Siempre Solutions

Steve Morgan – www.siempresolutions.co.uk

UMBRACO CERTIFIED DEVELOPER

How to Create an Umbraco Website from Flat HTML Files

Revision History

Version	Release Date	Author	Comment / Amends
1.0	20/04/2014	Steve Morgan	Initial Release.

1 Introduction

The idea of this document is to take you step by step through an Umbraco website build. It will allow you to take any website "template" (e.g. flat HTML, CSS and JS) and install it into a fresh Umbraco and wire up the sections that need content managing in Umbraco. Umbraco is a seriously powerful CMS but many find the learning curve of installing a website from scratch a little too much – this guide aims to explain the mysteries!

We avoid using one of the starter kits as when it comes to building your own site these don't give you an understanding of the basics of Umbraco Document Types and Templates and how these work together to build pages.

2 What You'll Need

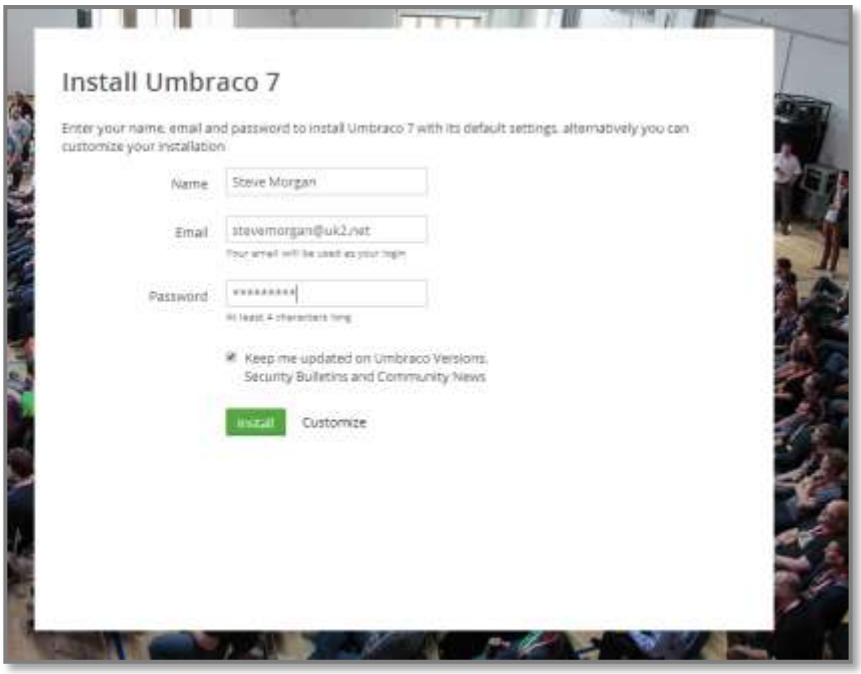
To take you through a demo of installing a basic site in Umbraco you need the following:

- A clean, empty installation of Umbraco – e.g. no starter site installed, see the notes below what to do when running through the installation wizard. Use the latest main 7.X download. Follow the installation steps in the documentation http://our.umbraco.org/documentation/Installation
- A copy of Initializr – a HTML5, responsive template which is a nice start for any website. https://github.com/verekia/initializr-template/archive/master.zip if you prefer you can use your own flat HTML files but my examples will use this.

3 Getting Started

3.1 Installing an Empty Umbraco

This guide doesn't cover the installation of Umbraco – follow the instructions in http://our.umbraco.org/documentation/Installation. When you see the first splash screen click ***customize*** – either fill in your MS SQL blank DB credentials or use the CE option – then on the final screen use the "***No thanks I do not want to use a starter website***".



Install Umbraco 7

Enter your name, email and password to install Umbraco 7 with its default settings, alternatively you can customize your installation

Name: Steve Morgan

Email: stevemorgan@uk2.net

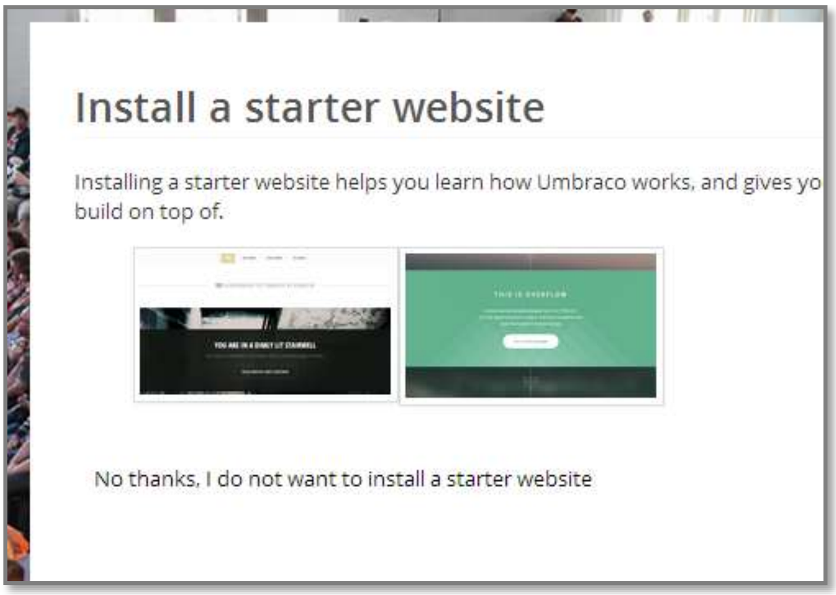
Your email will be used as your login

Password: *********

At least 4 characters long

[✓] Keep me updated on Umbraco Versions, Security Bulletins and Community News

Figure 1 - Installation Splash Screen - note the Customize link



Install a starter website

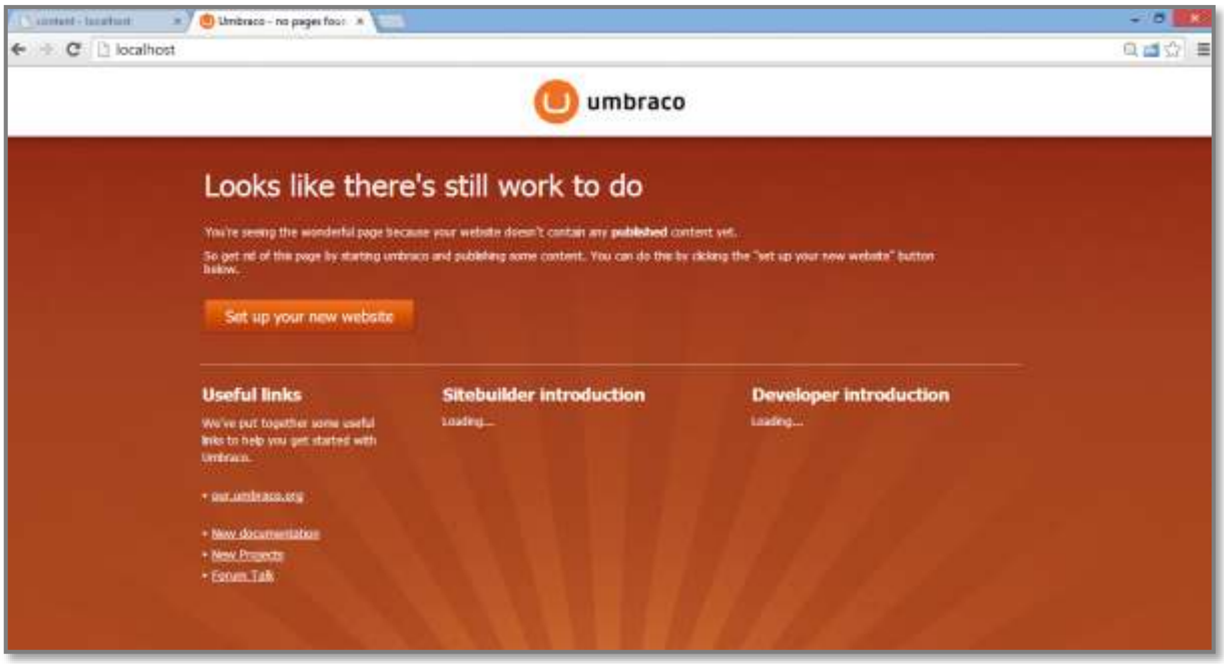
Installing a starter website helps you learn how Umbraco works, and gives yo
build on top of.

No thanks, I do not want to install a starter website

Figure 2 - Install a starter website - No Thanks!

3.2 Checking you have an Empty Umbraco Install

When you hit your local host address (http://localhost or whatever you've set up) you should see the Umbraco empty page screen.



Looks like there's still work to do

You're seeing the wonderful page because your website doesn't contain any **published** content yet.

So get rid of this page by starting umbraco and publishing some content. You can do this by clicking the "set up your new website" button below.

Useful links

We've put together some useful links to help you get started with Umbraco.

- our.umbraco.org
- New documentation
- New Projects
- Forum Talk

Sitebuilder introduction

Loading...

Developer introduction

Loading...

Figure 3 - This is correct – we have a blank, empty Umbraco website!

If you can see the Umbraco Starter kit site you've missed the option to install Umbraco with no starter site.

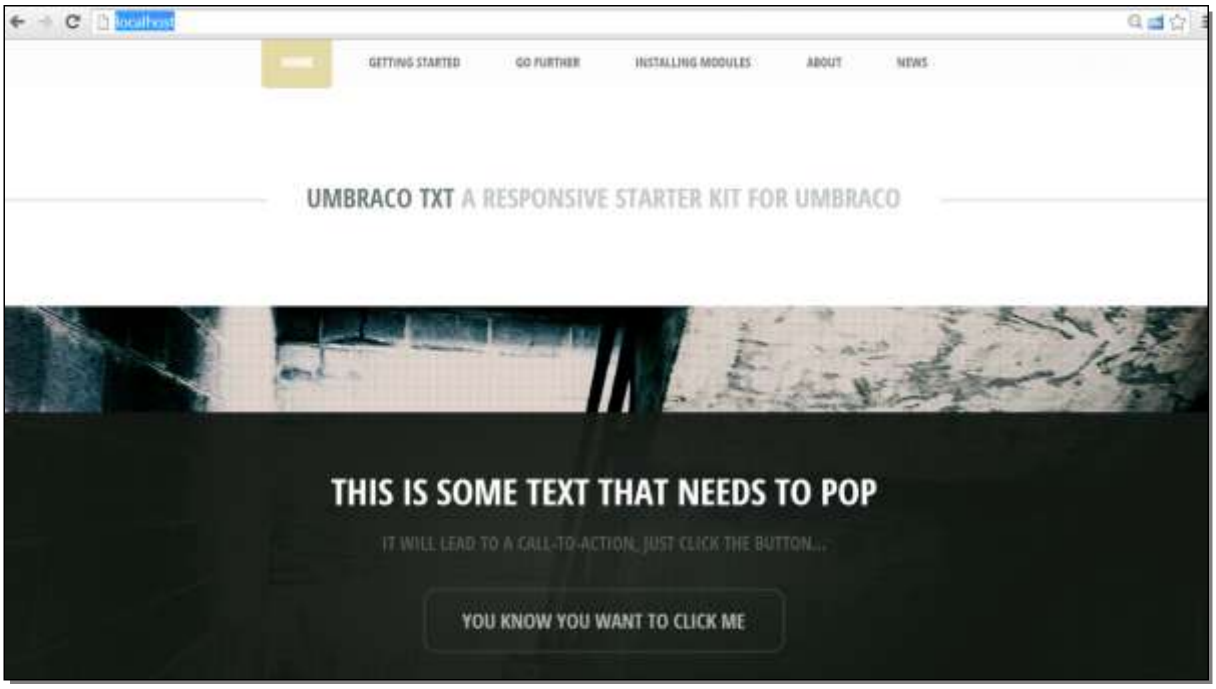
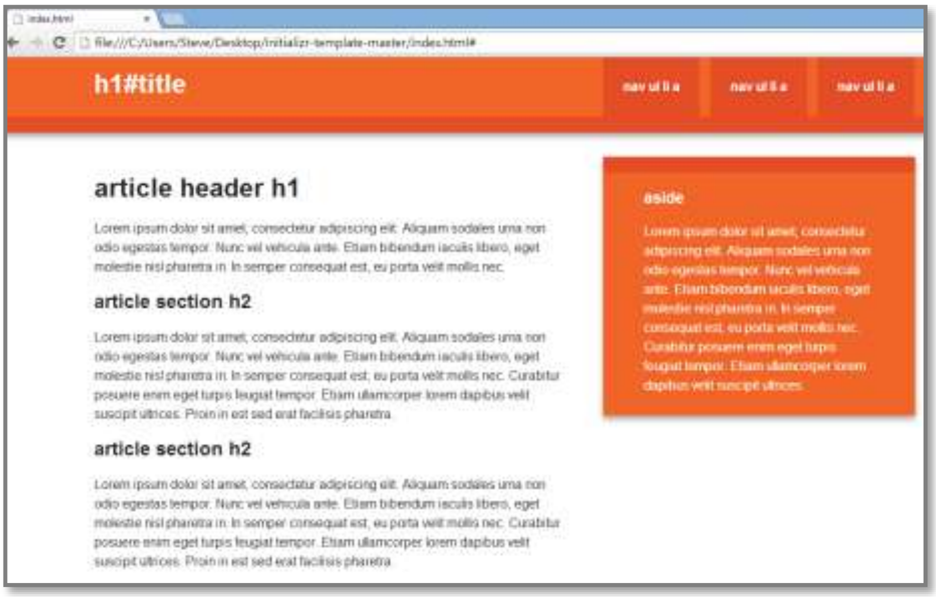

localhost
HOME
GETTING STARTED
GO FURTHER
INSTALLING MODULES
ABOUT
NEWS
UMBRACO TXT A RESPONSIVE STARTER KIT FOR UMBRACO
THIS IS SOME TEXT THAT NEEDS TO POP
IT WILL LEAD TO A CALL-TO-ACTION, JUST CLICK THE BUTTON...
YOU KNOW YOU WANT TO CLICK ME

Figure 4- You should NOT see this!

You need to reinstall Umbraco if you can see the starter kit – if you did a manual install you can delete all files in the directory where your local host is being served from, copy the Umbraco zip contents back in and then hit localhost.

3.3 Preparing the Initializr Template Site

Now unzip the Initializr contents to a folder onto your desktop (or a place of your choosing). Now open the ***index.html*** from this directory in your preferred browser to see the template – you can see it's full of lovely filler text with dummy links. We're going to turn this into a fully fledged, working site!


index.html
file:///C:/Users/Steve/Desktop/initializr-template-master/index.html#
h1#title
nav ul li a
nav ul li a
nav ul li a

article header h1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis nec.

article section h2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis nec. Curabitur posuere enim eget turpis feugiat tempor. Etiam ullamcorper lorem dapibus velit suscipit ultrices. Proin in est sed erat facilisis pharetra.

article section h2

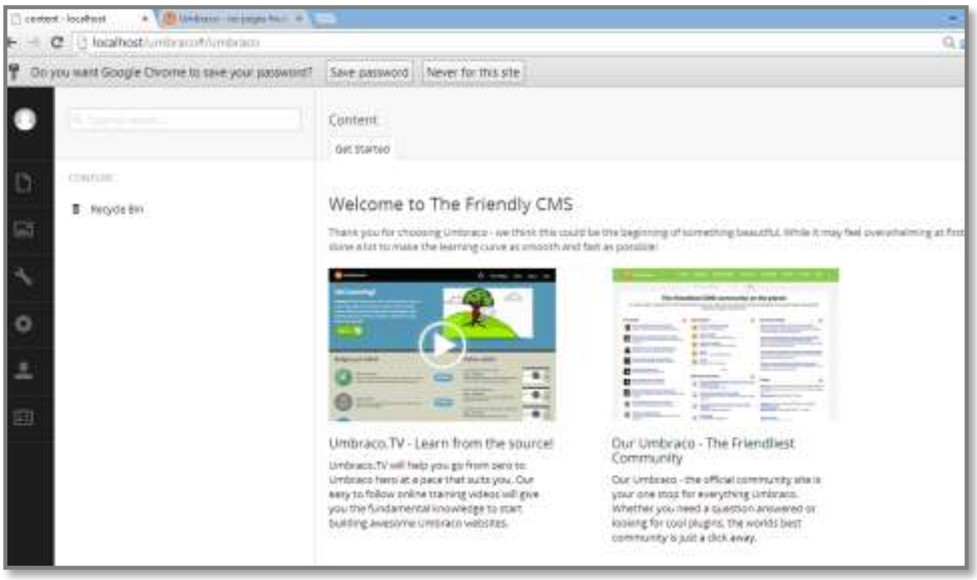
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis nec. Curabitur posuere enim eget turpis feugiat tempor. Etiam ullamcorper lorem dapibus velit suscipit ultrices. Proin in est sed erat facilisis pharetra.

aside

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis nec. Curabitur posuere enim eget turpis feugiat tempor. Etiam ullamcorper lorem dapibus velit suscipit ultrices.

