Fall 2013

Small Business Template

Michael Totleben
Governors State University

Follow this and additional works at: http://opus.govst.edu/capstones

Part of the Computer Sciences Commons

SMALL BUSINESS TEMPLATE

By

Michael Totleben

B.S., Illinois State University 2010

PROJECT

Submitted in partial fulfillment of the requirements

For the Degree Master of Computer Science,
With a Major in Computer Science

Governors State University
University Park, IL 60484

2013

Table of Contents

Acknowledgement:

Abstract:

This document contains instructions on how to create a template for a small business website using a content management system called Orchard, ecommerce software called Magelia, all running on a Windows Azure cloud instance (or locally/other). Since Orchard is not commonly used, documentation

for its setup, which is a custom configuration, is put in a step-by- step manner. Many times the standard is first show how to perform the task first view example then just give the technically spec for further operations of the same type. The documentation for Magelia, however, is already fabricated extensively and will be linked to when needed. The overall object of this documentation is to create and implement a dynamic, easy-to-use site for a small business.

Project Requirements:

The idea is to create a website that uses Orchard 1.7 in an ASP.NET environment. It will use web hosting and be online via Windows Azure cloud. The idea is to create a site that would fit the needs of a typical web-based small business that once configured would be easy to use. The site will have:

- Users and User Groups
- Security
- A Forum
- Host Videos
- Host Pictures
- Have an MSSQL Database backend
- A store

In the initial requirements a master page and email support was listed. However, due to the fact that Orchard does not use a master page (it fulfills a similar function with "layers" to be described later) and Windows Azure does not support email hosting so email hosting was removed.

1. What is Orchard?

Orchard is an ASP.NET based open source content management system. A content management system is a tool used to create websites using primarily a GUI and often implements modules to expand or improve its core functionality. Orchard itself is a series of ASP.NET MVC projects combined together that use a database with a schema defined by migrations. This means the database is not strictly hierarchal or has a set schema since each module can add its own tables to the database. This also means virtually all content items are stored in the database (except media files) meaning pages are not actually web pages but a series items referenced in the database. NOTE: That Orchard is constantly upgrading and changing its source code, so always make sure modules or functions are compatible with the next upgrade is compatible with the currently implemented modules.

1.1 Orchard Terminology:

It is important to understand the basic terminology when dealing with a CMS. An Orchard site is essentially a collection of content parts working together to form the UI, run queries, show images, create content etc. Here are some general terms commonly used in Orchard.

Content Item:

A content item is an individual object in orchard. An example of this would be a page.

Content Field:

A content field is information that can be added to a content type. Such examples would include a taxonomy field, creation date or tags.

Content Part:

A content part is an individual piece of a content Item. An example of this would be a text field or comment area on a page.

Content Type:

A content type is the classification of the object. A content item is an instance of content type.

Index:

 The index in Orchard is a site-wide catalog of information of every content item type marked for indexing. It updates periodically so new content that has a content type that is set to searchable in the index can be searched.

Layer:

A Layer is a set of widgets that apply to a particular page or series of pages. A layer for image gallery pages would be set to all pages named under the image gallery standard url.

Menu:

 A menu is a panel of buttons used for navigation. There are two types of menus primarily used in this site; static menus in which content items are assigned manually or dynamic menus which are populated by an injected taxonomy.

Module:

A module is essentially an ASP.NET MVC 3 project that typically contains a certain directed function. Orchard is made up from a series of these, particularly the core modules. Modules can be added on for additional functionality. For instance, a module having to do with Open Authentication provide the code needed to perform logins with twitter or Facebook with little coding required.

Taxonomy:

A Taxonomy is a classification system typical used in this application for dynamic menu types. A site typically has one active taxonomy at a time and can be used for structured queries and general organization.

Theme:

A theme is the sites overall layout that defines the locations of the zones which in turn widgets can be placed.

Widget:

A widget is something that can be added to the page or equivalent. Examples of widgets are menus and search fields. Widgets can be placed in zones on particular layer or multiple layers depending upon the layer rules.

Zone:

A zone is a particularly spot in a theme that a widget can be placed. Zones are the same on every layer of a theme but anything occupying a zone only appears on its assigned layer.

2. What is Magelia Web Store?

Magelia Web Store is a self-contained web store platform useful for managing inventory, orders and products. It is normally can be used standalone but in this case it will be the backend for store management and will be connected to Orchard which will act as the display end and which adds additional functionality that a business site might use.

3. What is Windows Azure?

Windows Azure is cloud based software as a service that is excellent for handling the needs of almost any size business. It has a wide range of services ranging from basic web hosting, creating a virtual private network in the cloud, database hosting, virtual machines, binary large object storage, Active Directory functionality and more. One of its biggest advantages is it is accessible via a web that simplifies deployment and so a user typically only has to deal with one main interface to see and monitor their used services. It also has tools for building web sites with windows web matrix and MSSQL server connectivity making working on the cloud relatively comfortable as developing locally. It can setup and link large portions of these services with one click helping ease the usually many of the externalities of deployment. It also has the ability to increase usage on demand. So instead of having to buy more hardware when demand increases a user can simple add a service at will have it functional relatively quickly.

4. Setup:

This portion of the document explains how to setup with two options for the template site. The first one requires Windows Azure, Orchard v 1.7.1 and Magalia Web Store version 2.4 or higher. The second option is a local setup which has similar requirements but without the Azure aspect and uses the portions of the azure section to supplement it.

4.1Windows Azure Setup:

The first thing that needs to be done is to create a Windows Azure account. A thirty day free trial is available for those wishing to test it.

Once you are logged in you will see the primary dashboard:

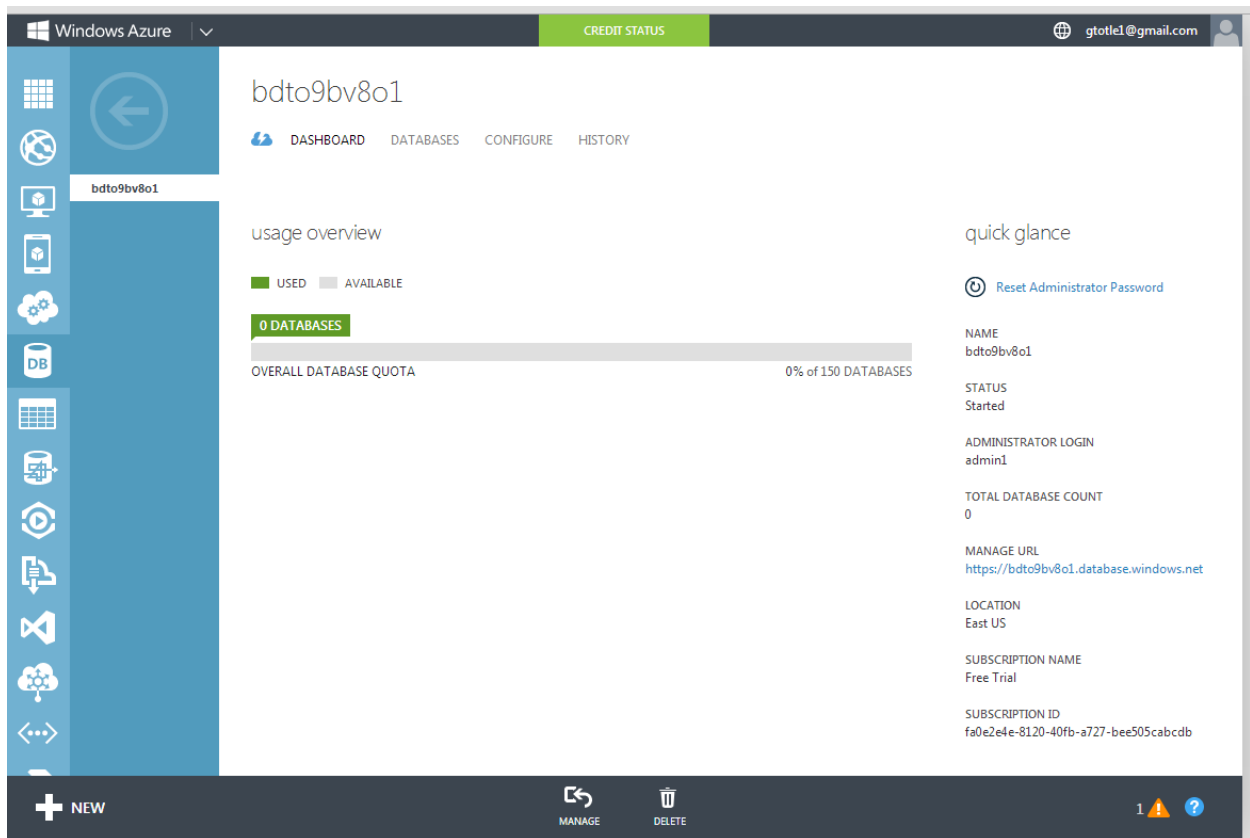
4.1.1 Database Servers:

First setup the database servers and then the MSSQL Azure databases. Two database servers are recommended for the business template site; one for orchard and one for Magelia Web Store. Note that this setup here must be done for each the Orchard and Magelia sites preferable with each site having its own database server and database. The created databases will be blank due the fact that Orchard and Magelia use Migrations to populate their databases on their instantiation.

First, go to The SQL database tab on the right.

Then in the top center there will be a "servers" tab. Click it and create new.
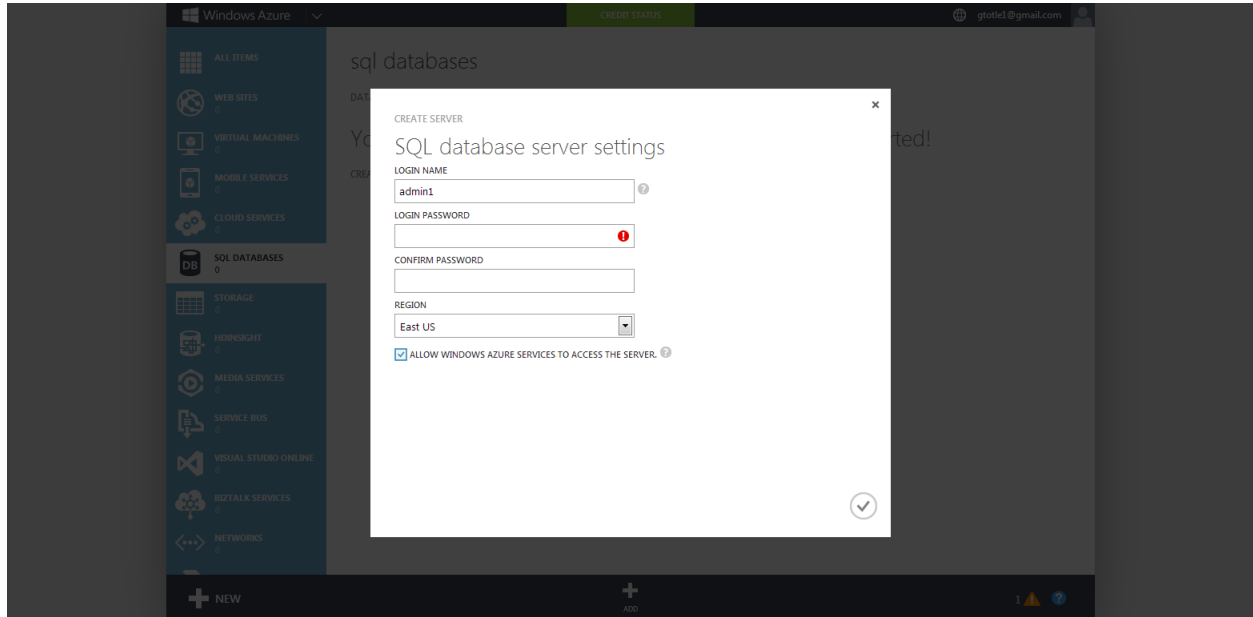
Make sure that the box to allow to Azure services to access the server is checked. The servers will be linked to the websites later. For the purpose of this document, the user name for Azure services will be Admin1. Make sure to store your password info although it is changeable via the Azure dashboard. Also, choose the region closets to what represents your location.

Once the server is created, notice that it is selectable in the sub tab. Above in the top menu choose databases and create. Under "Settings" choose "Web" for database type which has a limit of 1 gig. The name for this database will be Orchard or Magelia, depending   on for which site you are setting up the database.
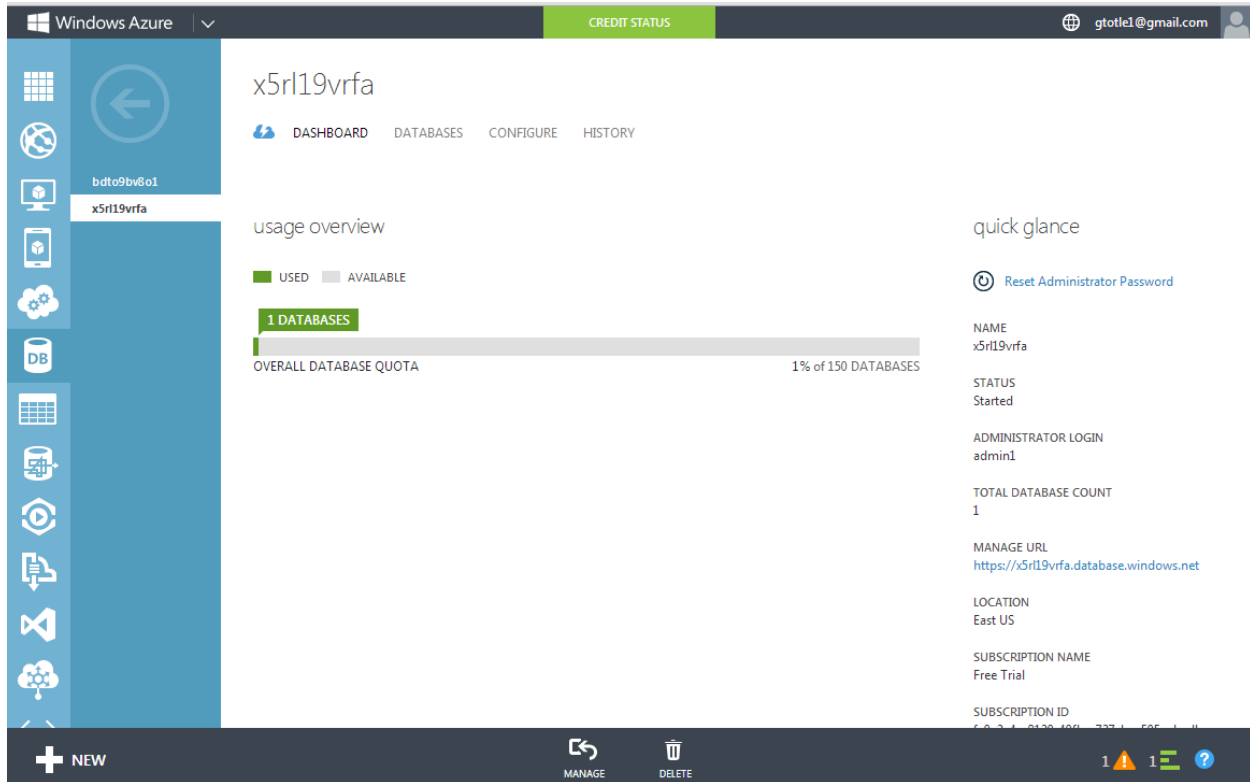
When finished there should be two database servers each with a corresponding empty database.

4.1.2 Website Setup:

There are 2 websites that need to be setup; the Orchard website and the Magelia dashboard. The Orchard website is the front end of the application that will be viewed by both the administrator and the normal users while the Magelia dashboard is the backend site only to be viewed by the administrator.

