**Exercises**

Contents

[Sitecore Overview and Data Definition 1](#_Toc457051315)

[Sitecore Development – Create a simple website 6](#_Toc457051316)

[Overview 6](#_Toc457051317)

[Template Creation 6](#_Toc457051318)

[Item Creation 11](#_Toc457051319)

[Layout File Creation 15](#_Toc457051320)

[Creating the main layout 15](#_Toc457051321)

[Creating Rendering Files 15](#_Toc457051322)

[Sitecore Layout and Rendering Components 16](#_Toc457051323)

[Sitecore Layout 16](#_Toc457051324)

[Rendering Components 17](#_Toc457051325)

[Sitecore Placeholder Settings 18](#_Toc457051326)

[Adding Rendering components to a template in Sitecore 19](#_Toc457051327)

[Publishing the Sitecore site 22](#_Toc457051328)

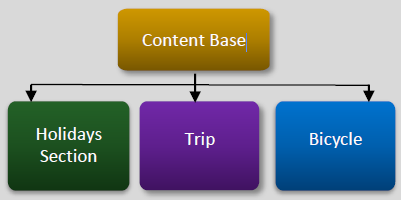
# Sitecore Overview and Data Definition

1. Creating Data Templates

* Content Base template
  + Name: Content Base
  + Path: master/sitecore/Templates/User Defined.
  + Field section and fields

|  |  |
| --- | --- |
| **Field section** | |
| Content Details | |
| **Field name** | **Field type** |
| Heading | **Single-Line Text** |
| Main Content | **Rich Text** |
| Main Image | **Image** |

1. Data Template Inheritance

* Trip template
  + Name: Trip
  + Path: master/sitecore/Templates/User Defined.
  + Base Template: Content Base
  + Field section and fields

|  |  |
| --- | --- |
| **Field section** | |
| Trip Details | |
| **Field name** | **Field type** |
| Price per person | **Single-Line Text** |
| Start Date | **Date** |

* Holidays Section template
  + Name: Holidays Section
  + Path: master/sitecore/Templates/User Defined.
  + Base Template: Content Base
* Bicycle template
  + Name: Bicycle
  + Path: master/sitecore/Templates/User Defined.
  + Base Template: Content Base

1. Create Content Items based on Holidays Section and Trip templates

* In Sitecore Explorer, locate the Home item under /BasicSitecore/master/sitecore/content/
* Right-click on the Home item and click Add > New Item…
* Search for Holidays Section or Trip in the Add New Item dialog
* Fill the Name and click OK
* Fill the data and click Save (or Ctrl + S)

1. Standard Values

* Right-click the Content Base data template and click Create Standard Values. This will create a \_\_Standard Values item as a child of Content Base.



* Type **$name** into the **Heading** field
* Right-click the **Main Content** field name (or click the dropdown arrow) and click Add ‘Lorem Ipsum…’, or type your own sample content.
* Create Content Items based on Trip template to see data is populated.

Notes: All OOTB tokens

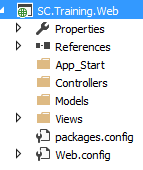
* $name Name of the item
* $id ID of the item $parentid ID of the parent of the item
* $parentname Name of the parent of the item
* $date Current date (ISO format)
* $time Current time (ISO format)
* $now Current date and time (ISO format)

1. Insert Options

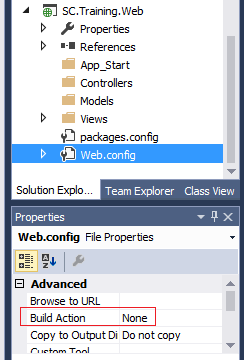
* Add Trip as an Insert Option to the Holidays Section data template
* Remember to assign Insert Options on the data template’s standard values.

1. Create a Visual Studio Project

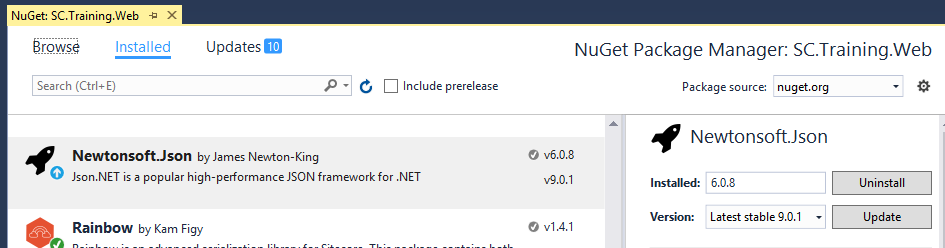
* Create an empty MVC project
* Update all references with Copy Local = False
* Delete all items except items in the below image



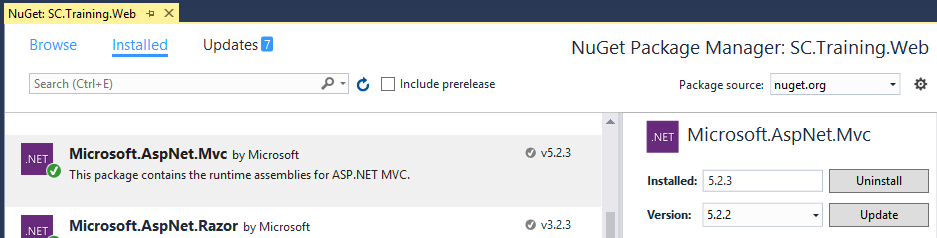
* Change build action of **Web.config** to **None** as below



* Change Nuget packages related to **Newtonsoft.Json** and **System.Web.Mvc**
  + Change version of Newtonsoft.Json to 6.0.8



