

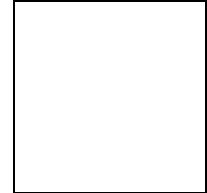
dotCMS Training Documents

From Structures to Pages:

A Step-by-Step Tutorial for creating Templates in dotCMS

Last Updated: May 01, 2007





Content

Overview	3
Prerequisites.....	3
Definitions.....	3
Scenario.....	4
Plan Out What Goes Where	4
Order of Creation	5
Step 1. Create a Content Structure.....	6
Step 2. Create a Container.....	8
Step 3. Create a Template:.....	11
Step 4. Create Homepage:.....	12
Step 5. Create Detail Page:.....	12

Overview

Prerequisites

For this tutorial you must have a good understanding of HTML. You must also be familiar with creating pages in the CMS and adding contents to a container. To learn how to do this, please refer to the user's manual.

Definitions

- **Template:** Templates are layouts available to users when building new HTML, XHTML or XML pages. The code that makes up a dotCMS template looks very much like standard HTML or XML code. The only difference between templates and HTML is that templates include *containers* which act as server side includes.
- **Container:** Containers in dotCMS perform two functions. First, they act like server-side include files and can be used to place reusable HTML elements across templates and pages. A good use of this type of container function is for things such as global headers and footers, main menu bars/navigation and other site wide-type content. Second, containers can define where *content* (from the *Content Repository*) can be placed on a *template*, how that *content* should be formatted and whether that *content* can be edited through the inline editing mode.
- **Content Repository:** dotCMS is built upon the idea that content should be totally separated from pages and should be reusable by selected user across a site or sites.
- **Structures:** Structures serve to hold and organize information in fields created within dotCMS. The CMS Administrators can use these Structures to specify Content Types to be displayed by Containers. These Content Structures break up information into various fields (Title, Subtitle, Description, Body, etc.) and then display some or all of these fields when it is associated with a page of your website.
- **Content:** A contentlet is a piece of content composed of fields given in a Structure. Think of a structure like an income tax form and the contentlet as your particular information that you fill in.
- **inode:** unique identifying number given to Templates / Containers / Contentlets and almost every other piece of information in the CMS.

- **Widget:** A piece of content that dynamically displays other content in it.
- **Content Types:** A particular Structure, ex. “A biography content type”.

Scenario

Homepage News Listing

- Must be able to manually add, remove and reorder news items

News Detail Page

- Must dynamically show news story based on inode passed in link.
- If an inode is not passed, then the latest news item will be displayed.
- On the right hand side there should be a listing of the latest news stories, ordered by published date.

Plan Out What Goes Where

We are assuming at this point we have the html, css, and images ready for the page. It's a good idea to know what the relationship is between the Template, Container, Contentlet (Illustration 1).

The page template will contain the standard sections of the page, such as the head tag, the

Note: For chunks of HTML such as a header or footer that are shared among templates, we can create containers that have no contentlets, making the template easier to manage and more streamlined.

page header, the page footer and other elements of a page that are standard for all the pages that use that specific template.

For the first part of this tutorial, we are going to create the news listing on the homepage. If we look at the design of the homepage, the container for the News Items will be a part of the page that has a container. There are other containers on this page, but for the sake of this



Illustration 1: The red box shows the boundaries of the template. The blue line shows the boundaries of the news listing container and the green boxes show the boundaries of the code part of the container that loops the contentlet that we add to the container.

tutorial we will go into detail on the “News Items” container.

Order of Creation

1. Structure
2. Containers
3. Template
4. Page
5. Contentlets

The order of creating Structure, Container, Template, Page then Contentlet is important. Each one depends on what was built before it. Variables from a structure are used in the container created after it. When creating a template you need to find the containers that you're going to include in it. When you create a page, you need to select a template. When you have that page, you can then create/re-use contentlets. Keeping this order saves a lot of time and makes things less confusing. If certain containers, templates, contentlets, etc. are already created and are going to be re-used you obviously don't have to create it, but still maintain the correct order.

Step 1. Create a Content Structure

Note: If you already have a structure you are going to use, procede to step 2.

1. Click on the Structures tab.
2. Click on the “Create New Structure” link.
3. Fill in “Add Name” and “Description” Fields
4. Click “Save Structure”. Once the structure is saved, you can start adding fields to it.