4.1.3.1 Orchard Setup:

There are two ways to install orchard. The first is through the App Gallery in Windows Azure. The second is to download the zip file from https://orchard.codeplex.com/releases/view/115110. 1.7.1 is preferred. NOTE: If the App Gallery version is 1.8 or above it will not work due to the shift in code. So the user will have will have to download the zip file from the link above and migrate the files manually. If that must be done the first steps and Magelia installation will show how to do it.
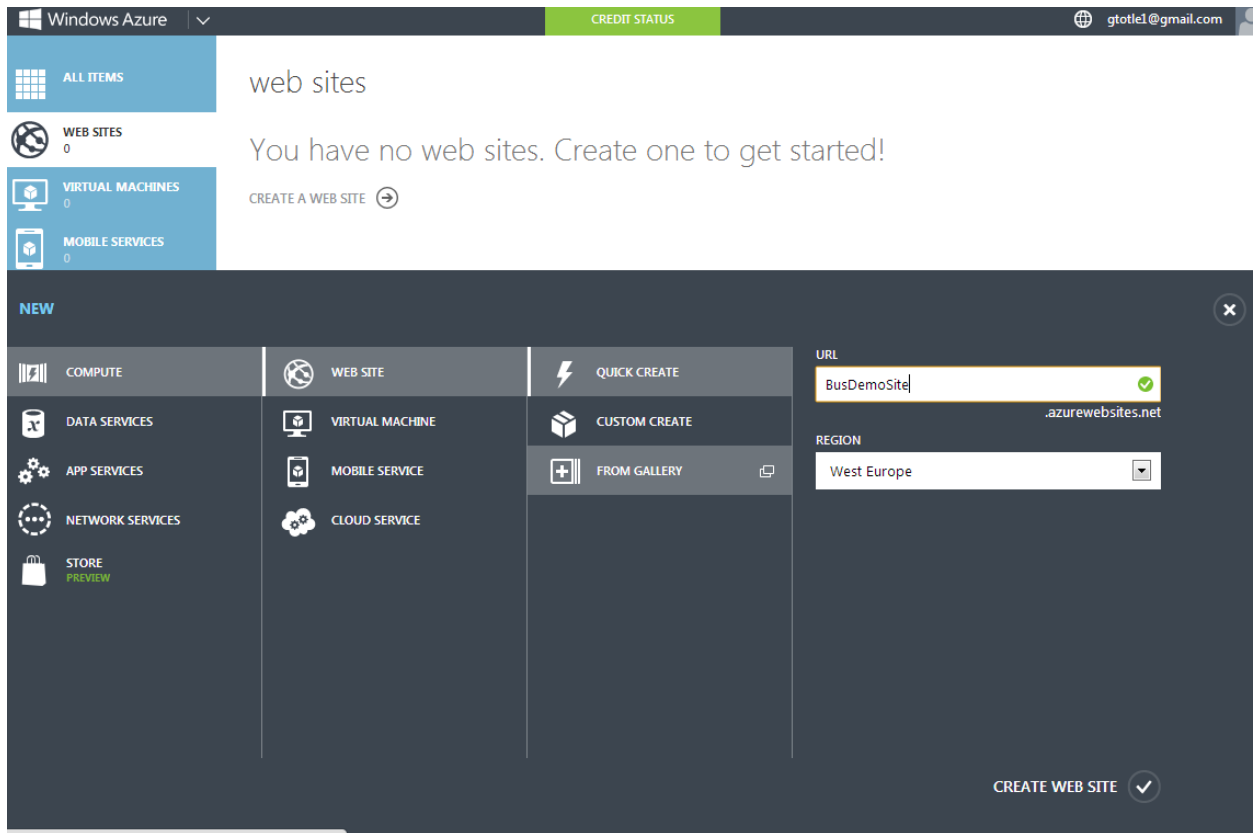
For this example the name of the site will be BusDemoSite. Each site in the Azure Cloud has to have a unique URL so you will have to use a different one. Also the make sure the region is the same as your database servers, i.e. East USA in this example.
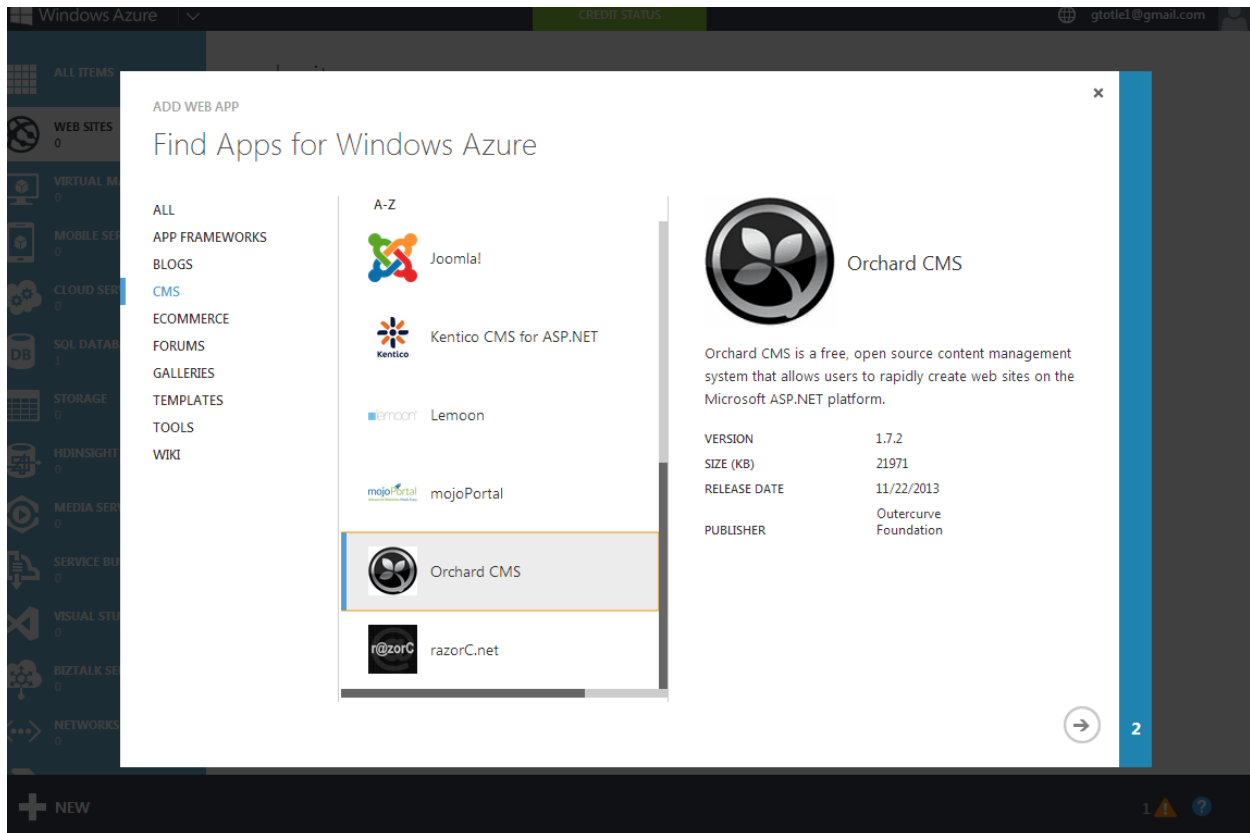
Select Orchard under the CMS menu and hit the left arrow to continue. It may ask you again to define the name of the site, url and region. Just re-enter the data you put in the previous step if this is necessary.

If you need to download the source code because orchard is no longer in 1.7.1 refer to steps in the Magelia site setup in the next section then until the site is launched, then refer back to here.

Next, click on the website that was just created and in the above menu click "linked resources". Then click the arrow button to link a resource.
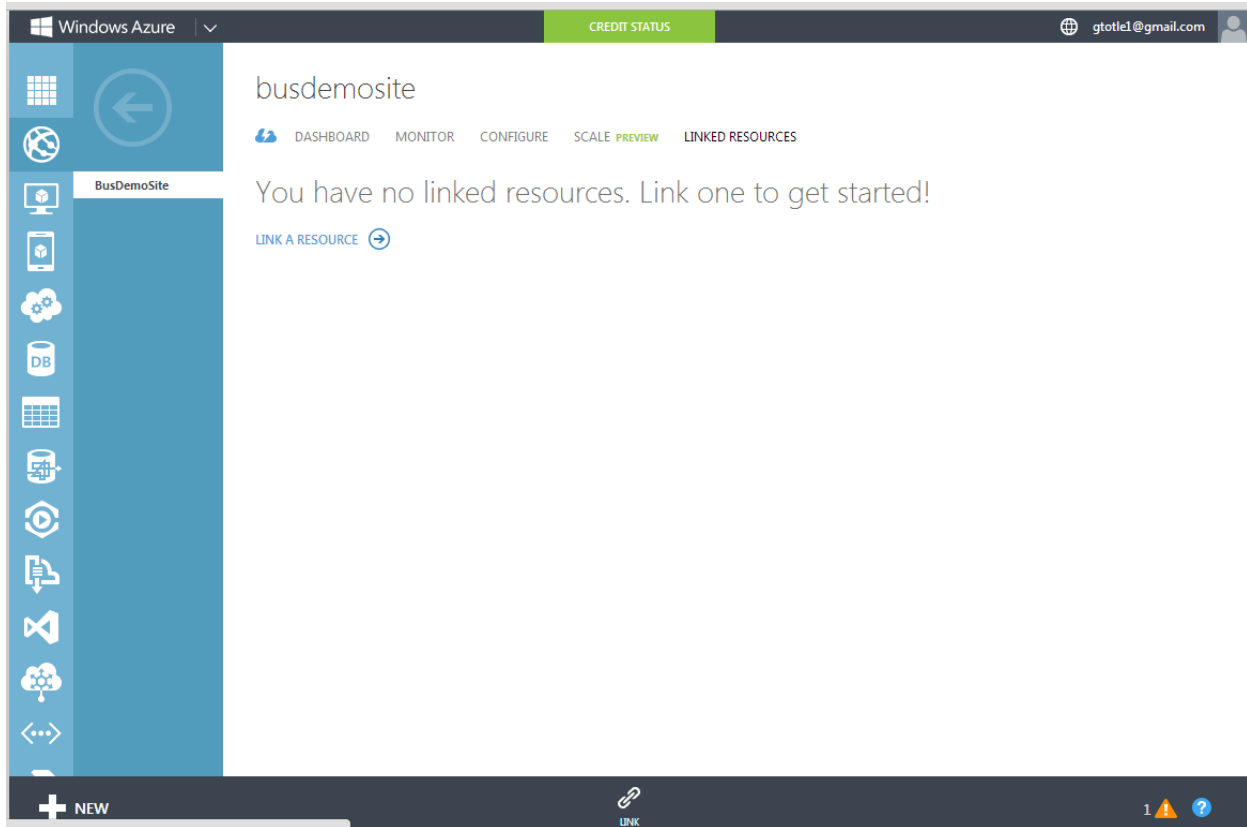
Once that has been clicked a dialogue box with appear asking to link which resources. Choose existing resources, SQL Server, then choose the Orchard Database from the server created for this purpose.(or Magelia if you are doing this for the Magelia site)
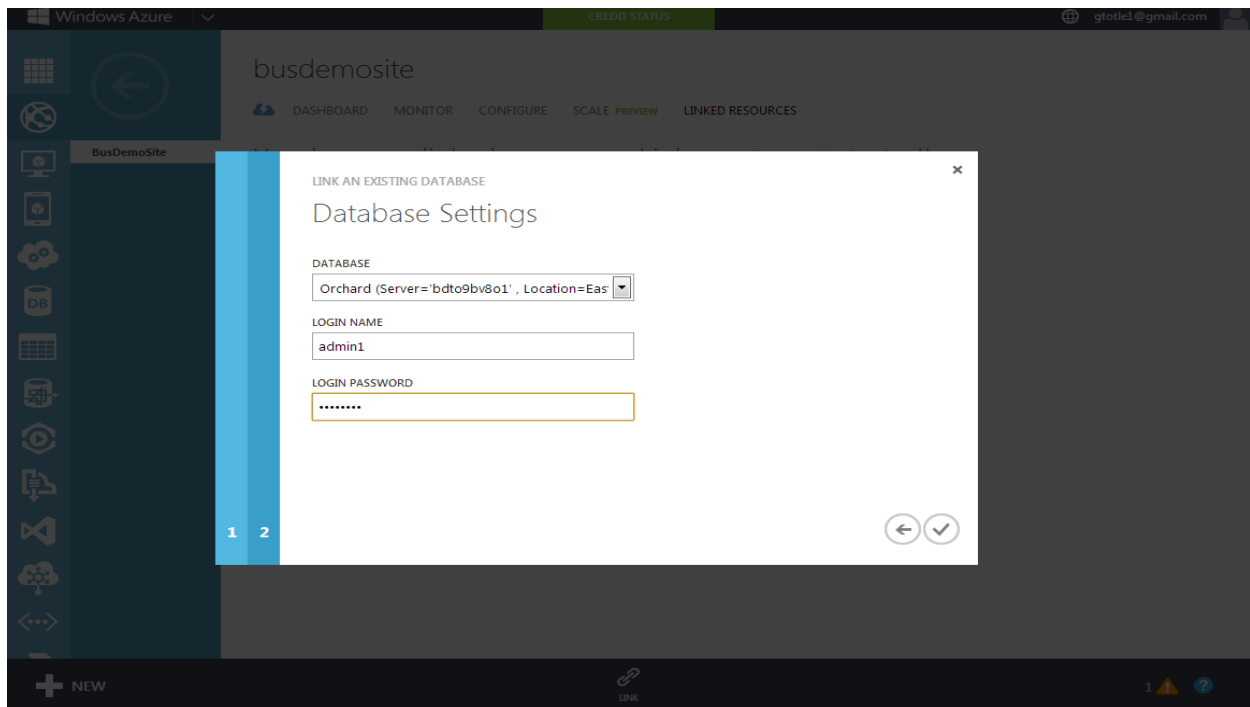
Then it should show that the resources have been successfully linked. Click over to the dashboard page via the top menu. Note that on the next screen you will need a connection string to the appropriate database which can be found one the dashboard on the bottom right under "quick glance" connection strings. Finally, hit browse at the bottom of the screen to launch the web browser and the initial site setup.
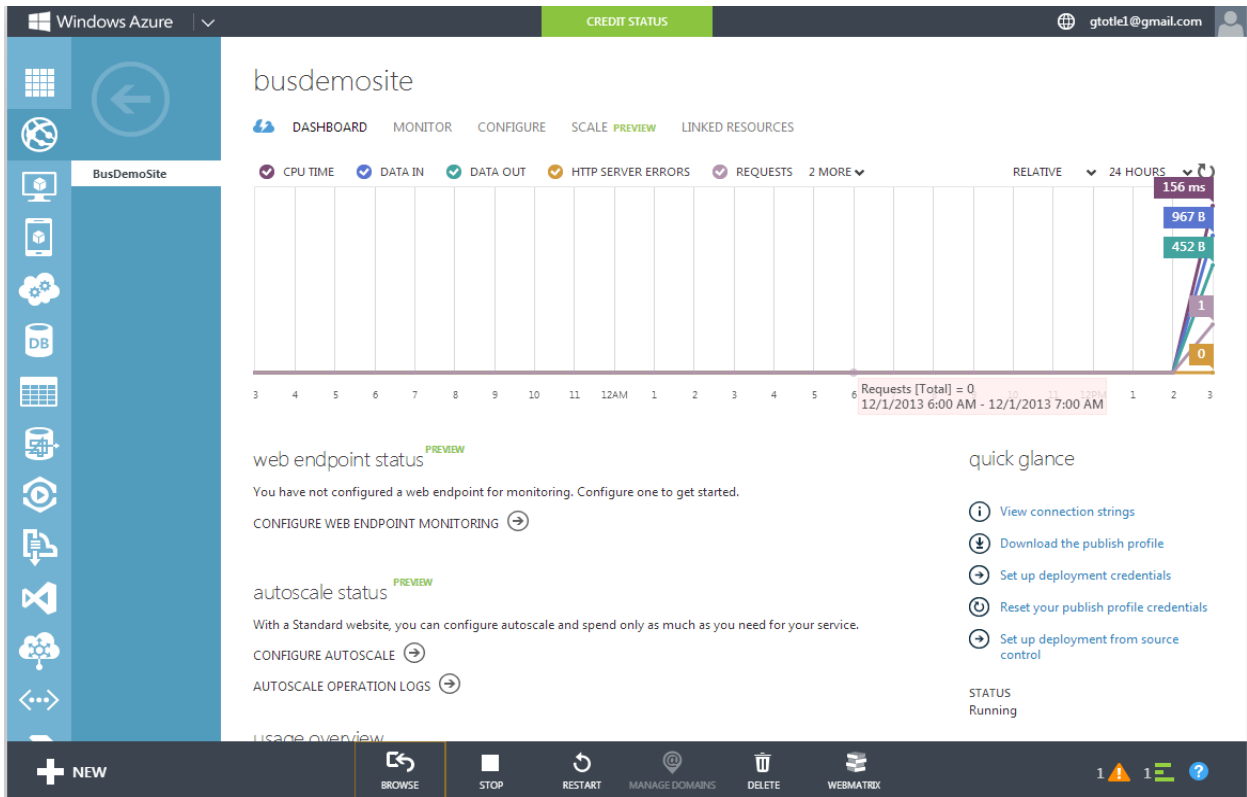
Once the web browser has been launched there will be a few fields to fill in:

Get Started

Please answer a few questions to configure your site.

What is the name of your site?

Business Demo Site

Choose a user name:

admin

Choose a password:

••••••••••

Confirm the password:

••••••••••

How would you like to store your data?

○ Use built-in data storage (SQL Server Compact)

● Use an existing SQL Server, SQL Express database

○ Use an existing MySql database

Connection string

Data Source=tcp:bdto9bv8o1.database.windows.net,1433;Initial Catalog=Orchard;User ID=adm

Data Source=sqlServerName;Initial Catalog=dbName;Persist Security Info=True;User ID=userName;Password=password

The site can be named anything, same with the user default name and password. However," use an existing SQL server" must be selected and the connection string can be found on the dashboard and pasted in the field below. No table prefixes are required and the Orchard default recipe should be selected. (Lower on the page outside the screen shot). Then click "Finish Setup".

You should end up with up with a page similar to the one below, this concludes the orchard portion of the Azure setup. For calibrating orchard itself, see the Configuring Orchard section.
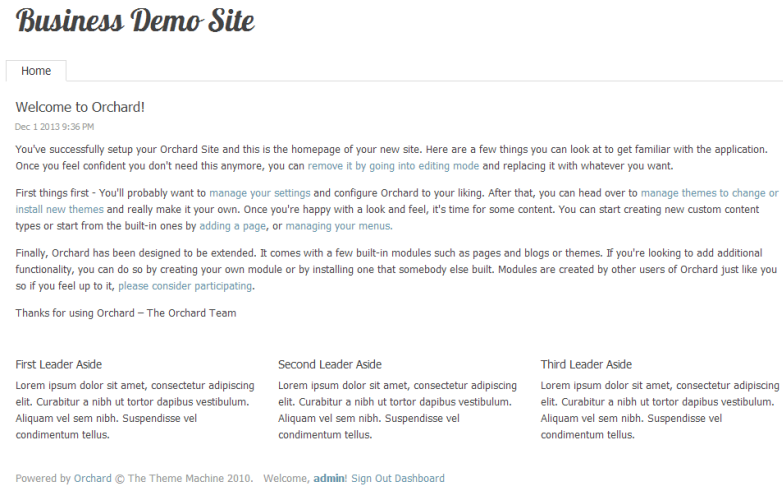
4.1.3.2 Magelia Webstore Setup:

This portion of the document is for setting up the Magelia Dashboard backend that will be viewed by the administrator to manage the commerce portion of the site. If no commerce portion is required than this step can be skipped. Make sure Web Matrix is downloaded because it will be specifically required for this step.