* + Change version of System.Web.Mvc to 5.2.3

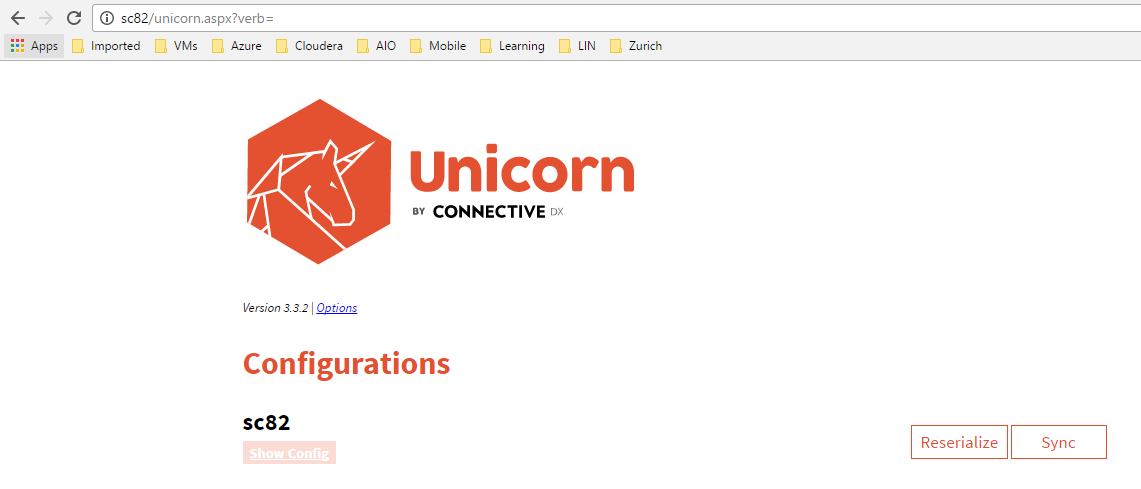


* Ensure assembly versions be configured correctly with above version changes in **SC.Training.Web\Views\Web.config**

**Notes:** A similar **Web.config** file can be found from Sitecore web folder **\Website\Areas\Social\Views**

* Custom Profile for Publishing Web project
  + Publish method: File System
  + Target location: <http://BasicSitecore>
  + Publish web project

1. Serialization
   * Install Unicorn Nuget
   * Configure Serialization Items
   * Publish project and perform serialization (<http://BasicSitecore/unicorn.aspx>)



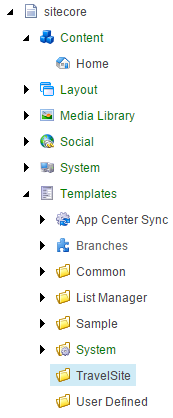
# Sitecore Development – Create a simple website

## Overview

To keep it simple this tutorial will be creating a web site for a simple travel agency. It will include generic content pages with information about the company and the services they offer (about, contact etc.), along with a custom destination item which uses a custom template/layout in Sitecore to showcase the various destinations which the agency offers travel to.

## Template Creation

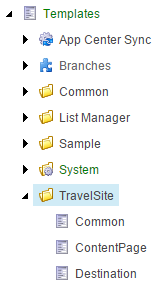
Log into Sitecore administration and open up the content editor. On the right hand menu expand the Templates item, right click it and create a Template Folder named TravelSite. By creating a folder to hold all custom templates, we make content syncing between environments easy. It also makes it easier for another developer to work with.

Đăng nhập vào quản trị Sitecore và mở trình soạn thảo nội dung. Trên menu bên phải mở rộng mẫu mục, nhấp chuột phải vào nó và tạo một thư mục mẫu có tên TravelSite. Bằng cách tạo một thư mục để giữ tất cả các mẫu tùy chỉnh, chúng tôi thực hiện đồng bộ hóa nội dung giữa các môi trường dễ dàng. Nó cũng làm cho nó dễ dàng hơn cho một nhà phát triển để làm việc với.

Right click the Travel Site folder and click Insert > New Template, do this 3 times and call them the following (they can all inherit from the standard template):

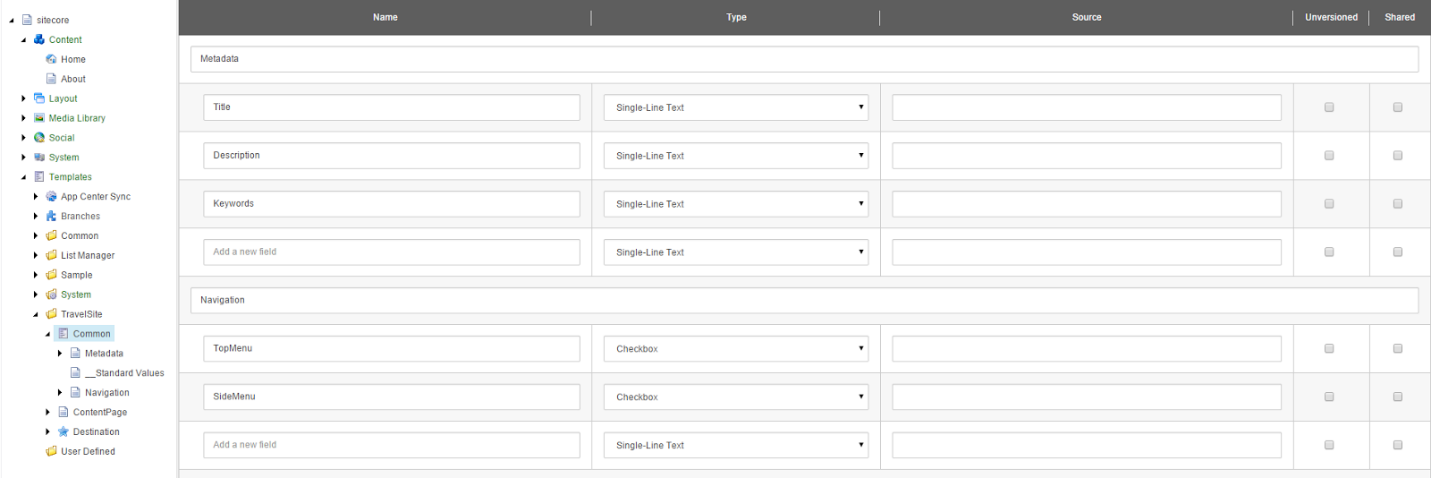
**• Commons:sẽ có các trường phổ biến trên các mẫu mục khác (chẳng hạn như tiêu đề, siêu dữ liệu và cài đặt điều hướng).**

