

Twig

Présentation

1. Twig ?
2. La syntaxe
3. Les extensions

- **Twig est un moteur de template moderne pour PHP**
 - Rapide
 - Sécuriser
 - Idéal pour les intégrateurs
 - Flexible
 - Le langage est moins verbeux que le php.

Déclaration de twig dans symfony2

```
# app/config/config.yml
```

```
framework:
```

```
#....
```

```
templating: { engines: ['twig'] }
```

La syntaxe – variables simples

PHP

```
<?php echo $var ?>
```

```
<?php echo htmlspecialchars($var, ENT_QUOTES, 'UTF-8') ?>
```

Twig

```
{{ var }} ou {{ var| escape }}
```

affichera: `<b class="orange">orange`

```
{{ var| raw }} (évite l'échappement)
```

affichera: **orange**

`{{ ... }}` utiliser pour afficher des variables

`{% ... %}` utiliser pour exécuter les fonctions de twig (if, for, set, block, ...)

`{# ... #}` les commentaires

La syntaxe – les boucles

PHP

```
<?php if(count($users) > 0): ?>  
    <?php foreach($users as $user): ?>  
        <?php echo $user->getName() ?>  
    <?php endforeach ?>  
<?php else: ?>  
    No user have been found.  
<?php endif ?>
```

Twig

```
{% for user in users %}  
    {{ user.name }}  
{% else %}  
    No user have been found.  
{% endfor %}
```

Remarquez le raccourci de twig: `$user->getName()` en `user.name`

La syntaxe – les boucles

Boucle **for** simple sur un tableau d'objets:

```
{% for line in lines %}  
    <li>{{ line.title }}</li>  
    <li>{{ line.getTitle() }}</li>  
{% endfor %}
```

Boucle **for** simple sur un tableau:

```
{% for line in lines %}  
    <li>{{ line['title'] }}</li>  
{% endfor %}
```

Boucle **for** avec clé:

```
{% for key, line in lines %}  
    <li>key={{ key }}, content={{ line.title }}</li>  
{% endfor %}
```

La syntaxe – if

if simple sur un objet et utilisation des conditions **and** et **or** :

```
{% if count > 10 %}  
    <div> {{ count }} supérieur à 10</div>  
{% elseif count >= 1 and count <= 10 %}  
    <div> {{ count }} entre 1 et 10</div>  
{% elseif count == 0 or count < 0 %}  
    <div> vide </div>  
{% endif %}
```

Savoir si une variable est non définie:

```
{% if variable is not defined %}  
{% endif %}
```

La syntaxe – assets et path


asset et **path** sont les fonctions les plus utilisées dans twig

asset permet de sélectionner un fichier localisé dans le dossier **/web** :

```
<link type="text/css" rel="stylesheet" href="{{ asset('css/style.css') }}" />
```

path génère une URL en fonction des routes répertoriées dans l'application

```
<a href="{{ path ("produit", { "id" : 12 } ) }}">lien</a>
```



```
/**
 * @Route ("/produit/{id}", name="produit")
 * @Template()
 */
public function produitAction()
{
    return array();
}
```


La syntaxe – les filtres

Un filtre permet de transformer l’affichage d’une chaîne à l’aide d’une fonction twig prédéfinie.

Formater un nombre

```
{{ 500421|number_format(2, ',', '') }}
```

```
//500 421
```

Mettre une majuscule à la première lettre d’une chaîne.

```
{{ nom|capitalize }}
```

```
//Nom
```

Formater une date (instance de classe DateTime)

```
{{ madate|date("d/m/Y") }}
```

```
//12/10/2013
```

Retour à la ligne

```
{{ montexte|nl2br }}
```

```
/*
```

Titre:

Mon texte.

```
*/
```

La syntaxe – les filtres

Equivalent de **strtolower** en php:

```
{{ 'SALUT' | lower }}
```

```
//salut'
```

Equivalent de **strtoupper** en php:

```
{{ 'salut' | upper }}
```

```
//SALUT
```

Première lettre en majuscule sur tous les mots:

```
{{ 'bonjour tout le monde' | title }}
```

```
//Bonjour Tout Le Monde
```

Equivalent de **trim** en php:

```
{{ ' hello! ' | trim }}
```

```
//"hello!"
```

```
{{ ' hello! ' | trim('!') }}
```

```
// " hello "
```

La syntaxe – les filtres

Formule de calcul

```
{% set var = valeur * 100 / nbr %}
```

Enregistrer un tableau

```
{% set montableau['element'] = 'value' %}
```

```
{% set montableau = montableau|merge( { 'element' : 'value' } ) %}
```

Un equivalent en php à **count(array())**

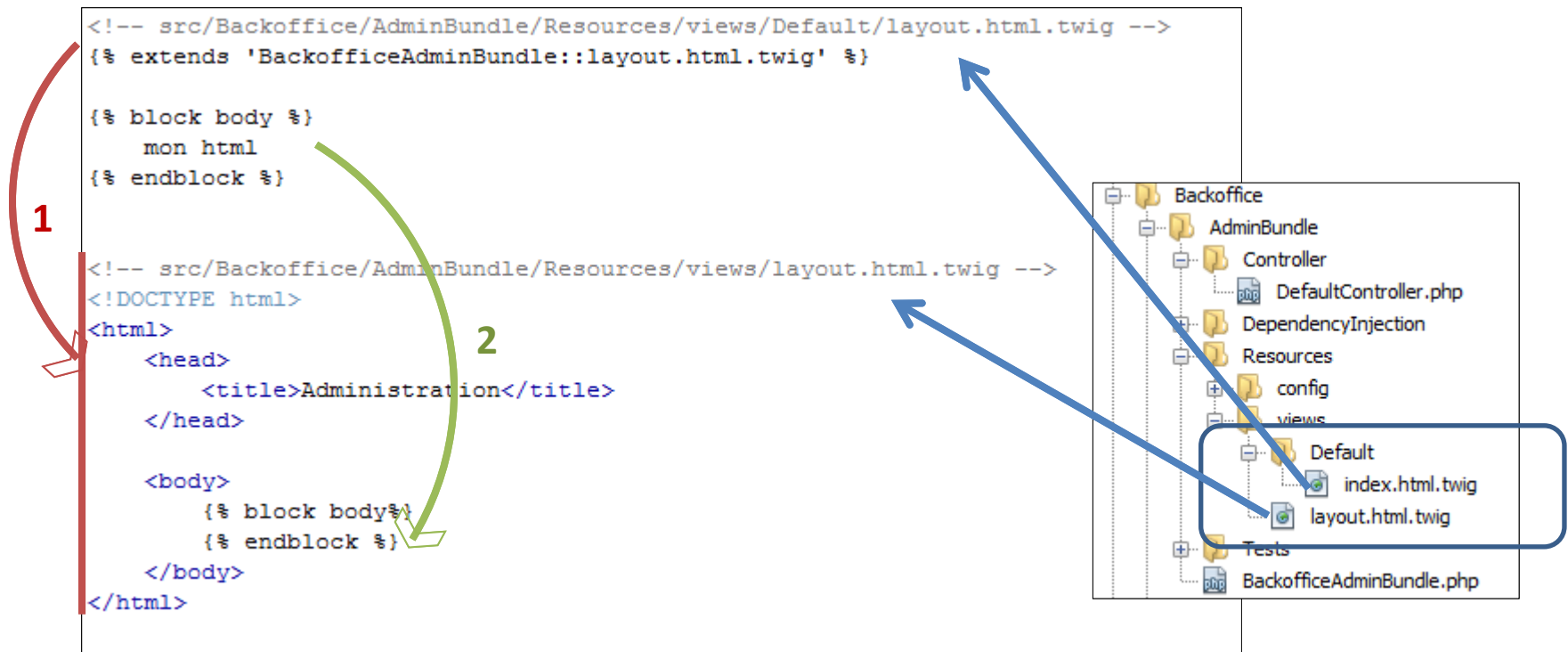
```
{% if liste_langs|length > 0 %}
```

```
{% endif %}
```

Introduction à extends et aux blocks

Gérer la fonction **block** de twig:

- 1: cherche le fichier layout.html.twig
2. Remplace le block **body de layout** par le contenu du block **body de l'index**



La syntaxe – Les tags

Permet de sélectionner le fichier squelette de la page html : (doit être la première balise du template)

Fichier à partir de: **app/Resources/views**

```
{% extends '::file.html.twig' %}
```

Fichier à partir d'un bundle:

src/Admin/ProjetBundle/Resources/views /layout.html.twig

```
{% extends 'AdminProjetBundle::layout.html.twig' %}
```

src/Admin/ProjetBundle/Resources/views /Default/fichier.html.twig

```
{% extends 'AdminProjetBundle:Default:fichier.html.twig' %}
```

Nous utiliserons la fonction '**block**' afin d'insérer du contenu dans le squelette twig.

```
{% block content %} html de block par défaut {% endblock %}
```

Si vous avez besoin de récupérer le contenu d'un bloc d'un template parent portant le même nom, vous pouvez utiliser la fonction **{{ parent() }}**. C'est utile si on souhaite compléter le contenu du bloc parent au lieu de le réécrire totalement.

```
{% block sidebar %}
```

```
<h3>Titre</h3>
```

```
... {{ parent() }}
```

```
{% endblock %}
```

La syntaxe – Les tags

Inclure un template dans un autre

```
{% include 'MonBundle:Default:template/content.html.twig' %}
```

inclure un template dans un autre en modifiant des variables, **only** indique que les autres variables ne sont pas transmises .

```
{% include 'MonBundle:Page:template/content.html.twig' with {'foo': 'bar'} only %}
```

Appeler une action d'un bundle depuis un template twig

```
{% render 'MonBundle:Default:action' with {'page': page} %}
```

!! Depuis symfony2.2

```
{% render controller('MonBundle:Default:action', {'page ': 'page'}) %}
```

Définir et utiliser une variable twig

```
{% set img = 'cheval' %}
```

```
{{ img }} .jpg
```

Permet par exemple de récupérer le nom de la route

```
{{ app.request.attributes.get('_route') }}
```

Dans twig, pour faire un var_dump() de \$varname

```
{% debug expression.varname %}
```

La syntaxe – autres

Exemple de coloration de tableau une ligne sur 2 avec le **test divisibleby** :

```
{% if( nombre is divisibleby(2) ) %}  
    #EFEFEF  
{% else %}  
    #FFFFF  
F{% endif %}
```

Gestion des droits utilisateur dans Twig:

is_granted : Indique si l'utilisateur possède un droit (boolean)

Attention, l'utilisateur doit être identifié pour obtenir des roles.

```
{% if(is_granted('ROLE_ADMIN')) %}  
{% if(is_granted('ROLE_ADMIN ') == false) %}
```

La syntaxe – Variables globales

Symfony2 envoie à twig une classe "**app**" permettant d'accéder à pas mal d'informations de l'application.

app.security

- .token
 - .key
 - .user
 - .roles
 - .authenticated
 - .attributes
- .alwaysAuthenticate

- .requestUri
- .baseUrl
- .basePath
- .method
- .format
- .uri
- .locale

app.user (entity User)

app.request

- .attributes
 - .get('_route')
- .query
- .server
- .files
- .cookies
- .headers
- .content
- .languages
- .charsets
- .acceptableContentTypes
- .pathInfo

app.session

- .defaultLocale
- .saved

app.environment

app.debug

app.kernel

- .cachedir
- .logdir
- .rootdir