TEMPLATE IN MWCMS

DANIELE UGOLETTI - GRUPPOMETA

MINISTERO PER I BENI E LE ATTIVITÀ CULTURALI

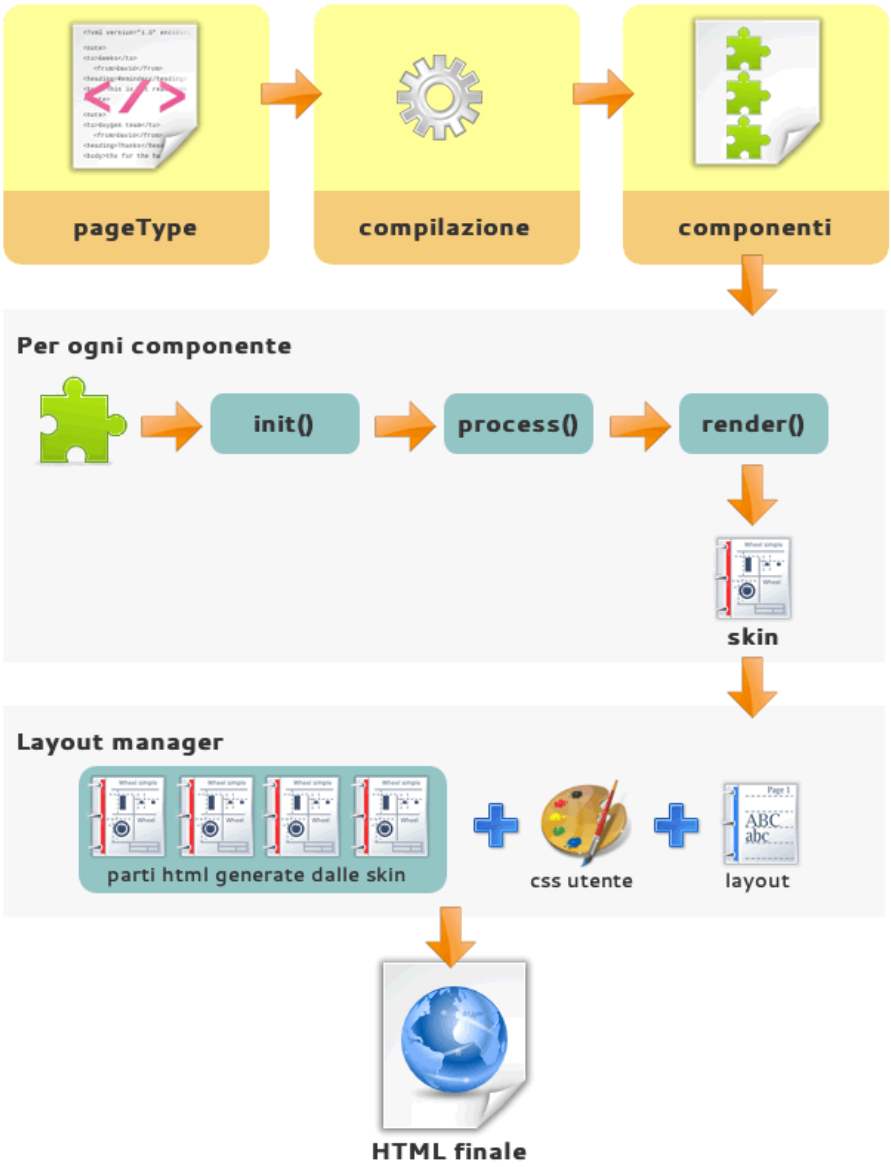
Si prega di comunicare eventuali errori o inesattezze riscontrate, scrivendo una mail a: otebac@beniculturali.it

CONTENTS

1

I TEMPLATE IN MWCMS

Un sito gestito con MWCMS è completamente personalizzabile in tutti i suoi aspetti, quindi è possibile modificare il layout oppure quello che viene disegnato da un componente. Per capire bene questo concetto partiamo dallo schema di esecuzione di una pagina:



Riassumiamo gli step fondamentali, per una spiegazione più approfondita si rimanda al documento "Architettura MWCMS":

- *quando avviene una richiesta per una pagina, MWCMS cerca il pageType associato alla pagina.*
- *il pageType contiene una serie di componenti*
- *alcuni componenti, la maggior parte di quelli usati in MWCMS, producono un frammento HTML delegando il rendering ad un file di skin*

- *ogni componente ha un attributo editableRegion che indica la zona in cui deve essere messo il frammento HTML da lui generato, questo zone saranno gestite dal LayoutManager (vedere capitolo successivo).*

- *Quando tutti i componenti hanno prodotto il loro frammento HTML, viene eseguito il LayoutManager che assembla questi frammenti con il layout del sito.*

Da questo schema si individuano due attori fondamentali: skin e layout, quindi per MWCMS un template è un insieme di skin e di layout.