**• ContentPage : đây là trang nội dung chung**

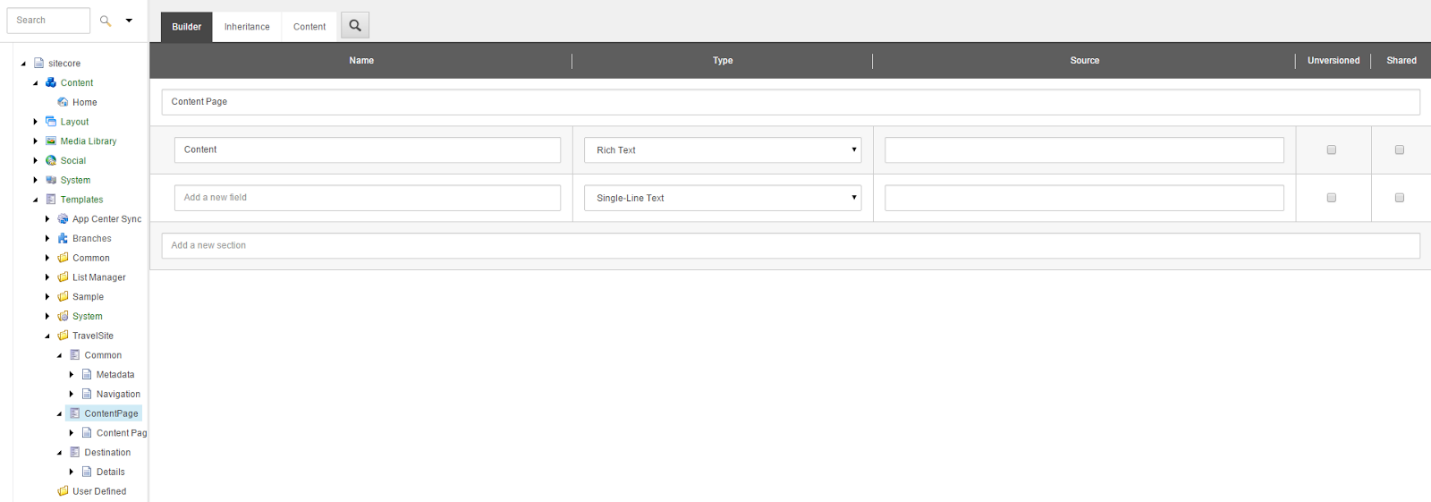
**• Destination: đây là điểm đến du lịch**Your content tree to the left will now look like the following:  


Now the field structure for the templates will be defined, enter the following details into the corresponding template (to keep things simple for right now, only the text and check box types are being used, however list types are ideal for fields like country).

