**Document Types**

|  |  |  |  |
| --- | --- | --- | --- |
| **Content type** | **Inheritance 1** | **Inheritance 2** | **Fields** |
| BasePage |  |  | Meta title (use node name if empty) – text string  Meta keywords – text string  Meta description – text area  Hide in menu – bool  **Footer**  Column 1   |  |  | | --- | --- | | Content picker |  |   Column 2   |  |  | | --- | --- | | Content picker |  |   \*Phone – text string \*Email – text string Other- text area |
|  | Home |  | **Sliders**   |  |  | | --- | --- | | \*Image (100 wide x 200 wide) | Media picker  /media/home sliders | | Red text | text string | | Yellow text | text string | | White text | text string | | Link text | text string | | Link | Content picker |   **Red feature tab**  Visible - bool  \*Text – text area Link text – text string Link – content picker  **Get involved tab** Visible - bool  \*Donate link – content picker  \*Ride link – content picker  \*Volunteer link – content picker  **News tab**  Visible – bool  \*Bold heading – text string  \*Text - text string  Link text – text string  Link – content picker  **Quick Link Images tab**   |  |  | | --- | --- | | \*Image (100 wide x 200 wide) | Media picker  /media/home quick links | | Title | text string | | Link | Content picker | |
|  | Page |  | **Main content tab**  \*Rte   * Images with caption (pick many)   **Side bar tab**  \*rte   * Margin highlight box (pick one) * Margin image gallery (pick many) |
|  |  | Internal Top level |  |
|  |  |  |  |
| Partner |  |  | Type – \*Partner drop down  Diamond  Platinum  Gold  Cancer Awareness Packs and Books  Category Jersey Partners  Essential Partners  Gold Community Partners  Silver Community Partners  Bronze Community Partners  \*Name – text string  \*Image – media picker (/media/partners)  Link - url  Order - float |
| ImageWithCaption |  |  | \*Image – media picker (/media/page-content) Caption – text string Link – text string |
| Margin highlight box |  |  | \*Red text – text string  \*Grey Number – text string  Description - text string  Link – content picker  Image – media picker (/media/margin highlight boxes) |
| Margin image gallery |  |  | \*Image – media picker (/media/margin image gallery)  Description - text string  Link – content picker |
| Menu item |  |  | Header image – media picker (/media/header-images)  Background image – media picker (/media/background-images) |

**Content**

|  |  |  |
| --- | --- | --- |
| Menu item |  | Tree |
|  | Page | Tree |
| Data |  | Tree |
|  | Partners | Grid |
|  | ImageWithCaption | Show as grid |
|  | Margin Highlight boxes | Grid |
|  | Margin image gallery | Grid |

All the possible web service requests in the front-end:

Get information from the server:

* Search result/ search in site, team membersu
* Donations amount in header for all active tours + target for all active tours
* Donations raised for a specific tour
* Tours/Events List
* Tour Route information
* Tour Roles
* Team members in each role/stage/participation type
* Members search result
* Event information
* Events search result, should be able to filter by date and state
* Group rides, filter by state, city and date

Send information to the server:

* Donation

WEB API

Add Web API to the existing MVC Project:

Add a new role “SERVICE” and assign a user to that role.

Add Required namespaces: System.Web.Http.WebHost , System.Web.Http.WebHost

If you add a web api controller above assemblies will be added to the project automatically.