ANATOMIA DI UN TEMPLATE

MWCMS ha la possibilità di gestire più template e tramite l'interfaccia d'amministrazione è possibile scegliere quale utilizzare. Un template può anche essere configurabile tramite l'interfaccia d'amministrazione, per esempio impostare l'immagine di testata o i colori.

I template si trovano nella cartella static/org_minervaeurope_museoweb/templates/ attualmente sono distribuiti 3 template: Template 1 2-col, Template 1 3-col, Template 2.

STRUTTURA DEI FILE PRESENTI NELLA CARTELLA DEL TEMPLATE

Per capire com'è strutturato un template analizziamo il Template 2, il contenuto della cartella static/org_minervaeurope_museoweb/templates/T2/ è il seguente:

```
/
        assets/
        locale/
        skins/
        class color.inc.php
        home 3cols.php
        home.php
        page 3cols.php
        page.php
        preview.jpg
        Template.php
        Template.xml
        TemplateAdmin.xml
```

Per essere un template riconosciuto da MWCMS devono essere presenti i file evidenziati in grassetto, di seguito il dettaglio del significato dei file e cartelle:

ASSETS/

Cartella contenente i file di supporto (css, immagini) del template, questa cartella è specifica per il template 2, non è necessaria se si sviluppa un template nuovo.

LOCALE/

Se il template ha una parte di amministrazione per personalizzarlo e si vuole localizzare l'interfaccia nella varie lingue è necessario mettere in questa cartella i file di localizzazione, vedere il documento “Localizzazione MWCMS” per indicazioni sul sistema di localizzazione in MWCMS.

SKINS/

Cartella contenente tutti i file di skin.

CLASS_COLOR.INC.PHP

File specifico del Template 2 non necessario se si sviluppa un template nuovo.

HOME_3COLS.PHP

File di layout per il layout a 3 colonne delle home, file specifico del Template 2 non necessario se si sviluppa un template nuovo.

HOME.PHP

File di layout della home.

PAGE_3COLS.PHP

File di layout per il layout a 3 colonne delle pagine interne, file specifico del Template 2 non necessario se si sviluppa un template nuovo.

PAGE.PHP

File di layout delle pagine interne.

PREVIEW.JPG

Anteprima usata in amministrazione per la selezione del template.

TEMPLATE.PHP

File contenente il codice per applicare le personalizzazioni al layout, questo file lega il suo funzionamento al file TemplateAdmin.xml.

TEMPLATE.XML

File XML con la definizione dei componenti, si può considerare questo file come una porzione di pageType, questi componenti vengono aggiunti in automatico al pageType che si sta visualizzando. Questo file è molto importante perché permette di modificare il comportamento standard di MWCMS, per esempio è possibile variare il funzionamento delle barre di navigazione per un particolare template.

TEMPLATEADMIN.XML

File XML con la definizione dei componenti utilizzati in amministrazione per disegnare la pagina di personalizzazione del template (Template e Colori).

2

LAYOUT

Il layout è un file PHP contenente il codice HTML con il layout della pagina, non c'è un vincolo su come è costruito questo layout o su come deve essere il css, lo sviluppatore ha una completa libertà, l'unico vingolo è che nel file sia presente il codice PHP per il disegno dell'editableRegion. Senza questo codice la pagina inviata al browser sarà vuota.

Il codice per il disegno di un'editableRegion è un semplice `print` o `echo` specificando il nome della region:

```
<?php print( $miaRegion ); ?>
```

Nel capitolo precedente abbiamo visto che i layout necessari sono due: **home.php** e **page.php**. Di seguito verranno elencate le region usate per ogni file, se un layout non implementa una regione, MWCMS non dà errore e la pagina viene renderizzata comunque, l'unico effetto che si ha è che il contenuto previsto per quella region non sarà visualizzato.

REGION IN PAGE.PHP

DOCTITLE

Sostituisce il tag `<title>titolo della pagina</title>` dell'HTML

METATAGS

Disegna i metatag della pagina

HEAD

Solitamente usato per inserire codice javascript o css nell'head della pagina.

HIDDENNAV

Navigazione nascosta da usare per la navigazione assistita con tastiera.