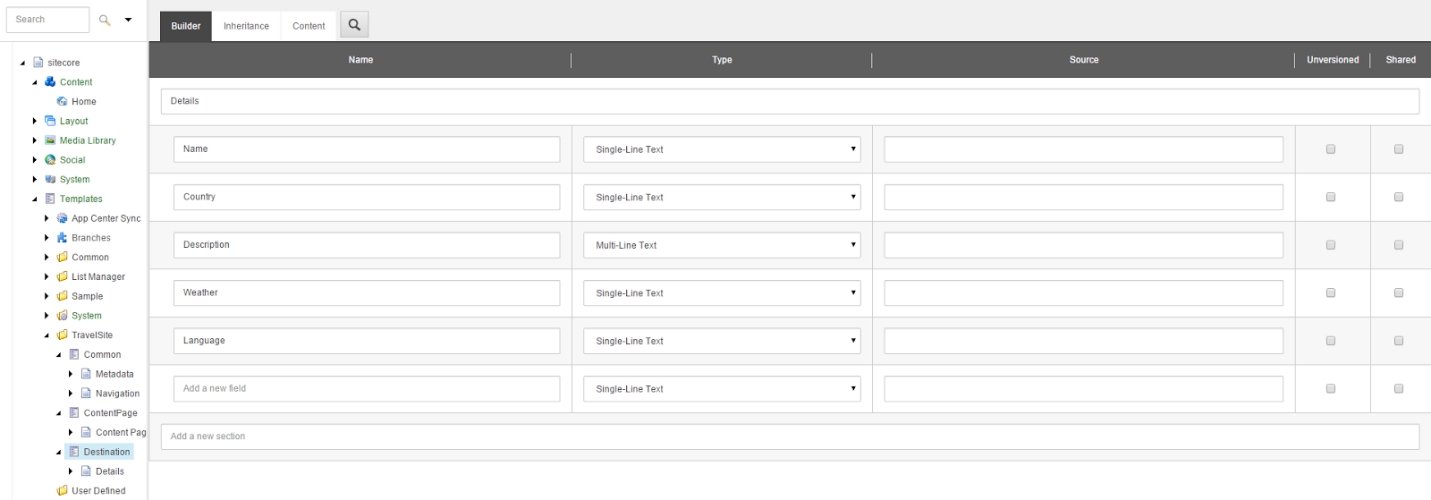
* Common



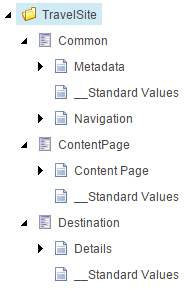
* ContentPage



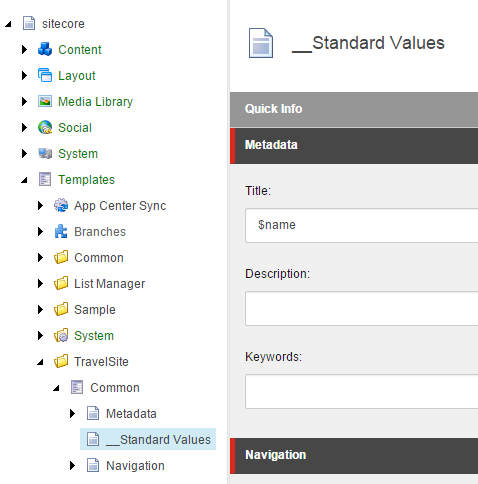
* Destination



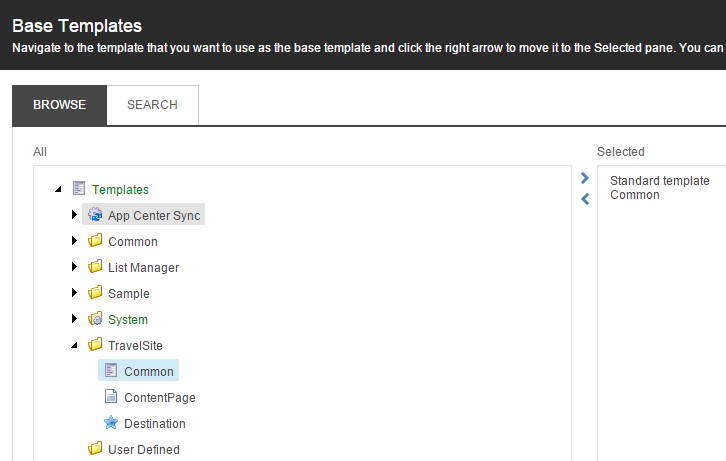
Click on each template and from the top Sitecore administration menu click Options > Standard values. This creates the standard values configuration for each template where we can set the default field settings, layouts and sublayouts. The folder structure should now look like the following:



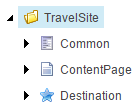
On the Common \_\_Standard Values file, set the Title field to "$name", this will default that title field to the name of the Sitecore item. The reason we use this field is because the title may want to be output to the page with special characters (apostrophe for example), and Sitecore does not allow these in item names. On the Common standard values file, tick the TopMenu and SideMenu check boxes.



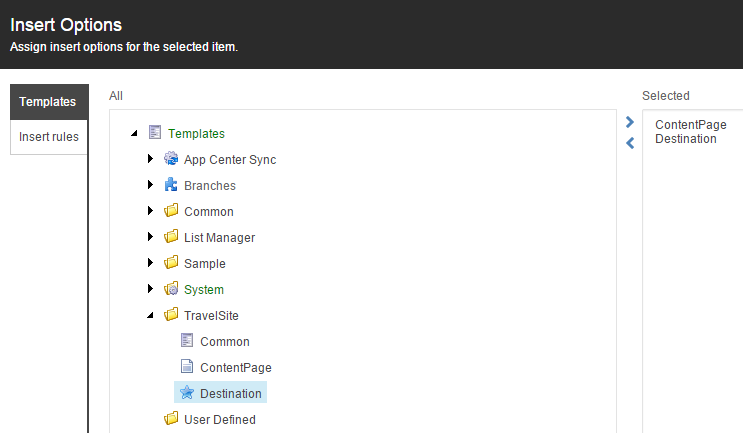
We also need to assign the Common template as a base template to ContentPage and Destination. Select each item and from the top menu Options and then Base templates (Template). Now add the Common template and click Next to save the changes.



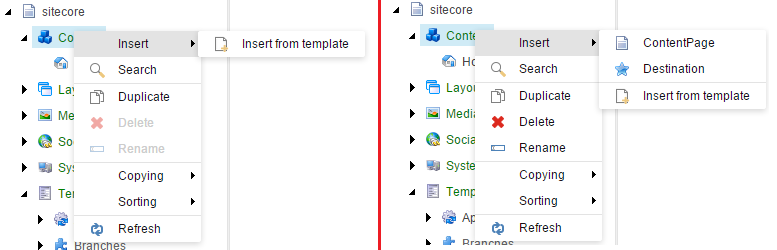
To give a nice user experience, we will also assign some icons to the ContentPage and Destination templates. This is done by selecting the templates, then from the top menu selecting Configure > Icon.



On both the ContentPage and Destination templates standard values we are going to set the insert options. This is what allows content authors to right click an item of this template type and have the templates appear there as insert-able right away. To set the insert options click on the standard values for ContentPage and Destination and from the Sitecore menu select Configure > Insert Options (Assign) and select both ContentPage and Destination Templates.