First download the Magelia source code from http://www.magelia.org/store/license.aspx?edition=community

Then create a new site for the Magelia dashboard. In this instance its MageliaDemo.

The fields required are a bit different to be filled in. Here is an example of what it should look like, notice the database name and the server as well the location. The admin name and password for the database server are also required.
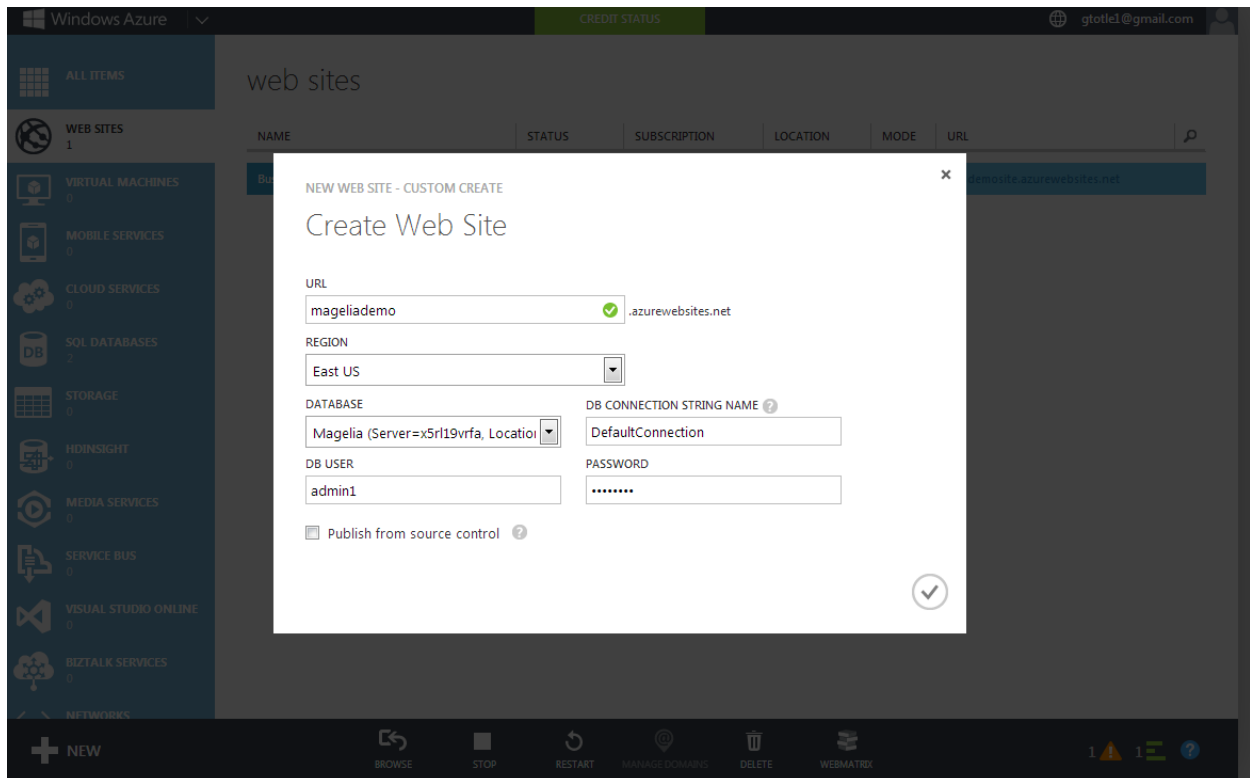
Once the site is created, select it and launch web matrix found at the bottom of the screen. This is what should come up.
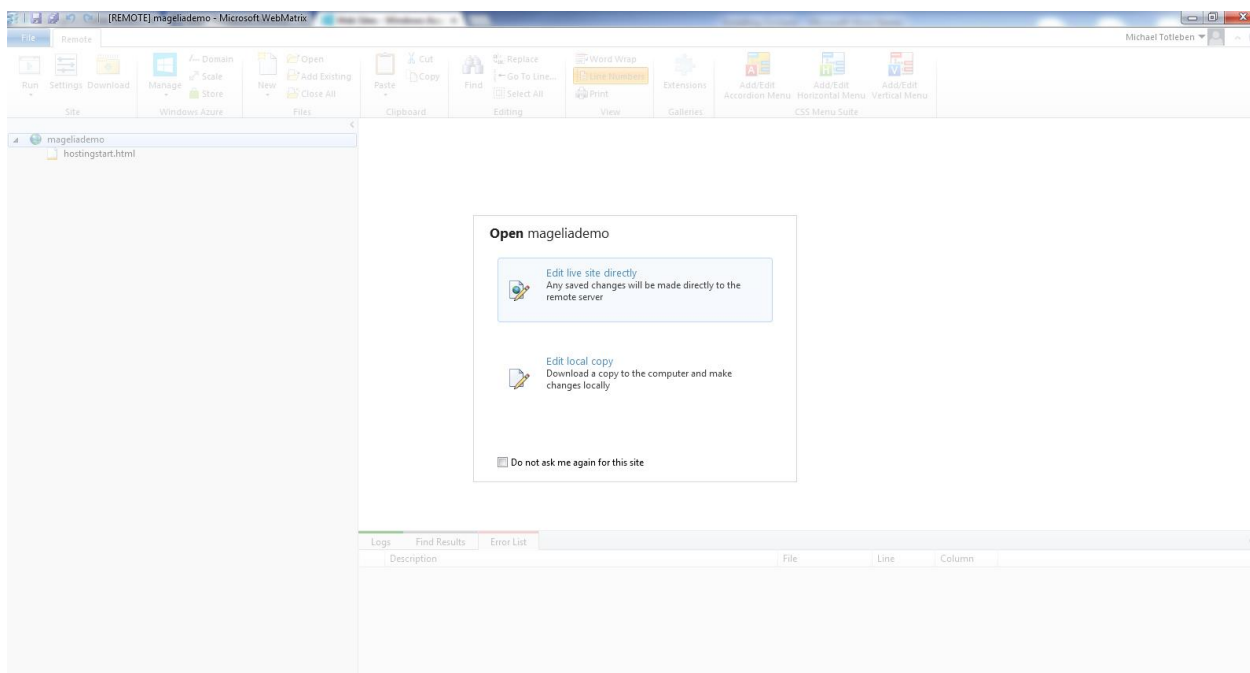
Choose Edit live site directly and delete' hostingstart.html'

Find the Magelia zip folder and extract it to easy to find location. Then select copy all the files and folders within the Magelia folder (but not the actual folder itself) and drag and drop them on the world shaped mageliademo.(or whatever your sites name is) This should take a few minutes to perhaps even a half hour or more depending on the connection being used. The end result should look like this:
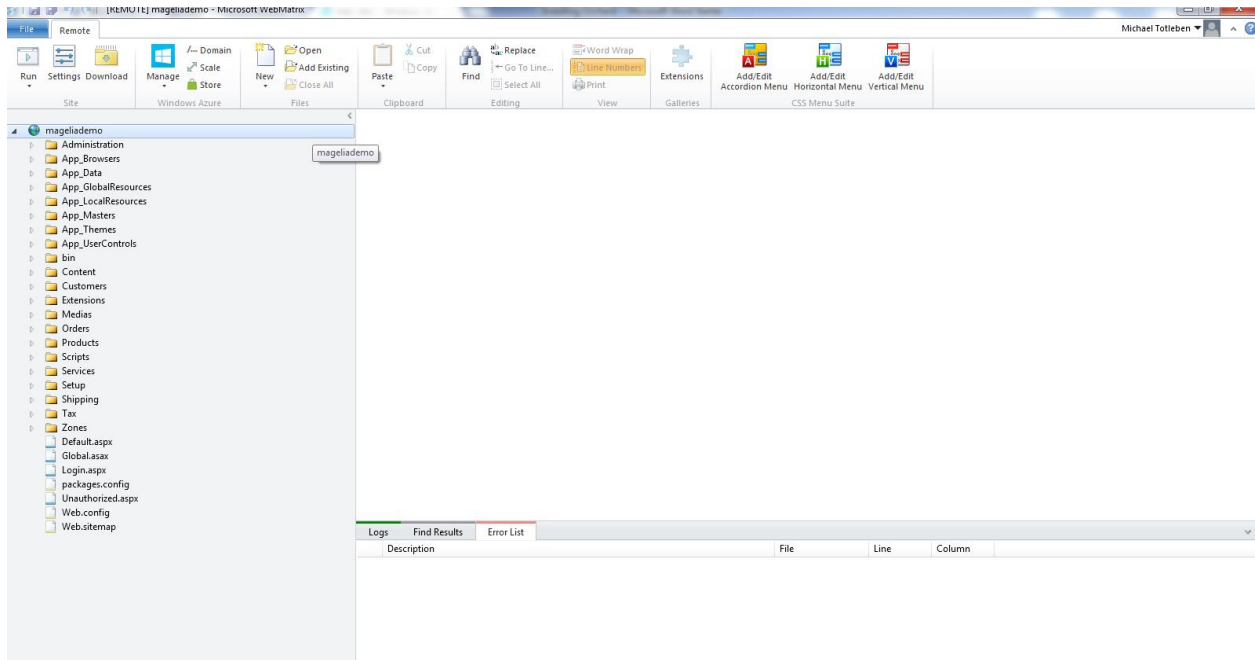
Then click run in the top left hand side of web matrix. This may take a few minutes to initially load.

A Screen should come up with the Magelia Logo in which it should show three options; choose Community edition(which is free) on the left side of the page.
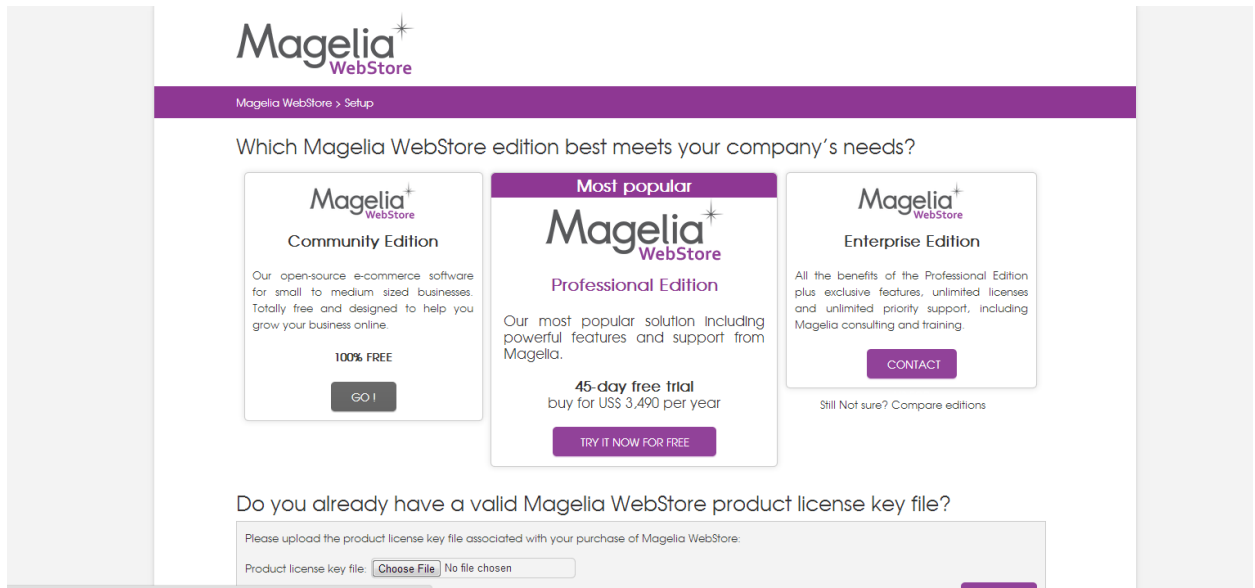
The next screen you will be prompted to create a Magelia account which is free. Do so and then login or just login if you already had made an account in the past. Remember to store your password.

Next the store configuration must be filled out. The customers will not be able to see company name or store name but it is good to name them anyway for your own reference. Under store parameters chose "Sample Catalogs and parameters" to get basic warehouse and shipping data already configured. NOTE: that in this demo sample customers and orders will be checked for demostration purposes. Then hit next.

The next screen is merely asks for your email and password along with an adminstrator name default of which is admin. Enter your password(make sure to keep it somewhere so you wont forget) and email you would want Magelia to send messages to, ie a business email.

On the next screen the database must be formally connected to the site application side. Make sure to uncheck sql express auto attack database and to select "use an existing blank database". If you named your blank Magelia database Magelia fill that in for the database name. Since the database was not configured the Database adminstration credentials and the database access credentails should be the

same by default(both for look almost the same, second form is truncated by screen shot.)

Database settings

Database 'Magelia' exists.

SQL Server name

Please enter the name of the SQL Server where you would like to install WebStore.

SQL Server name or IP: *   x5rl19vrfa.database.windows.net,1433

☐ Use SQL Express auto-attach database

Existing or new database

○ Create a new database
◉ Use an existing blank database (new installation)
○ Upgrade or migrate an existing database

Database name: *   Magelia

Database administration credentials

○ Windows authentication mode
◉ SQL Server account

Username: *   admin1

Password: *   ••••••••

TEST DATABASE CONNECTION

Database access credentials

Indicate the account that will be used by Magelia WebStore to access the database. If the account does not exist, it will be created.

Quick & easy = check SQL Express auto-attach database

For quick and easy installation, simply check the box "SQL Express auto-attach database." A new database will be created in the App_Data folder.

Database administration credentials

Specify the account that will be used to create the database. If you wish to create a new database, your account must be enabled to do so. If you choose to use an existing blank database, your account must be the owner of this database.

The log-in information used will not be stored in the system once the setup process is complete.

Database access credentials

Specify the account that will be used by Magelia WebStore to access the database. If you choose Windows authentication mode, the user currently running the setup (*IIS APPPOOL\mageliademo*) will be given the right to access the Magelia WebStore database. If you choose SQL Server account mode, a new SQL account will be created if it does not already exist with

To get the SQL Server simply go back to the Magelia Database under the database tab and toward the bottom . Its advisable on the magelia screen to test the database connection to make sure it is working correctly. If the passwords or access names are incorrect you can click on the server tab(as shown in the database set up) to modify this.

Once you entered all the data and hit continue setup it may take up to 30 minutes or more for the sight to build depending on the connection but when it is finished the view of the dashboard should be simialar to this:

4.2 Local Setup:

To setup the site locally one first needs to download Windows Web matrix from
http://www.microsoft.com/web/webmatrix/. You can download orchard via the app gallery in Web
Matrix or download the source code at from https://orchard.codeplex.com/releases/view/115110.
NOTE: Gallery version is 1.8 or above it will not work due to the shift in code. Parts of this installation are
very similar to the Azure installation neglecting the Azure linking database to site. Then, download the
Magelia source code from http://www.magelia.org/store/license.aspx?edition=community and drag and
drop the top (but not the zip folder) into Web Matrix.

4.2.1 Database Servers:

Depending upon the need of the user during the Orchard and Magalia setup you have a choice of
choosing a compact server or MSSQL Server depending on the needs of the user(as shown on page 14
for Orchard and page 20 Magelia will work.) The rest disregarding the Azure references will show the
appropriate choices for installation. For the MSSQL server merely create empty databases named
Orchard and Magalia respectively and use the connection strings for those similar to the azure setup
above.

5. Configuring Orchard:

This section shows a user how to configure Orchard from a fresh install as described earlier in the
documentation.
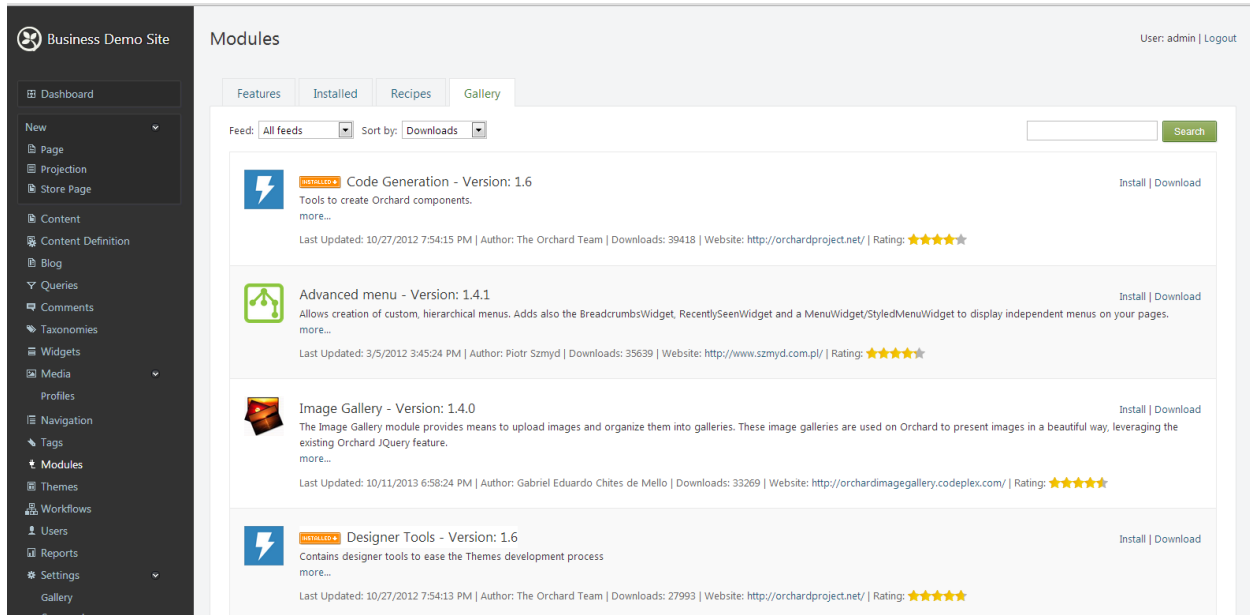
First, it is best to view the dashboard; from there most tasks will begin.