TOPNAV

Region usata per disegnare la metanavigazione.

LANGUAGES

Region usata per disegnare la barra del cambio delle lingue.

LEFTSIDEBAR

Region usata per la colonna sinistra.

RIGHTSIDEBAR

Region usata per la colonna di destra.

BREADCRUMBS

Region per le molliche di pane.

CONTENT

Contenuto della pagina.

AFTERCONTENT

Region alla fine di tutti i contenuti disegnati dai pageType, viene usata per inserire i commenti delle pagine.

SIDEBARADDRESS

Viene disegnato l'indirizzo ed i copyright del sito.

TAIL

Ultima editableRegion della pagina, prima della chiusura del body, attualmente viene usata per inserire il codice di Google Analytics.

REGION IN HOME.PHP

DOCTITLE

Sostituisce il tag `<title>titolo della pagina</title>` dell'HTML

METATAGS

Disegna i metatag della pagina

HEAD

Solitamente usato per inserire codice javascript o css nell'head della pagina.

HIDDENNAV

Navigazione nascosta da usare per la navigazione assistita con tastiera.

TOPNAV

Region usata per disegnare la metanavigazione.

LANGUAGES

Region usata per disegnare la barra del cambio delle lingue.

LEFTSIDEBAR

Region usata per la colonna sinistra.

BOXLINK

Region usata per i box con i link

TEXT

Testo della home

NEWS

Elenco news della home

EVENTS

Elenco eventi della home

FEED

Link per feed RSS della home.

AFTERCONTENT

Region alla fine di tutti i contenuti disegnati dai pageType, viene usata per inserire i commenti delle pagine.

SIDEBARADDRESS

Viene disegnato l'indirizzo ed i copyright del sito.

TAIL

Ultima editableRegione della pagina, prima della chiusura del body, attualmente viene usata per inserire il codice di Google Analytics.

3

SKIN

I file di skin sono contenuti nella sottocartella “skins” del template, questa cartella contiene molti file, uno per ogni componente, si è cercato di mantenere una convenzione nei nomi dei file, per esempio la skin del risultato della ricerca per il modulo Opere si chiama Catalog_list.html, la skin per il dettaglio dell'opera di chiama Catalog_entry.html.

Le skin in MWCMS utilizzano come linguaggio di definizione PHPTAL, questo è un esempio di skin:

```
<div class="boxNewsEvents" tal:condition="php: count(RecordSetBox['records'])">
        <h3 tal:content="RecordSetBox/title" />
        <div tal:repeat="item RecordSetBox/records">
                <h4 class="date" tal:content="structure item/newsdetail startDate" />
                <h4><a href="" tal:content="structure item/newsdetail title" tal:attributes="href
item/  url  ; structure title item/newsdetail title"></a></h4>
                <p><span tal:omit-tag="" tal:content="structure item/newsdetail bodyShort" /><a
tal:attributes="href item/  url  ; structure title item/newsdetail title" class="moreRight"><span
tal:omit-tag="" tal:content="php:   T('MW MORE')"/></a></p>
        </div>
        <div class="clear"></div>
</div>
```

Come si può vedere il codice è HTML con degli elementi in più (tal:...) che indicano l'azione da svolgere, per la sintassi completa del linguaggio TAL si rimanda alla documentazione del sito ufficiale:

http://phptal.org/manuals.html

Come abbiamo visto le skin sono associate ad un componente che a sua volta è definito in un pageType, ma come avviene quest'associazione? Esistono due modi: skin definita all'interno del pageType, skin definita esternamente. In MWCMS tutte le skin sono definite esternamente proprio per rendere la personalizzazione del template più semplice.

SKIN DEFINITA ALL'INTERNO DEL PAGETYPE

```
<?xml version="1.0" encoding="iso-8859-1"?>
<glz:Page xmlns:glz="http://www.glizy.org/dtd/1.0/" xmlns:gm="it.gruppometa.metacms.*" id="Page"
templateType="php" templateFileName="page.php" defaultEditableRegion="content">
        <glz:Import src="Common.xml"/>
        <glz:LongText id="text" label="Testo" rows="20" cols="75" htmlEditor="true" forceP="true"
skin="{miaSkin}"/>
        <glz:SkinDefine id="miaSkin"><![CDATA[
..... codice della skin
        ]]></glz:SkinDefine>
</glz:Page>
```

Per definire una skin all'interno di un pageType è necessrio:

- definire nel componente a cui si vuole agganciare una skin l'attributo `skin="{idDellaSkin}"` dove `idDellaSkin` è l'id del componente `glz:SkinDefine` contenente la skin.

- Inserire la skin dentro il componente `glz:SkinDefine` questo componente deve avere l'id come indicato sopra.

SKIN DEFINITA ESTERNAMENTE

```
<?xml version="1.0" encoding="iso-8859-1"?>
<glz:Page xmlns:glz="http://www.glizy.org/dtd/1.0/" xmlns:gm="it.gruppometa.metacms.*" id="Page"
templateType="php" templateFileName="page.php" defaultEditableRegion="content">
        <glz:Import src="Common.xml"/>
        <glz:LongText id="text" label="Testo" rows="20" cols="75" htmlEditor="true" forceP="true"
skin="miaSkin.html"/>
</glz:Page>
```

Per definire una skin esterna è necessario:

- definire nel componente a cui si vuole agganciare una skin l'attributo

  `skin=”nomeFileDellaSkin”`

- creare il file nella cartella skins del template.

4

COME MODIFICARE UN TEMPLATE DI MWCMS

Durante lo sviluppo di un sito può essere necessario fare alcuni interventi di modifica del layout o delle skin rispetto al template standard distribuiti con MWCMS, per fare questa operazione si consiglia di non modificare il template ma di duplicarlo e di crearne uno nuovo. Il vantaggio in questa operazione è di non perdere le modifiche fatte quando ci sarà un aggiornamento di MWCMS.

Questi sono i passaggi da fare per la duplicazione di un template, come esempio prenderemo il template 2.

1. andare nella cartella static/org_minervaeurope_museoweb/templates/

2. duplicare la cartella “T2” e rinominarla in “T2 modificato”

3. aprire i file dentro “T2 modificato/locale” e sotituire la riga `"T2" => "Template 2"`, con `"T2 modificato" => "Template 2 modificato"`, questa modifica indichera all'amministrazione del CMS il nuovo nome del template.

4. Andare in amministrazione nella sezione “template e colori” e selezionare il nuovo template.

A questo punto MWCMS utilizzerà il nuovo template, quindi è possibile fare qualsiasi modifica al layout o alle skin, per esempio potremmo voler modificare i dati visualizzati nel risultato della ricerca del modulo news, apriamo il file skin/news_list.html:

```
<div class="searchResults" tal:condition="php: !is_null(RecordSetList['records'])">
	<h3 tal:content="structure RecordSetList/title"/>
	<span tal:omit-tag="" tal:condition="php: count(RecordSetList['records'])">
		<div tal:repeat="item RecordSetList/records" tal:attributes="class item/__cssClass__">
			<div class="around">
				<h4><a href="" tal:attributes="href item/__url__; structure title item/newsdetail_title"
tal:content="structure item/newsdetail_title">news</a></h4>
				<h4 class="subtitle" tal:content="php: MW_formatDate(item['newsdetail_startDate'])" />
				<span tal:content="structure item/newsdetail_bodyShort" tal:omit-tag=""/>
			</div>
		</div>
	</span>
	<span tal:omit-tag="" tal:condition="php: !count(RecordSetList['records'])">
		<div class="around" >
			<p tal:content="php:__T('MW_NO_RECORD_FOUND')"></p>
		</div>
	</span>
	<div class="clear"></div>
</div>
```

Se vogliamo togliere la data basta rimuovere la riga evidenziata e salvare il file, allo stesso modo è possibile aggiungere informazioni. Le informazioni disponibili variano da modulo a modulo e si può far riferimento alle tabelle del database.

5

COME CREARE UN NUOVO TEMPLATE

Per creare un nuovo template si consiglia di partire da un template esistente in questo modo sono presenti tutti i file di skin necessari applicazione, se non si volesse usare un template esistente è necessario implementare tutte le skin contenute nella cartella skins, quindi eseguire la duplicazione come indicato nel capitolo precedente; in questo modo sono definite tutte le skin ed i file necessari, poi è necessario fare un po' di pulizia:

1. cancellare la cartella `assets`

2. cancellare la cartella `locale`, cancellando questa cartella il nome usato in amministrazione corrisponderà al nome della cartella del template.

3. cancellare i file Template.php, home_3cols.php, page_3cols.php, TemplateAdmin.xml

4. sostituire il file di anteprima preview.jpg, La dimensione del file è 125 x 250

5. aprire il file Template.xml e lasciare solo questo codice:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<glz:EmptyComponent>

</glz:EmptyComponent>
```

Fatte queste operazioni si ha un template funzionante ma senza stili da poter modificare.