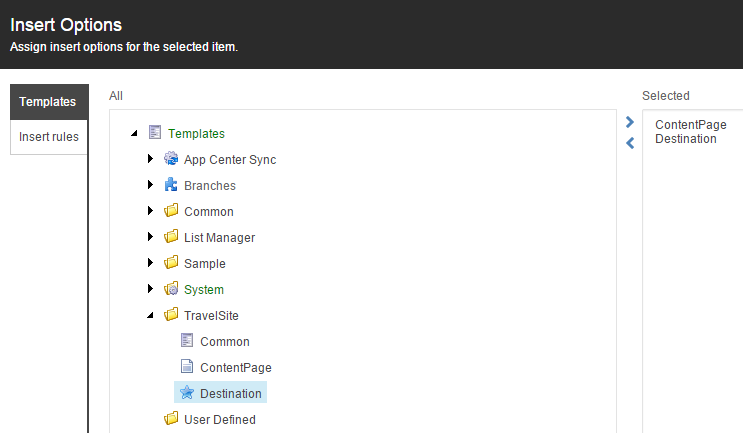


This will mean that whenever a user right clicks and selects insert on any item of template type ContentPage or Destination they will be able to insert another ContentPage or Destination without having to manually find the template in the tree.



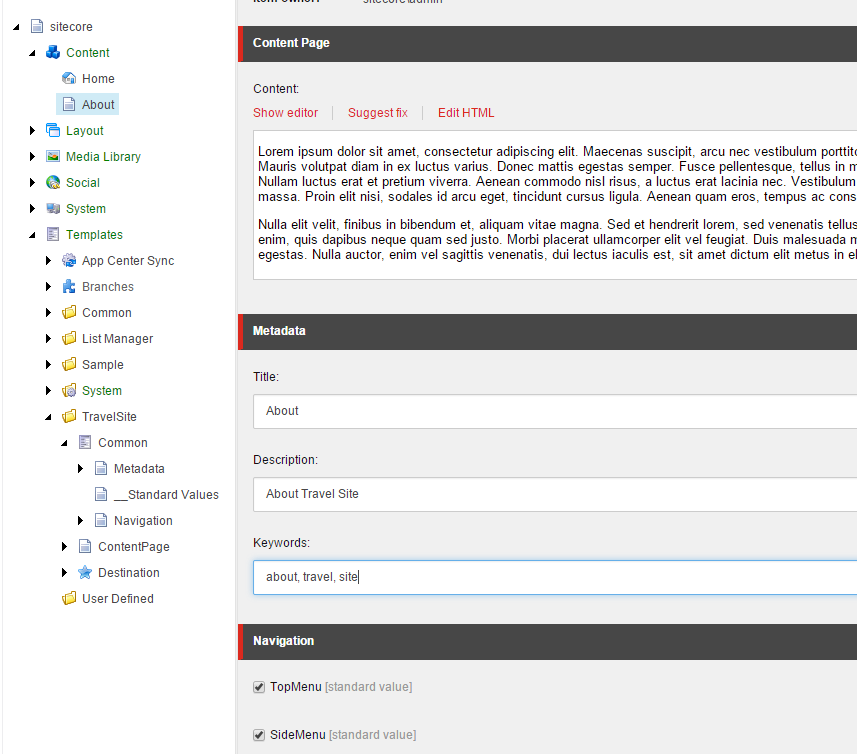
## Item Creation

Log into Sitecore administration and load up the content editor. Select the main Content item and from the top menu select Configure and Assign (Insert Options). Select the ContentPage and Destination templates as the insert options for this item.

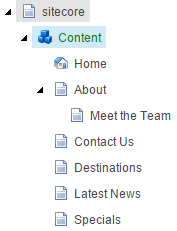


Select the Home item under Content and from the top menu select Configure and then Change (Template). Now select the ContentPage template found in the TravelSite folder, you will get a message indicating that fields will be lost, click Next as these fields were from the default Sitecore template (not required).

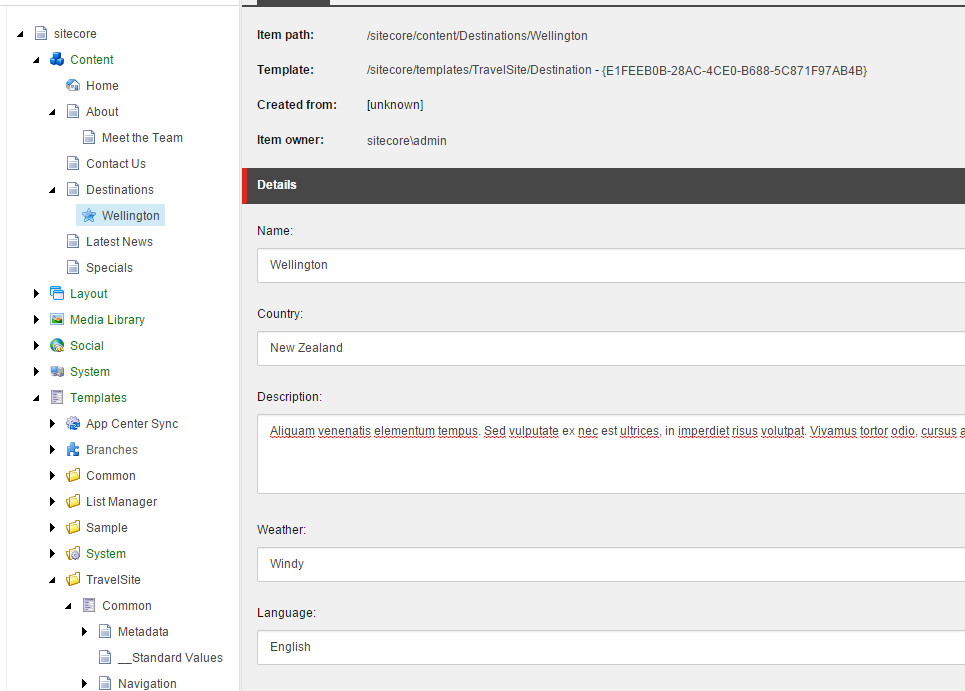
Now we are going to create some simple content pages. Right click on the Content item and then Insert > ContentPage and give the item the name "About". Under the Content field enter some HTML content (click Edit HTML) for the page, the title attribute has pre-populated from the standard values, however you will need to enter a description and keywords.



