TWIG

David Annebicque

2016-2017

1 Présentation de TWIG

Twig est un moteur de templates qui a été créé par SensioLabs, les créateurs de Symfony. On le retrouve nativement dans les frameworks Symfony et Drupal8, mais il peut être installé sur la majorité des frameworks ainsi que dans un environnement PHP.

Toute la documentation officielle `http://twig.sensiolabs.org/` ou `http://iner-dukoid.developpez.com/tutoriels/web/moteur-template-twig/` dont une partie de ce document s'inspire.

1.1 Avantages

- Twig permet de séparer la présentation des données du traitement: Twig permet de définir en dehors de la page web des filtres que l'on pourra appliquer à la donnée.
- Twig permet la personnalisation de page web : Un block menu, un block recherche, un block contenu... le tout défini dans un template lui-même héritable.
- Twig permet de rendre les pages web plus lisibles, plus claires. Twig par son langage est moins invasif que PHP et se substitue à celui-ci.
- Twig est rapide.
- Twig apporte de nouvelles fonctionnalités.
- Twig est facile à apprendre

Listing 1 – Exemple d'une boucle en PHP

```php
<?php
foreach ($items as $value)
{
        if ($value.active)
        {
?>
        <img src="<?php echo $value->url;?>"/>

        <?php
        }
}
?>
```

Listing 2 – Le même exemple avec TWIG

```
{% for value in items if value.active %}
        <img src="{{ value.url }}"/>
{% endfor %}
```

2 Le langage TWIG

2.1 Les syntaxes

Il existe 3 syntaxes dans TWIG :
— {% %} : pour faire une action : un test, une boucle, ...
— {{ }} : pour afficher quelque chose
— {# #} : pour commenter quelque chose

2.2 Affichage des données

Pour TWIG, le type de données est indéfférent. Il peut être un tableau, une variable ou un objet. S'il peut afficher quelque chose, il le fera.

Listing 3 – Exemple d'affichage de propriétés d'un objet en PHP

```php
<?php
echo $produit->getNom();
echo $produit->getCategorie()->getNom();
echo $produit["nom"];
?>
```

Listing 4 – Le même exemple avec TWIG

```
{{ produit.nom }}
{{ produit.categorie.nom }}
{{ produit["nom"] }}     ou     {{ produit.nom }}
```

Il est possible de concaténer des variables, mais aussi d'effectuer des transformations (mise en majuscule, en minuscule, ...)

Listing 5 – Exemple d'affichage de variable en PHP

```php
<?php
echo "Description du produit:"  .
$produit->getDescription();

$texte = $greeting . strtolower($name);
echo  $texte;

echo strtolower($greeting . $name);
?>
```

Listing 6 – Le même exemple avec TWIG

```
{{ "Description du produit:" ~ produit.description }}

{% set texte = greeting ~ name|lower %}
{{ texte }}

{{ (greeting ~ name)|lower }}
```

Dans TWIG l'utilisation des filtres (transformations) se fait en utilisant le caractère | (ALT-GR + 6). Vous avez une liste de filtre pré-définie sur le site de TWIG. Vous pouvez créer d'autres filtres.

2.3 Les tests conditionnels

Listing 7 – Un test en PHP

```php
<?php
if ($produit == 'ART123')
{
        //code ici
}
?>
```

Listing 8 – Le même exemple avec TWIG

```
{% if produit == 'ART123 %}
        code ici
{% endif %}
```

Il est possible de faire des and, or, note, defined (existe), starts with (commence avec), ends with (termine avec), in, empty, matches

Listing 9 – Exemples de tests conditionnels

```
{% if produits is empty %}
        il n'y a plus de produit
{% endif %}

{% if ((a==1 and b>0) or not c==0) and d is defined %}
        {% set resultat = (d + a * b) / c %}
        {{ resultat }}
{% endif %}

{% if 'Fabien' starts with 'F' %}
        commence par F
{% endif %}

{% if 'Fabien' ends with 'n' %}
        Finis par n
{% endif %}

- regular expressions:
{% if phone matches '/^[\\d\\.]+$/' %}
        format telephone ok
{% endif %}

{% if 5 not in [1, 2, 3] %}
        5 non présent
{% endif %}
```

Il existe aussi :
— is null: si est null ;
— is constant: comparer si est une constante ;
— is divisible by(x): si est divisible par x ;
— is even: si est pair ;
— is odd: si est impair ;
— is iterable: si est du type itérable (comme une liste) ;
— is same as: comparer 2 variables (en php correspond ===).

2.4 Les boucles

Listing 10 – Exemples de tests conditionnels

```
- affiche 0123456789
{% for i in 0..9 %}  // pareil que {% for i in range(0, 9) %}
        {{ i }}
{% endfor %}

{% for produit in produits %}
        {{ produit.nom }}
{% endfor %}

- avec une condition:
{% for produit in produits if produit.etat==1 %}
        {{ produit.nom }}
{% endfor %}

- boucle for avec condition vide:
{% for article in articles %}
        {{ article.nom }}
{% else %}
        pas d'article trouvé
{% endfor %}

- clés et valeurs:
{% for key, value in table %}
        {{ key }} {{ value }}
{% endfor %}

- les 10 premiers users:
{% for user in users|slice(0, 9) %}
        <li>{{ user.username }}</li>
{% endfor %}

- bascule:
{% for i in 0..10 %}
        <div class="{{ cycle(['fond-noir', 'fond-blanc'], i) }}">       //  fond-noir / fond-blanc  /  ...</div>
{% endfor %}
```

La variable loop permet de récupérer des informations sur la boucle

Listing 11 – Variable de boucle loop

```
{{ loop.index }}        // Numéro de l'itération courante
↪ en commençant par 1
{{ loop.index0 }}         // Numéro de l'itération
↪ courante en commençant par 0
{{ loop.revindex }}        // Nombre itérations restantes
↪ avant la fin en commençant par 1
{{ loop.revindex0 }}        // Nombre itérations restantes
↪ avant la fin en commençant par 0
{{ loop.first }}        // La première itération?  True ou
↪ false
{{ loop.last }}           // La dernière itération? True
↪ ou false
{{ loop.length }}        // Nombre total d'itérations
```