
|   |-- examples.twig
|   |-- index.html.twig
|   `-- menu.html.twig
`-- tree.txt
```

Le script *bin/renderTemplate.php*

```php
#!/usr/local/bin/php
<?php

require_once __DIR__ . '/../vendor/autoload.php';

use Twig\Loader\FilesystemLoader as TwigLoader;
use Twig\Environment as TwigEnvironment;

$loader = new TwigLoader(__DIR__ . '/../templates');
$twig = new TwigEnvironment($loader, [
    'cache' => __DIR__ . '/../cache',
    'debug' => true]);

isset($argv[1]) || die("The name of the template is missing.\n");
isset($argv[2]) || die("A json string for filling the template is
 ↪   needed.\n");

$fields=json_decode($argv[2],true);
echo $twig->render($argv[1], $fields);
```

La template templates/base.html.twig

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8" />
        <title>{{ title }}</title>
    </head>
    <body>
        <div class='container'>
        {% block body %}{% endblock %}
    </div>
{% block javascripts %}{% endblock %}
</body>
</html>
```

Le « rendu »

```
> bin/renderTemplate.php base.html.twig '{"title":"Bonjour"}'
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8" />
        <title>Bonjour</title>
    </head>
    <body>
        <div class='container'>
            </div>
</body>
</html>
```

La template compilée I

Fichier `cache/3e/3e77703d5875a759f0a4f8be47630f5fd60f09c75f0555655ee1f7152e351f9a.php` :

```php
<?php

/* base.html.twig */
class __TwigTemplate_96876972e9a20b172d232732a09dbc32bb253aa330a55eb0f31d73b4782032dd extends
  ↪    Twig_Template
{
    private $source;

    public function __construct(Twig_Environment $env)
    {
        parent::__construct($env);

        $this->source = $this->getSourceContext();

        $this->parent = false;

        $this->blocks = array(
            'body' => array($this, 'block_body'),
            'javascripts' => array($this, 'block_javascripts'),
        );
    }

      ...
```

```
  ...
    protected function doDisplay(array $context, array $blocks = array())
    {
        // line 1
        echo "<!DOCTYPE html>
<html>
    <head>
        <meta charset=\"UTF-8\" />
        <title>";
        // line 5
        echo twig_escape_filter($this->env, ($context["title"] ?? null), "html", null, true);
        echo "</title>
    </head>
    <body>
        <div class='container'>
        ";
        // line 9
        $this->displayBlock('body', $context, $blocks);
        // line 10
        echo "    </div>
";
        // line 11
        $this->displayBlock('javascripts', $context, $blocks);
        // line 12
        echo "</body>
</html>
";
    }

    // line 9
    public function block_body($context, array $blocks = array())
    {
    }

    // line 11
    public function block_javascripts($context, array $blocks = array())
    {
    }

  ...
```

```
    ...
  public function getTemplateName()
  {
      return "base.html.twig";
  }

  public function isTraitable()
  {
      return false;
  }

  public function getDebugInfo()
  {
      return array (  56 => 11,  51 => 9,  45 => 12,  43 => 11,  40 => 10,  38 => 9,  31 =>
↪    5,  25 => 1,);
  }

    ...
```

```
    ...
    public function getSourceContext()
    {
        return new Twig_Source("<!DOCTYPE html>
<html>
    <head>
        <meta charset=\"UTF-8\" />
        <title>{{ title }}</title>
    </head>
    <body>
        <div class='container'>
        {% block body %}{% endblock %}
    </div>
{% block javascripts %}{% endblock %}
</body>
</html>
", "base.html.twig",
  ↪    "/Users/lsantoca/AmuBox/DW/latex/ch10_twig/code/twigDemo/templates/base.html.twig");
    }
}
```

Non HTML templates

On peut utiliser twig pour composer du texte autre que HTML :

```
{# templates/mail.twig #}
Bonjour {{ prenom }},

veuillez retirer votre colis chronopst.fr

Cordialement,

        votre tricheur préféré
```

```
> bin/renderTemplate.php email.twig '{"prenom":"Luigi"}'
Bonjour Luigi,

veuillez retirer votre colis chronopst.fr

Cordialement,

        votre tricheur préféré
```

Plan

Introduction

Compilation des templates Twig

Le langage Twig

Twig

Documentation :

https://twig.symfony.com/doc/2.x/
https://twig.symfony.com/doc/2.x/templates.html

- Commentaires :

```
{# Commentaire #}
```

- Affichage de variables (et des expressions) :

```
        <li>Hello {{nom}}</li>
```

- Tags (marqueurs/balises) pour le flux de contrôle :

```
<ul>
    {% for nom in noms %}
        <li>Hello {{nom}}</li>
    {% endfor %}
</ul>
```

Affichage des objets et des tableaux, filtres

```
{{ user.surname }}
{{ user['surname'] }}
{{ attribute(user, 'second-name') }}
```

Filtres :

```
{{ user.surname | upper }}
{{ user.surname | raw }}
{{ list | default(0..10) | join(', ') }}
```

```
>   bin/renderTemplate.php tableaux.twig
  ↪    '{"user":{"surname":"<a>Coucou</a>","second-name":"Luigi"},"list":{"0":"a","1":"b"}}'
&lt;a&gt;Coucou&lt;/a&gt;
&lt;a&gt;Coucou&lt;/a&gt;
Luigi

&lt;A&gt;COUCOU&lt;/A&gt;
<a>Coucou</a>
a, b
```

```
>   bin/renderTemplate.php tableaux.twig
 ↪   '{"user":{"surname":"<a>Coucou</a>","second-name":"Luigi"}}'
&lt;a&gt;Coucou&lt;/a&gt;
&lt;a&gt;Coucou&lt;/a&gt;
Luigi

&lt;A&gt;COUCOU&lt;/A&gt;
<a>Coucou</a>
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
```

Liste de tous les filtres :

https://twig.symfony.com/doc/2.x/filters/index.html

Opérateurs

```
{{ 7 + 8 }}
{% spaceless %}
    {% if 7 < 1 %}
        vrai
    {% else %}
        faux
    {% endif %}
{% endspaceless %}
```

```
>  bin/renderTemplate.php ops.twig "{}"
15
faux
```

Voir la doc.