5.1 Installing a Module:

To install a module the primary method is downloading it from the Orchard gallery.
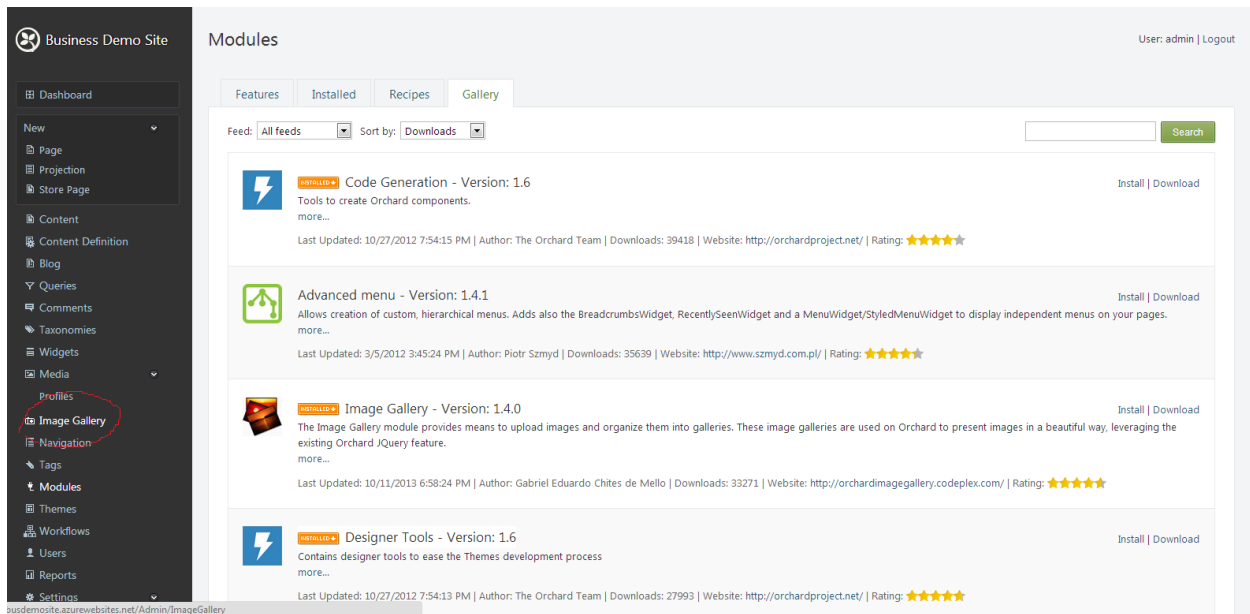
To access the Orchard gallery, click modules on the dashboard and then the tab called "gallery".
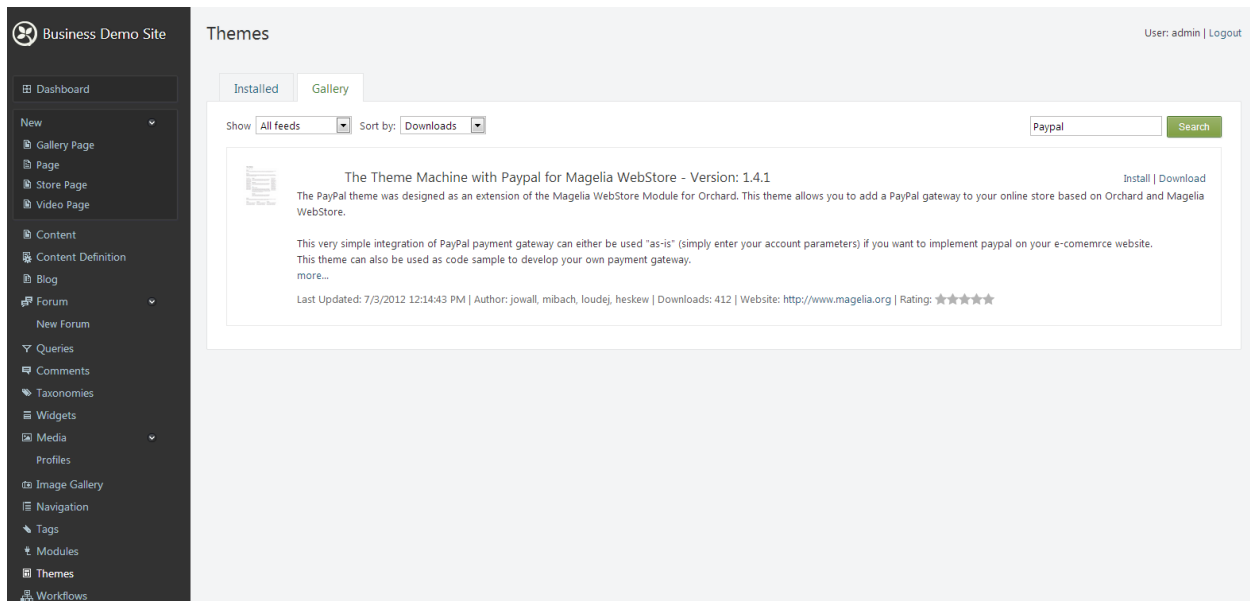

This displays a list of modules for download which can be searched via search field and filters. Click install on the Image gallery module. Then you should see a green check box asking if you want to enable the module. Click yes. Then you will notice that in the dashboard menu Image Gallery has appeared.

5.2Choosing a Theme:

Themes are the overall layout that define zones in which widgets will be placed. By default the theme is "Theme machine". Click on the dashboard button "themes" and the hit the tab "gallery". This installation will require the Magelia Paypal theme for Orchard so paypal can work. When this theme is found, click install.

Then going back to the themes menu you will see the themes listed in which you can set one as the current theme. Themes are also where you can customize the overall CSS of the website via the .css file.

5.3 Preparing Taxonomies:

In orchard taxonomies are used to organize content items and can be used to populate dynamic menus as well be used for organization in queries.

5.3.1 Setting up the Basic Taxonomy:

To create a taxonomy click on the taxonomies button in the menu and enter a title and hit save. The permalink will self-generate.
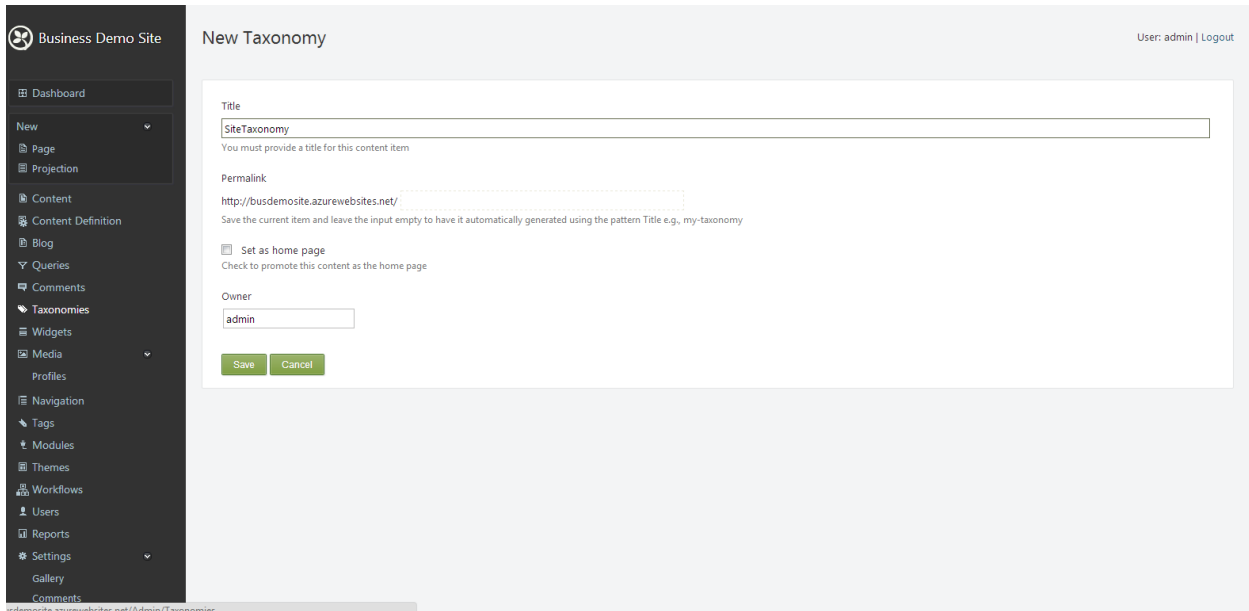
Once that has completed, a bar with the taxonomy name on it will appear, click edit and on the next screen click add term. This site will have three primary terms for taxonomies and as many sub terms as the end user desires. In this case, a root term is being created. Having a root term is not required unless you want to create a taxonomy menu that includes all sub menus.
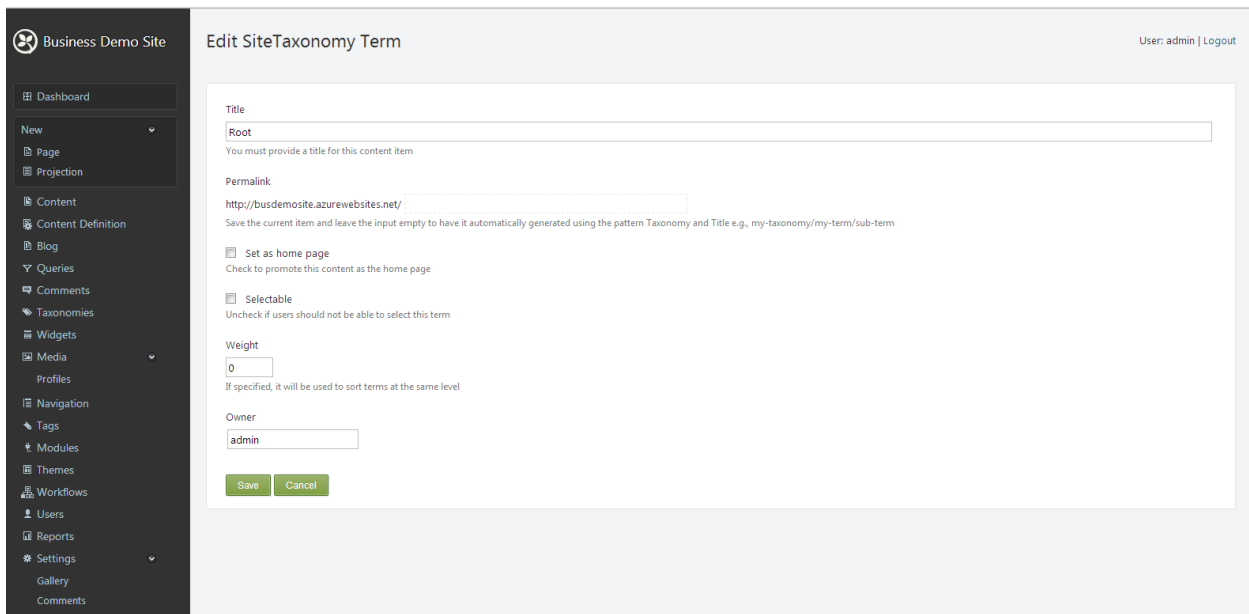
When creating a sub term that can be selected choose the appropriate parent term to create a proper hierarchy. An example is shown below:
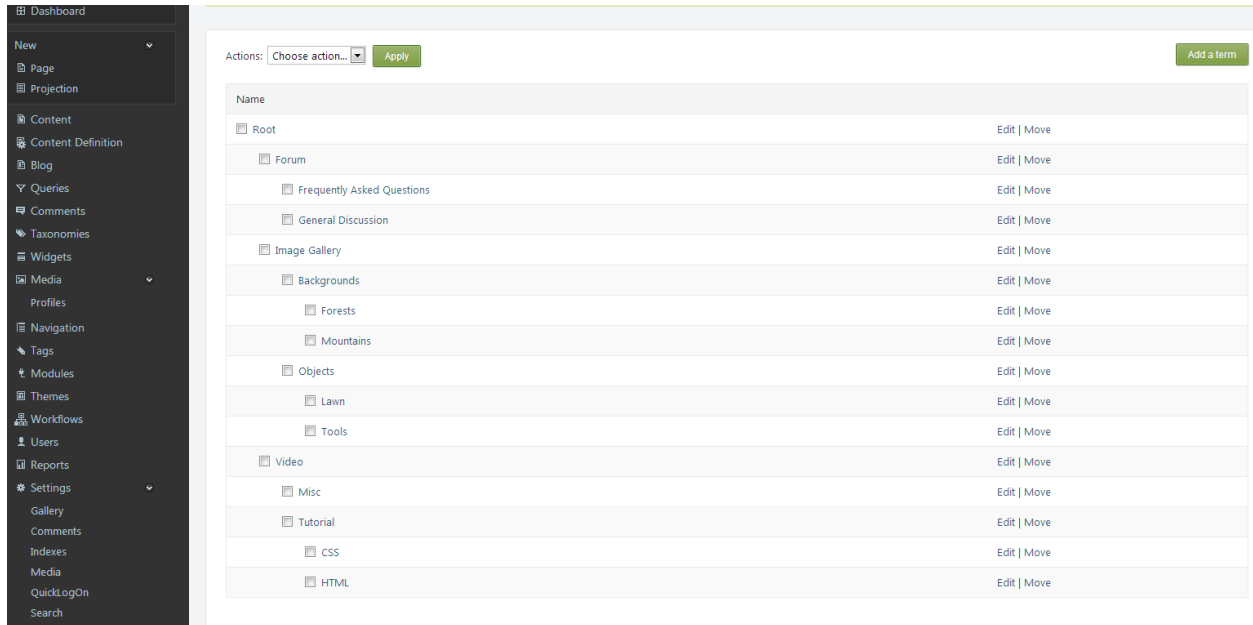
You will be able to add content to taxonomies as it is created.

5.3.2 Setting up a Taxonomy Menu:

A taxonomy menu is a menu injected with terms of taxonomy. This makes menus dynamic and update as more terms are added. To create one first, go the navigation button on the dashboard. Hit create new menu, in this case it's the gallery menu.
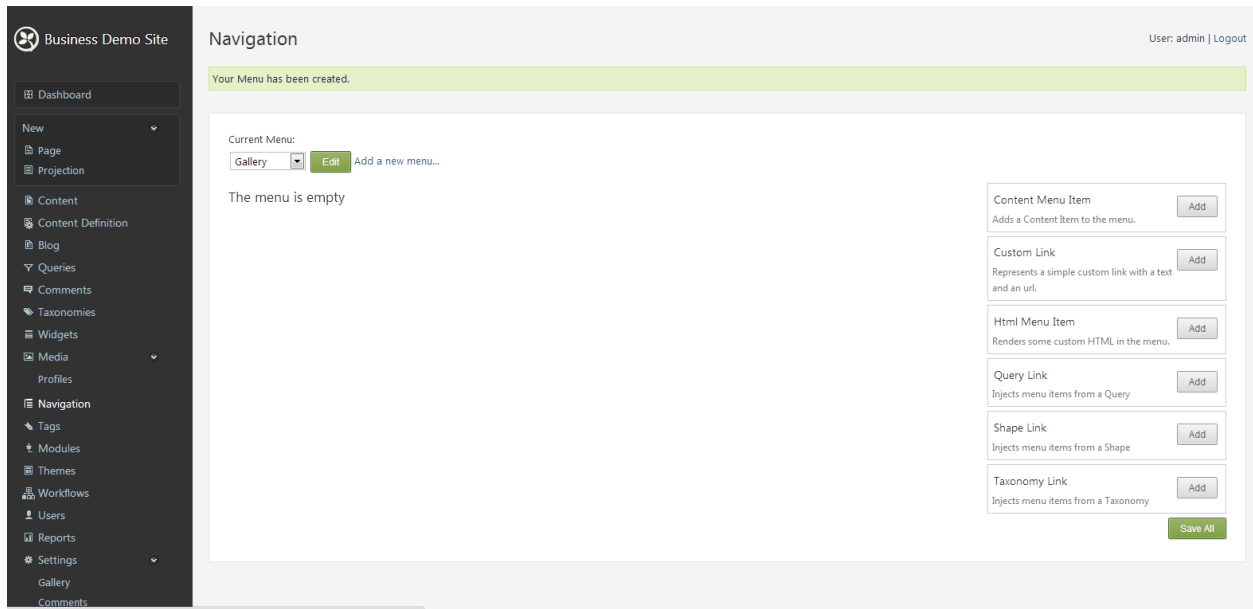
Then click "add" in the taxonomy link.
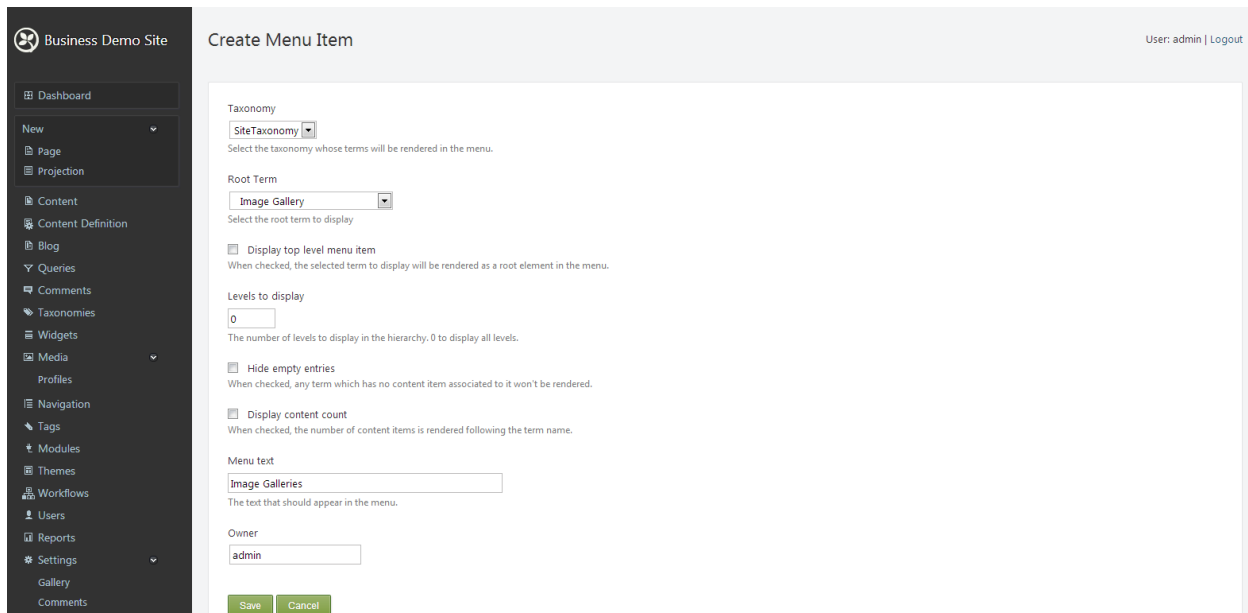
The root term is the term that the menu will be based off of; most of the time it is not necessary to display this term so leave it unchecked. The rest of the check boxes are optional to modify to the user's needs.
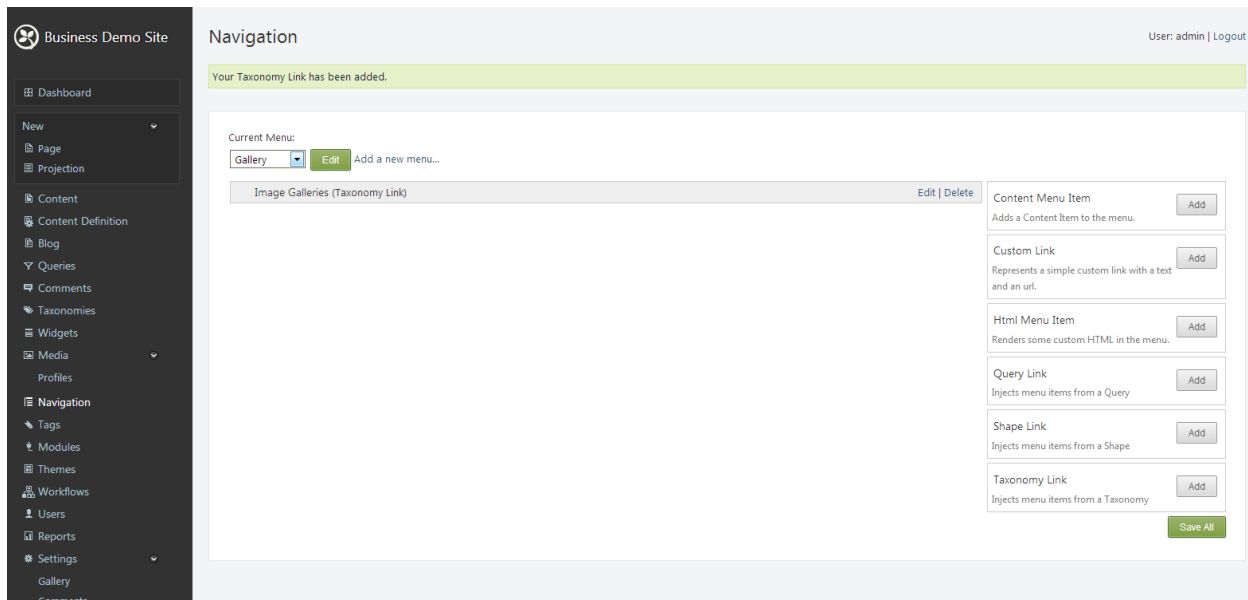
The end result should look like this. Now the menu can be added to a widget in the layer section.

