

Hoofdstuk 6 – De structuur van een template

Ook worden hacks voor Internet Explorer 6 of Internet Explorer 7 (bijvoorbeeld workarounds voor de vreemde quirks mode) in aparte bestanden geplaatst. Alleen onder Internet Explorer worden deze bestanden toegevoegd aan de webpagina. Tot slot is er ook een speciaal bestand `print.css` voor als men een pagina of de website wil afdrukken.



Voorbeelden van CSS

Op de website www.csszengarden.com is te zien hoe krachtig CSS kan zijn. De website demonstreert enkele honderden uiterst uitgebalanceerde huisstijlen die allemaal gebaseerd zijn op dezelfde XHTML-code. Het is het ultieme voorbeeld van de scheiding tussen design en functionaliteit – de scheiding tussen CSS-opmaak en XHTML-structuur.

TemplateDetails.xml

Het XML-bestand `templateDetails.xml` bevat zogenoemde metadata met betrekking tot de template. De inhoud van het bestand wordt nooit rechtstreeks op de webpagina getoond, maar dient als extra informatie voor Joomla! om op de juiste wijze om te gaan met de template.

Het bestand bevat bijvoorbeeld informatie over de maker van de template en onder welke licentie de template wordt verspreid. Ook is er informatie terug te vinden over de bestanden waar de template uit is opgebouwd (inclusief JavaScript-bestanden, CSS-bestanden en afbeeldingen). Deze informatie wordt onder meer gebruikt door de Joomla!-installer in het beheergedeelte van Joomla!.

Een voorbeeld van een `templateDetails.XML`-bestand wordt hierna in detail besproken.



Hoofdletter

Let erop dat de naam van dit bestand wordt geschreven met een hoofdletter D.

Een XML-header

Ook al is het niet verplicht voor de juiste werking van het XML-bestand, het kan verstandig zijn het XML-bestand te beginnen met een officiële XML-regel die ook aangeeft welke tekenset wordt gebruikt. Door hier UTF-8 te gebruiken, kunnen bijvoorbeeld in de naam van de template of in de naam van de auteur vreemde Nederlandse tekens gebruikt worden zoals `ë` of `é`.

```
<?xml version="1.0" encoding="utf-8"?>
```

Hoofdstuk 7 – Het opbouwen van index.php

Zoals misschien opvalt wordt bij de parameters `xml:lang`, `lang` en `dir` PHP-code gebruikt. Dit is erg handig; u hoeft zelf niet te bedenken welke taal op welk moment gebruikt wordt. Joomla! vult dit automatisch in.

De XHTML-header

De XHTML-code binnen de `<html>`-tags wordt opgedeeld in een header en een body. De header wordt aangegeven met de tags `<head>` en `</head>`. De body wordt aangegeven met de tags `<body>` en `</body>`.

Binnen de XHTML-header wordt extra informatie gegeven over het XHTML-document en worden verwijzingen naar de nodige CSS- en JavaScript-bestanden opgenomen. Het mooie van Joomla! is dat het merendeel van de headertags automatisch worden gegenereerd.

Dit wordt binnen de template mogelijk gemaakt door de speciale Joomla!-tag `<jdoc>`. De `<jdoc>`-tag zien we later ook terug in de XHTML-body.

Een headersectie ziet er als volgt uit:

```
<head>
<jdoc:include type="head" />
<link rel="stylesheet" href="templates/<?php echo $this->template
?>/css/template.css" type="text/css" />
</head>
```

De JDocument-tag van het type head

De tag `<jdoc:include type="head">` wordt door Joomla! vervangen door alle `<head>`-tags die nodig zijn voor een bepaalde webpagina. Dit gedeelte is geheel dynamisch. De enkele regel binnen de template staat uiteindelijk gelijk aan meer regels:

- Een `<title>`-tag met de titel van de website of een titel van de pagina.
- Meer metatags met een paginaomschrijving, keywords (trefwoorden), een tekensetspecificatie en een robotsindicatie voor zoekmachines.
- Een koppeling naar de RSS-feed voor de getoonde pagina.
- Een koppeling naar het favicon (met de naam `favicon.ico`), een pictogram dat onder meer in de adresbalk van de browser wordt getoond.
- Koppelingen naar eventuele CSS-bestanden en JavaScript-bestanden (zoals bij een component van derden).