Figure 5 - The Initializr Template

Log into your Umbraco installation (e.g. go to http://localhost/umbraco in your browser). You should be faced with an empty Umbraco installation – but where to start!?



Welcome to The Friendly CMS

Thank you for choosing Umbraco - we think this could be the beginning of something beautiful. While it may feel overwhelming at first [illegible] done a lot to make the learning curve as smooth and fast as possible.

Umbraco.TV - Learn from the source!

Umbraco.TV will help you go from zero to Umbraco hero at a pace that suits you. Our easy to follow online training videos will give you the fundamental knowledge to start building awesome Umbraco websites.

Our Umbraco - The Friendliest Community

Our Umbraco - the official community site is your one stop for everything Umbraco. Whether you need a question answered or looking for cool plugins, the worlds best community is just a click away.

Figure 6 - A barren, empty Umbraco installation

4 Document Types

4.1 Data first – nothing in = nothing out!

Step 1 of any site is to create a "***Document Type***" – after a few installations you'll speak this terminology but at the start it's a little bit bewildering. A ***Document Type*** is a data container in Umbraco where you can add data fields / attributes where the editor user can input data and Umbraco can use it to output it in the relevant part of a "***template***" (more on these later).

Document Types are infinitely extendable but usually you'll add data fields something like the following:

- Page title
- Sub Heading
- Body Text
- Meta Title
- Meta Description
- ...

Each ***Data Field*** has a type - e.g. a text string or a number or rich text body... we'll come to this later.

4.2 Creating your first Document Type

1. Right, let's get busy. Go to the ***Settings*** menu in Umbraco. This is the third button on the left hand black menu with the spanner. Then you'll see a long list of settings – don't worry about these yet, we'll introduce them as we need them.

2. ***Document Types*** is strangely positioned as the last option in the list yet it's always the starting point for any Umbraco build. Hover over the ***Document Types node*** and you'll see three dots ***...*** , click this to see the menu. Then click ***+ Create*** button.

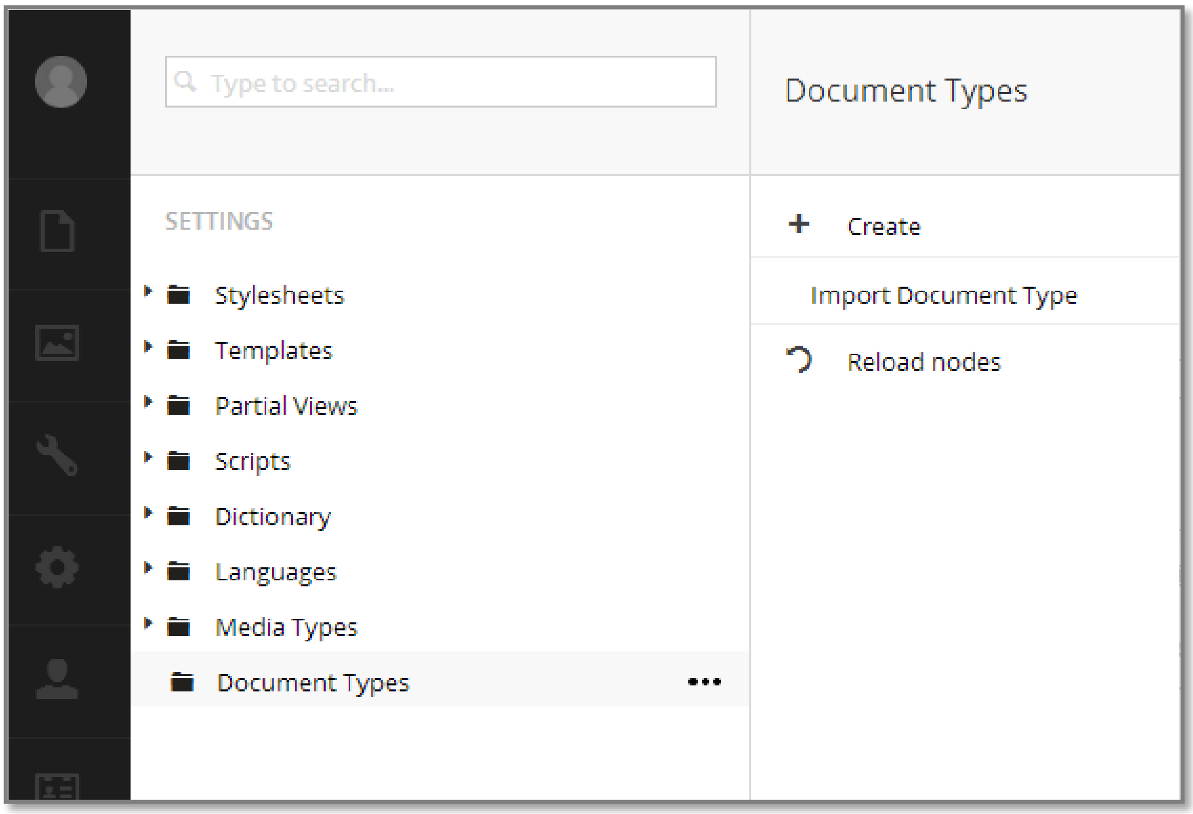


Figure 7 - Creating a Document Type

3. Ignore the ***Master Document Type*** drop down for now. Give our new ***Document Type*** the ***Name*** = "*HomePage"* and ensure the ***Create matching template*** checkbox is checked. Click ***Create***.

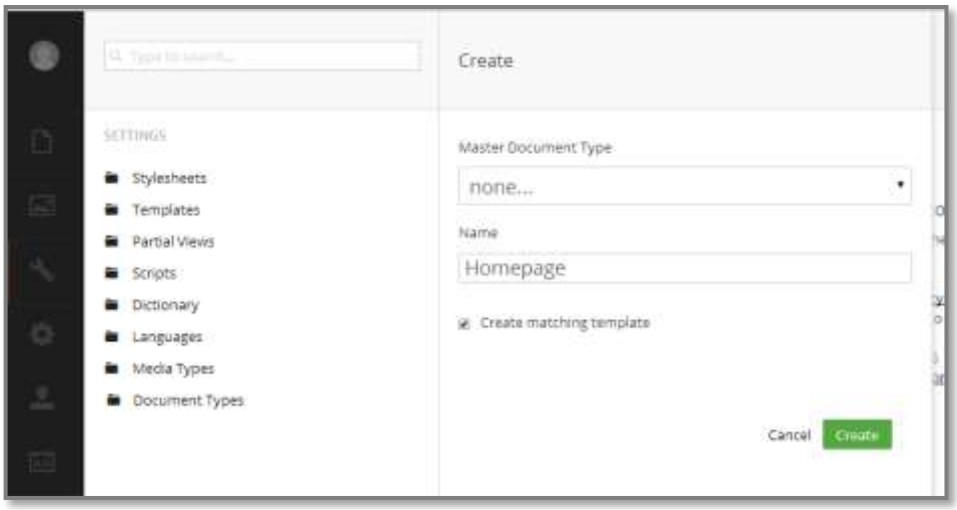


Figure 8 – Name your Document Type

4. Umbraco now adds a ***Document Type*** to the tree under the node. You'll see four tabs ***Info***, ***Structure***, ***Generic properties***, ***Tabs***. Click ***Info*** (should already be selected) and then the ***Choose...*** link next to the ***Icon*** label. Enter "*home*" into the search and you'll have a house icon – this will help our editors in the ***Content*** tree later.

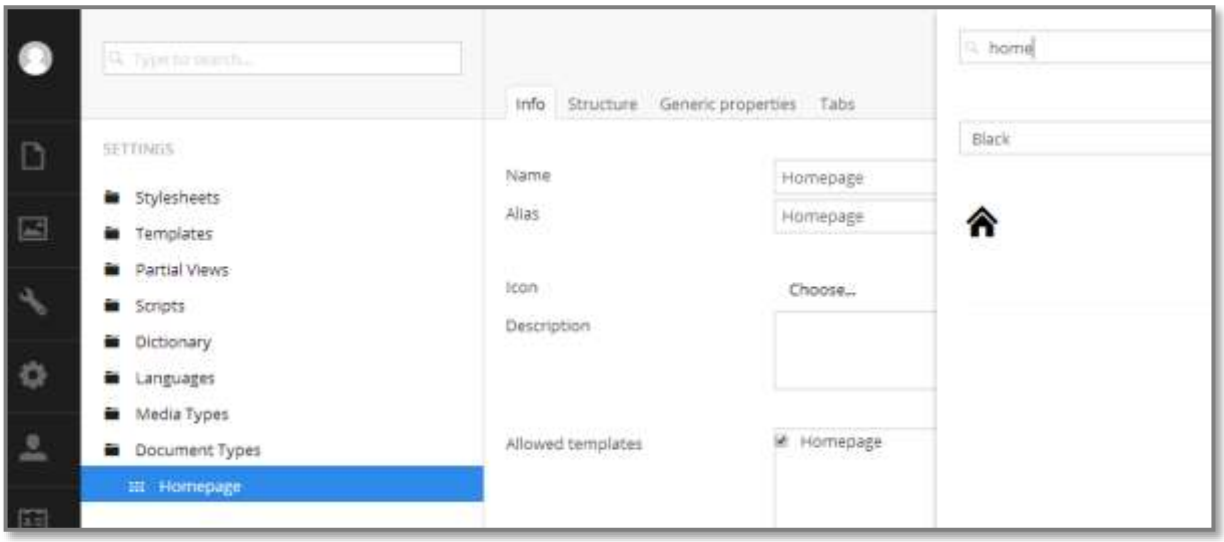
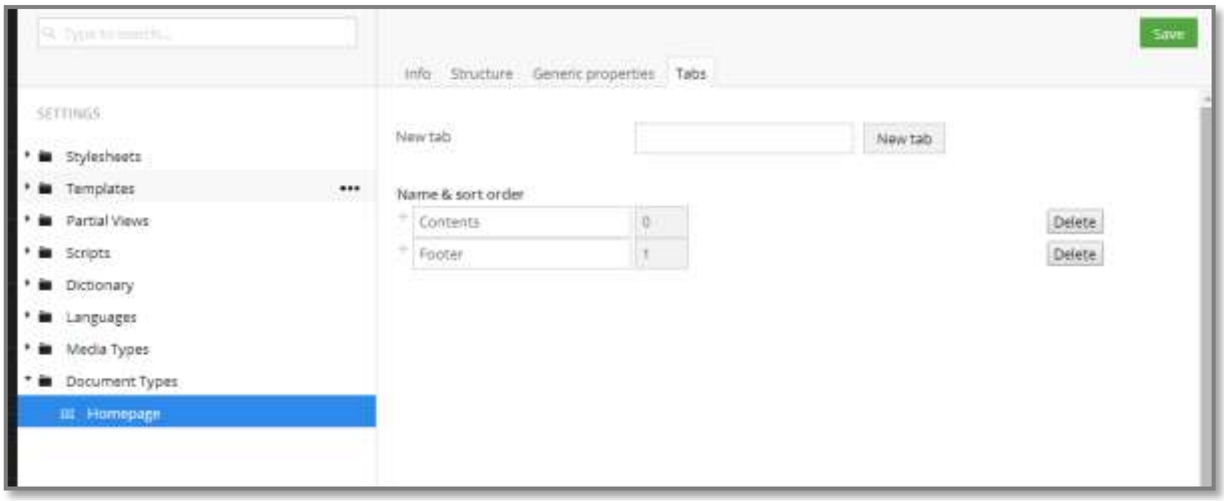


Figure 9 - Adding an Icon to Document Type

5. Enter in the ***Description*** field "*This is our homepage template*". This text is used to help the user select the correct document type later.

6. Next click the ***Structure*** tab and check ***Allow at root***. This will allow us to create a homepage at the root (simple huh?).

7. Next we go to the ***Tabs*** tab. Create a new tab called "*Contents*" and then another called "*Footer*" (enter the name and click the ***New tab*** button remembering to click ***Save*** after).

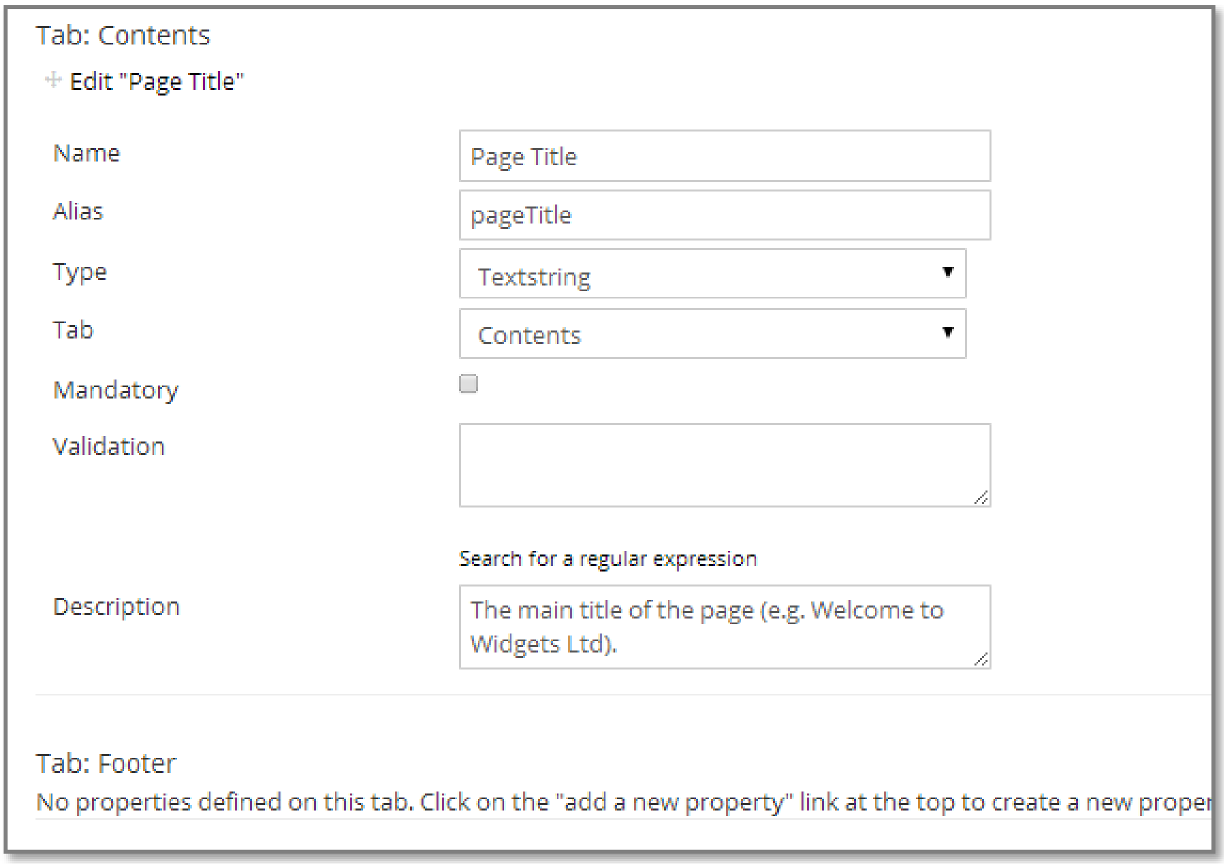


Name & sort order

Contents	0
Footer	1

Figure 10 - Document Types - Adding Tabs

8. Now go to the ***Generic properties*** – this tab is where we create the data containers that the homepage will need and use. Click on the link ***Click here to add a new property***. Enter the ***Name*** "*Page Title*". When you move to the next field you'll see Umbraco helpfully gives you the alias "pageTitle". The ***Type*** is defaulted to "*Textstring*" and ***Tab*** = "*Contents*" (remember, we just created that!). ***Description*** again helps the editor so we'll fill this in "*The main title of the page (e.g. Welcome to Widgets Ltd).* "



Tab: Contents

Edit "Page Title"

Name	Page Title
Alias	pageTitle
Type	Textstring
Tab	Contents
Mandatory	☐
Validation	
	Search for a regular expression
Description	The main title of the page (e.g. Welcome to Widgets Ltd).