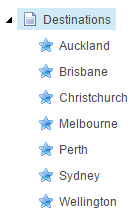
Do the same step as above for some other general content pages, I added 5 more content pages.



Now right click the destinations item and select Insert > Destination, give the destination the name of "Wellington". Enter some details like the following image (please note that for this tutorial we have used simple content types rather than lookups or selections for country and weather to keep things simple).



Now enter some other destinations (with multiple countries). I have entered 7 destinations across New Zealand and Australia.



## Layout File Creation

### Creating the main layout

For this tutorial I will be using the clean blog design, available from Start Bootstrap (mirror of download available here). To integrate this into the solution, follow these steps:

* Delete: the mail folder, about.html, contact.html, LICENSE, post.html, README.md
* Create a folder in the root of the project called "Resources" and place the following folders inside it (remember to include these in the project):
  + css
  + fonts
  + img
  + js
  + less
* Under Shared folder, create MainLayout.cshtml
* On the MainLayout.cshtml page use the following code (taken from index.html) as attached



You will notice that there is a registration included for the Sitecore namespace and that 2 Sitecore placeholders have been defined (TopNavigation and Main). These placeholders will allow us to add rendering components onto the layout in Sitecore. There is also a field output in the title tags, with DisableWebEditing set to true because meta tags can't be edited in the browser by content authors. For on page text fields, this value would not need to be set and the user could edit the fields on page.

### Creating Rendering Files

In the Views folder of the project create a folder named "Content" and add a new partial View named "MainContent.cshtml". On the view add in the following code:



This is simply outputting the HTML from the Content field on the ContentPage Sitecore items. For more advanced logic you would use C# code to get the fields, perform logic and then output to a repeater or other control.

Create the DestinationContent Partial View in the same Content folder.



In the Models folder of the project, create a class named NavMenuItem with the following code:



In the Controllers folder of the project, create a new controller class with the following code:



Notes: In the navigation controller we are getting the main Sitecore content item (by GUID - you will need to update to the GUID for your content item). Then we get all of the children of content and add them to the list of menu items. This list of menu items is then bound to the repeater on the user control.

In the partial view TopNavigation.cshtml, add the below code:



Now you can publish the solution to your Sitecore web site, this can be done by right clicking on the project and clicking Publish.

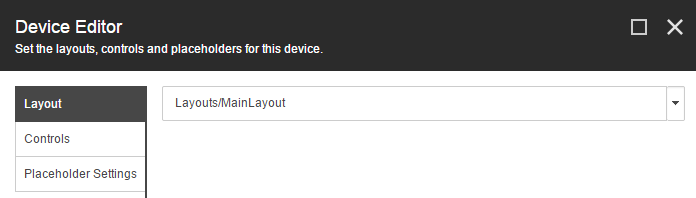
## Sitecore Layout and Rendering Components

### Sitecore Layout

A layout is basically a master page for Sitecore, it is required for Sitecore items to render in the browser and has placeholders for sublayouts to render as well. To define a layout, load up Sitecore administration and then the content editor. In the content tree expand the Layout item and right click Layouts and click Insert > Layout. Give the layout a name of "MainLayout" and click Next, then click Next again twice as the default locations are okay. Now to connect this layout with the file you created in the visual studio project, change the Path to "/Views/Shared/MainLayout.cshtml".



Select the ContentPage template's standard values from the content tree and then select Details (under presentation) off of the top menu. Now on the layout tab (after clicking edit under default) select the MainLayout created in the previous step. By setting this value, by default (because of standard values) any ContentPage item in Sitecore will use the MainLayout layout.

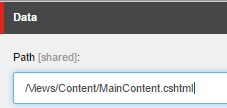


Do the same for the Destination template's standard values.

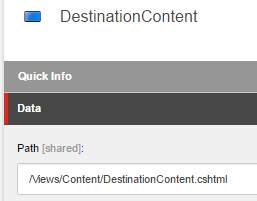
### Rendering Components

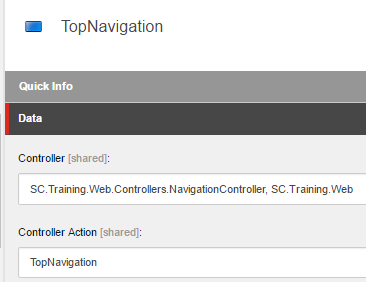
Sitecore Rendering component can be thought of as widgets (or web parts in SharePoint), they distinct items (with code behind) that can be added to Sitecore layout. A Sitecore item can only have one Layout (although it can be different based on desktop, mobile or print), but can have many rendering components - even multiple copies of the same one.