5.4 Setting up Projections:

Projections are ordered lists created by queries that can be set as individual pages or embedded in widgets. Projections rely on queries to get data to populate themselves. Fields in queries can range from the content name, tags, body content or its taxonomy type.

29

5.4.1Designing Orchard Queries:

Queries in Orchard can be set up by clicking the queries tab and selecting "add new query". Then add the name of the query. The query is currently blank, to modify it click edit.

In the editing tab you can choose a filter, a sort description and a layout. In this case since image galleries are sorted first by their creation date and second by their name. The layout will be set to be undefined so it render like a list in a widget, in which the widget can override the queries defined layout. Note: that if this query is run now it will return no results because they are no content items defined in the taxonomy as a gallery. On this site these types of queries will be run as part a widget defined in the layers section.

5.5 Creating Pages:

Pages are what content in orchard is placed on via layers or direct user input into fields such as the body, tags etc. The primary pages include the Home Page (which is already created so it merely needs to be edited), the Gallery Page, the Video Page, the Forum Page and the Store Page. In order to create a page under the "new" tab on the dashboard click page:
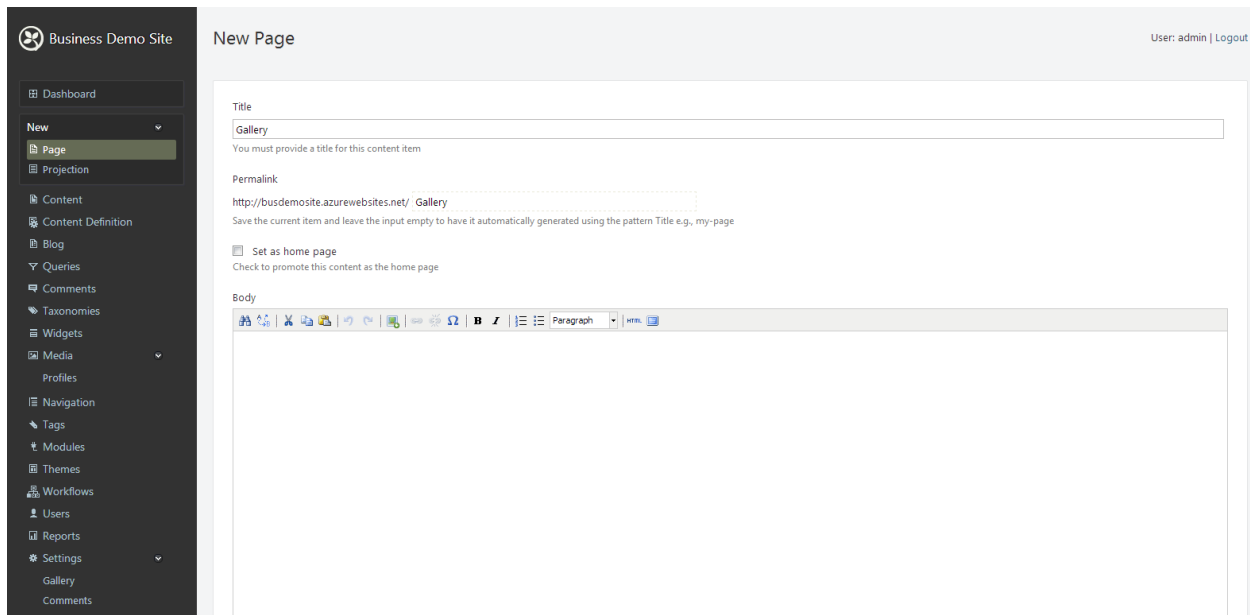

Upon scrolling down there will be a check box named "show on menu". Check yes and choose the main menu. You can both save and publish later (saving without publishing makes the page not visible to non-dashboard capable users) but for the purposes of this documentation just choose publish and do not create a layer for this page. Layers will be created in a later section.

5.5.1 The Home Page:

The Home Page is the simplest page by default seeing that its function is to be the first page viewed by a person who enters the site. It comes preloaded with text that will need to be removed. It can be customized with various things to cater to the needs end user but for purposes of this document not much is required. NOTICE: unlike other pages it does NOT have a permalink.

First, on the dashboard click content then on the left there should be an edit text and click it.

31
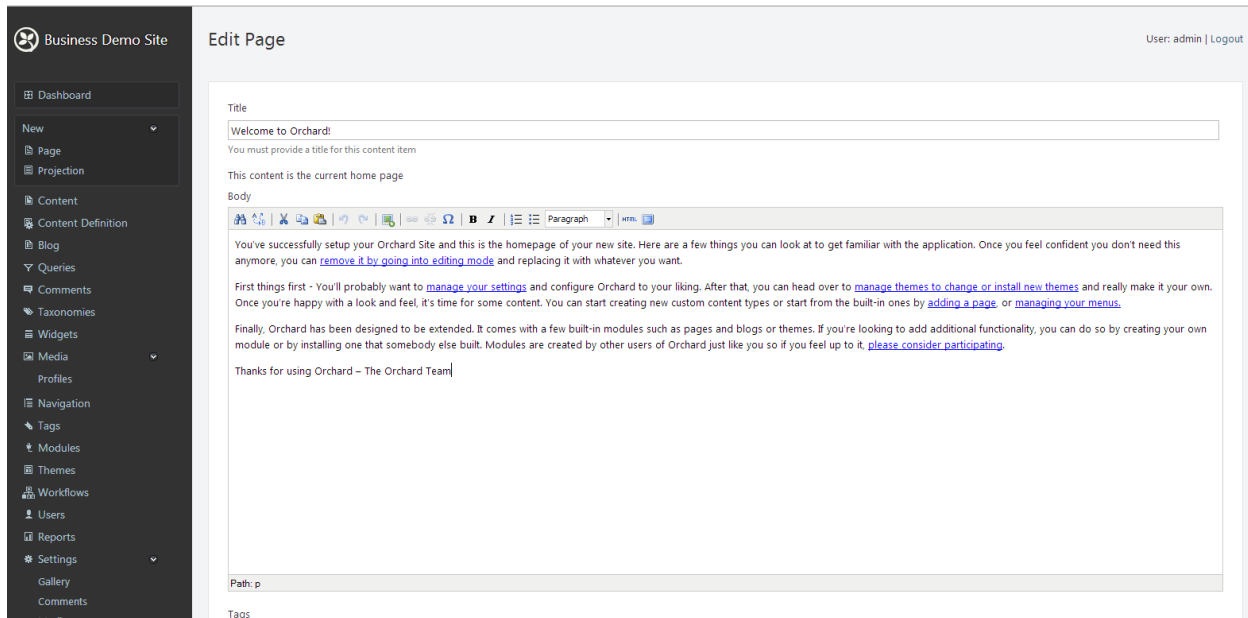
As it can be seen the body has default text and a default title. These can be erased and rewritten in this case the title is going to be Business Demo Home Page and the text is going to be a simple welcome message which can have media added later simply by clicking on the picture plus button . This scrolling down this page can also be added to a menu in which the menu that will be chosen is the default navigation menu and the name on the menu will be "Home". As with all pages you have the option to publish now or save for later and not publish it yet making the changes not visible. For our purposes publish now is the button to click.

5.5.2 The Gallery Page:
The Gallery Page is the primary page for the image gallery. It will be covered by the Gallery Main layer and will be featured on the main menu.

Title: Gallery

Permalink field: Gallery

Name on Menu: Gallery

5.5.3 The Video Page:
The Video is the primary page for videos. It will be covered by the Video Main layer and will be featured on the main menu.

Title: Videos

Permalink field: Video

Name on Menu: Videos

5.5.4 The Forum Page:

The Forum Page is the primary page for the Forums. It will be covered by the Forum Main layer and will be featured on the main menu.

Title: Forums

Permalink field: Forum

Name on Menu: Forums

5.5.5 The Store Page:

This the page on which the Magelia store will be integrated.

Title: Store

Permalink field: Store

Name on Menu: Store

5.6 Setting up the Layers:

The Layers are one of the primary factors in determining what widgets are viewed on what pages. They essentially can set the layout for a single page, set group of pages or the entire site.
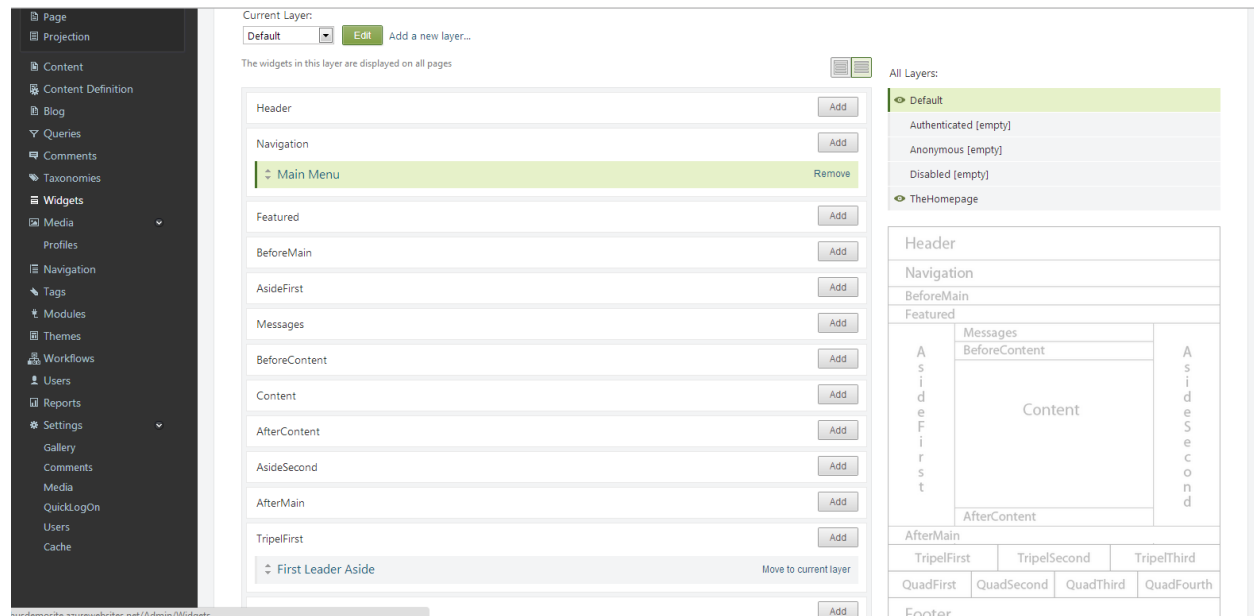
Notice the top shows which of the current layers is selected. Notice that the green bars are widgets in their corresponding zones (illustrated to right) the currently selected layer and the blue are the not selected layers. The eyeball menu on the right shows which layers are visible in this current menu. If the layer is deselected in the eye menu any widgets in the layer will not show up in the zones. To delete a widget select from the drop down above its native layer (In this case the homepage layer). First Delete the Triple first, second and third widgets (which are those 3 boxes at the bottom of the home page). Unless you want to keep them and say edit their content then just simply click on them and you would see something similar to this which shows the basic format of a typical widget. Viewing the widget

format can be useful later when dealing with other widgets.

5.6.1 Layer Rules:

Each layer has a set of rules to which it is to be viewed, particularly regarding which urls they apply to. This is an example of the format a layer rule will have. A * after the url means any sub directory is also included. Or is used if there are multiple different url types the layer governs. Note: if you did not create a root in the taxonomy do not include it in the urls where it is otherwise noted.

5.6.2The Default Layer:

The default layer is the layer is the layer visible on all pages within the site. Setting up the default layer sets the baseline for the site.

For our purposes the default layer will contain the following widgets:

The Main Menu:

A menu widget in the navigation zone will be populated manually by content items as they are created, particularly the main pages.

Search:

 A basic Search field widget to occupy the header zone that can be used to search content in the site,

Quick Logon:

The Quick Logon widget allows users to login as an authenticated user using a Facebook, Google or Twitter account.

Magelia Shopping cart widget:

This widget should be placed in second aside and represents a count of how many items are in the cart if any. This particularly widget configuration will not be covered in this section but the section referencing Magelia usage.

Note: these widgets can be moved to different zones depend on the needs of the end user but recommend they stay on the default layer.