Tags pour les structures de contrôle

La liste des tags.

Exemple avec `set`, `if`, `for` :

```
{% set x = 0 %}
{% for y in range(low=6,high=0,step=1) %}
    {% if y > x %}
        {{ x }}
    {% else %}
        {{ y }}
    {% endif %}
    {% set x = x+1 %}
{% endfor %}
```

Remarquez

- les noms des paramètres de la fonction `range`,
- la liste des fonctions natives de twig.

Héritage et inclusions

Tags : include, block

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8" />
        <title>{% block title %}Welcome!{% endblock %}</title>
        <link rel="stylesheet" href="{{ asset('css/bootstrap.css') }}" />
        <link rel="stylesheet" href="{{ asset('css/style.css') }}" />
        {% block stylesheets %}
        {% endblock %}
        <link rel="icon" type="image/x-icon" href="{{ asset('favicon.ico') }}" />
    </head>
    <body>
        {% include 'menu.html.twig' %}
        <div class='container'>
        {% block body %}{% endblock %}
    </div>
{% block javascripts %}{% endblock %}
</body>
</html>
```

Template incluse

```
{# views/top_bar.html.twig  #}
<div class="navbar navbar-static-top">
    <div class="container">
        <div class="navbar-collapse collapse">
            <ul class="nav navbar-nav navbar-right">
                {%if app.user is not null %}
                    <li>
                        Bonjour {{ app.user.username }}
                    </li>
                    <li>
                        <a href="{{ path('admin_exercice_index')}}">Aller au CRUD</a>
                    </li>
                    <li>
                        <a href="{{ path('fos_user_security_logout')}}">Se déconnecter</a>
                    </li>
                {% else %}
                    <li><a href="{{ path('fos_user_security_login')}}">Se connecter</a></li>
                    <li><a href="{{ path('fos_user_registration_register')
  ↪    }}">S'inscrire</a></li>
                {% endif %}
            </ul>
        </div>
    </div>
</div>
```

Template qui hérite

Tags : `extends`, `block`

```
1  {% extends 'base.html.twig' %}
2
3  {% block body %}
4      <h1>Liste des exercices</h1>
5
6      <table class='table'>
7          <thead>
8              <tr>
9                  <th>Id</th>
10                 <th>Titre</th>
11                 <th>Texte</th>
12                 <th>Origine</th>
13                 <th>Auteur</th>
14                 <th>Motscles</th>
15                 <th>Actions</th>
16             </tr>
17         </thead>
```

```
18      <tbody>
19      {% for exercice in exercices %}
20          <tr>
21              <td><a href="{{ path('admin_exercice_show', { 'id': exercice.id }) }}">{{ exercice.id
    ↪     }}</a></td>
22              <td>{{ exercice.titre }}</td>
23              <td>{{ exercice.texte }}</td>
24              <td>{{ exercice.origine }}</td>
25              <td>{{ exercice.auteur }}</td>
26              <td>{{ exercice.motscles }}</td>
27              <td>
28                  <ul>
29                      <li>
30                          <a href="{{ path('admin_exercice_show', { 'id': exercice.id }) }}">voir</a>
31                      </li>
32                      <li>
33                          <a href="{{ path('admin_exercice_edit', { 'id': exercice.id }) }}">modifier</a>
34                      </li>
35                  </ul>
36              </td>
37          </tr>
38      {% endfor %}
39      </tbody>
40  </table>
```

```
42      <ul>
43          <li>
44              <a href="{{ path('admin_exercice_new') }}">Créer un nouvel exercice</a>
45          </li>
46      </ul>
47  {% endblock %}
```

Remarques :

- `asset` et `path` ne sont pas des fonctions natives twig. Elles sont des extensions de Symfony.

Définitions de macros

```
{# templates/table.twig #}
{% macro headerRow(columNames) %}
    {% spaceless %}
            <tr>
                {% for columnName in columNames %}
                    <th>{{ columnName }}</th>
                {% endfor %}
            </tr>
    {% endspaceless %}
{% endmacro %}

{% macro row(entries) %}
    {% spaceless %}
        <tr>
            {% for entry in entries %}
                <td>{{ entry }}</td>
            {% endfor %}
        </tr>
    {% endspaceless %}
{% endmacro %}
```

Utilisation des macros

```
{# tempates/index2.html.twig #}
{% extends 'base.html.twig' %}
{% import 'table.twig' as table %}
{% block body %}

    <h1>Liste des exercices</h1>

    <table class='table'>
        <thead>
            {{ table.headerRow(['Id','Titre','Texte','Origine','Auteur','Motscles']) }}
        </thead>
        <tbody>
            {% for exercice in exercices %}
                {{ table.row([
                    exercice.id,
                    exercice.titre,
                    exercice.texte,
                    exercice.origine,
                    exercice.auteur,
                    exercice.motscles]) }}
            {% endfor %}
        </tbody>
    </table>

{% endblock %}
```

Pour terminer

Quelques lectures :

- Un tutoriel par Robin Dupret,
- Un autre tutoriel par Alexandre Bacco.