Tab: Footer

No properties defined on this tab. Click on the "add a new property" link at the top to create a new proper

Figure 11 - Creating our pageTitle Data Type

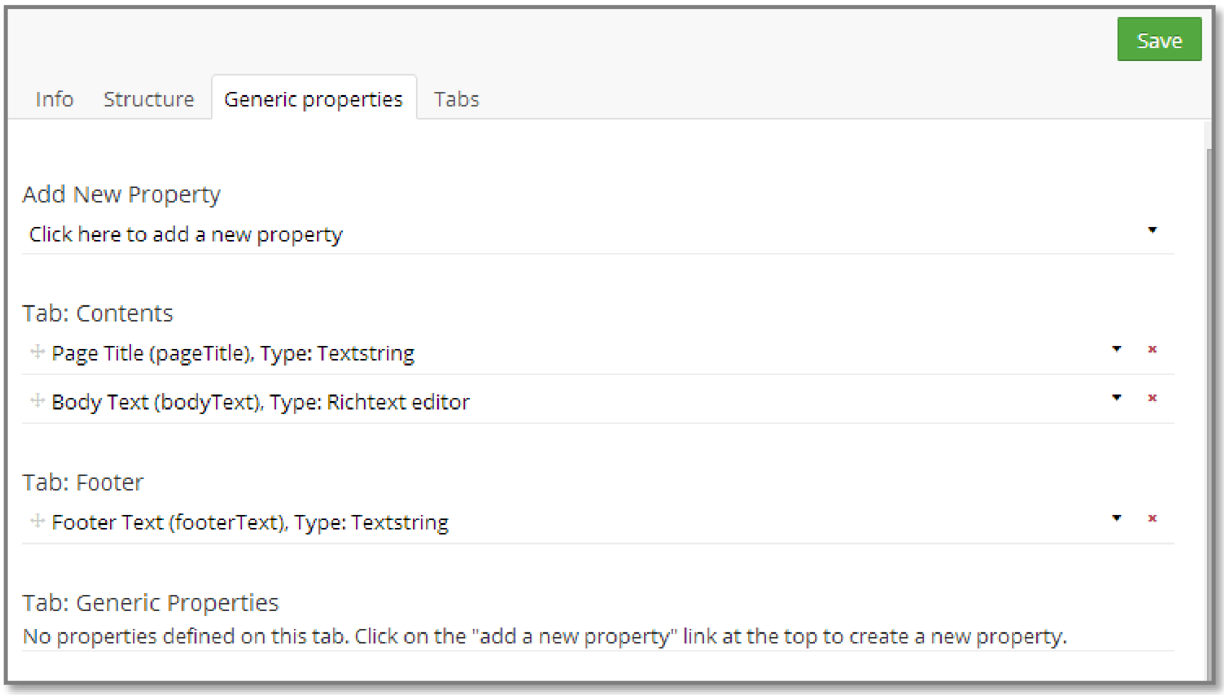
9. Ignore the rest of the fields for now and click the green ***Save*** button at the top right.

10. Repeat this step, clicking the ***Click here to add a new property link*** at the top of the ***Generic Properties tab*** and create these (remembering to click ***Save*** each time):

Name:	Body Text
Alias:	bodyText
Type:	Richtext editor *(click the arrow on the Type field!)*
Tab:	Contents
Description:	The main content of the page.

Name:	Footer Text
Alias:	footerText
Type:	Textstring
Tab:	Footer *(remember to change this!)*
Description:	Copyright notice for the footer.

11. You should now have a ***Generic Properties tab*** that looks like this:



Add New Property

Click here to add a new property

Tab: Contents

Page Title (pageTitle), Type: Textstring

Body Text (bodyText), Type: Richtext editor

Tab: Footer

Footer Text (footerText), Type: Textstring

Tab: Generic Properties

No properties defined on this tab. Click on the "add a new property" link at the top to create a new property.

Figure 12 - Generic Properties Tab of your Homepage Document Type

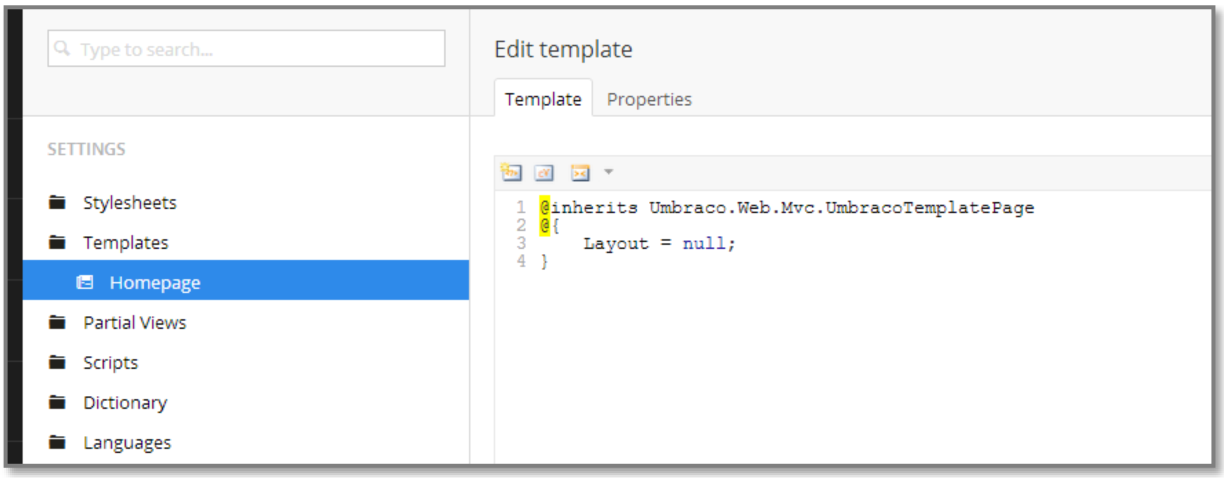
We've now created our first Document Type – Umbraco needs three things to create a webpage and this is the first and most important. It takes the data inside an instance of the Document Type and merges it with a template – we'll edit our template next.

5 Creating (Editing) Your First Template

1. Next click the expand node icon (it's the small triangle) ▸ behind the ***Settings > Templates folder*** – you should then see a child node titled "*Homepage*" – we created this automatically when we created the ***Document Type*** (remember that checkbox?).

NOTE – in early versions of 7.1 the tree doesn't automatically refresh to show you this, if it's missing try a hard refresh of Umbraco (Ctrl+F5) – should be fixed soon.

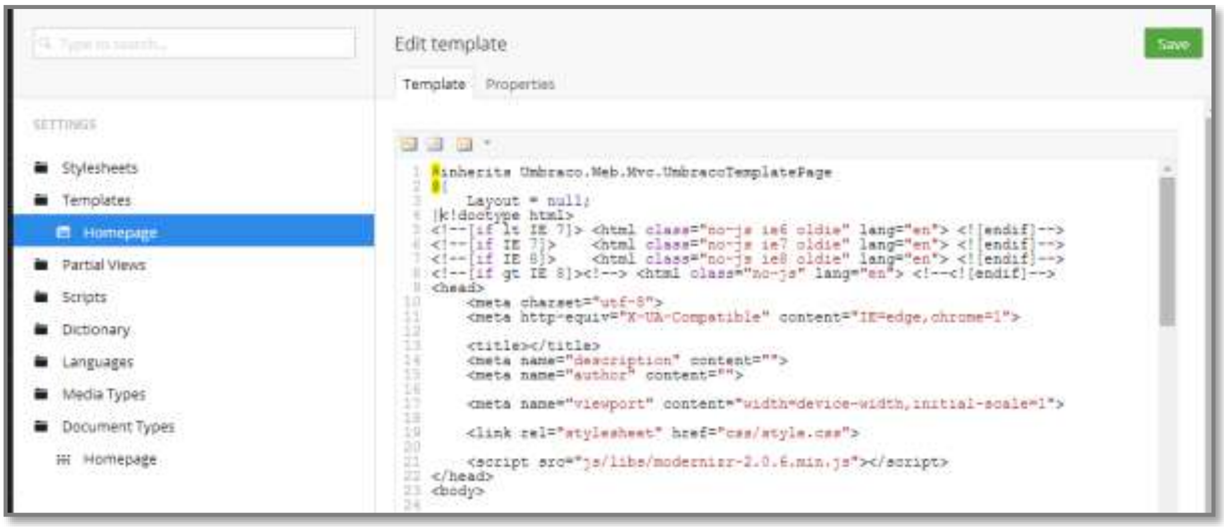
Clicking on the ***Homepage node*** will load the template – which, except for a little bit of Razor code, is empty!



```
@inherits Umbraco.Web.Mvc.UmbracoTemplatePage
@{
    Layout = null;
}
```

Figure 13 - Empty Homepage Template

2. Leaving the code that's there (if you don't understand it, don't worry!) let's copy our template code in. Open up ***index.html*** from the ***Initializr*** template in your favourite text editor (Notepad++ is good). Copy and paste the whole thing into this template **after** the closing curly brace "}". Your template should now look like below:



```
@inherits Umbraco.Web.Mvc.UmbracoTemplatePage
@{
    Layout = null;
<!doctype html>
<!--[if lt IE 7]> <html class="no-js ie6 oldie" lang="en"> <![endif]-->
<!--[if IE 7]>    <html class="no-js ie7 oldie" lang="en"> <![endif]-->
<!--[if IE 8]>    <html class="no-js ie8 oldie" lang="en"> <![endif]-->
<!--[if gt IE 8]><!--> <html class="no-js" lang="en"> <!--<![endif]-->
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

    <title></title>
    <meta name="description" content="">
    <meta name="author" content="">

    <meta name="viewport" content="width=device-width,initial-scale=1">

    <link rel="stylesheet" href="css/style.css">

    <script src="js/libs/modernizr-2.0.6.min.js"></script>
</head>
<body>
```

Figure 14 - Homepage template

3. Click the ***Save*** button.

We now have a template. That's two out of the three stages complete for our first page.

6 Creating Your First Content Node

Our third and final stage to creating our first page in Umbraco is to create a content node where a content editor can add the content and then Umbraco will use this along with the Document Type and Template to serve up an HTML page to web visitors.

1. We're now ready to create our first page in Umbraco! Click the ***Content*** button (first option in the left hand menu).

2. Hover over the grey text ***CONTENT*** and you'll see the three dots ***...*** – click this. If you've done everything correctly so far you should see the option to create our Homepage!

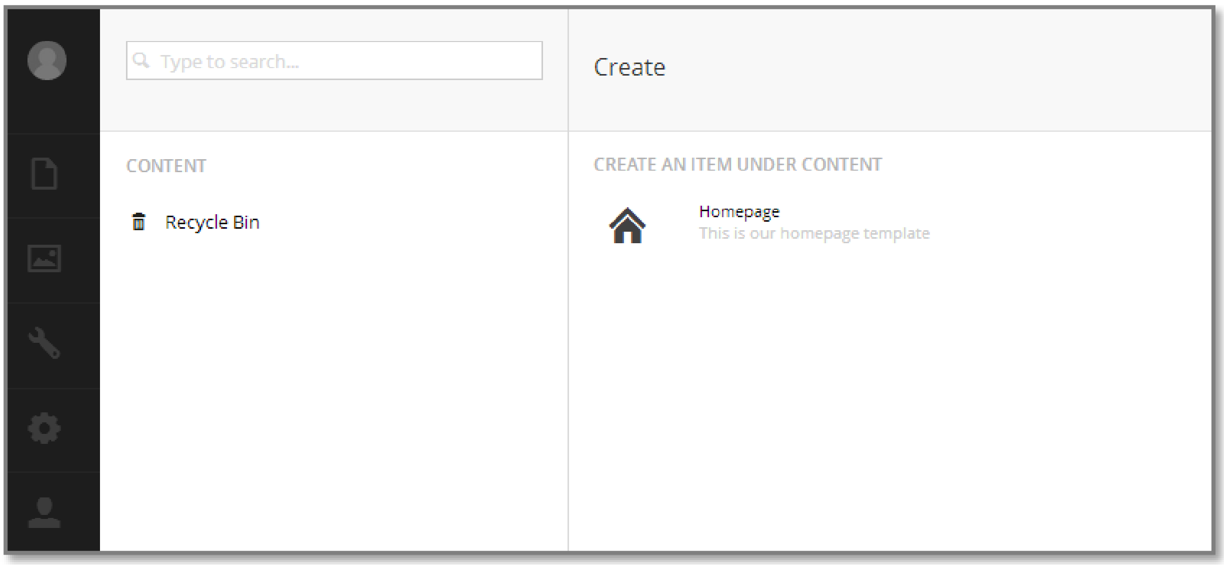


Figure 15 – Create a Homepage

If you can't see this then don't panic – check that ***Settings> Document Types > Homepage node*** > ***Structure tab*** > ***Allow at root*** is checked.

3. Let's create our homepage. Click the icon and you'll see what we've just been setting up – our document type is now going to drive our homepage content – it gives us and the editors the fields they need.

4. In red at the top you'll see "*Enter a name...*" click this and enter a name (on some browsers this is hard to see as a field – an Umbraco bug has been raised for this). We're going to call this "*Homepage*".

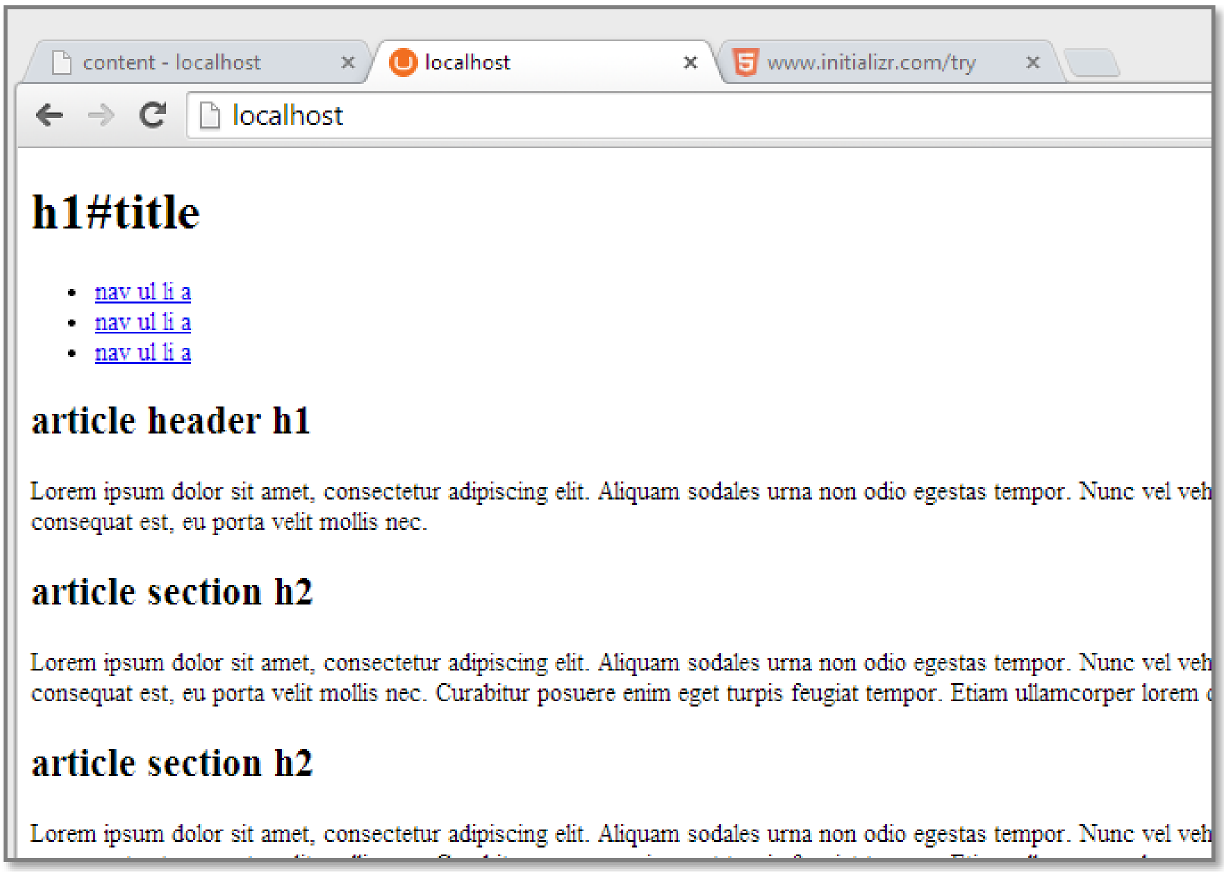
5. Fill in the following on the ***Contents*** tab:

Page Title	Welcome to Widgets Ltd
Body Text	Hello world! We can write what we like here!
Widgets Ltd 2014"	

Click the Footer tab and enter:

Footer Text	"Copyright Widgets Ltd 2014"

6. Now click the green ***Save and publish*** button. The menu will reload with our homepage node under the ***CONTENT*** label. And here's the good bit... go and refresh your webpage in your browser http://localhost – the default Umbraco page will be gone and we'll see a very bare, unstyled page! We're getting there!



h1#title

- nav ul li a
- nav ul li a
- nav ul li a

article header h1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel veh
consequat est, eu porta velit mollis nec.

article section h2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel veh
consequat est, eu porta velit mollis nec. Curabitur posuere enim eget turpis feugiat tempor. Etiam ullamcorper lorem

article section h2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel veh

Figure 16 – An Unstyled Homepage

6.2 CSS and Javascript

1. Looking at our homepage we're obviously missing the CSS and JS from the Initializr template. To include this navigate to your Umbraco directory in Windows Explorer and copy over the ***css|style.css*** file into the ***Umbraco|Css*** folder (replace Umbraco with wherever your Umbraco instance is being served from – e.g. "*C:|inetpub|wwwroot*". Now refresh your webpage in your browser and you'll see a more styled page.

NOTE – you could use the Umbraco UI to create your CSS file. ***Settings > Stylesheets*** > ***... > + Create*** and create a stylesheet called style (don't add the file suffix .css) and paste the CSS in I find it easier to copy the CSS. Using either method should be noted does NOT include them in your HTML markup automatically – Umbraco produces clean output and this means you only wire up what you want and need.

2. Next copy the ***scripts*** folder from the **js** directory of the initializr template to the ***Umbraco|Scripts*** directory – we'll have to update the template to look in **\Scripts** instead of **\js**. To do this go to ***Settings > Templates > Homepage*** and change line 21 to say "*scripts/...*" and click ***Save***.

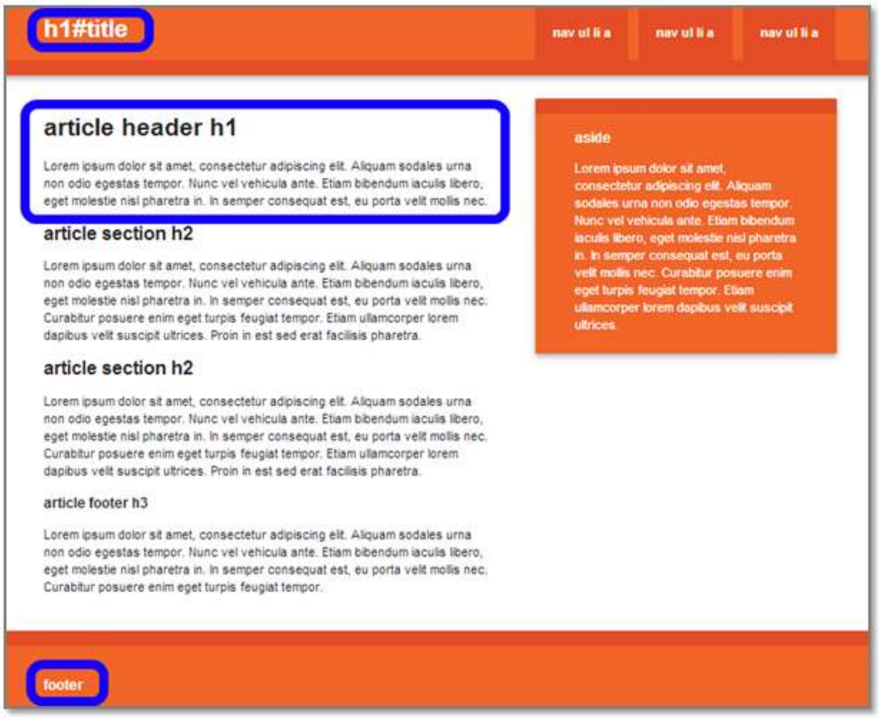
```
<script src="scripts/libs/modernizr-2.0.6.min.js"></script>
```

NOTE – you can also use the UI interface to create your JS files too ***Settings > Scripts > ... > + Create*** (again don't add the suffix but select .js from the ***Type*** dropdown) the reference in your template should be "*scripts/myfile.js*".

3. Now in dev tools when looking at the http://localhost page you should find that the network tab doesn't report any missing assets - if it does have a look and fix any typos / check the files are in the right places!

6.3 Outputting the Umbraco Data Fields

What you'll notice though is that our content we added to the homepage isn't being output. We need to wire up the data properties to the template. Let's look at our template and identify where the data fields we created before should go.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis nec.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis nec. Curabitur posuere enim eget turpis feugiat tempor. Etiam ullamcorper lorem dapibus velit suscipit ultrices. Proin in est sed erat facilisis pharetra.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis nec. Curabitur posuere enim eget turpis feugiat tempor. Etiam ullamcorper lorem dapibus velit suscipit ultrices. Proin in est sed erat facilisis pharetra.

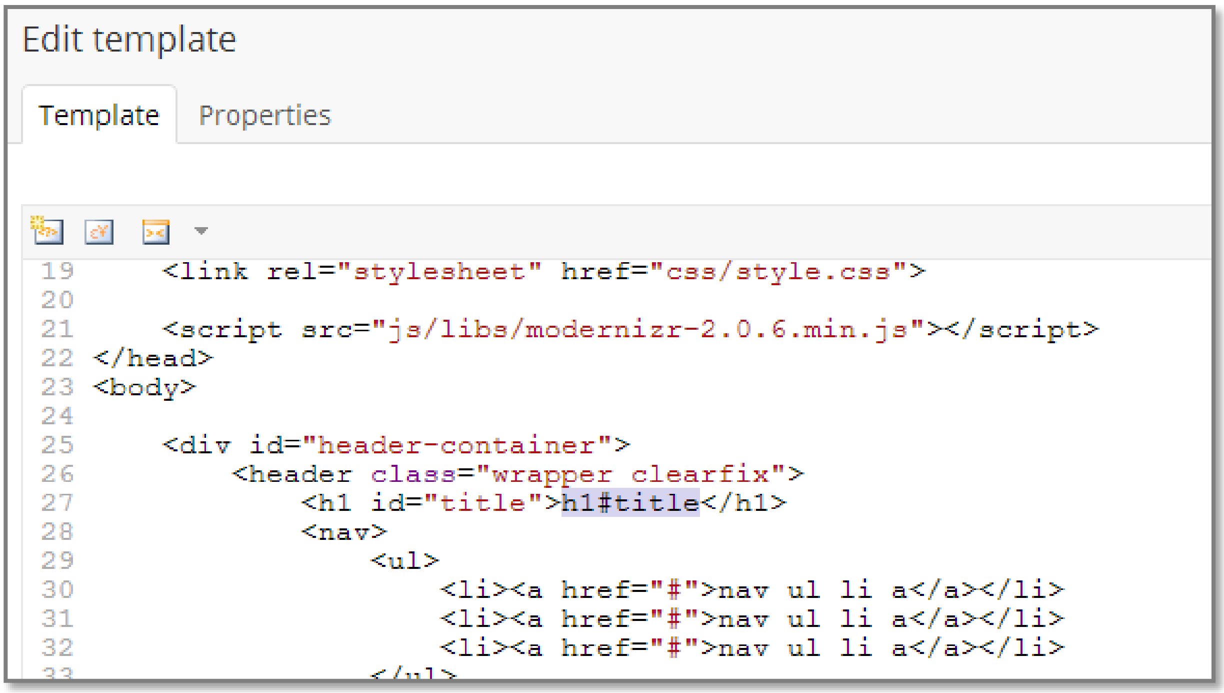
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis nec. Curabitur posuere enim eget turpis feugiat tempor.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis nec. Curabitur posuere enim eget turpis feugiat tempor. Etiam ullamcorper lorem dapibus velit suscipit ultrices.

Figure 17 - Where our Data Fields Content Should be Output

We've marked in blue where we want our data field content to be output. Now we need to wire up the relevant fields.

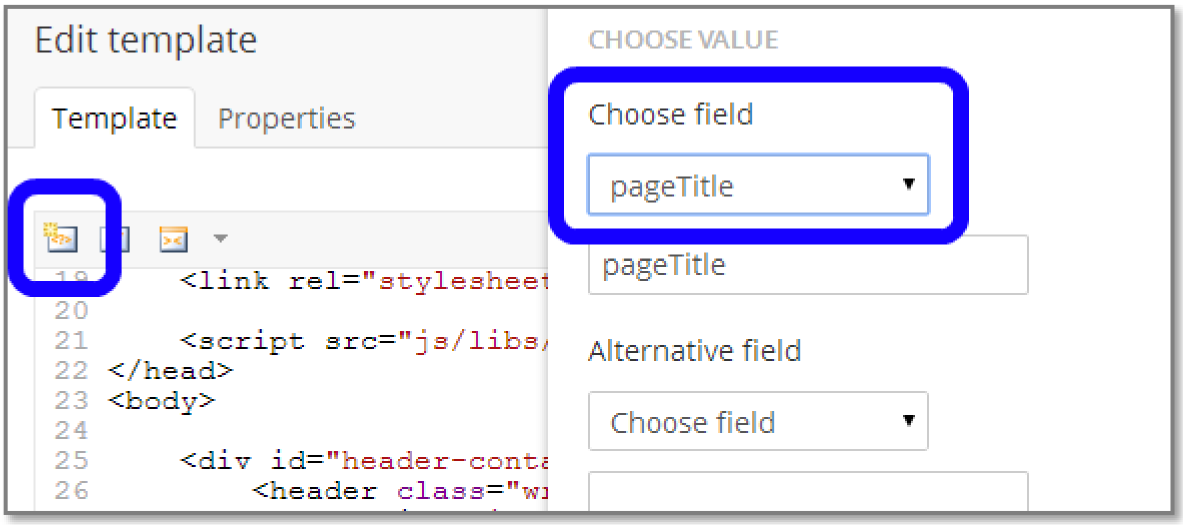
1. Go to the ***Settings > Templates > Homepage***. Scroll down and highlight the text "h1#title" around line 27.



```
    <link rel="stylesheet" href="css/style.css">

    <script src="js/libs/modernizr-2.0.6.min.js"></script>
</head>
<body>

    <div id="header-container">
        <header class="wrapper clearfix">
            <h1 id="title">h1#title</h1>
            <nav>
                <ul>
                    <li><a href="#">nav ul li a</a></li>
                    <li><a href="#">nav ul li a</a></li>
                    <li><a href="#">nav ul li a</a></li>
```

Figure 18 - Preparing to replace the hardcoded text with an Umbraco Page Field

2. Click the button ***Insert Umbraco Page Field*** and under the ***Choose field*** drop down select ***pageTitle*** from the ***Custom Fields*** section.



Choose field: pageTitle

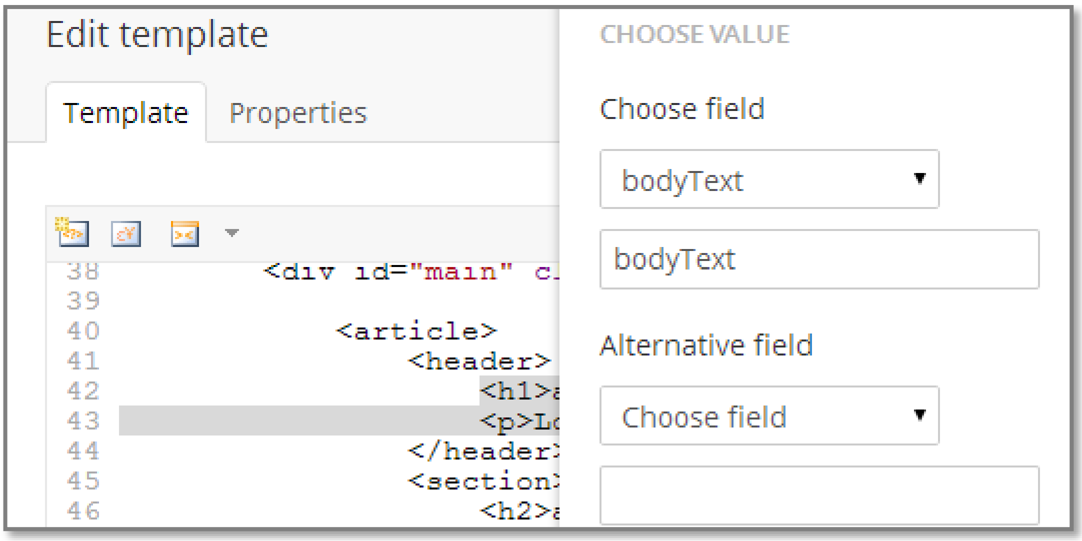
pageTitle

Alternative field: Choose field

Figure 19 - Umbraco Page Field

3. Click the green ***Insert*** button then the ***Save*** button.

4. Next do the same for the content between the "*<header></header>*" tags (around lines 42 -43) using field ***bodyText***. Again click the ***Insert*** and then ***Save*** buttons.


Edit template

Template | Properties

```
        <div id="main" c
        
            <article>
                <header>
                    <h1>a
                    <p>Lo
                </header>
                <section>
                    <h2>a
```

CHOOSE VALUE

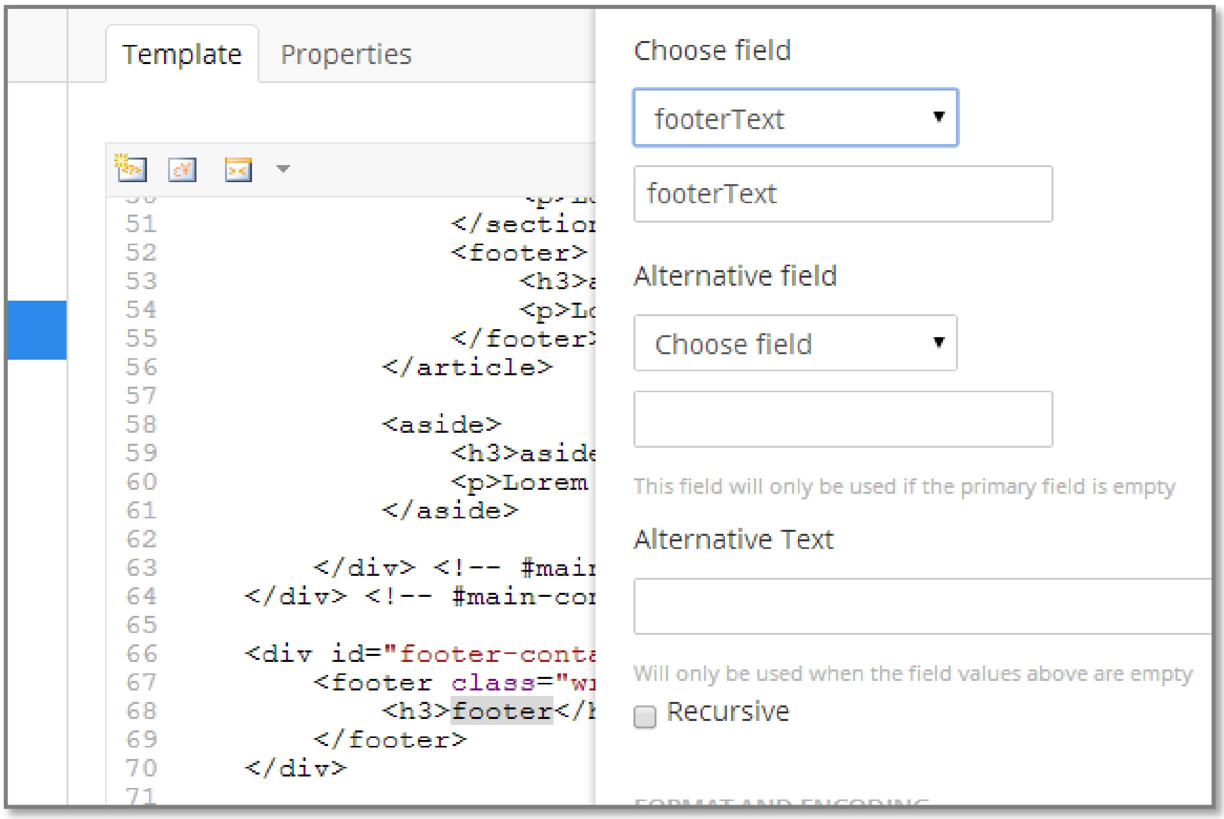
Choose field: bodyText

bodyText

Alternative field: Choose field

Figure 20 - Replacing the bodyText with the Umbraco Page Field

5. Finally we do the footer – between the <h3></h3> tags in the footer div (line 68).


Template | Properties

```
51                  </section
52                  <footer>
53                      <h3>a
54                      <p>Lo
55                  </footer>
56              </article>
57              
58              <aside>
59                  <h3>aside
60                  <p>Lorem
61              </aside>
62              
63          </div> <!-- #mair
64      </div> <!-- #main-cor
65      
66      <div id="footer-conta
67          <footer class="wr
68              <h3>footer</h
69          </footer>
70      </div>
```

Choose field: footerText

footerText

Alternative field: Choose field

This field will only be used if the primary field is empty

Alternative Text

Will only be used when the field values above are empty

☐ Recursive

Figure 21 - Footer Text

6. Now go and reload your homepage... viola! We have content!   Now, we could go back and add two tabs called Article1, Article 2, Article Footer each containing a title and content field and wire these to the relevant places in the template. However this would limit the content manager to always having to have these sections. This might be OK but we could also use child nodes – we'll learn about those later.

7 Creating More Pages

7.1 Using a Maintainable Template Structure

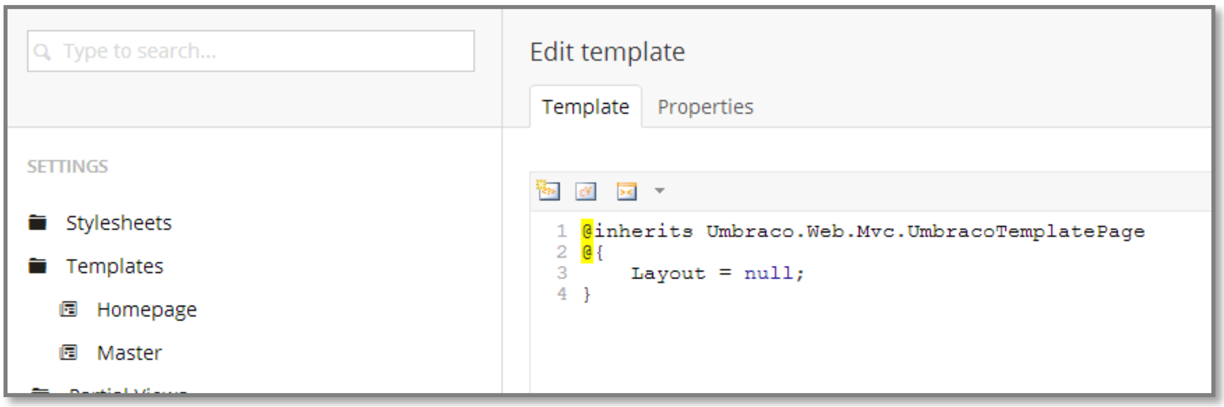
We've seen how to create a ***Document Type***. We have a simple three page site, Home, News and Contact Us. We could easily just create three ***Document Types*** and leave ***Create matching template*** checkbox checked to also create three matching templates. Then we'd copy the same HTML code, job-lot into each template. This would work – on a very simple site it actually has some merits however once a site starts to grow this would lead to problems – for instance changing anything in the menu needs to be done on each template - it also means we'd need the user to set the footer on each page etc.

Umbraco provides us with an elegant solution to keeping a consistent base template – those familiar with MVC will recognise it.

To start we're going to unpick a little bit of what we did in creating the homepage to sit the homepage template under a master.

7.2 Create a Master Template

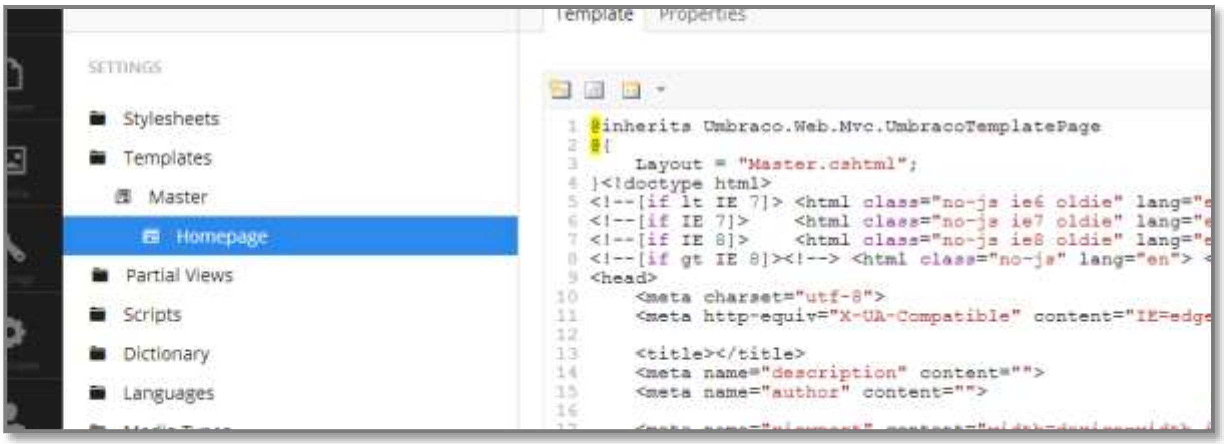
1. Go to the ***Settings > Templates*** and open up the tree. At the moment we just have our ***Homepage*** template. Hover over the ***Templates*** menu and click the menu ***...*** button. Create a new template called Master, click ***+ Create*** and then give it the name "*Master*". Remember to click ***Save***.



```
@inherits Umbraco.Web.Mvc.UmbracoTemplatePage
@{
    Layout = null;
}
```

Figure 22 - Master Template

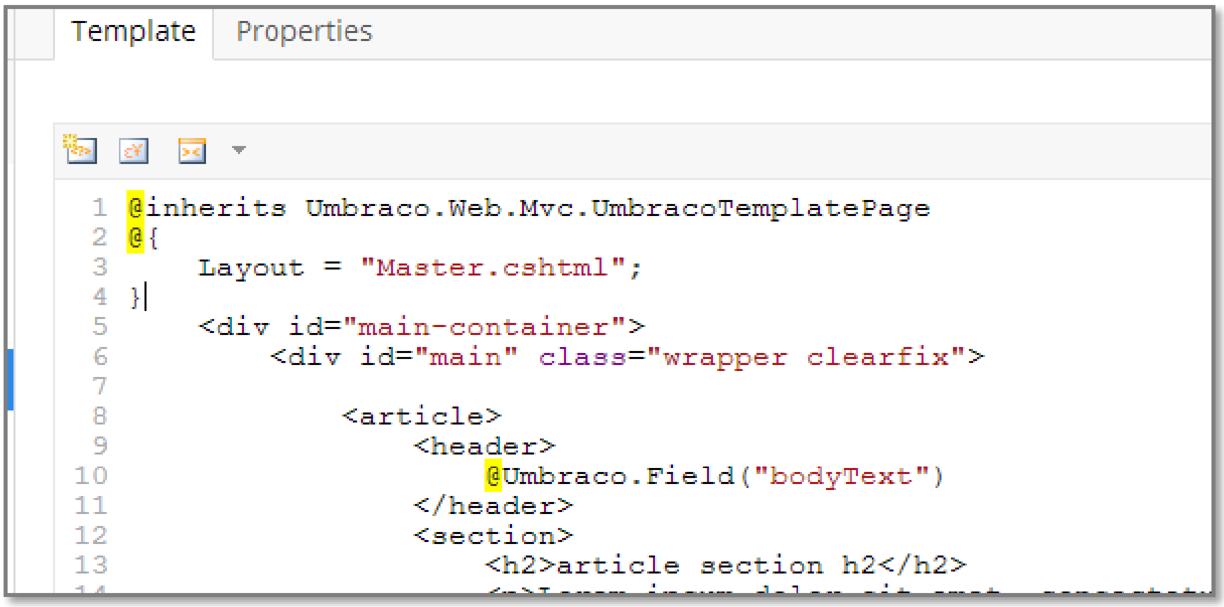
2. Now we're going to move the ***Homepage*** template under the ***Master*** template. To do this select the ***Settings > Homepage node*** and from the ***Properties tab >Master template drop down*** select "*Master*" and then click ***Save***. This will update the Razor code section to change "*Layout = null*"; to "*Layout = "Master.cshtml""* (you may have to click off the ***Homepage node*** and back on to see this update – a bug that will be fixed in a future release of Umbraco).



```
@inherits Umbraco.Web.Mvc.UmbracoTemplatePage
@{
    Layout = "Master.cshtml";
}<!doctype html>
<!--[if lt IE 7]> <html class="no-js ie6 oldie" lang="
<!--[if IE 7]>    <html class="no-js ie7 oldie" lang="
<!--[if IE 8]>    <html class="no-js ie8 oldie" lang="
<!--[if gt IE 8]><!--> <html class="no-js" lang="en"> <
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge

    <title></title>
    <meta name="description" content="">
    <meta name="author" content="">

```

Figure 23 - Homepage Template now sits under the Master

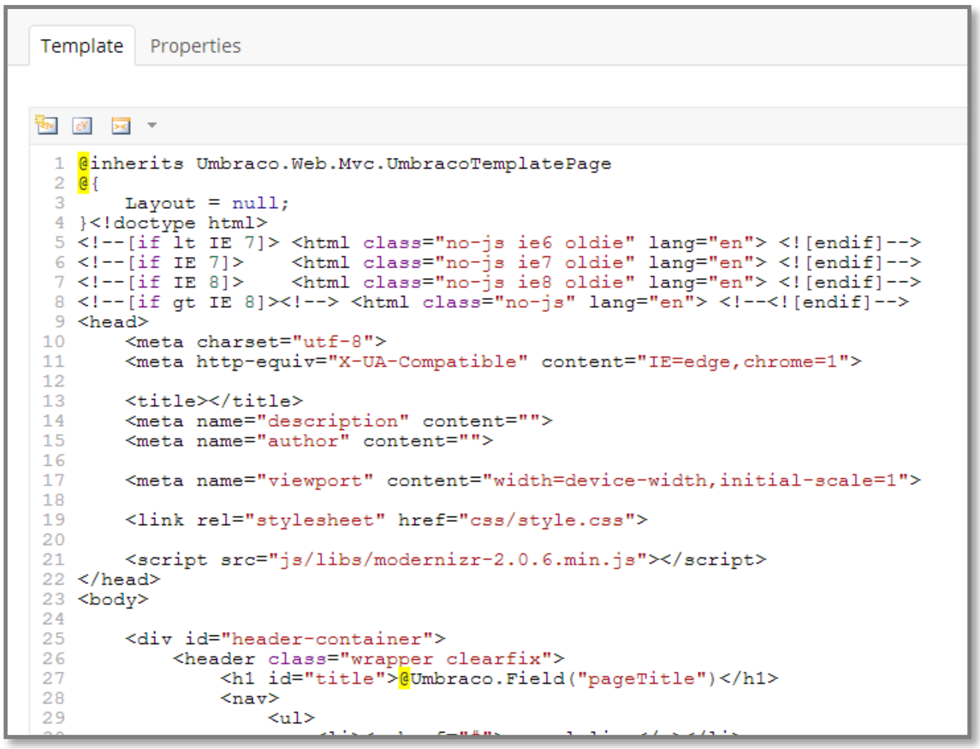
3. Now we need to move the parts of our HTML template that are common across all pages into the ***Master***. This is where as a developer you might need to use your brain as it will be slightly different for different websites – e.g. do all pages have a <div id="main"> as so can we put this in the master or does this belong to only certain pages? For this site we'll assume this is part of the child page. Cut everything from the closing curly brace to line 37 *<div id="main-container">* - we're going to move the header and nav of the site to the master template. Cut this and click ***Save***.



```
@inherits Umbraco.Web.Mvc.UmbracoTemplatePage
@{
    Layout = "Master.cshtml";
}
    <div id="main-container">
        <div id="main" class="wrapper clearfix">

            <article>
                <header>
                    @Umbraco.Field("bodyText")
                </header>
                <section>
                    <h2>article section h2</h2>
```

Figure 24 - Homepage Template After Cutting the Header

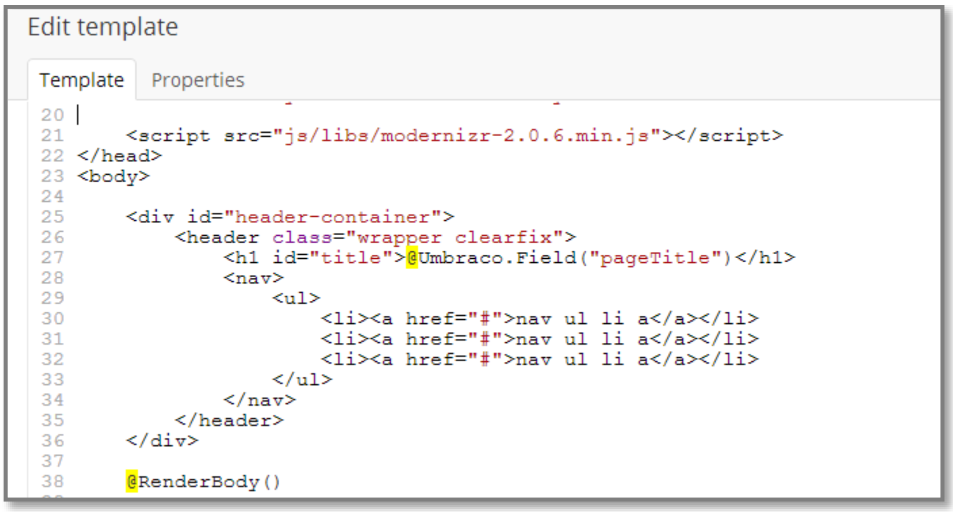
4. Now click on your ***Master*** template and paste this HTML markup after the closing curly brace and remember to click ***Save***.



```
 1 @inherits Umbraco.Web.Mvc.UmbracoTemplatePage
 2 @{
 3     Layout = null;
 4 }<!doctype html>
 5 <!--[if lt IE 7]> <html class="no-js ie6 oldie" lang="en"> <![endif]-->
 6 <!--[if IE 7]>    <html class="no-js ie7 oldie" lang="en"> <![endif]-->
 7 <!--[if IE 8]>    <html class="no-js ie8 oldie" lang="en"> <![endif]-->
 8 <!--[if gt IE 8]><!--> <html class="no-js" lang="en"> <!--<![endif]-->
 9 <head>
10     <meta charset="utf-8">
11     <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
12
13     <title></title>
14     <meta name="description" content="">
15     <meta name="author" content="">
16
17     <meta name="viewport" content="width=device-width,initial-scale=1">
18
19     <link rel="stylesheet" href="css/style.css">
20
21     <script src="js/libs/modernizr-2.0.6.min.js"></script>
22 </head>
23 <body>
24
25     <div id="header-container">
26         <header class="wrapper clearfix">
27             <h1 id="title">@Umbraco.Field("pageTitle")</h1>
28             <nav>
29                 <ul>
```

Figure 25 - Master Template after Pasting the Header

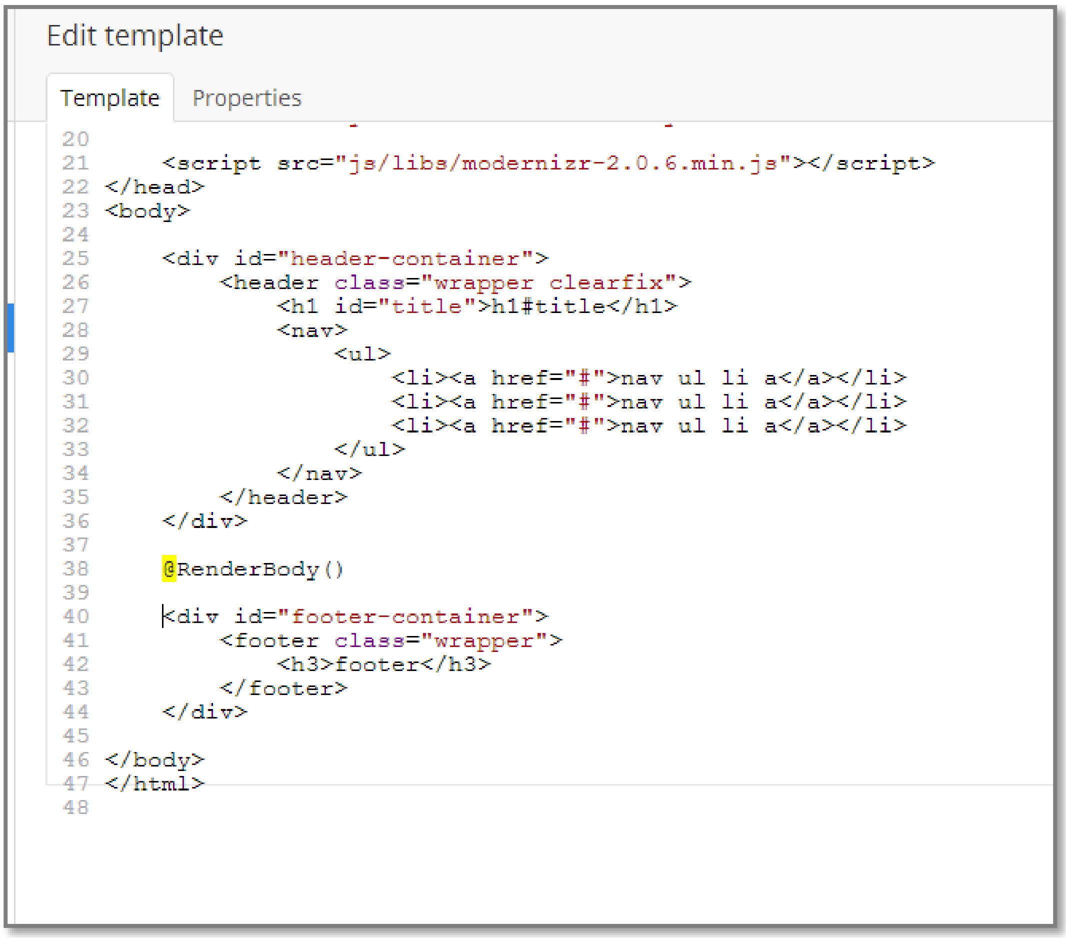
5. At the end of this markup we need to tell Umbraco to insert the child template's content – this is done by adding the code ***@RenderBody()*** at the end (around line 37). Click ***Save***.



```
20 |
21     <script src="js/libs/modernizr-2.0.6.min.js"></script>
22 </head>
23 <body>
24
25     <div id="header-container">
26         <header class="wrapper clearfix">
27             <h1 id="title">@Umbraco.Field("pageTitle")</h1>
28             <nav>
29                 <ul>
30                     <li><a href="#">nav ul li a</a></li>
31                     <li><a href="#">nav ul li a</a></li>
32                     <li><a href="#">nav ul li a</a></li>
33                 </ul>
34             </nav>
35         </header>
36     </div>
37
38     @RenderBody()
```

Figure 26 - Adding RenderBody() to the Master Template

6. Now we'll do the same with the footer content. Cut everything from the opening of the *footer-container* div (approximately line 33) from the ***Settings > Templates > Homepage > template tab***, click ***Save*** and then paste this into the ***Master*** template under the ***@RenderBody*** field we've just added. Remember to click ***Save***.


Edit template

Template Properties

```
    <script src="js/libs/modernizr-2.0.6.min.js"></script>
</head>
<body>

    <div id="header-container">
        <header class="wrapper clearfix">
            <h1 id="title">h1#title</h1>
            <nav>
                <ul>
                    <li><a href="#">nav ul li a</a></li>
                    <li><a href="#">nav ul li a</a></li>
                    <li><a href="#">nav ul li a</a></li>
                </ul>
            </nav>
        </header>
    </div>

    @RenderBody()

    <div id="footer-container">
        <footer class="wrapper">
            <h3>footer</h3>
        </footer>
    </div>

</body>
</html>
```

Figure 27 – Completed Master Template

7. Now we've done a lot of work – and what we should see if we refresh our localhost page is nothing has changed! If you have a compilation error you've perhaps mistyped ***@RenderBody()***. If you're missing any content (header or footer) check that what you have in the templates matches the following:

```
@inherits Umbraco.Web.Mvc.UmbracoTemplatePage
@{
    Layout = null;
}<!doctype html>
<!--[if lt IE 7]> <html class="no-js ie6 oldie" lang="en"> <![endif]-->
<!--[if IE 7]>    <html class="no-js ie7 oldie" lang="en"> <![endif]-->
<!--[if IE 8]>    <html class="no-js ie8 oldie" lang="en"> <![endif]-->
<!--[if gt IE 8]><!--> <html class="no-js" lang="en"> <!--<![endif]-->
<head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

        <title></title>
        <meta name="description" content="">
        <meta name="author" content="">

        <meta name="viewport" content="width=device-width,initial-scale=1">

        <link rel="stylesheet" href="css/style.css">

        <script src="js/libs/modernizr-2.0.6.min.js"></script>
</head>
<body>

        <div id="header-container">
                <header class="wrapper clearfix">
                        <h1 id="title">@Umbraco.Field("pageTitle")</h1>
                        <nav>
                                <ul>
                                        <li><a href="#">nav ul li a</a></li>
                                        <li><a href="#">nav ul li a</a></li>
                                        <li><a href="#">nav ul li a</a></li>
                                </ul>
                        </nav>
                </header>
        </div>

        @RenderBody()

        <div id="footer-container">
                <footer class="wrapper">
                        <h3>@Umbraco.Field("footerText")</h3>
                </footer>
        </div>

</body>
</html>
```

Figure 28 - Complete Master Template

```
@inherits Umbraco.Web.Mvc.UmbracoTemplatePage
@{
    Layout = "Master.cshtml";
}
        <div id="main-container">
                <div id="main" class="wrapper clearfix">

                        <article>
                                <header>
                                        @Umbraco.Field("bodyText")
                                </header>
                                <section>
                                        <h2>article section h2</h2>
                                        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum
iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis
nec. Curabitur posuere enim eget turpis feugiat tempor. Etiam ullamcorper lorem dapibus velit
suscipit ultrices. Proin in est sed erat facilisis pharetra.</p>
                                </section>
                                <section>
                                        <h2>article section h2</h2>
                                        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum
iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis
nec. Curabitur posuere enim eget turpis feugiat tempor. Etiam ullamcorper lorem dapibus velit
suscipit ultrices. Proin in est sed erat facilisis pharetra.</p>
                                </section>
                                <footer>
                                        <h3>article footer h3</h3>
                                        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum
iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis
nec. Curabitur posuere enim eget turpis feugiat tempor.</p>
                                </footer>
                        </article>

                        <aside>
                                <h3>aside</h3>
                                <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum iaculis
libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis nec.
Curabitur posuere enim eget turpis feugiat tempor. Etiam ullamcorper lorem dapibus velit
suscipit ultrices.</p>
                        </aside>

                </div> <!-- #main -->
        </div> <!-- #main-container -->
```

Figure 29 - Complete Homepage Template

If you're new to these concepts then I don't think what we've just done is going to make much sense until we make our next page.

7.3 Creating More Pages Using the Master – Contact Us

1. We're now going to make a simple page where we'll just put our contact details. For added functionality you might want to look at replacing this with a contact us form – see

http://our.umbraco.org/projects/tag/contact%20form . For now let's create a simple page – a page where the user can provide a title and some rich text. This is very similar to our homepage document type at the moment but we're assuming that you'll go and develop this into something very specific (e.g. adding the featured article and other article content blocks).

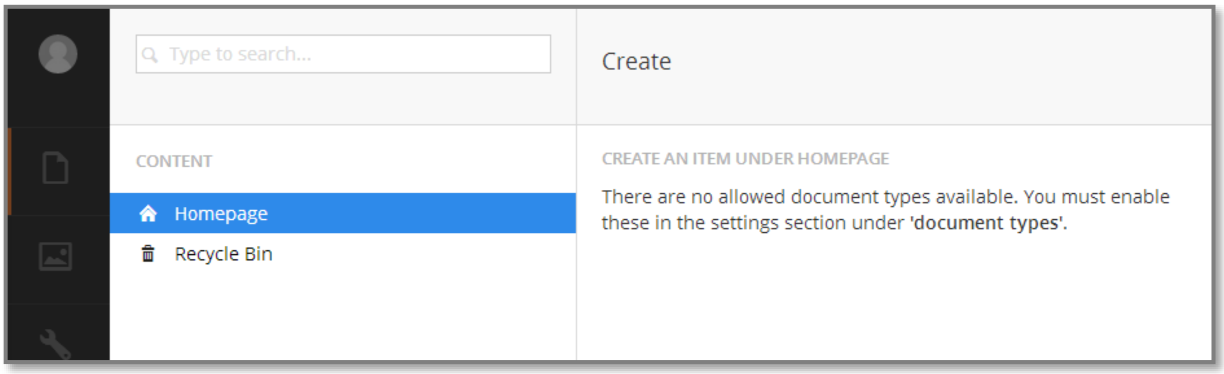
2. Go to ***Settings > Document Types*** (hover) ***> ... > + Create*** . Let's create one called "*Simple Content Page*". Leave the ***Master Document Type*** drop down to "*none...*" (the use of allows you to inherit property types from parents, I'd recommend you don't use it unless there is a definite need) but we'll create a matching template so leave this checked.

3. Firstly let's select an ***Icon*** – type the word "*Content*" into the filter and select the document icon. In description type "A simple content page", leave the ***Allowed Templates*** as it is (e.g. only ***Simple Content*** page checked). Click ***Save***.

4. Now click on the ***Settings > Templates (hover) > ...*** and then ***Reload Nodes.*** Click on your ***Simple Content Paqe node*** and then the ***Properties tab***. Change the ***Master template*** drop down to select value "*Master*" – this will mean that we'll get the header and footer from the master just as we do in the Homepage template. Click ***Save*** then load the ***Template tab*** you should see the portion of Razor code has updated to say *"Layout ="Master.cshtml""* if it hasn't updated itself click on a different node and then back again to reload it. Now add the following HTML to the template and click ***Save***.

```
<div id="main-container">
        <div id="main" class="wrapper clearfix">
                <section>
                        <h2>Header goes here</h2>
                        <p>Content goes here</p>
                </section>
        </div> <!-- #main -->
</div> <!-- #main-container -->
```
Figure 30 - Contact Us Template HTML

5. Now let's create a page using our new ***Document Type*** and ***Template*** – go to **Content > Homepage (hover) > ... > Create**. Oh but we can't! We'll see an error like that below:



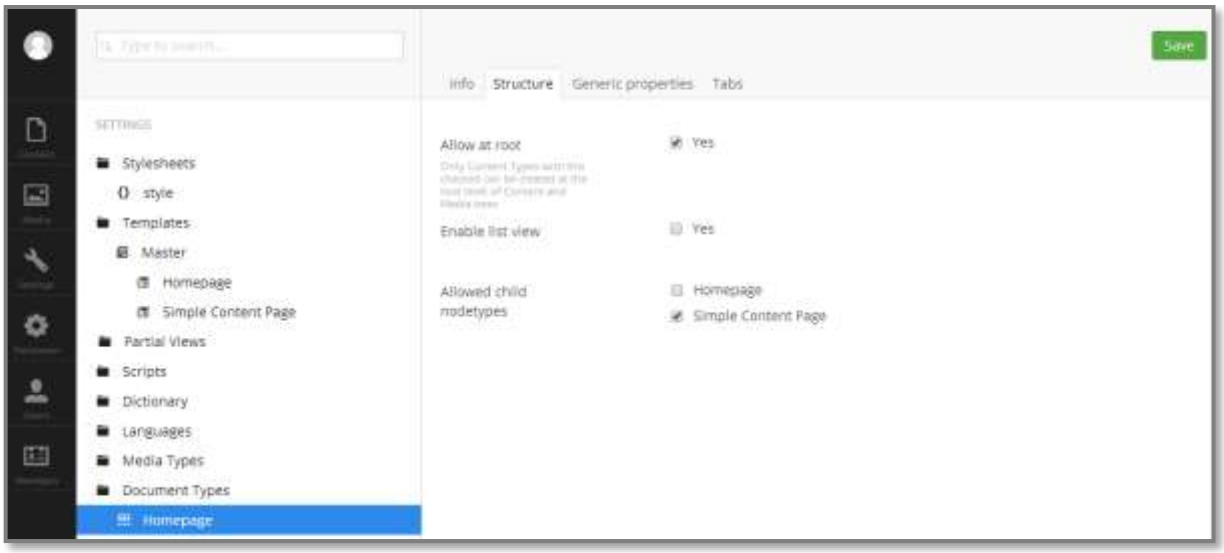
CREATE AN ITEM UNDER HOMEPAGE

There are no allowed document types available. You must enable these in the settings section under '**document types**'.

Figure 31 - Umbraco Content Error - No Document Types Available

6. This is by design – Umbraco limits the editors to only being able to create content in the places that you, the developer, allows. This will stop a user from breaking a site design (or an entire site) when they create a new homepage under the news container node later! Unfortunately this functionality also confuses a lot of new Umbracians – hence why we show you this error here.

Go to ***Settings > Homepage*** on the ***Structure*** tab you'll see a list of checkboxes under a label ***Allowed child nodetypes*** (be careful not to confuse this with the ***Info tab's Allowed templates*** – we'll cover that later). So we need to allow users to be able to create child nodes below our Homepage of type "*Simple Content Page*". Check the box and hit ***Save***.



Type to search...

SETTINGS

- Stylesheets
  - style
- Templates
  - Master
    - Homepage
    - Simple Content Page
- Partial Views
- Scripts
- Dictionary
- Languages
- Media Types
- Document Types
  - Homepage

Save

Info | Structure | Generic properties | Tabs

Allow at root — ✓ Yes
Only Content Types with this checked can be created at the root level of Content and Media trees

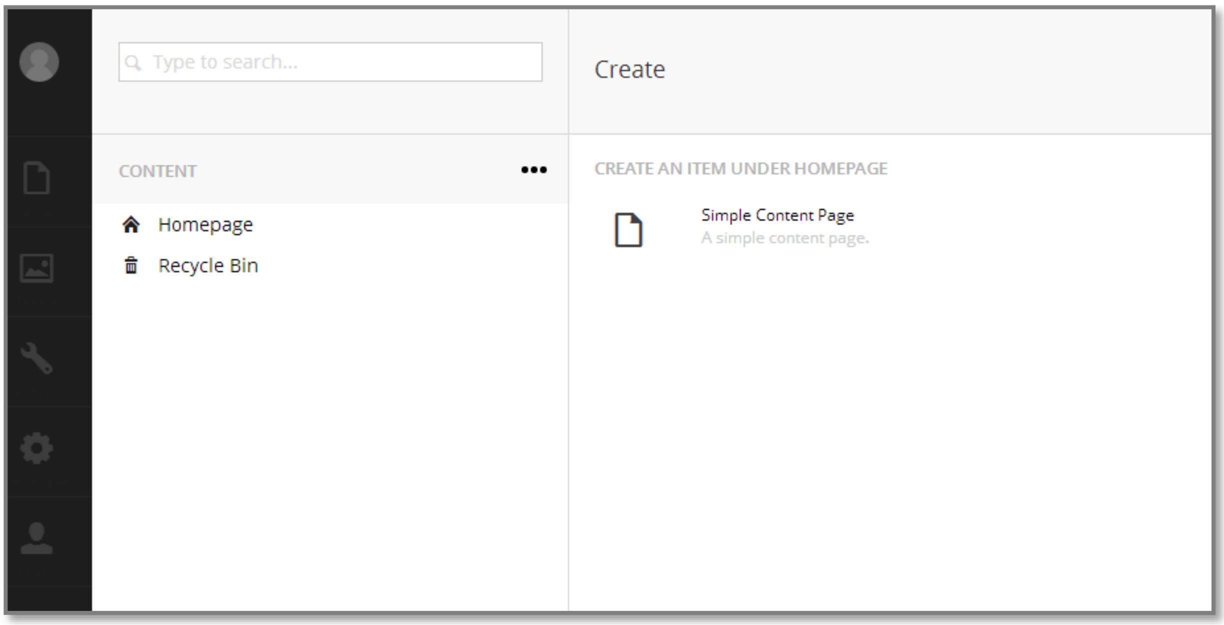
Enable list view — Yes

Allowed child nodetypes — Homepage; ✓ Simple Content Page

Figure 32 - Homepage - Allowed Child Nodetypes

7. So there is the confusing bit – first we create the ***Simple Content Page*** but then we have to allow it to be created under the document type that will sit above it – e.g. we create our new ***Document Type*** but then have to update the ***Homepage*** settings to be able to use it. We'll do this again later when we create an Articles container and Article item – we'll need to allow the Article items under the container. Simple – perhaps not but you'll get used to it!

8. Now go back **Content > Homepage (hover) > ... > Create –** now we have the **Simple Content Page**! Select this and enter a name (red text at the top). You can see that we only have a ***Properties tab*** here – no data. This is different to the document type for the homepage as we've not yet created any tabs nor data properties (e.g. no bodyText or pageTitle!). Click ***Save and Publish.***



Type to search...

CONTENT •••

- Homepage
- Recycle Bin

Create

CREATE AN ITEM UNDER HOMEPAGE

Simple Content Page
A simple content page.

Figure 33 - Creating Our Contact Us Page

9. Our ***Content tree*** will now reload and there will be a ***Contact Us*** page under the homepage. This is the recommended structure for most sites – your primary level 1 pages will sit under the Homepage. Go look at this page – if you look at the ***Properties tab*** you can see a ***Link To document row*** – click this. You might find an unstyled page again. This is because the template designers have assumed that your site will be a flat structure – e.g. all pages sitting at the same level

so the browser can’t find the CSS or JS at the page level below the homepage. You need to update the ***Master*** template to add a leading slash on the JS and CSS source lines.

E.g. change the lines in the Master Template:

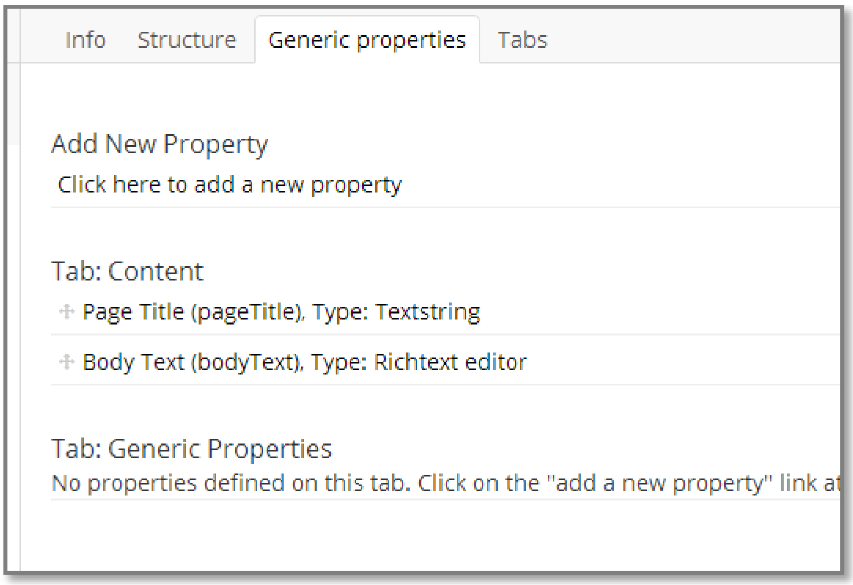
CSS	`<link rel="stylesheet" href="css/style.css">`
	`<link rel="stylesheet" href="/css/style.css">`

To:

JS	`<script src="js/libs/modernizr-2.0.6.min.js"></script>`
	`<script src="/js/libs/modernizr-2.0.6.min.js"></script>`

10. ***Save*** the template changes and reload your ***Contact Us page***. We’ll now have a pretty empty looking page.

11. Let’s add two simple fields – ***pageTitle*** (type = Textstring) and ***bodyText*** (type Rich Text Editor). Follow the instructions in creating the Homepage document type if you’re not sure how to do this. Then wire these fields up – by editing the Template, again follow the Homepage section if this isn’t yet second nature to you yet!

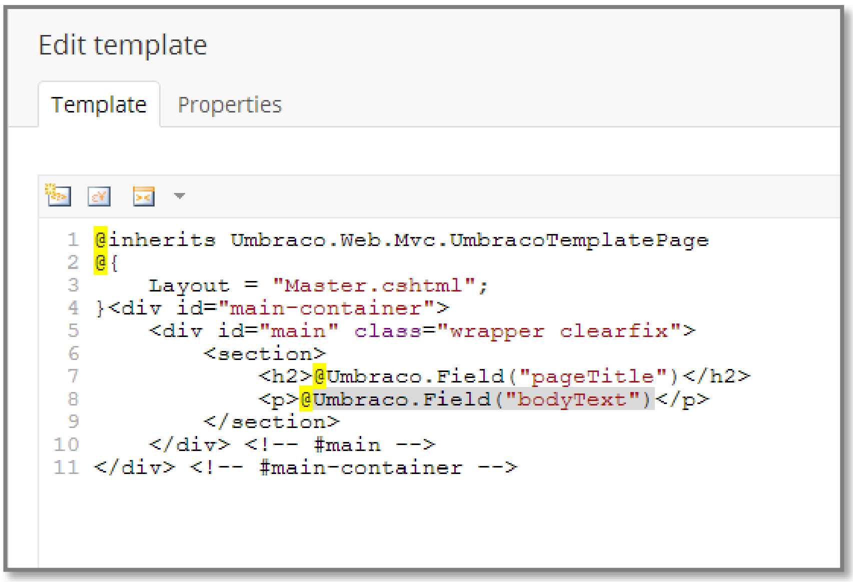


Add New Property
Click here to add a new property

Tab: Content
Page Title (pageTitle), Type: Textstring
Body Text (bodyText), Type: Richtext editor

Tab: Generic Properties
No properties defined on this tab. Click on the "add a new property" link at

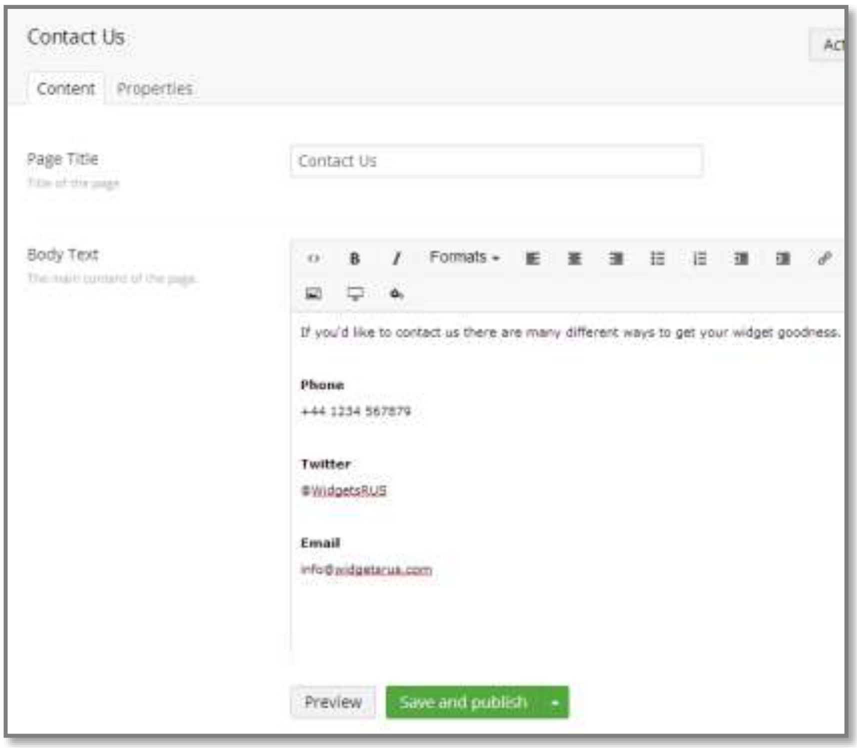
Figure 34 - Contact Us Generic Properties



```
@inherits Umbraco.Web.Mvc.UmbracoTemplatePage
@{
    Layout = "Master.cshtml";
}<div id="main-container">
    <div id="main" class="wrapper clearfix">
        <section>
            <h2>@Umbraco.Field("pageTitle")</h2>
            <p>@Umbraco.Field("bodyText")</p>
        </section>
    </div> <!-- #main -->
</div> <!-- #main-container -->
```

Figure 35 - Contact Us Template with Data Fields

12. Now add some content under the ***Content > Homepage node > Contact Us node***. Click ***Save and Publish*** and you should have a slightly more interesting page when you reload it!



If you'd like to contact us there are many different ways to get your widget goodness.

Phone

+44 1234 567879

Twitter

@WidgetsRUS

Email

info@widgetsrus.com

Figure 36 - Contact Us with Some Data

7.4 Using Data Fields from the Homepage

1. What you may notice is that the footer is now empty – we don't have our content from our Homepage node. We need to tell Umbraco to get the content from the parent ***Homepage*** tab. To do this we edit the template (the Master).

Highlight the Umbraco field in the footer <h3> tags and then click the ***Insert Umbraco page field*** button again. here is where all of the options we ignored earlier come into play – choose footerText again from the ***Choose field dropdown*** but this time we'll check the Recursive checkbox. This tells Umbraco that if the field doesn't exist at the node level for the page we're requesting (e.g. Contact US) it will look up the content tree (so in our example go to the ***Homepage*** for this content) – this means you could also create a ***footerText*** element at the Contact Us page if you wanted the editor to override the site wide default but for fields like this it's not normally used. Click ***Insert*** and you'll see a different bit of Razor is added : *@Umbraco.Field("footerText", recursive: true)*

2. Click ***Save*** and reload our Contact Us page.

7.5 Master Template – The Menu

1. Now let's fix the menu – there are two ways of doing this, you could have Umbraco dynamically create a menu from the pages it has in the Content Tree, so that when an editor creates a page it automatically appears or, more simply you can hardcode it. We're going to hardcode this for now (it's a good idea as you start building a site to hard code this so you can move around testing before you replace this) and we'll leave it to you as an exercise to do this later. Edit your ***Master template*** – edit the <li> items under the <nav> tags to say:

```
<nav>
        <ul>
                <li><a href="/">Home</a></li>
                <li><a href="/contact-us">Contact Us</a></li>
                <li><a href="/articles">Articles</a></li>
        </ul>
</nav>
```

Figure 37 - Master Template - Menu / Nav Section

2. ***Save*** your changes and let's test our menu. You'll find that clicking on the Article link throws an Umbraco error as we've not created this page yet. Let's do that now.

8 Parent Pages with Infinite Children

Having an Articles Parent page and a number of associated child articles which the editors can add to freely is a good example of the power of Umbraco. We'll assume our fictional company, Widgets Ltd, write about ten articles a month and want the articles page to act like a blog (e.g. you could use this functionality for a blog or news and events pages)!

8.1 Creating the Articles Parent and Article Items

1. Create two new Document Types "*Articles Home*" and "*Articles Item*" ***Document Types Settings > Document Types (hover) > ... > + Create*** . Remember to leave ***Master Document Type*** = "*none..*".

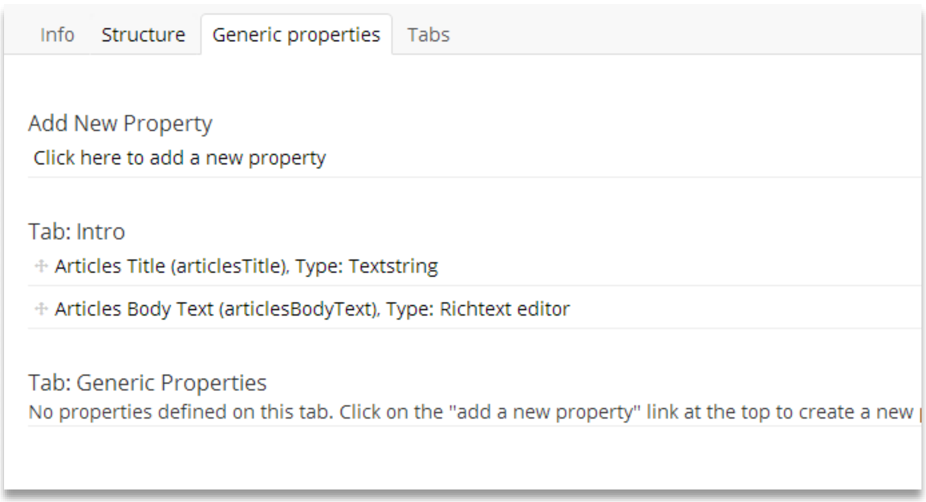
2. Create the following ***Tabs*** and ***Data Properties***:

Articles Main

Tab = Intro

"Articles Title" – Type = Textstring

"Articles Body Text" – type = Rich Text Editor



Add New Property
Click here to add a new property

Tab: Intro
Articles Title (articlesTitle), Type: Textstring
Articles Body Text (articlesBodyText), Type: Richtext editor

Tab: Generic Properties
No properties defined on this tab. Click on the "add a new property" link at the top to create a new

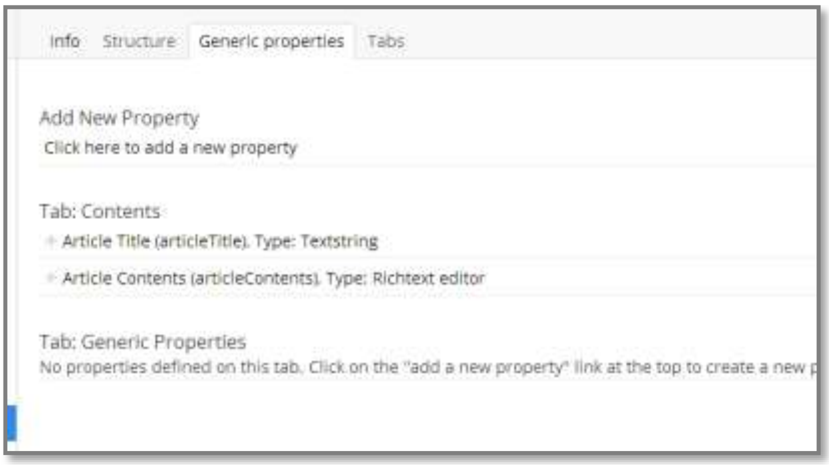
Figure 38 - Articles Main Document Type Data Properties

Articles Item

Tab = Contents

"Article Title" – Type = Textstring

"Article Contents" – type = Rich Text Editor



Add New Property
Click here to add a new property

Tab: Contents
Article Title (articleTitle), Type: Textstring
Article Contents (articleContents), Type: Richtext editor

Tab: Generic Properties
No properties defined on this tab. Click on the "add a new property" link at the top to create a new

Figure 39 – Article Item Document Type Data Properties

3. Now go to the ***Settings > Document Types >Articles Main node > Structure tab > Allowed child nodetypes*** and check the ***Articles Item***. This allows us to create items under the main (which acts as a parent container). We also need to allow the ***Articles Main node*** to be created under the ***Homepage node*** (do this in the ***Settings > Document Types > Homepage node > Structure tab > Allowed child nodetypes –*** don't check the ***Articles Item*** – only the main should be at this level).

4. Now go to ***Content > Homepage node (hover)> ...*** and create a node called "*Articles*" of type ***Articles Main*** (if you don't have this option go back and check your allowed child nodes – did you forget to click ***Save***)? Give the Articles node some content and a title and then create a couple of article item content nodes under this node (***Content > Homepage node > Articles node (hover) > ...***

5. Now you should have a content tree that looks like this below (obviously with your own content node names that you added before). Let's go update our templates we just created (automatically when we created the Document Types). First update them to use the Master as a parent ***Settings > Templates > Articles Main node > Properties tab > Master template dropdown*** = "Master" – do the same for the Articles Item remembering to click ***Save***.

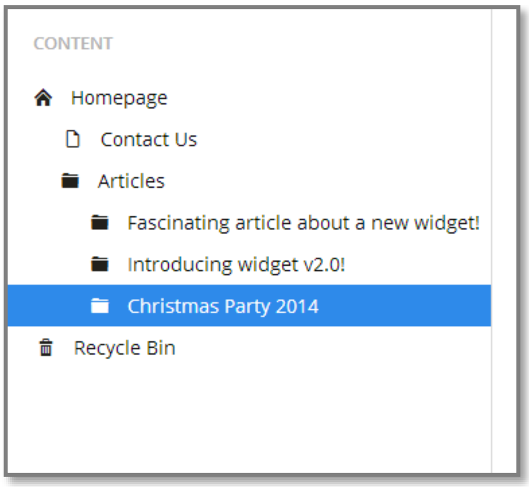
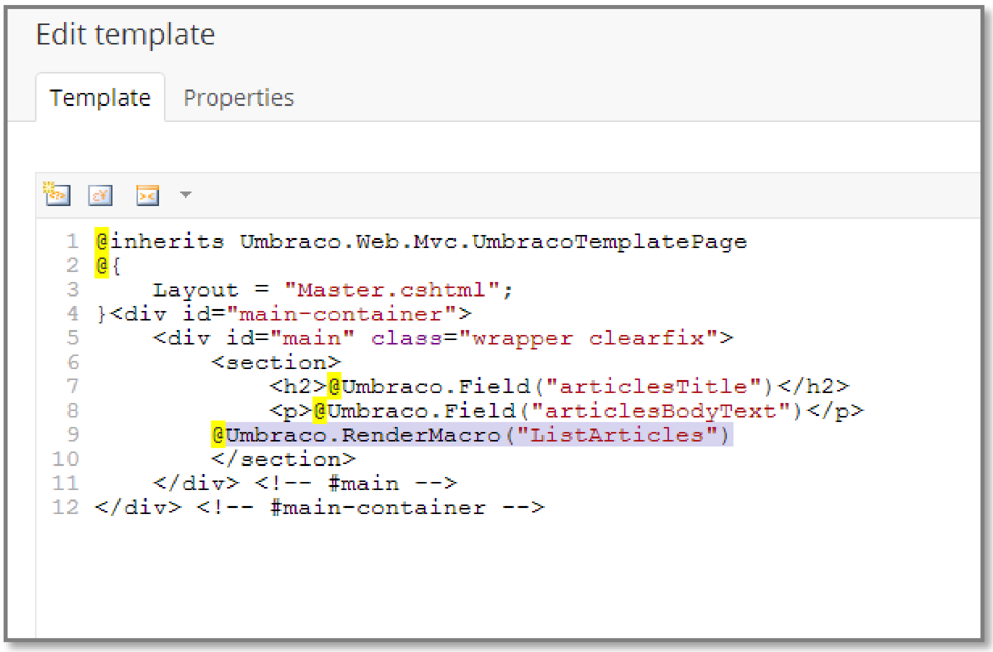


Figure 40 - Content Tree With Articles

6. Copy the template content from the ***Simple Content Page*** template and paste this into each (clicking ***Save***). Then replace the Page fields with the relevant e.g. ***articlesTitle*** and ***articlesBodyText*** for the ***Articles Main*** and the ***articleTitle*** and ***articleContents*** for ***Article Item***.

7. If we now go and check our Articles Main page in the browser we should see our content. Now we need to list the children under this so that we can see a list of our articles. Umbraco makes this easy for us but we need to use a bit of Razor.

8. Click on the ***Developer*** menu from the left hand side menu and then hover over the ***Partial View Macros Files node*** to get the more menu ***...*** then ***click + Create***. Name this "*listArticles*" and select the "*List Child Pages Ordered By Date*" in the ***Choose a snippet*** field and click ***Create***.



```
@inherits Umbraco.Web.Mvc.UmbracoTemplatePage
@{
    Layout = "Master.cshtml";
}<div id="main-container">
    <div id="main" class="wrapper clearfix">
        <section>
            <h2>@Umbraco.Field("articlesTitle")</h2>
            <p>@Umbraco.Field("articlesBodyText")</p>
        @Umbraco.RenderMacro("ListArticles")
        </section>
    </div> <!-- #main -->
</div> <!-- #main-container -->
```

Figure 41 - Template for Articles Parent with the Macro Code

9. Now all we have to do is wire up the Articles main page to list our child articles. Edit the Articles Main template ***Settings > Templates node > Master node > Articles Main node > Template tab***.

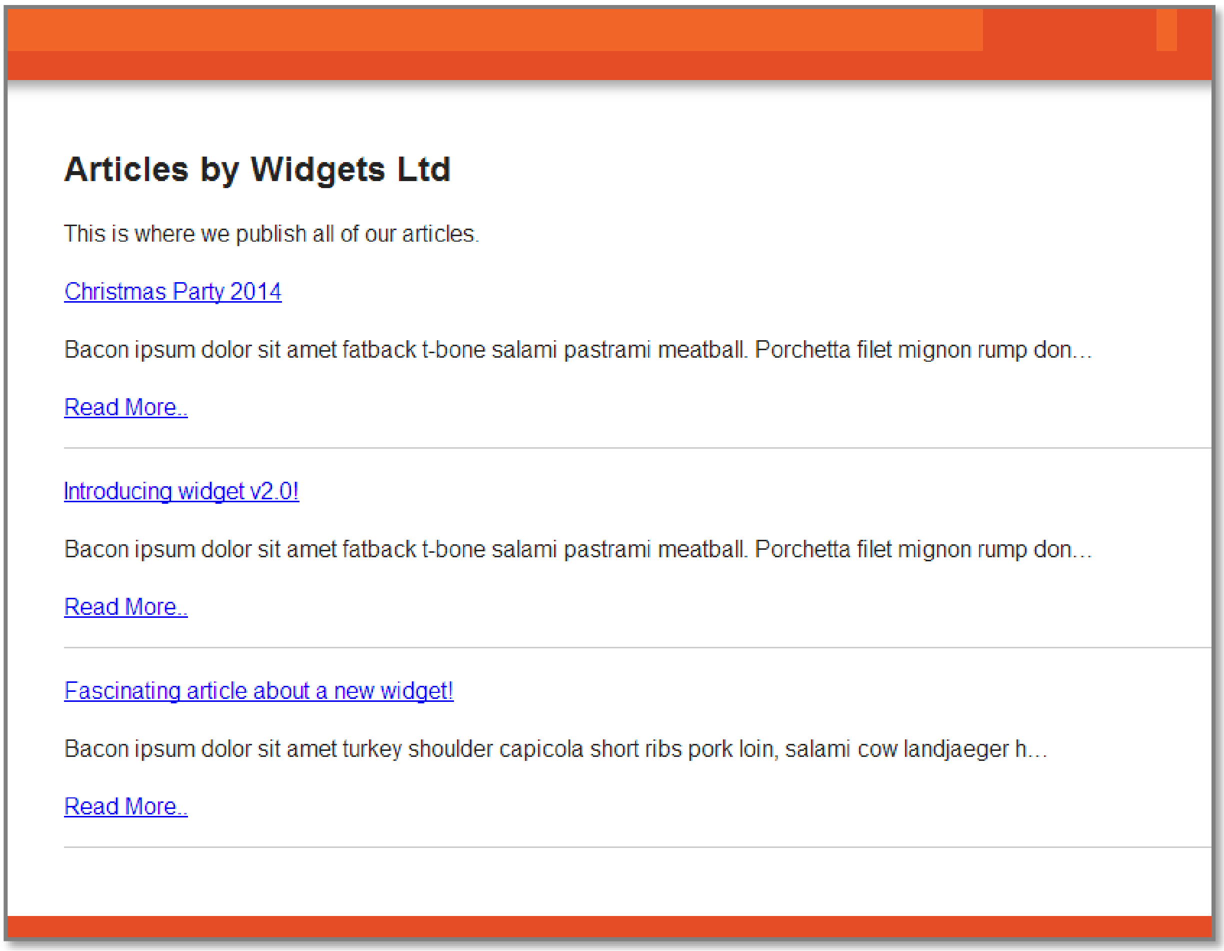
   Under the paragraph tags enter a carriage return and then click the ***Insert Macro*** button then click ***Save***.

10. Check what we have on our ***Articles*** page now - we're really getting somewhere! Let's make it a bit more real world – I'll leave the understanding of this to Razor lessons / The Umbraco videos but it will finish our site off nicely – edit the Partial you just created – ***Developer > Partial View Macro Files > listArticles.cshtml*** and change the content to be:

```
@inherits Umbraco.Web.Macros.PartialViewMacroPage



       @* OrderBy() takes the property to sort by and optionally order desc/asc *@

    @foreach (var page in CurrentPage.Children.Where("Visible").OrderBy("CreateDate desc"))
    {
               <div class="article">
               <div class="articletitle"><a href="@page.Url">@page.Name</a></div>
                      <div
class="articlepreview">@Umbraco.Truncate(@page.ArticleContents,100) <a href="@page.Url">Read
More..</a></div>
               </div>
               <hr/>
    }
```

Figure 42 - Improved Macro for listArticles

11. Now check this in the browser!


Articles by Widgets Ltd

This is where we publish all of our articles.

Christmas Party 2014

Bacon ipsum dolor sit amet fatback t-bone salami pastrami meatball. Porchetta filet mignon rump don...

Read More..

Introducing widget v2.0!

Bacon ipsum dolor sit amet fatback t-bone salami pastrami meatball. Porchetta filet mignon rump don...

Read More..

Fascinating article about a new widget!

Bacon ipsum dolor sit amet turkey shoulder capicola short ribs pork loin, salami cow landjaeger h...

Read More..

Figure 43 - A Completed Articles Page

9 Conclusions

What we now have is a fully fledged working site! Hopefully this guide should have given you all of the basics to see how to create your own site in Umbraco. For the majority of sites you'll create they won't be much more complicated than this. We've not written any code nor have we needed to work much out of the Umbraco UI.

This is not the limit of Umbraco – you've barely scratched the surface in this guide.

- For further reading / exercises I'd recommend working through the video tutorials: http://umbraco.com/help-and-support/video-tutorials.aspx
- There is also a wiki (of which the intention is to use this guide to populate the "Creating a basic site with Umbraco" section). http://our.umbraco.org/documentation
- For help and guidance on anything this guide doesn't cover the forums are a great place to start:
- http://our.umbraco.org/forum
- This guide was produced by Steve Morgan of Siempre Solutions – for more info contact him via http://www.siempresolutions.co.uk

Happy Umbraco-ing!