5.6.2.1 Configuring the main menu:

For the most part this menu gets configured as needed depending on which pages the user gets setup so it will not be setup directly but though the process of configuring the other pages. By default it should possess home is its sole field. To add more content items (ie pages) you add content menu items which give a list of selectable content menu items which will be used for the main menu.
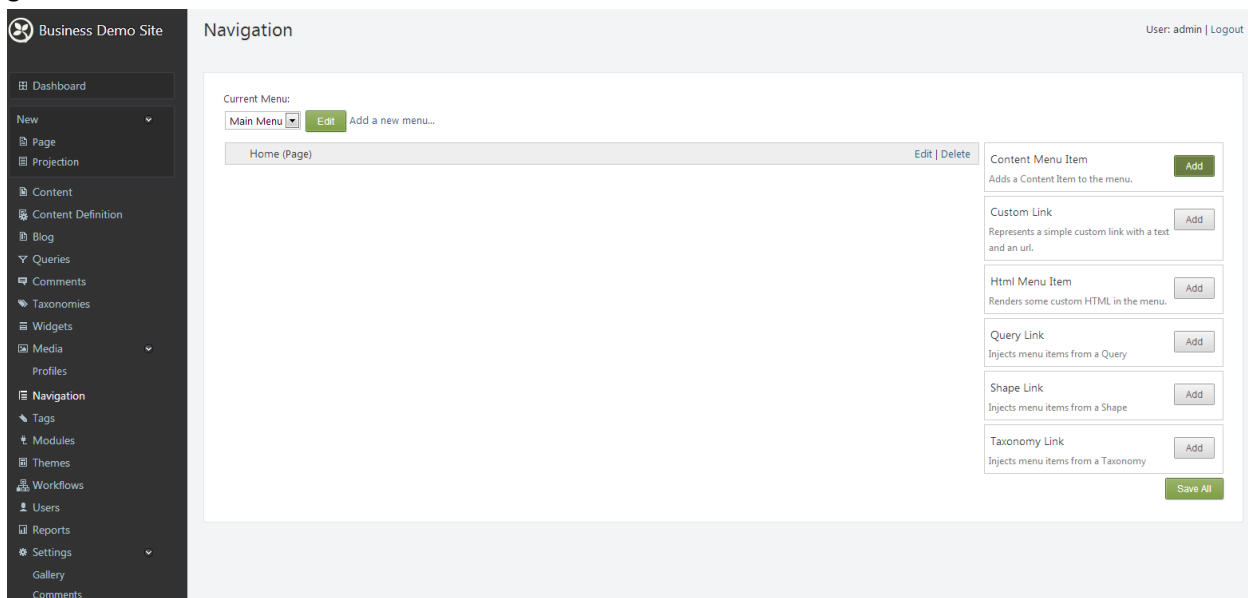
5.6.2.2 Configuring the search widget:

First, the features of the search module have to be activated. To do this click on modules on the dashboard and select the appropriate features under the search module as follows. Hit Execute to enable.
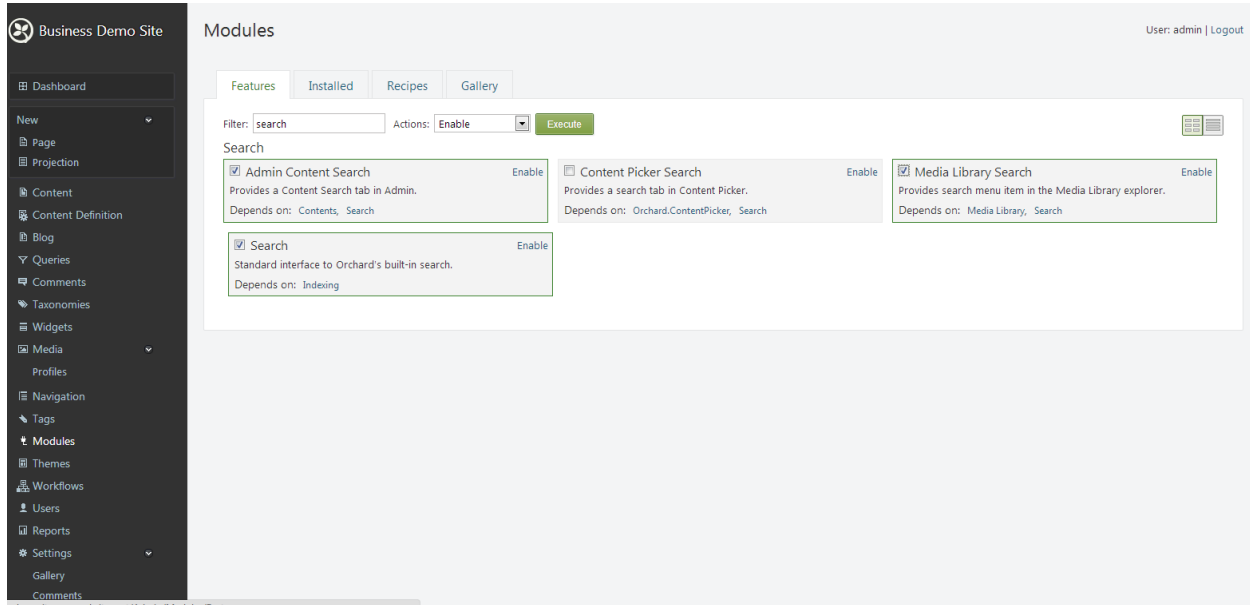
It will then ask you to enable a module called Lucene for indexing, type it into the filter and enable it.

5.6.2.2.1 Creating the search index:

The simplest way to create the search index is to under the same page click on the recipes tab. Then click execute to create a default search index which will be named "search" by default. Towards the bottom of the dashboard under settings, click on search to make sure that "search" is the selected index. Also note that it would be good to check type, taxnomyField, and title as search properties..
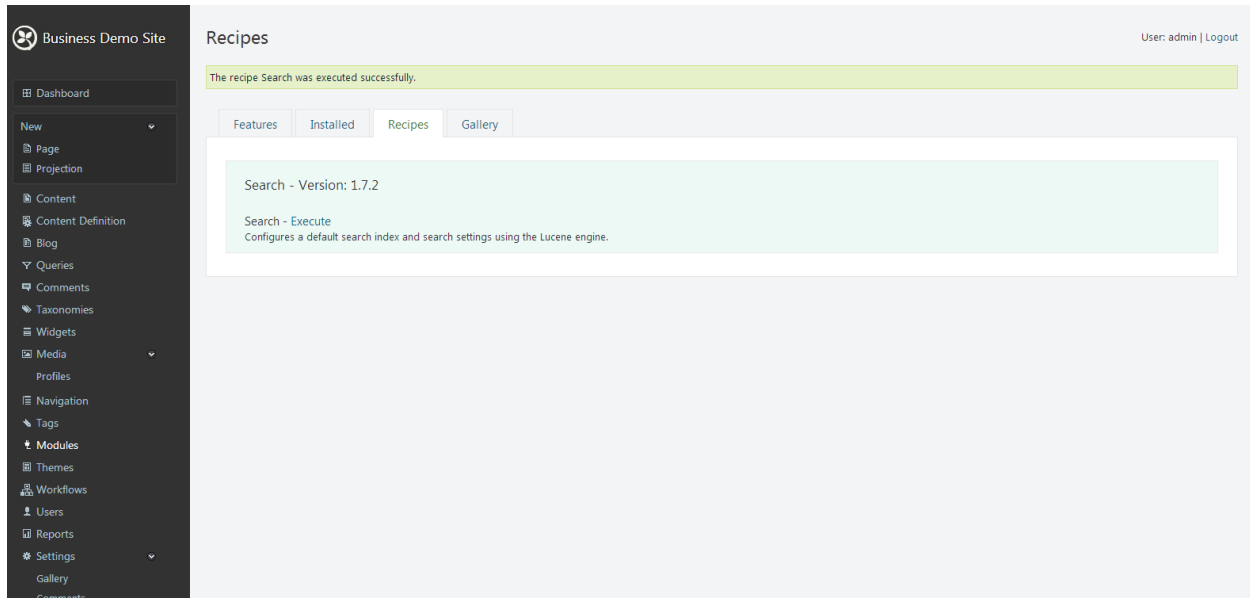
Now go back to the widgets tab that shows the zones and layers. Click "add" in the navigation zone and choose search form.
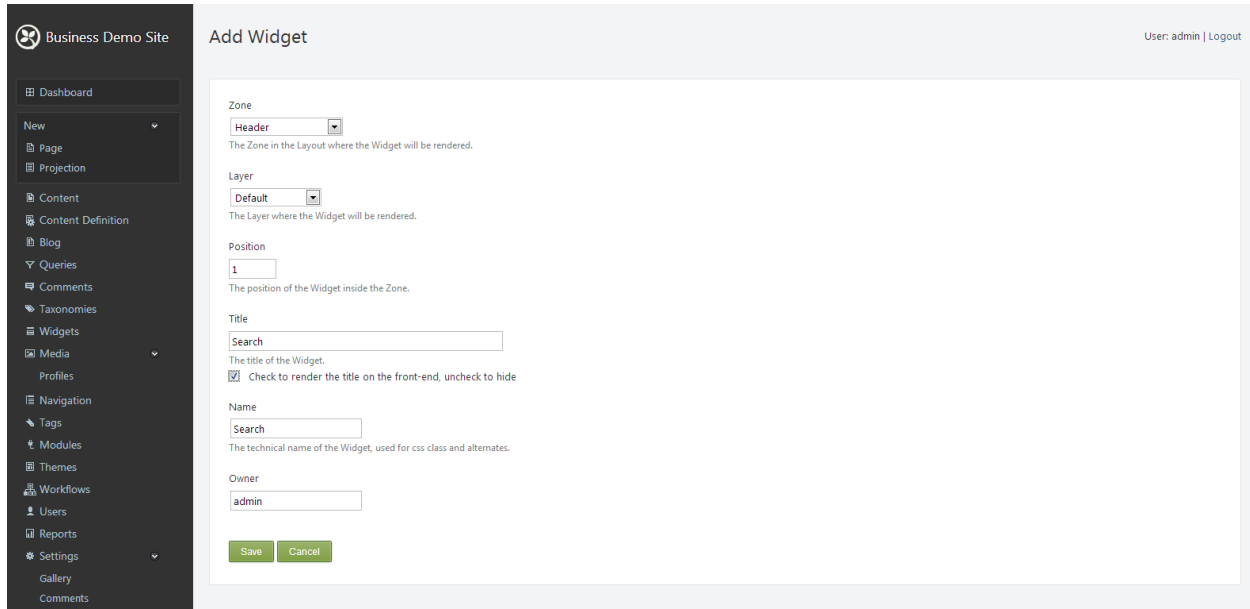
Notice the zone and layers are both described and changeable via this menu. The title and the name should be as similar as possible for consistency. Depending on the object showing its name on the UI is up to the user's needs. Once save is clicked, it will return to the widgets menu and show it has been added.
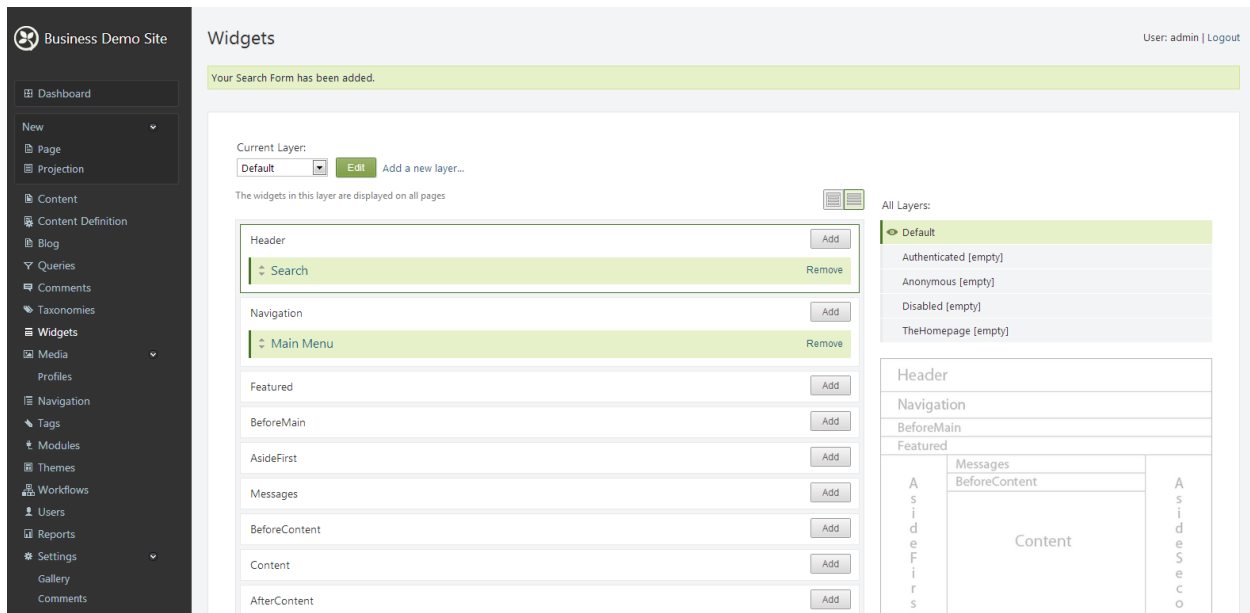
5.6.2.3 Configuring the Quick Logon:

The quick logon allows users to login using Facebook, Google and twitter. It requires the installation of 2 separate modules: RM.QuickLogOn.OAuth - Version: 1.0.7 and RM.QuickLogOn.OAuth - Version: 1.0.3 which the ladder is dependent on. Click the modules tab, then gallery and type the module name or

close into the search.

Install 1.0.3 first and hit the checkbox yes to enable. Then install version 1.0.7 and enable quick logon, then the same for Facebook, Google and twitter. Other Quick logons are available but will not be addressed here.

Note: The check marks are not necessary they are just a visual aid to show which features to have active from these modules. Once they are active at the bottom of the dashboard there should now be a QuickLogon item on the menu. Click it.

Though the instructions are somewhat thorough on how to set these up this document will show some details that might not be readily apparent through the default screen. For all of these you have to register as developer with each of the corresponding sites.

5.6.2.3.1 Google OAuth:

Once you go the link in step one it should look like this; if you have no app click create app or API project(in this instance its named BusLogin) and return to this page then you will be able to get Google auth to work once you fill in the callback url given by the module in the previous screen and taking the client id and client secret from this one.
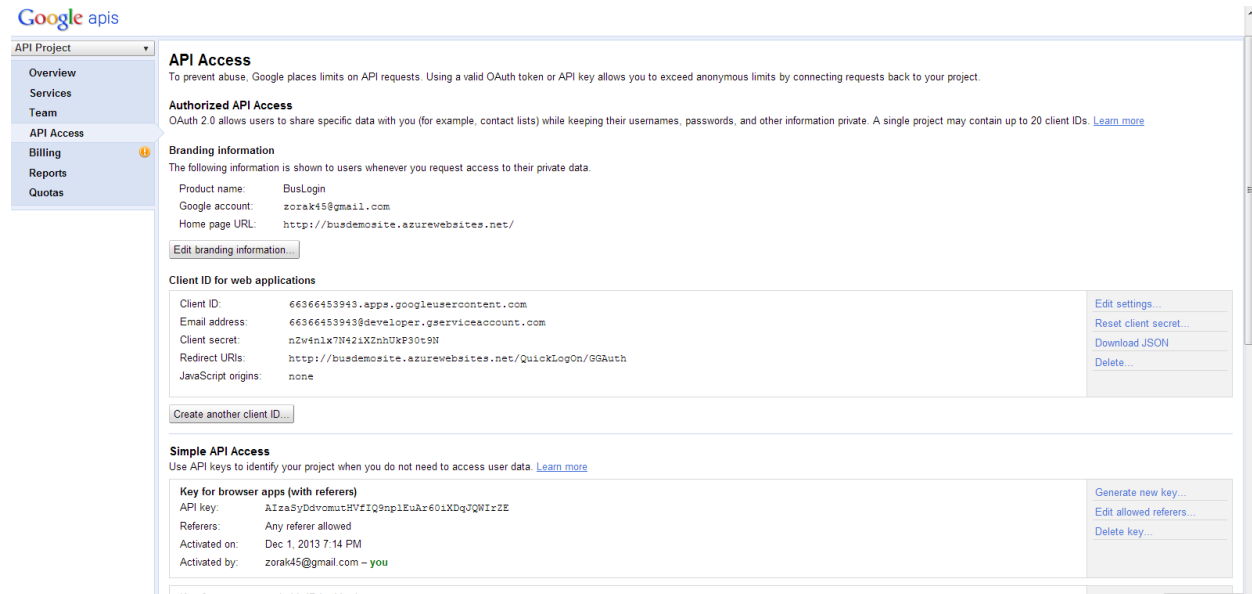
5.6.2.3.2 Facebook OAuth

Facebook OAuth is similar to setup. First create and name an app and put in the callback url given by the module. You will also need to fill a few other fields listed below and make sure Facebook login for site is checked. NOTE: Make sure that sandbox mode is turned off or only developers can use Facebook login.
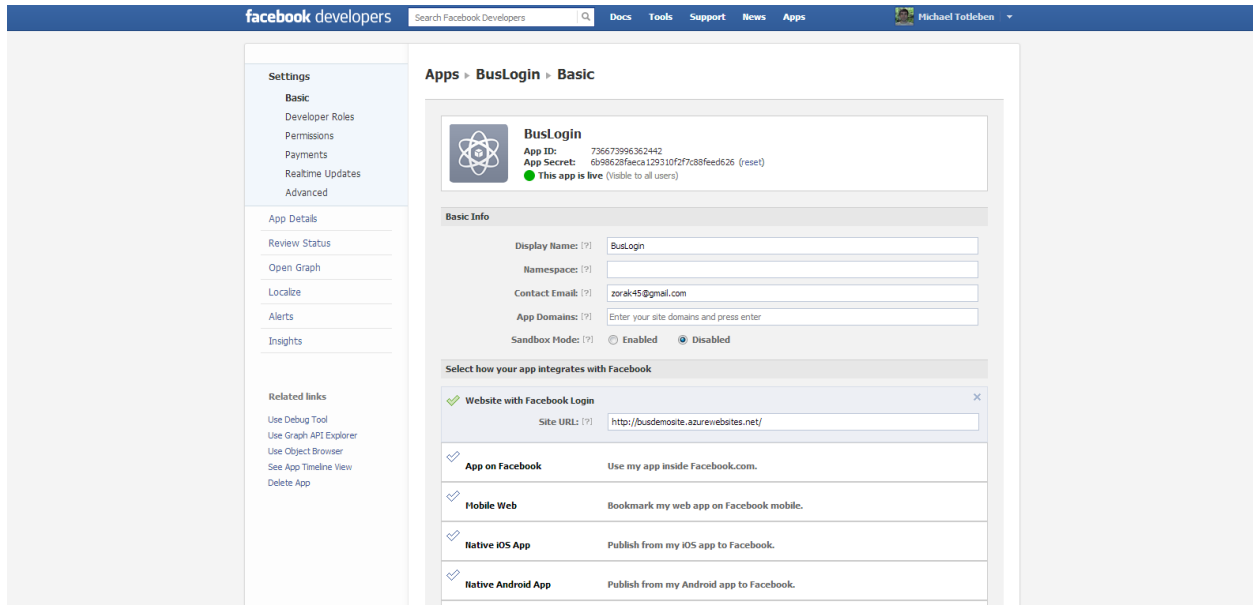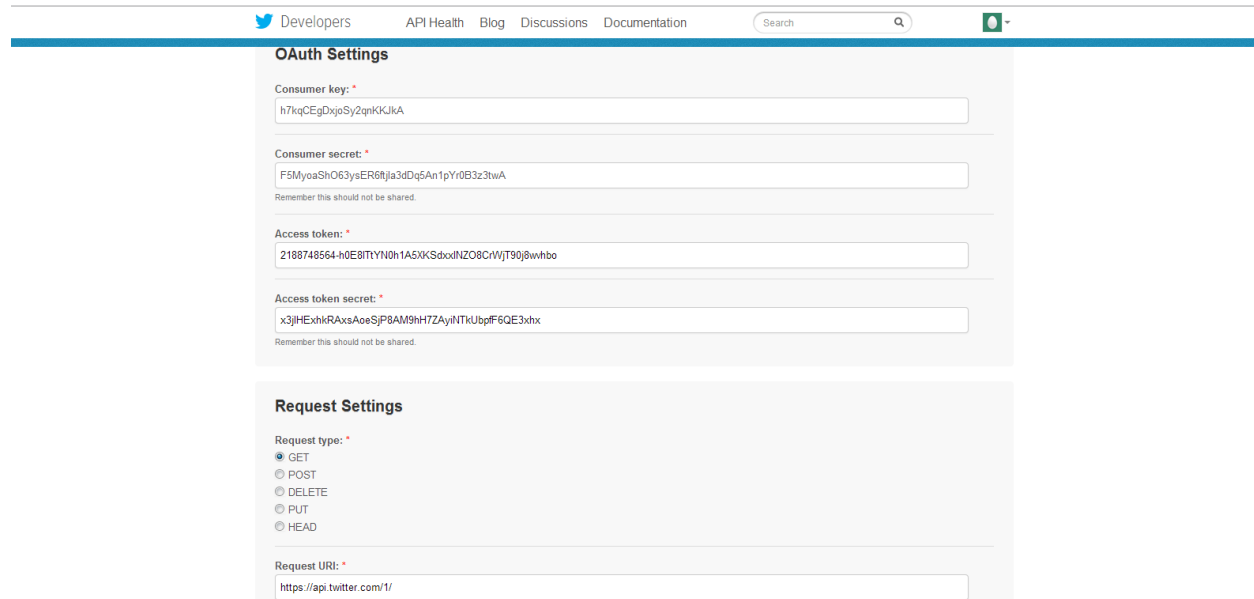
5.6.2.3.3 Twitter OAuth:

Once again it functions similar to the Google and Facebook open auth but also requires creating an app. First, go under the settings tabs and fill in the appropriate fields. Note that callback url field is on this page but is requires some scrolling so its outside the screen shot.
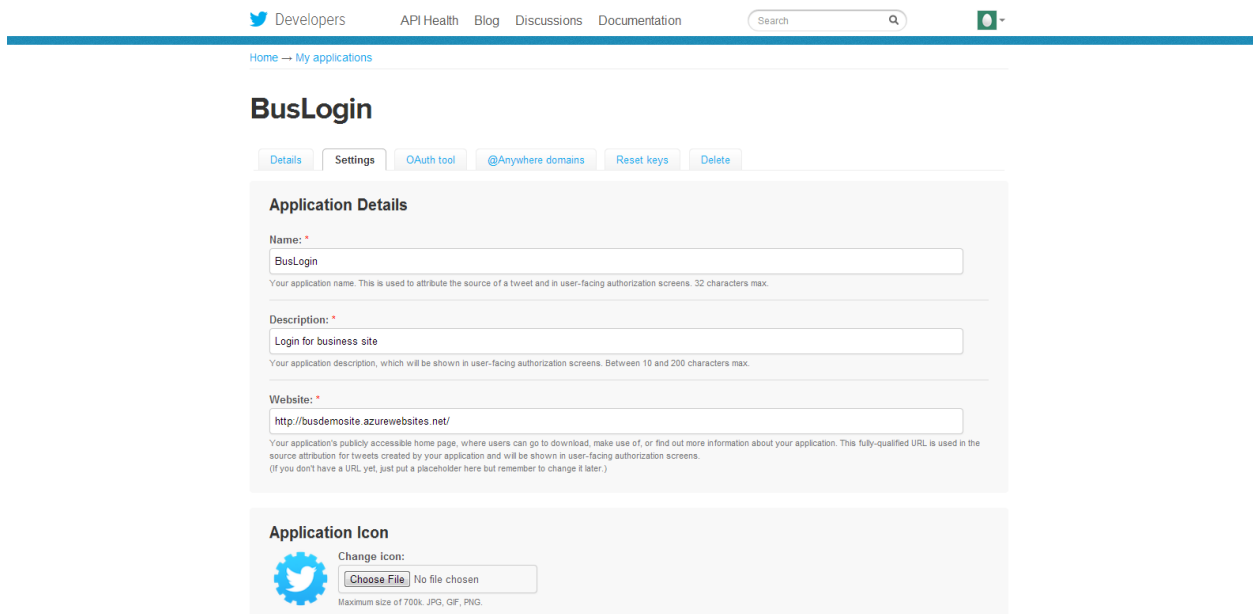
The second part of the Twitter configuration requires that you click on the "Oauthtool" tab to get the rest of the data required. Hit save when finished with all three.

5.6.2.3.4 Widget Creation:

To create the widget simply go to the widgets menu, select "add" in the navigation zone and choose quick logon widget on the default layer. Unchecking display title is recommended.

5.6.2.4 Magelia Cart Widget:

See the Magelia's Orchard module documentation to set this up in section 5.9.

5.6.3 The Home Page Layer:

By default in this setup there aren't any special widgets assigned to this layer since it's simply a home page. Its rules are set by default that widgets created on its layer only appear on the homepage.

5.6.4 Image Gallery Layers:

The image gallery layers is the layer for image gallery pages. There are two layers that have to with the image gallery. Image Gallery Main and Image Gallery sub.

5.6.4.1 Image Gallery Main:

This layer is the layer for the image gallery when it is selected via the main menu giving both a taxonomy menu on the right and a projection of Image galleries in the content area.

5.6.4.1.1 Layer Rule:

url '~/Gallery'

5.6.4.1.2 Taxonomy Menu:

The taxonomy menu is a widget located in the aside first zone. Create a Menu called "Gallery" and have it use a taxonomy link. Make the root term "Gallery" and do not check display root term.

5.6.4.1.3 Gallery Projection:

The gallery projection will be a projection widget located in the content zone with the gallery query selected:

Gallery widget query parameters:

Filters: Categorized with one of Image Gallery

Sort Criteria: Order by creation date ascending, Order by Title Part Ascending.

5.6.4.2 Image Gallery Sub:

This layer is the layer used when a gallery is selected by the taxonomy menu or selected directly though the projection in the content area.

5.6.4.2.1 Layer rule:
  url("~/sitetaxonomy/root/image-gallery*") or url("~/gallery/*")

5.6.4.2.2 Taxonomy Menu:

The taxonomy menu is a widget located in the aside first zone. Name it "GallerySub" (and have title just "Galleries" if you want it show) and have it use the already created menu "Gallery".

5.6.4.2.3 Content:

This is simple the content of the page i.e. the particular Gallery Page that will appear in the content zone.

5.6.5 Video Page Layers:

The Video Page layer is the layer for video pages. There are two layers that have to with the Video Page Main and Video Page sub.

5.6.5.1 Video Page Main:

This Layer is the layer for the Video Page when it is selected via the main menu giving both a taxonomy menu on the right and a projection of Video Pages in the content area.

5.6.5.1.1 Layer Rule:
url '~/Video'

5.6.5.1.2 Taxonomy Menu:

The taxonomy menu is a widget located in the aside first zone. Create a Menu called "Videos" and have it use a taxonomy link. Make the root term "Video" and do not check display root term.

5.6.5.1.3 Video Projection:

The Video Projection will be a projection widget located in the content zone with the Video query selected:

Video widget   query parameters:

Filters: Categorized with one of Video

Sort Criteria: Order by creation date ascending, Order by Title Part Ascending.

5.6.5.2 Video Page Sub:

This layer is the layer used when a video page is selected by the taxonomy menu or selected directly though the projection in the content area.

5.6.5.2.1 Layer rule:

 url("~/sitetaxonomy/root/video*") or url("~/video/*")

5.6.5.2.2 Taxonomy Menu:

The taxonomy menu is a widget located in the aside first zone. Name it "VideoSub" (and have title just "Videos" if you want it show) and have it use the already created menu "Video".

5.6.5.2.3 Content:

This is simple the content of the page i.e. the particular Video Page that will appear in the content zone.

5.6.6 Forum Layers:

The Forum layer is the layer for Forum pages. There are two layers that have to with the Forums; Forum main and Forum Sub.

5.6.6.1 Forum Main:

This Layer is the layer for the Forum when it is selected via the main menu giving both a taxonomy menu on the right and a projection of Forums in the content area.

5.6.6.1.2 Layer Rule:

url '~/Forum'

5.6.6.1.3 Taxonomy Menu:

The taxonomy menu is a widget located in the aside first zone. Create a Menu called "Forum" and have it use a taxonomy link. Make the root term "Forum" and do not check display root term.

5.6.6.1.4 Forum Projection:

The Forum Projection will be a projection widget located in the content zone with the Forum query selected:

Forum widget   query parameters:

Filters: Categorized with one of Forum

Sort Criteria: Order by creation date ascending, Order by Title Part Ascending.

5.6.6.2 Forum Sub:

This layer is the layer used when a forum is selected by the taxonomy menu or selected directly though the projection in the content area.

Layer rule: url("~/sitetaxonomy/root/forum*") or url("~/forum/*")

5.6.6.2.1 Taxonomy Menu:

The taxonomy menu is a widget located in the aside first zone. Name it "ForumSub" (and have title just "Forum" if you want it show) and have it use the already created menu "Forum".

5.6.6.2.2 Content:

This is simple the content of the page ie the particular Gallery Page that will appear in the content zone.

5.6.7 Magelia Layer(Store Layer):
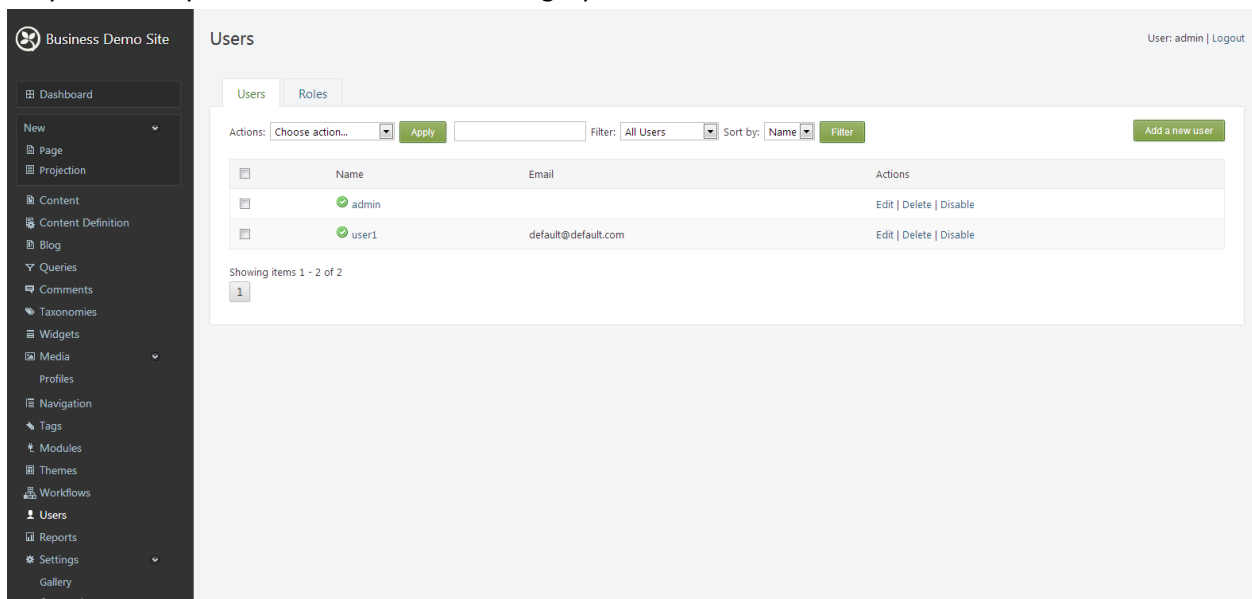
The Magelia setup within orchard has its own documentation and it will be referenced in the Magelia/Orchard Linking section.

5.7 Security/User Settings

Orchard uses a system very similar to a standard asp.net configuration when it comes to users and security.

5.7.1 Users:

Users can be created via the users tab. When they are created they can be assigned roles. By default they will merely be in the authenticated category.

Users in the site can be authenticated or anonymous. If it is desired that users can create their own accounts then under the "settings" tab on the dashboard select "users" and check permissions.
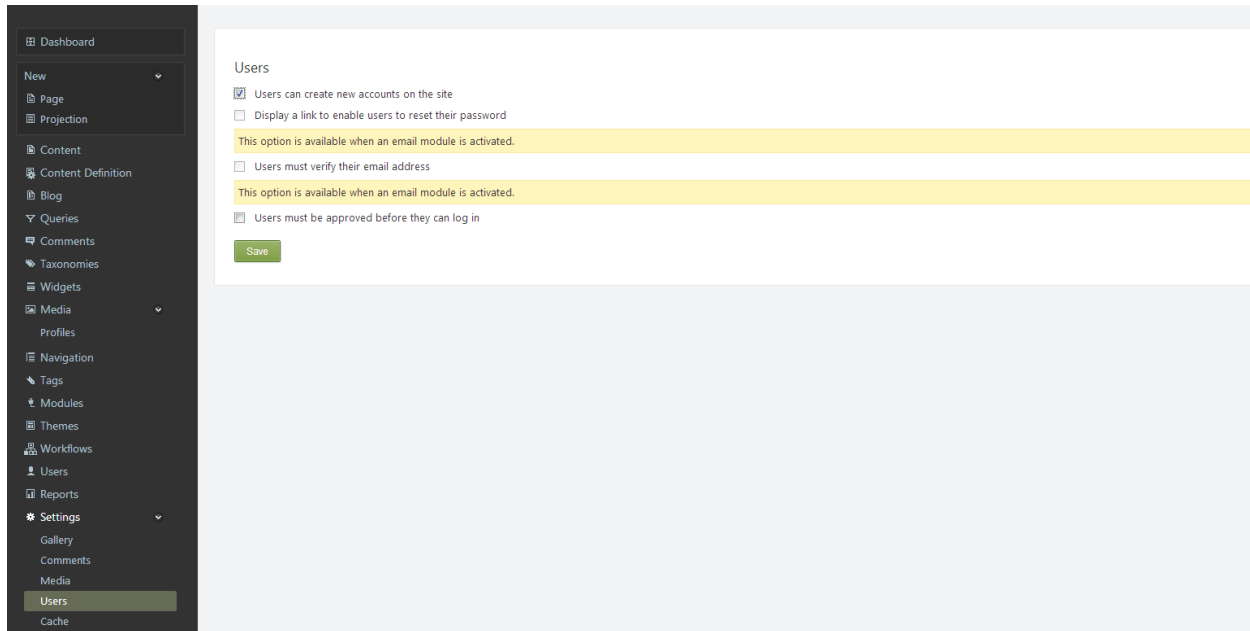
5.7.2 Roles

Roles are setup similar in other ASP.net applications can be added and modified to the needs of the administrator.
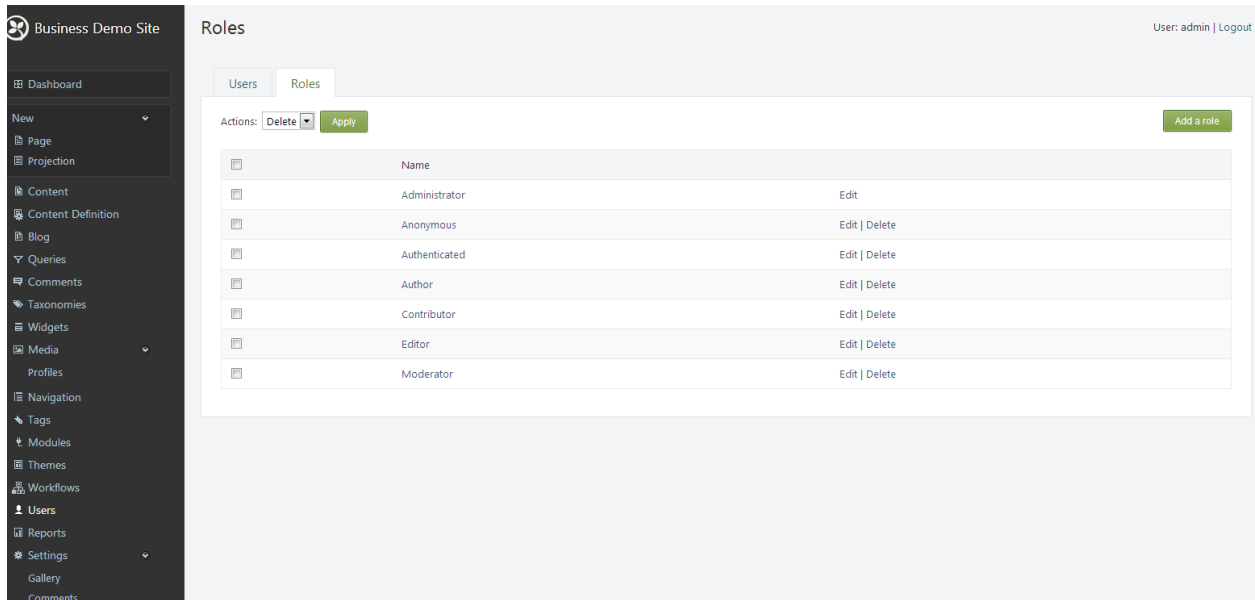
Under the roles tab, check that authenticated users can publish their own content. This will allow then to post in forums.