Add the Config file to the App\_Start Folder

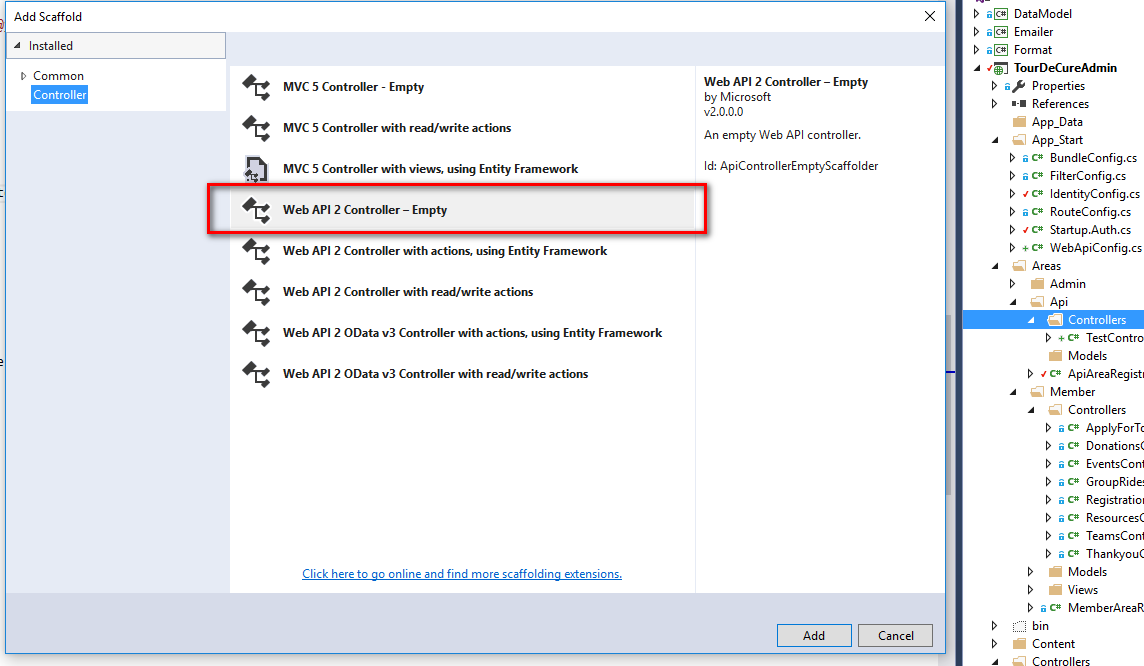
WebApiConfig.cs

Only stores routing for the api

Add Below lines to the Global.asax.cs at the beginning of the Application\_Start

GlobalConfiguration.Configure(WebApiConfig.Register);

Add a web api 2.0 Controller



Controller class should have Authorize filter if necessary.

Add below lines to Startup.Auth.cs in App\_Start Folder

OAuthAuthorizationServerOptions OAuthServerOptions = new OAuthAuthorizationServerOptions()

{

AllowInsecureHttp = true,

TokenEndpointPath = new PathString("/token"),//it’s the rout to get a token

AccessTokenExpireTimeSpan = TimeSpan.FromDays(1),

Provider = new Utilities.SimpleAuthorizationServerProvider()//You have to implement this Class

};

// Token Generation

app.UseOAuthAuthorizationServer(OAuthServerOptions);

app.UseOAuthBearerAuthentication(new OAuthBearerAuthenticationOptions());

Code for the SimpleAuthorizationServerProvider.cs

using Microsoft.AspNet.Identity;

using Microsoft.AspNet.Identity.EntityFramework;

using Microsoft.Owin.Security.OAuth;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Security.Claims;

using System.Threading.Tasks;

using System.Web;

using TourDeCureAdmin.Models;

using Microsoft.AspNet.Identity.Owin;

namespace TourDeCureAdmin.Utilities

{

public class SimpleAuthorizationServerProvider : OAuthAuthorizationServerProvider

{

public override async Task ValidateClientAuthentication(OAuthValidateClientAuthenticationContext context)

{

context.Validated();

}

public override async Task GrantResourceOwnerCredentials(OAuthGrantResourceOwnerCredentialsContext context)

{

context.OwinContext.Response.Headers.Add("Access-Control-Allow-Origin", new[] { "\*" });

var manager = HttpContext.Current.GetOwinContext().GetUserManager<ApplicationUserManager>();

IdentityUser user = await manager.FindAsync(context.UserName, context.Password);

if (user == null)

{

context.SetError("invalid\_grant", "The user name or password is incorrect.");

return;

}

bool hasRole = await manager.IsInRoleAsync(user.Id, ApplicationRoleManager.SERVICE);

if (!hasRole)

{

context.SetError("invalid\_grant", "Wrong role");

return;

}

var identity = new ClaimsIdentity(context.Options.AuthenticationType);

identity.AddClaim(new Claim("sub", context.UserName));

identity.AddClaim(new Claim("role", "user"));

context.Validated(identity);

}

}

}

Examples below are javascript ajax sample to send a request to the web service.

When you want to call a service call, first you need to request a token by sending a post request to /token url with username and password of the user with “Serice” role.

var loginData = {

grant\_type: 'password',

username: 'petras.surna@yart.com.au',

password: 'Arbuzas1'

};

$.ajax({

type: 'POST',

url: '/token',

data: loginData

}).done(function (data) {

sessionStorage.setItem(tokenKey, data.access\_token);

Test();

}).fail(showError);

Then you can use that token to call other web service actions. You need to add the token to the header of the request like below:

var token = sessionStorage.getItem(tokenKey);

var headers = {};

if (token) {

headers.Authorization = 'Bearer ' + token;

}

$.ajax({

type: 'POST',

url: '/Api/Test/Foo',

headers: headers

}).done(function (data) {

alert(data);

}).fail(showError);