To define a Rendering component, load up Sitecore administration and then the content editor. In the content tree expand the Layout item and right click Renderings and click Insert > Rendering Folder and name it "TravelSite". Right click the newly created folder and click Insert > View Rendering, give it a name of "MainContent as below details:

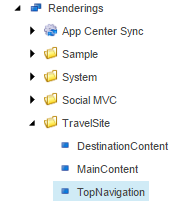


Do the same for Destination Content (View Rendering) and Top Navigation (Controller Rendering) details as below:





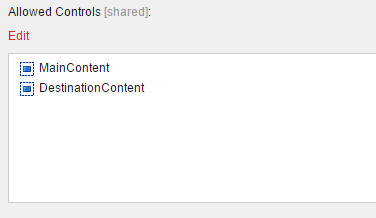
Finally, we a tree like below:



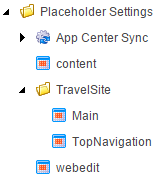
### Sitecore Placeholder Settings

In the main layout cshtml file in visual studio, 2 Sitecore placeholders were referenced. We now need to reference these in Sitecore to allow them to be used. In the content tree expand the Layout item and right click Placeholder Settings and click Insert > Placeholder Settings Folder and name it "TravelSite". Right click the newly created folder and click Insert > Placeholder, give the placeholder a name of "Main".

The placeholder has a setting called Allowed Controls, this allows you to manually define which controls can be added to this placeholder. If left blank, there are no restrictions however if you add selected Rendering components, only those ones can be added. Click edit and select the MainContent and DestinationContent renderings.



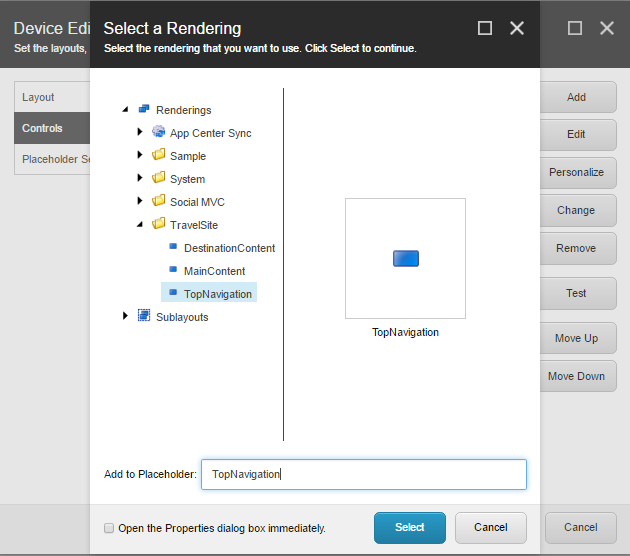
Create another placeholder called TopNavigation and set it's allowed controls to allow the Navigation rendering. Your content tree should look like the following:



### Adding Rendering components to a template in Sitecore

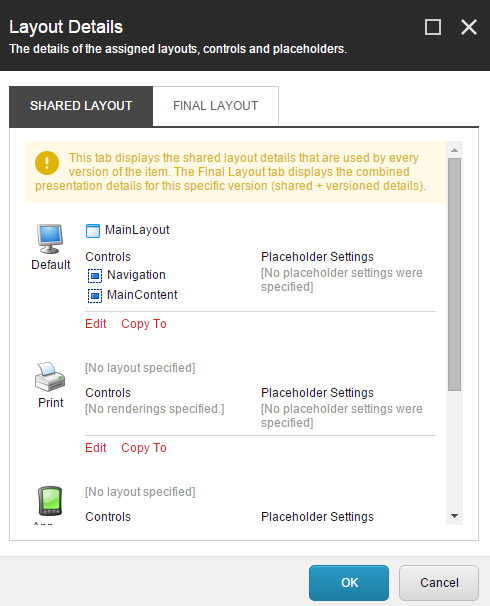
Now that we have rendering components and placeholder settings defined, we could add the required renderings to the content items created earlier in this tutorial. However instead of doing it manually multiple times, like anything else the renderings can be defined at the template level.

Select the ContentPage template's standard values from the content tree and then select Details (under presentation) off of the top menu. On the Controls tab (after clicking edit under default), click Add, then add the Navigation rendering to the TopNavigation placeholder.

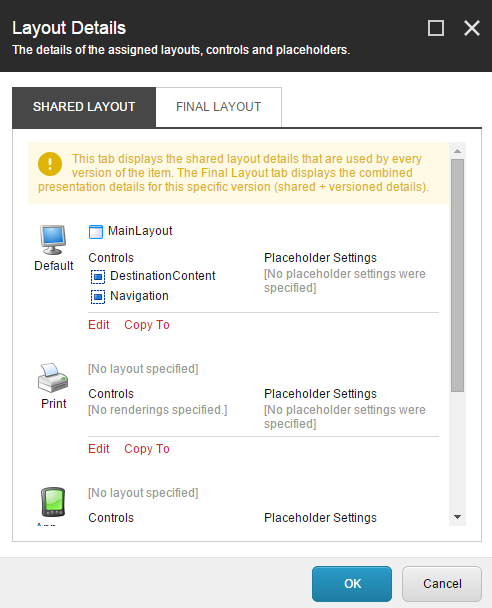


Do the same for the MainContent rendering, but add this one to the Main placeholder.