Een logo als achtergrond

Een <div>-tag geeft u als volgt een achtergrondafbeelding:

```
div#top {
  background-image: url(../images/logo.png) ;
  background-repeat: no-repeat ;
  background-position: top left ;
  width: 100% ;
  height: 200px ;
}
```

Een box in een andere box rechts uitlijnen

We hebben al het centreren van een box binnen een andere box besproken. U kunt echter ook de binnenste box op een specifieke plek binnen een andere box uitlijnen. De beste manier om dit voor elkaar te krijgen is met de CSS-property `position: absolute`.

```
div#top_right {
  position: absolute ;
  top: 10px ;
  right: 10px ;
  width: 200px ;
}
```

Kolommen met <div>-tags

De <div>-tag main bestaat uit twee kolommen: left en right. Standaard worden <div>-tags onder elkaar geplaatst, terwijl het principe van kolommen natuurlijk is dat ze naast elkaar staan. We gebruiken de CSS-definitie float om het standaardgedrag van de <div>-tags left en right te veranderen:

```
div#main {
  width: 100% ;
}
div#left {
  float: left ;
  width: 190px ;
}
```

Voor deze module wordt het juiste menu gekozen, er kan een titel ingesteld worden en eventueel een bepaalde menustijl. De splitmenufunctie wordt ingesteld met de parameters Startniveau en Eindniveau. Het hoofdmenu moet alleen menu-items bevatten van level nul, dus hiervoor stellen we het Startniveau in op 0 en het Eindniveau op 1.



Afbeelding 10.8 De menumodule voor het submenu.

Het submenu moet alleen menu-items bevatten met een level groter dan nul, dus hiervoor stellen we het Startniveau in op 1 en het Eindniveau op bijvoorbeeld 6. De maximumwaarde voor zowel Startniveau als Eindniveau is negen. Met andere woorden: een level kan alleen van nul tot en met negen reiken, dus een menu heeft maximaal tien niveaus.

Splitmenu's in templates

U kunt zelf splitmenu's maken via Modulebeheer, maar er zijn ook templates beschikbaar waarbij het splitmenu al is ingebouwd. Het voordeel hiervan is dat bij het maken van een menu de opdeling van menu-items over meer moduleposities meteen al werkt.

Het gebruik van splitmenu's in templates is meestal beperkt tot een keuze voor welk menu het mechanisme toegepast wordt. Standaard is dit het menu mainmenu, maar dit kan eventueel via de templateparameters aangepast worden. Het voordeel van het gebruik van dergelijke vooraf gedefinieerde splitmenu's is dat u als gebruiker niet over de functionaliteit hoeft na te denken. Het werkt direct.

Verderop in het boek wordt uitgelegd hoe het Joomla! Framework gebruikt wordt om met PHP-code een eigen splitmenu in een template te zetten.

Hoofdstuk 12 – Tips voor webdesign

Transparantie

Een element kan een vaste kleur hebben, maar u kunt ook doorzichtigheid toepassen. Transparantie wordt vaak op boxen toegepast met de eigenschap `background-color`.

```
div.mijnbox {  
    background-color: transparent;  
}
```

De box is op deze manier volledig doorzichtig en krijgt de achtergrondkleur van de achterliggende box mee. Als een box geen achtergrondkleur krijgt, is die meestal automatisch transparant.

Semitransparantie

Een box kan ook semitransparent gemaakt worden. De box is niet geheel doorzichtig of geheel ondoorzichtig, maar krijgt een eigen achtergrond waar gedeeltelijk doorheen gekeken kan worden. Helaas werkt dit onder verschillende browsers anders. Semitransparantie staat binnen CSS bekend als *opacity*.