5.8 Sub Page Creation/Creating New Content Types:

The sub pages are specialized pages that exist to cater to a specific function. These functions are an Image gallery page, a video page, a forums page and a store page. For the most part they are similar to a standard page but with a few special parts and fields. NOTE: Aside from the store pages all subpages should be indexed in the search index. Also, under content definition it is useful to uncheck projections as creatable in the new menu because the end user will not need them to be accessed in that fashion.

5.8.1 Store Page Creation:

The simplest page that will be created is a store page. It is exactly the same as a standard page except for the name. The justification for this is that Magelia requires various pages to implement it store and to filter out those pages from standard pages they will be given a different name. Also, it complies with the standard that all subpages are their own form of page. Note: this does not include the Main store page which is a standard page.

First go the Content Definition tab in the Dashboard and click it, then create new content type.

Name the content type Store Page and choose the content parts as shown below.

These are Common, Autoroute, Body, Menu, Title. Hit save at the bottom of the screen and the content type is created.

5.8.2 Image gallery Page Creation:

Creating an Image gallery page is similar except it allows tags, is indexed in search has an Image Gallery Part and a taxonomy field. To add taxonomy field click add field and choose taxonomy. Name it "Taxonomy Field" for simplicity. Note that it is implied the image gallery module has been installed already.

<u>Parts:</u>

Common

Autoroute

Comments

Image Gallery

Navigation

Publish Later

Body

Tags

Title

Fields:

Taxonomy: an example of what all content with taxonomy field should have check and selected.

5.8.3 Video Page Creation:

The Video Page is a page on which videos are presented. To create a video page selected the following parts.

Parts:

Common

Autoroute

Comments

Navigation

Publish Later

Body

Tags

Fields:

Taxonomy

5.8.4 Forum Page Creation:

The Forum Page is a sub page for forums. Install the Forum module "Forums - Version: 1.0.1" from the gallery and modify content definition for forums by adding a taxonomy field and checking searchable in

the index. It's optional to make them creatable since they are creatable already via the dashboard when the module is installed.

5.9 Magelia/Orchard Linking:

Magelia has created its own documentation regarding the linking process which can be found here:

http://www.magelia.org/Media/Default/Documentation/20120702-Magelia-WebStore-Module-for-Orchard-004.pdf
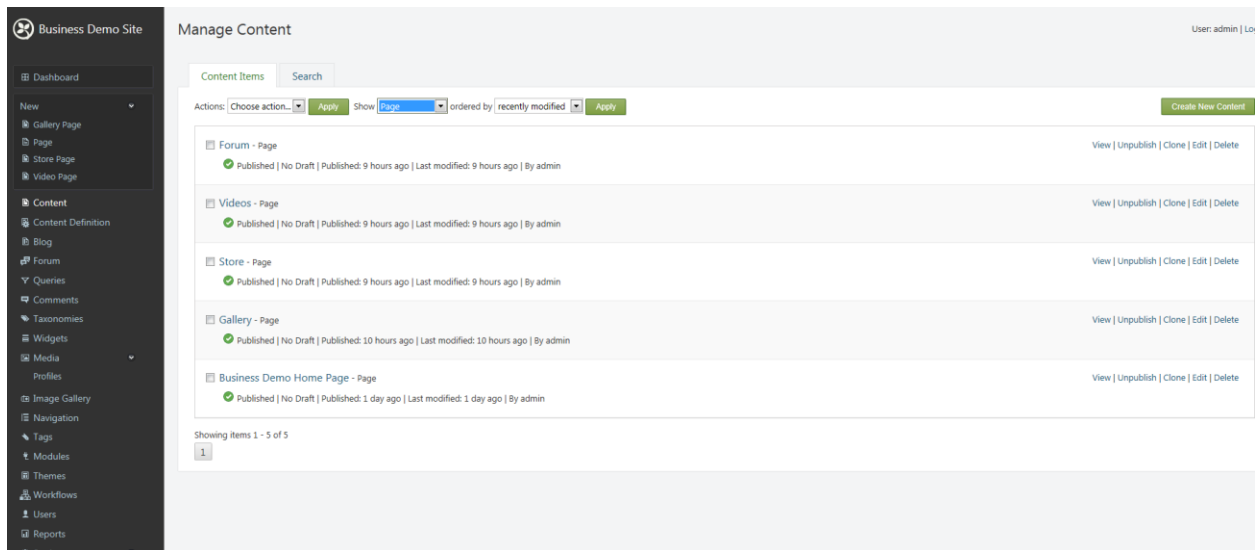
For implementation in this environment instead of using the home page as the location of the store use the main store page by creating a store layer instead of using the homepage layer. Any additional pages created product, basket etc. use the store page subtype.

Note that the Paypal step requires you have a Paypal business account setup.

6. Adding Content Pages:

Now that the site is essentially setup the user will want to add content pages.

6.1 Video Page:

In order to add a video page it is first recommended to have some media. Click the media library tab and create a folder called video. You can create sub folders if necessary. Then click import.
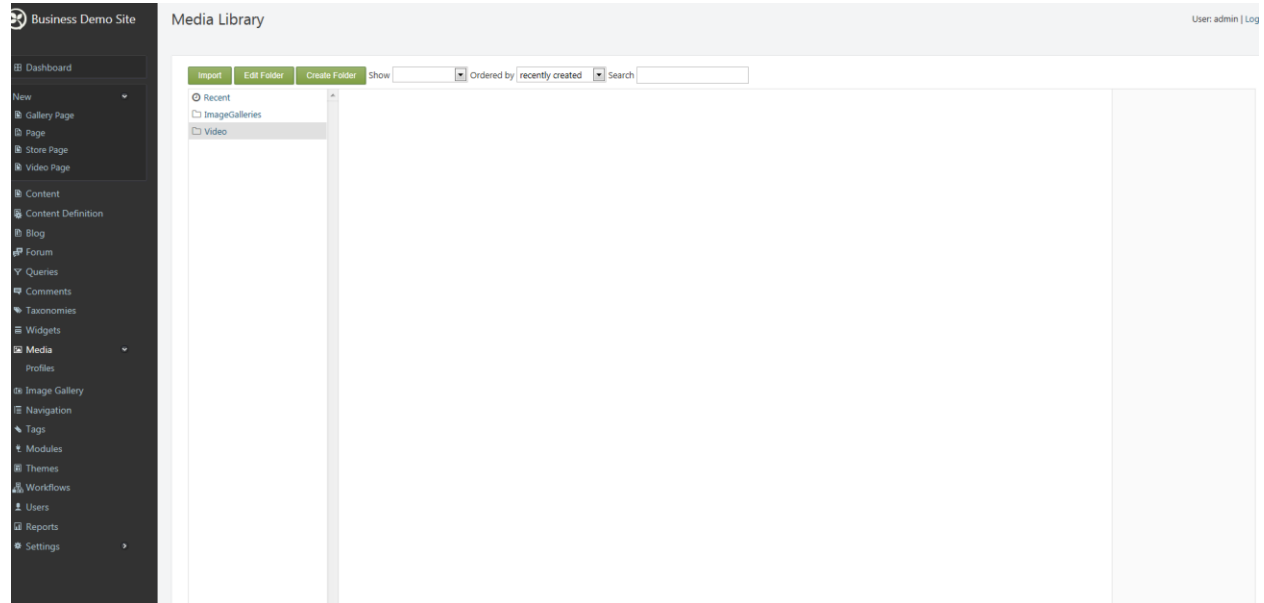
The Url and thumbnail for the video will appear. Hit close.
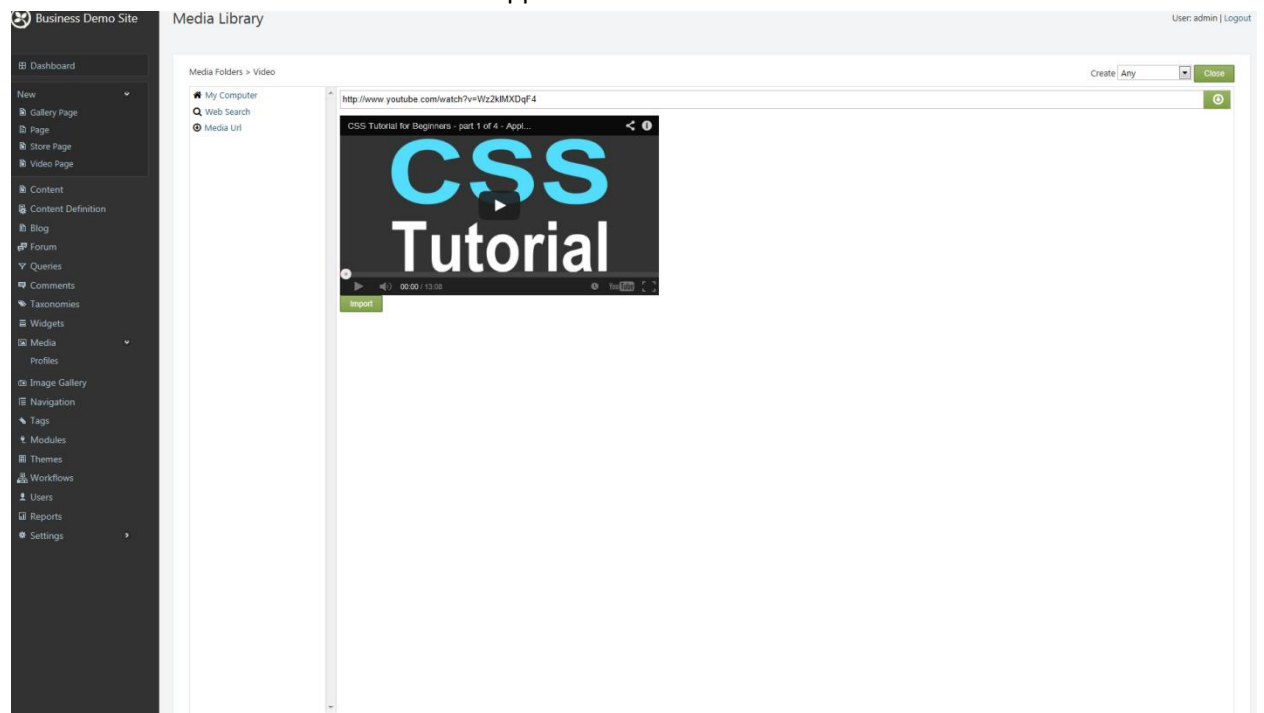
Then using the dashboard click Video Page under new:
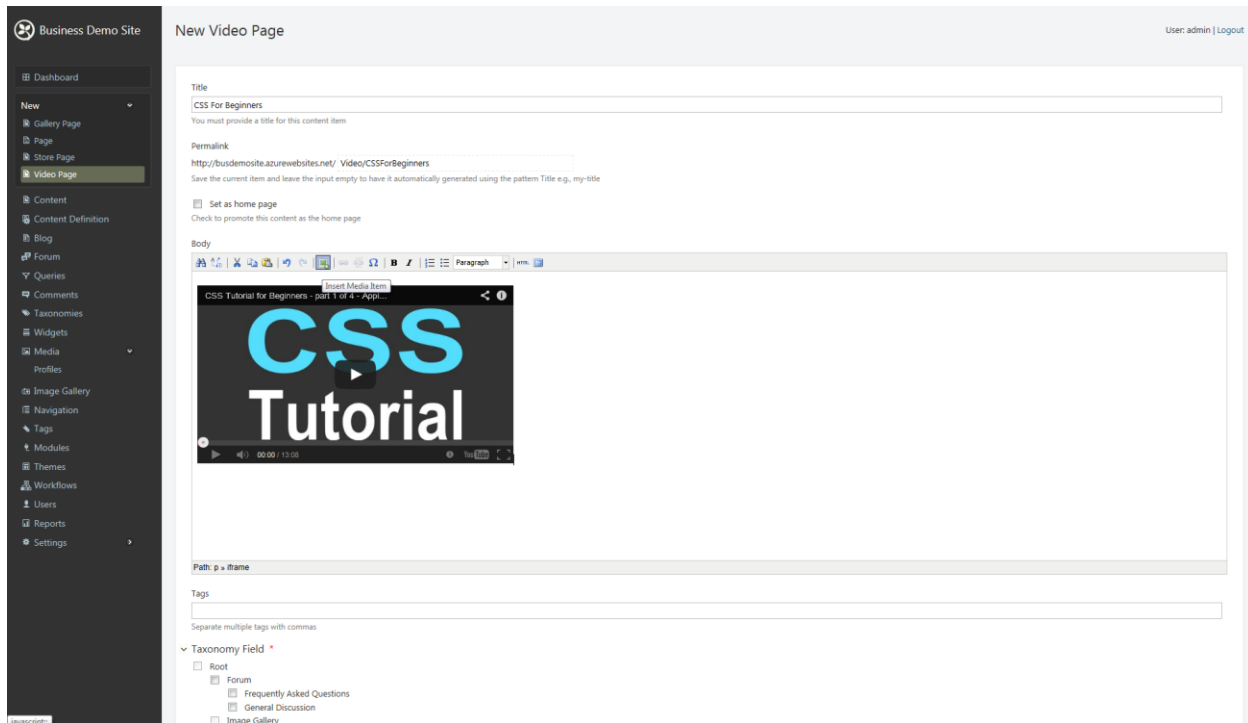
For all video pages make sure in the permalink to always put "video/" in front of the desired url. This will ensure it is governed by the video layers. Click "add media" and select the video you want displayed in the body. It should now be viewable on the video page and through its taxonomy menus. Comments can be added is checked by default.

6.2 Gallery Page:

The gallery page requires the existence of an image gallery to work. To create and image gallery click Image gallery on the dashboard. Name the Image gallery; in this case it will be mountains. Its easiest

upload all images in groups as a zip file. Hit upload.

You will then see the gallery populated like so.

Now click Gallery Page under new on the dashboard. Make sure to choose and check to display an image gallery on the page. Remember that all permalinks have "Gallery/ "in front of the desired url so they are

governed by the Gallery layers.
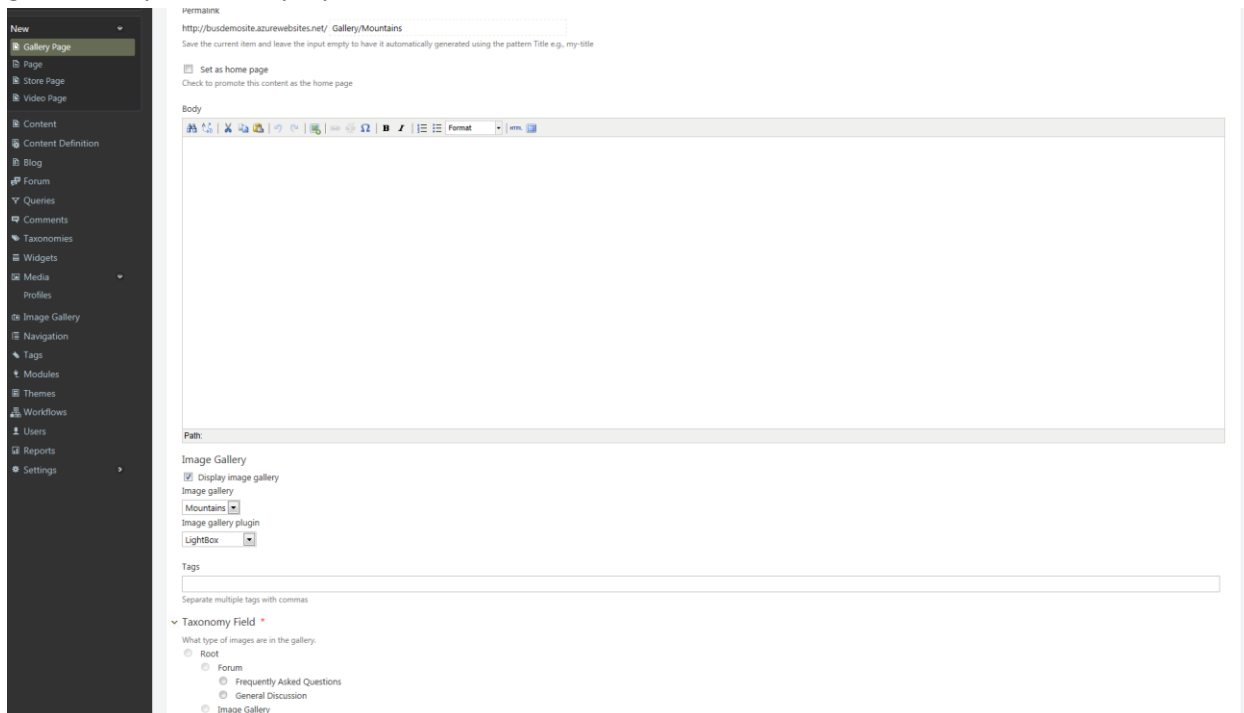
Make sure display image gallery is check with the appropriate gallery selected. Also, make sure the appropriate taxonomy field is selected. Comments can be added is checked by default.

6.3 Forum Page:

The forum is a page created by the forums module displayed on the forum page. To create forums click the forum button on the dashboard. Notice the url must begin with "forum/" to be governed under the

forum layers.

.

7. Magelia Dashboard User guide:

This is already prefabricated documentation is available at this link:

http://www.magelia.org/resources/introduction-to-magelia-webstore

8. Sources/Bibliography:

Windows Azure Programing for startups, Ricardo Becker.

Windows Azure Step by Step, Roberto Brunetti

Plural Site Orchard Training:

http://www.pluralsight.com/training/courses/TableOfContents?courseName=orchard-fundamentals

Orchard Project:

http://www.orchardproject.net/

Magelia.org

http://www.magelia.org/

Codeplex.com

www.Codeplex.com

Orchard Dojo:

http://orcharddojo.net/

Bitbucket Nawezet Commerce:

https://bitbucket.org/bleroy/nwazet.commerce

Windows Azure.com

http://www.windowsazure.com/en-us/documentation/