The added controls should now look like the following:

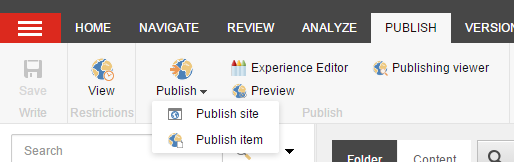


For the Destination template's standard values file, add the DestinationContent rendering to the Main placeholder and the Navigation rendering to the TopNavigation placeholder. It should now look like the following:

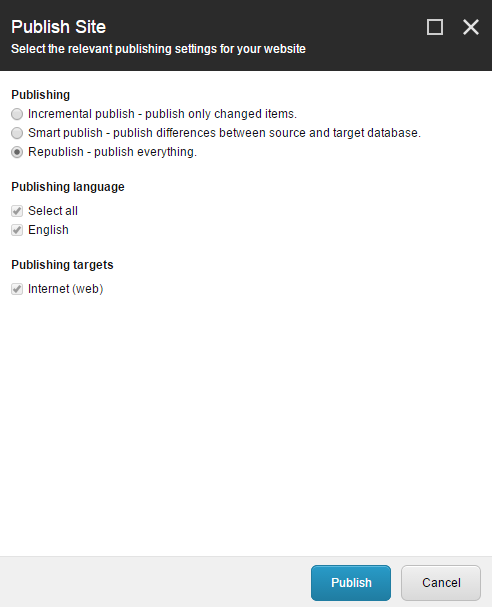


## Publishing the Sitecore site

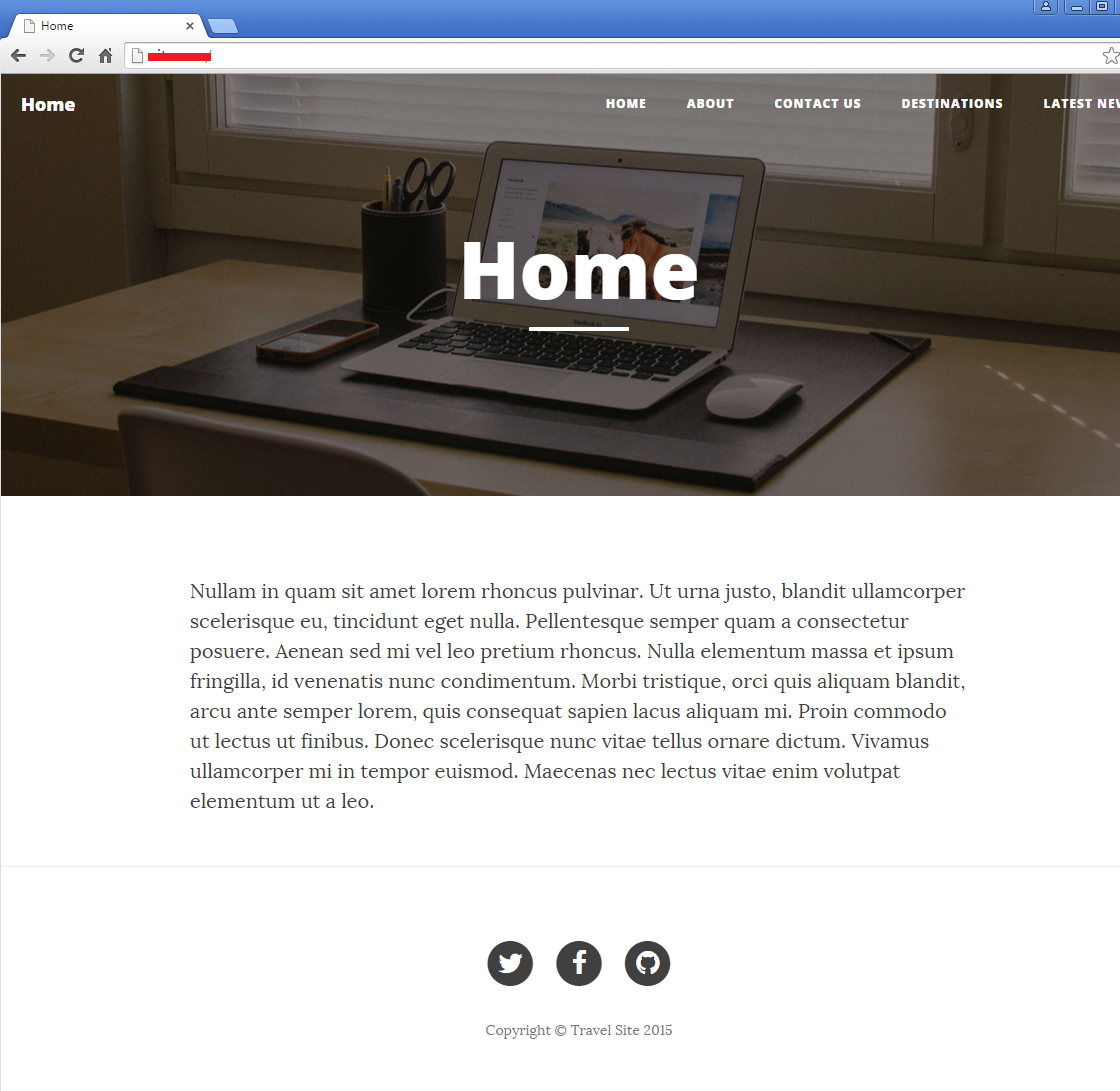
Now that the items have been created, they have a layout along with multiple sublayouts the Sitecore site can be published (please also ensure the visual studio solution has also been published). From the top menu select Publish > Publish and Publish Site.



As this is our first publish we will select "Republish everything" to ensure nothing it missed. Normally a smart publish would suffice as Sitecore picks up the changed items well. Click the publish button to publish the site (this may take several minutes).



Once complete your site should load in the browser and this shows:



Notice that the navigation is showing due to the Navigation rendering component, the content is showing due to the MainContent rendering and the page title (and browser title) is set due to the MainLayout.

The about page appears as follows:

