Curso POO Motores de plantillas: Smarty

Sumario

- 1 Motores de plantillas
  - 1.1 Smarty
  - 1.2 Plantillas Smarty
    - 1.2.1 Variables
    - 1.2.2 Estruturas de control

Motores de plantillas

Os motores de plantillas son un xeito sinxelo de separar nunha aplicación a lóxica de presentación da lóxica de negocio. Deste xeito, non será necesario que a persoa ou persoas que desenvolvan o interface web da aplicación coñezan a sintaxe e as etiquetas empregadas na linguaxe PHP; no canto terán que familiarizarse únicamente cun conxunto de etiquetas moito máis simple, e cunha mecánica de utilización máis cercana a das linguaxes de marcas.

Smarty

Un dos motores de plantillas que podemos empregar coa linguaxe PHP é Smarty (enlace á documentación en inglés). Para instalar Smarty nun servidor soamente temos que descargar a última versión e copiar os arquivos correspondentes á libraría nunha ruta do sistema. A continuación deberemos indicarlle ao intérprete PHP a localización da libraría; o máis habitual é configurar a variable **include_path** do sistema para engadir a ruta correspondente.

```
include_path = "...;/usr/local/lib/php/smarty"
```

Para poder chamar a unha plantilla Smarty dende unha páxina PHP, teremos que incluir o arquivo **Smarty.class.php**, instanciar un obxecto da clase Smarty, e configurar as rutas para 4 directorios.

```
<?php
require_once('Smarty.class.php');

$smarty = new Smarty();
$smarty->setTemplateDir(RUTA . '/smarty/templates/');
$smarty->setCompileDir(RUTA . '/smarty/templates_c/');
$smarty->setConfigDir(RUTA . '/smarty/configs/');
$smarty->setCacheDir(RUTA . '/smarty/cache/');
?>
```

As rutas anteriores correspóndense con:

- **setTemplateDir**. Especifica a ruta na que se almacenarán as plantillas Smarty do proxecto.
- **setCompileDir**. Especifica a ruta na que se almacenarán as plantillas compiladas do proxecto.
- **setConfigDir**. Especifica a ruta na que se almacenarán or arquivos de configuración do proxecto.
- **setCacheDir**. Especifica a ruta na que se almacenará o saída xerada por Smarty a partires das plantillas do proxecto.

É importante ter en conta que Smarty fai uso da característica de carga automática de clases de PHP, polo que no caso de empregar a función **__autoload** no noso proxecto deberemos cambiala pola función **spl_autoload_register**.

As plantillas Smarty gárdanse en arquivos que empregan normalmente extensión ".tpl". Unha vez configuradas as rutas, para chamar a unha plantilla dende Smarty empregaremos o método **display**:

```
<?php
require_once('Smarty.class.php');

$smarty = new Smarty();
$smarty->setTemplateDir(RUTA . '/smarty/templates/');
$smarty->setCompileDir(RUTA . '/smarty/templates_c/');
$smarty->setConfigDir(RUTA . '/smarty/configs/');
$smarty->setCacheDir(RUTA . '/smarty/cache/');

$smarty->display('index.tpl');
?>
```

Plantillas Smarty

As plantillas Smarty compóñense de HTML que contén etiquetas propias de Smarty intercaladas. As etiquetas das plantillas Smarty empregan por defecto chaves como delimitadores. Os principais elementos que podemos empregar nas etiquetas propias de Smarty son:

Variables

Deben ir precedidas do signo **$**. Poden ser elementos complexos como arrays ou obxectos, que empregan a mesma sintaxe que en PHP para acceder aos seus membros respectivos. As variables deben crearse antes de executar a plantilla empregando o método **assign** da libraría Smarty. Por exemplo:

Páxina PHP **index.php**:

```
<?php
require_once('Smarty.class.php');

$smarty = new Smarty();
$smarty->setTemplateDir(RUTA . '/smarty/templates/');
$smarty->setCompileDir(RUTA . '/smarty/templates_c/');
$smarty->setConfigDir(RUTA . '/smarty/configs/');
$smarty->setCacheDir(RUTA . '/smarty/cache/');

$smarty->assign('post', $post);
$smarty->display('index.tpl');
?>
```

Plantilla Smarty **index.tpl**:

```
<div class="data-post">
   <span>{$post->get_data()}</span><br />
</div>
```

As variables Smarty permiten que se lles apliquen **modificadores** que inflúen no seu comportamento. Estes especifícanse a continuación do nome da variable separados por unha barra vertical |. Algúns modificadores poden levar tamén parámetros, separados por dous puntos **:**. Por exemplo:

Plantilla Smarty **index.tpl**:

```
<div class="data-post">
   <span>{$post->get_data()|date_format:"%d"}</span><br />
</div>
```

Estruturas de control

Nalgúns casos pode ser necesario empregar nas plantillas algunha estrutura de control. Por exemplo, podemos necesitar que non se amose certo código HTML se o usuario non se atopa logueado no sistema. Ou repetir un conxunto de etiquetas HTML por cada elemento dun array. Smarty incorpora nas súas etiquetas algunhas funcións que nos permiten facer isto empregando a mesma sintaxe que PHP para avaliar condicións.

Plantilla Smarty **index.tpl** cunha expresión condicional:

```
...
{if $is_admin}
  <div class="borrar-comentario">
    ...
  </div>
{else}
  ...
{/if}
...
```

Plantilla Smarty **index.tpl** cun bucle:

```
...
{foreach $comentarios as $comentario}
  <div class="comentario">
    {$comentario->get_texto()|nl2br}
    ...
```

```
  </div>
{/foreach}
...
```

--Víctor Lourido 09:56 22 oct 2013 (CEST)