```
using Sabio.Web.Domain;
using Sabio.Web.Models.Requests.MyMedia;
using Sabio.Web.Models.Requests.Places;
using Sabio.Web.Models.Responses;
using Sabio.Web.Services;
using System.Collections.Generic;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using Microsoft.Practices.Unity;
using Sabio.Web.Services.Interface;
using WebApi.OutputCache.V2;
using Sabio.Web.Models.Requests.Pagination;

namespace Sabio.Web.Controllers.Api
{
    [RoutePrefix("api/places")]
    public class PlacesApiController : BaseApiController
    {
        public const string EntityIndex = "places";

        [Dependency]
        public IPlacesService _PlacesService { get; set; }


        //Post
        [Route(), HttpPost]
        [Authorize]
        public HttpResponseMessage Add(PlacesAdd Model)
        {
            if (!ModelState.IsValid)
            {
                return Request.CreateErrorResponse(HttpStatusCode.BadRequest, ModelState);
            }

            Model.UserId = UserService.GetCurrentUserId();

            Places newplace = _PlacesService.Insert(Model);

            ItemResponse<Places> Response = new ItemResponse<Places>();

            Response.Item = newplace;

            return Request.CreateResponse(HttpStatusCode.OK, Response);
        }

        //Get By ID
        [Route("{id:int}"), HttpGet]
        [CacheOutput(ServerTimeSpan = 120)]
        public HttpResponseMessage Get(int id)
        {
            if (!ModelState.IsValid)
            {
                return Request.CreateErrorResponse(HttpStatusCode.BadRequest,
                  ModelState);
            }

            Places newplace = _PlacesService.GetPlace(id);

            ItemResponse<Places> Response = new ItemResponse<Places>();

            Response.Item = newplace;

            return Request.CreateResponse(HttpStatusCode.OK, Response.Item);
        }

        //Get By Slug
        [Route("{Slug}"), HttpGet]
        //[CacheOutput(ServerTimeSpan = 120)]
        public HttpResponseMessage GetSlug(string Slug)
        {
            if (!ModelState.IsValid)
            {
                return Request.CreateErrorResponse(HttpStatusCode.BadRequest, ModelState);
            }

            Places newPlaceSlug = _PlacesService.GetPlaceSlug(Slug);

            ItemResponse<Places> Response = new ItemResponse<Places>();

            Response.Item = newPlaceSlug;

            return Request.CreateResponse(HttpStatusCode.OK, Response.Item);
        }

        //Get By List
        [Route(), HttpGet]
        public HttpResponseMessage GetAll([FromUri]PaginatedRequest model)
        {
            if (!ModelState.IsValid)
            {
                return Request.CreateErrorResponse(HttpStatusCode.BadRequest, ModelState);
            }

            int TotalCount = 0;

            List<Places> Listofplaces = _PlacesService.GetList(model, ref TotalCount);

            PagedItemsResponse<Places> Response = new PagedItemsResponse<Places>();

            Response.Items = Listofplaces;
            Response.CurrentPage = model.CurrentPage;
            Response.ItemsPerPage = model.ItemsPerPage;
            Response.TotalItemCount = TotalCount;

            return Request.CreateResponse(HttpStatusCode.OK, Response);


        //Update By ID
        [Route("{id:int}"), HttpPut]
        public HttpResponseMessage Put(PlacesUpdate Model, int id)
        {
            if (!ModelState.IsValid)
            {
                return Request.CreateErrorResponse(HttpStatusCode.BadRequest, ModelState);
            }

            _PlacesService.Update(Model, id);

            ItemResponse<Places> Response = new ItemResponse<Places>();

            Response.IsSuccessful = true;


            return Request.CreateResponse(HttpStatusCode.OK, Response);
        }

        //Delete By ID
        [Route("{id:int}"), HttpDelete]
        public BaseResponse Delete(int id)
        {
            var place = _PlacesService.GetPlace(id);

            if (place == null)
                return Fail($"Place with id {id} does not exist!");

            _PlacesService.Delete(id);

            return Success();
        }


    }
}
```