Het volgende voorbeeld toont een gele `<div>`-tag die voor veertig procent transparant is (in andere woorden zestig procent *opacity*). Merk op dat alle elementen bij de `<div>`-tag hiermee ook transparant worden.

```
div {  
    background-color: yellow ;  
    /* Firefox */  
    -moz-opacity: .60 ;  
    /* Opera */  
    opacity: .60 ;  
    /* Internet Explorer 5 + 6 */  
    filter: alpha( opacity = 60 );  
    /* Internet Explorer 7 */  
    filter: progid:DXImageTransform.Microsoft.Alpha(opacity=60);  
    width: 100% ;  
}
```

Merk op dat voor Internet Explorer-filters een definitie van `width` nodig is. Ontbreekt deze, dan werkt de semitransparantie onder Internet Explorer niet.

Veiligheid

Net zoals bij componenten en andere extensies is het verstandig het templatebestand te laten beginnen met een controle op de constante `_JEXEC`:

```
<?php
defined( '_JEXEC' ) or die( 'Access denied' );
?>
```

Op deze manier wordt voorkomen dat `index.php` rechtstreeks vanaf de url aangeroepen kan worden. Dit kan ongewenste effecten hebben, zeker zodra de template meer PHP-code bevat.



GET- of POST-variabelen

Mocht het nodig blijken om GET- of POST-variabelen uit te lezen in de template, dan kan dit op een veilige manier met de Joomla! API. Dit wordt later, bij het Joomla! Framework uitvoerig besproken.

XML-header

De header van een XHTML-compliant template begint met de volgende regel:

```
<?php echo '<?xml version="1.0" encoding="utf-8"?>' . '>' ?>
```

Hierbij wordt met PHP niets anders gedaan dan de volgende regel op het scherm afdrukken:

```
<?xml version="1.0" encoding="utf-8"?>
```

Een logische vraag is waarom dit met PHP moet worden gedaan en niet rechtstreeks met XHTML-code. De reden is dat PHP-code gewoonlijk begint met de tag `<?php`, maar dat er ook een methode bestaat waarbij PHP-code begint met de tag `<?>`. Deze laatste methode is te gebruiken als in de PHP-configuratie (het bestand `php.ini`) de optie `'short_open_tag'` aan staat. Het probleem is dan dat een regel beginnend met `<?xml` wordt opgevat als een PHP-begintag `<?>` gevolgd door de niet bestaande functie `xml`. PHP zal vervolgens de volgende foutmelding geven:

```
Parse error: syntax error, unexpected T_STRING in X on line X
```

U voorkomt dit door de hele XML-regel met PHP af te drukken.

Een afwijkende frontpage

Het is mogelijk om de opmaak van de frontpage van de website (ofwel de hoofdpagina van de voorkant van de website) te laten verschillen van die van andere pagina's. Het kan bijvoorbeeld handig zijn om op de voorkant de <div>-tag voor de component van grootte te laten veranderen. Ook kan het zijn dat u helemaal geen component aan de voorkant wilt tonen, maar alleen modules.

Hiervoor moet in de template worden gecontroleerd of de getoonde pagina de frontpage is. Voorheen was de frontpage een aparte component met de naam `com_frontpage`. Onder Joomla! 1.5 is frontpage een onderdeel van de overkoepelende component `com_content`, waarbij frontpage een zogenoemde view (binnen het MVC-model) is die bestaat naast andere views zoals `article` en `category`. Een view kan ook nog een lay-out bevatten, die weer een andere opmaak toevoegt.