(Illustration 2)

Structure Field Details (Illustration 3):

dotCMS
by dotmarketing

Welcome, Will Ezell!
Home - My Account - Personalize Pages - Sign Out

Home Content CMS Admin **Structures** Workflow Website Events

Community Page: CMS Administrators

Structures
View all Structures | Add New Structure | View all Relationships | Add New Relationship

Structures

Properties

Structure Name: Description:

☒ No Review ☐ Enable Review
Review Every: Day

Reviewer Role:

Permissions

Action	Label	Display	Variable	DB Column	Required	Searchable	Show in List	Sort Order
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	Headline	TEXT	NewsHeadline	text1	Yes	Yes	Yes	<input type="text" value="1"/>
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	Publish Date	DATE_TIME	NewsPublishDate	date1	Yes	No	Yes	<input type="text" value="2"/>
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	Expire Date	DATE_TIME	NewsExpireDate	date2	No	No	No	<input type="text" value="3"/>
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	Short Summary	TEXTAREA	NewsSummary	text_area1	Yes	No	No	<input type="text" value="4"/>
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	Body	WYSIWYG	NewsBody	text_area2	Yes	Yes	No	<input type="text" value="5"/>
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	Image	IMAGE	Image	integer1	No	No	No	<input type="text" value="6"/>

|

About - Disclaimer

Illustration 2: Structure with Fields Entered

- a) **Label:** the name given to the field within the Content Structure
- b) **Display Type:** Determines how a field displays in HTML on the “Add New Content” form, i.e. Text, Radio button, WYSIWYG, Select Drop Down, Text Area, etc.
- c) **Data Type:** determines the data type that will be entered into a field: Boolean (yes/no), Date, Decimal, Integer, Text, Long Text
- d) **Values:** Information entered here will be the values the User will be able to select from when adding Content in this field. It is only used for drop down list, radio buttons, checkbox, select boxes or multiselect boxes.
- e) **Validation:** Validation placed on any field will designate the characters and/or format of the data allowed in the field, i.e. email address, numbers, no HTML. Administrators can write their own regular expressions if needed for special validations.
- f) **Hint:** Displayed to the User as reminder as to what content will be allowed in the field or where the entered data will be used, i.e. example data or a detail page
- g) **Default Value:** The default value assigned to the field before user input.
- h) **Required:** when checked, the associated field requires validated data entry.
- i) **Searchable:** when checked, the Content Search will use this field in its search.
- j) **Show in Listing:** when checked, the content in this field will be displayed in the Content Search results on the backend.

The screenshot shows the 'dotCMS by dotmarketing' interface. The top navigation bar includes links for Home, Content, CMS Admin, Structures (active), Workflow, Website, and Events. A user greeting 'Welcome, Will Ezell!' is visible in the top right. Below the navigation bar, the 'Structures' section is active, showing options to 'View all Structures' or 'Add New Structure'. The 'Add/Edit Field' form is displayed, containing the following fields and options:

- Label:** Headline
- Display Type:** Text (selected from a dropdown)
- Data Type:** Boolean, Date, Decimal, Integer, Text (selected), Long Text
- Values:** A large text area for entering values.
- Validation:** A dropdown menu with '<--Select validation-->'.
- Hint:** A text input field.
- Default Value:** A text input field.
- Required:** ☒
- Searchable:** ☒
- Show in listing:** ☒

Buttons for 'Save Field' and 'Cancel' are at the bottom of the form. A 'Required Fields' indicator is present in the top right of the form area.

Illustration 3: Sample Field

Step 2. Create a Container

This container will be the listing on the homepage in which the contentlets can be manually added, reordered and removed. Each news item's title will link to the detail page and pass the inode of the contentlet to the detail page.

1. Click on the "Website" tab and then the "Container" tab.
2. Click "Add Container" button.
3. In the code field (read below in Container Details) we are going to put in the variables for the headline and a link to a detail page, also passing the inode of the contentlet in the URL. (`$!{NewsHeadline}`)

Container Details (Illustration 5):

- a) **Identity:** This number will be used when creating and editing Templates in the CMS. The number is assigned automatically and is unique to every Container
- b) **Host:** The host to which this Container is assigned
- c) **Title:** Title given to the Container by the user.
- d) **Description:** A short description of the Container. This is displayed only on the back end.
- e) **Max. Content:** The maximum number of pieces of Content that a user can place in a Container.
- f) **Content's Structure:** Determines the Structure that will be allowed to be used in this Container
- g) **Pre-Loop:** This field will contain the HTML used when the Container is parsed. This field will only be processed once, i.e. opening an HTML tag that will stipulate how the content will be formatted within the Container - `<div class="[ClassName]">`
- h) **Code:** The Code entered in this field will be processed for each piece of Content added to the Container (determined by the "Max. Content" field).
 - The fields within a Structure will be utilized according to the variables used here.
 - Each variable represents a Field in a Content Structure.

- To select a variable from a list of available Fields from the Structure, click “add variable” located to the right of the text area.
- This will pop a new window with all the Fields that make up the Structure that will be used in the designated Container. (Illustration 4)
- Clicking “Add | Current Value” under the desired Field Label will add a variable for that Field to the Code field and close the popup window.
- The steps can be repeated to add more fields.

The illustration shows a pop-up window titled "Add a Variable" with a list of fields and their corresponding code values. The fields are grouped into three sections: "Content Inode", "Headline Text Field", and "Short Summary Text Area". Each section has an "add" button and a "Current Value" field. The "Content Inode" section also has a "Text Field Code" field. The "Short Summary Text Area" section also has a "Textarea Code" field. The "Body Text Area" section is partially visible at the bottom.

Field Label	Current Value	Code
Content Inode	Content Inode Value	
Headline Text Field	Current Value	
		Text Field Code
Short Summary Text Area	Current Value	
		Textarea Code
Body Text Area		

Illustration 4: Add a Variable pop-up

- i) **Post-Loop:** Much like the Pre-Loop, it is processed after the Code field and only once when the Container is processed. Can be used to close the Container, i.e. `</div>`

The screenshot displays the dotCMS CMS interface. At the top, the dotCMS logo is visible, along with a navigation bar containing links for Home, Content, CMS Admin, Structures, Workflow, Website, and Events. A user greeting 'Welcome, Will Ezell!' and links for 'Home - My Account - Personalize Pages - Sign Out' are on the right. Below the navigation bar, the 'Website Browser' section shows tabs for 'Browser', 'HTML Pages', 'Files', 'Menu Links', 'Containers', and 'Templates'. The 'Containers' tab is active, showing 'View Working Containers | Add/Edit Container'. The 'Add/Edit Container' form is open, featuring three tabs: 'Properties', 'Permissions', and 'Versions'. The 'Properties' tab is selected, showing fields for Identity (128), Title (News Listing (homepage)), Description (News Listing (Homepage)), Max. Contents (5), Content's Structure (News Item), and Dynamic (unchecked). The 'Pre-Loop' field contains a code snippet for setting variables and generating an RSS feed. The 'Code' field contains a loop for displaying news items. The 'Post Loop' field contains closing tags for the loop and the container. The 'Notes' field is empty. At the bottom of the form are buttons for 'save', 'save and publish', 'check spelling all fields', and 'cancel'.

dotCMS
by dotmarketing

Welcome, Will Ezell!
Home - My Account - Personalize Pages - Sign Out

Home Content CMS Admin Structures Workflow Website Events

Community Page: CMS Administrators

Website Browser

Browser HTML Pages Files Menu Links Containers Templates

View Working Containers | Add/Edit Container

Add/Edit Container

Properties Permissions Versions

Identity: 128
Title: News Listing (homepage)
Description: News Listing (Homepage)
Max. Contents: 5
Content's Structure: News Item
Dynamic: ☐

Pre-Loop:

```
#set($today = $UtilMethods.htmlDateTimeToDate(""))  
#set($actualYear = $UtilMethods.dateToHTMLDate($today,"MMMM  
d, YYYY"))  
  
<link rel="alternate" type="application/rss+xml"  
title="Dotmarketing University News"  
href="http://$(host.hostname)/news/news_rss.dot" />  
  
<div class="news">  
<a href="/news/?inode=${ContentInode}">${NewsHeadline}</a></li>
```

Post Loop:

```
</ul>  
</div>
```

Notes:

save save and publish check spelling all fields cancel

Illustration 5: News Listing Container

Step 3. Create a Template:























1. Click on the “Website” tab.
2. Click on the “Templates” tab.
3. Click the “Add Template” button.
4. Add Containers to the template by placing the cursor where you want a container and pressing the “Add Container” button. This will bring up a popup window in which you can browse for the right container and it will place the needed code back in the template.
5. Once you have all the containers in the template, click “save and publish”.

Tips on Templates:

- *Putting width constraining elements within the template which wrap around the containers helps keep the page elements in the right place when in preview mode.*
- *A container can only be used once in a template.*

View All Working Containers

Any Structure ☐ Show Deleted

Status	Title	Mod. Date
 	Banner	10/30/06 9:40 AM
 	Blank Container	2/21/07 4:30 PM
 	Blog Container	11/10/06 3:58 PM
	Body (default)	3/7/07 2:00 PM
 	Column 1 (default)	3/7/07 2:00 PM
	Column 3 (default)	3/7/07 2:00 PM
	Column 4 (default)	11/6/06 11:43 AM
 	Footer	3/7/07 2:00 PM
 	Header	3/7/07 2:00 PM
	News Listing (homepage)	10/30/06 9:40 AM
 	Newsletter Sidebar	6/29/06 10:15 AM
	Newsletter Story	6/29/06 10:15 AM
	Promotion	10/30/06 9:40 AM
 	Side Nav	3/7/07 2:00 PM




Status Legend:
 live  archived  locked

Illustration 6: Popup Container Browser

Step 4. Create Homepage:

1. Look for the folder “home”, or create it if it isn't there, making sure to select the template we just created.
2. Once you save this page, double click on it to open the page in edit mode.
3. Now you can add already existing news items to the container or create new news items from the homepage.
4. Don't forget to click “save and publish” so you can check your changes live.

Step 5. Create Detail Page:

For the detail page we are going to use widgets to display the detail for the news story and another widget to list the latest news stories.

The Advantages of Widgets: Widgets have two big advantages. For one, they help us keep the number of unique templates down. The more templates you have in a list when a user creates a new page, the more confusing it can be. Templates should be created sparingly for creating different layouts or different themes to pages. It's important to try to re-use the templates as much as possible and use generic “Web Page Content” content types for the containers. By placing widget contentlets into these containers, you can pull lists of any kind of contentlet and display them using velocity code. It also has the advantage that the listings update

1. Go into / create “news” folder and create an index page using a generic body page template.

2. Add a new contentlet to the body. Here's the code we are going to use:

```
#set($inode = $request.getParameter('inode'))
```

this line sets the inode variable on the page to the inode from the URL using the `$request.getParameter`

```
#if($inode )
    #getContentMapDetail($inode)
#else
    #pullContent('+type:content +live:true +deleted:false
+structureInode:5 +languageId:1' '1' 'date1 asc')
#end
```

This if statement checks if there is an inode variable (passed from the URL in the previous line). It then uses the `#getContentMapDetail` to load the fields from that particular contentlet with that inode. If not it uses `#pullContent` to get the latest news story using a query (you can paste this from a contentlet search in the contentlet tab [Illustration 7]).

```
<h2>${content.NewsHeadline}</h2>
<span>${date.format('MM/dd/yyyy',$content.NewsPublishDate)}</span>
<hr/>
```

`$date.format` is a velocity method for formatting dates from the database's format to human friendly one.

```
#if($UtilMethods.isSet($content.Image) && ($content.Image != '0'))

#end
```

This if statement checks if there is an image attached to this contentlet by using `$UtilMethods.isSet` to see if that variable has been set.

```
${content.NewsBody}
```

3. Now we are done with the detail widget. Publish the page to view the results and click a news story link from the homepage to test it out.

Dotmar

Qatar C

College

Senate

Colorac
open-si

live

working

deleted

Show Query

About - Disclaimer

Qatar Grants Millions in Aid to New Orleans

College Admission Angst Finds a Forum on Web

Senate begins immigration debate

Colorado State University and Dotmarketing to develop a comprehensive open-source platform

live

working

deleted

locked

unlocked

Hide Query

+type:content +deleted:false +structureInode:5 +(languageId:1* +live:true)

Ordered by: date1 desc

(Hint 1: Change +working:true by +live:true in your query if you want to show only the live items)

(Hint 2: You can add date ranges to your query using the following syntax: +dateField: +modDate:[02/02/2006 TO 01/01/2100]

E.G.1: +modDate:[02/02/2006 TO 01/01/2100]

E.G.2: +date1:[12/21/1990 05:12:12 TO 01/01/2000 00:00:00])

About - Disclaimer

Illustration 7: Contentlet Search for all "News Items", results ordered by "Publish Date", "the Show Query" link is circled in red. Clicking on this will give you the parameters for the #pullContent('query' '# of results' 'order by'). The 'query' is marked green and the 'order by' is marked in blue.

- Next we are going to create a listing for the right hand column of the news detail page using this code:

```
#pullContent('+type:content +live:true +deleted:false  
+structureInode:5 +languageId:1' '5' 'modDate')
```

This line uses the #pullContent macro to get a list of the latest 5 news items in order of last modified date. For more details on the #pullContent macro see Illustration 7.

```
<link rel="alternate" type="application/rss+xml" title="Dotmarketing  
University News" href="http://${host.hostname}/news/news_rss.dot"  
>
```

```
<div class="news">  
<h2>Top Stories<a  
href="http://${host.hostname}/news/news_rss.dot"></a></h2>  
<ul>
```

```
#foreach($content in $list)  
<li><a  
href="/news/?inode=${content.inode}">${content.NewsHeadline}</a></  
li>  
#end
```

This foreach grabs the each contentlet in the list generated by the #pullContent macro and loops through each one printing out the variables for that particular contentlet in each pass.

```
</ul>  
</div>
```

5. We now save and publish the detail page.

Enjoy!