View	Lay-out	Omschrijving
article	-	Een standaardartikellay-out
article	form	Een formulier om een artikel toe te voegen
category	-	Een lijst van artikelen uit een bepaalde categorie
category	blog	Een lijst van artikelen in blogindeling
section	-	Een lijst van artikelen uit een bepaalde sectie
section	blog	Een lijst van artikelen in blogindeling
archive	-	Een lijst van gearchiveerde artikelen
frontpage	-	De voorpagina van de website in blogindeling

De PHP-code waarmee gecontroleerd kan worden of de getoonde webpagina de frontpage betreft kan er als volgt uit zien:

```
<?php if( JRequest::getVar('view') == 'frontpage' ) { ?>
...
<?php } ?>
```

extra debuginformatie op het scherm te tonen. Dit wordt mogelijk gemaakt met de functie `renderBacktrace()`. Het is verstandig deze melding alleen te tonen als debugging is geactiveerd in de globale configuratie.

```
<?php if ($this->debug) {  
echo $this->renderBacktrace();  
} ?>
```

De klasse JTemplate

In het Joomla! Framework is een aparte klasse gedefinieerd voor het behandelen van templates: `JTemplate`. Bij het interpreteren van de template wordt deze klasse gebruikt om `<jdoc>`-tags te vertalen naar daadwerkelijke HTML-code.

`JTemplate` gebruikt een niet-Joomla!-project met de naam `patTemplate`. `PatTemplate` is een open source PHP-templatingsysteem (naast andere templatingssystemen als `Smarty` en `FastTemplate`), gedistribueerd onder de Lesser General Public License (LGPL) en is met aanpassingen toegevoegd aan het Joomla!-project, een mogelijkheid die de kracht van open source demonstreert. De homepage van `patTemplate` is www.php-tools.de.

Het templatingsysteem biedt niet alleen een vereenvoudigde notatiewijze met XML-tags, maar ook extra mogelijkheden als caching (wat gebruikt wordt door de officiële Joomla! Cache).

Output via JDocument

Bij het genereren van output biedt Joomla! keuzen uit verschillende bestandsindelingen. De content kan in een gangbare XHTML-indeling gepresenteerd worden, maar ook in de vorm van RSS-feeds of een PDF-document dat gemakkelijk afgedrukt kan worden.

De klasse `JDocument` faciliteert het vertalen van content naar daadwerkelijke output. Hierbij worden subklassen gebruikt zoals `JDocumentHTML` (verantwoordelijk voor het vertalen van de Joomla!-template naar HTML-output inclusief content), `JDocumentPDF` (voor PDF-documenten), `JDocumentFeed` (voor RSS-feeds) en `JDocumentRAW` (voor overige bestandstypen).

Eerder werd de toepassing `JApplication` genoemd. Feitelijk is de klasse `JDocumentHTML` verantwoordelijk voor manipulatie van het HTML-document. Hierbij kan het toevoegen van metatags, CSS-stylesheets en JavaScript aan de

```

echo "<span class='" . $class . "_title'">" ;
echo $module->title ;
echo "</span>" ;
}
echo "<span class='" . $suffix . "'">" ;
echo $module->content ;
echo "</span>" ;
}

```

Deze stijl kan nu worden toegepast in de template:

```
<jdoc:include type="module" name="left" style="simplespan" />
```

Administrator template overrides

De Joomla! Administrator is ook opgebouwd uit modules en componenten, en de meeste van deze extensies zijn ook opgebouwd volgens het MVC-model. Dit betekent dat binnen een Joomla! Administrator-template ook template overrides geplaatst kunnen worden.

Voorbeeld: com_installer

Bij het installeren van een extensie is er de mogelijkheid om een extensie te installeren met een uploadknop, een serverfolder en direct vanuit een url. De laatste optie maakt het installeren van een extensie erg gemakkelijk, omdat het de tussenstap van het downloaden naar de eigen computer overbodig maakt.

De moeilijkheid hierbij blijft natuurlijk de vraag welke extensies geschikt zijn voor het installeren onder Joomla! 1.5? Het zou mooi zijn als de Joomla! Administrator een lijst toont met vooraf gedefinieerde extensies die probleemloos geïnstalleerd kunnen worden. Een core hack is hiervoor niet nodig. Dit probleem is eenvoudig op te lossen met een template override van de Administrator-component com_installer.

In de map administrator/templates/khepri/ moet een map html gemaakt worden met daarin de map com_installer (voor de template override van de Administrator-component com_installer). Het formulier om extensies te installeren wordt getoond met de view install, dus in de map com_installer maken we een extra map install.