```
1 using Sabio.Web.Domain;
 2 using Sabio.Web.Models.Requests.MyMedia;
 3 using Sabio.Web.Models.Requests.Places;
 4 using Sabio.Web.Models.Responses;
 5 using Sabio.Web.Services;
 6 using System.Collections.Generic;
7 using System.Net;
8 using System.Net.Http;
9 using System.Web.Http;
10 using Microsoft.Practices.Unity;
11 using Sabio.Web.Services.Interface;
12 using WebApi.OutputCache.V2;
using Sabio.Web.Models.Requests.Pagination;
14
15 namespace Sabio.Web.Controllers.Api
16 {
        [RoutePrefix("api/places")]
17
       public class PlacesApiController : BaseApiController
18
19
            public const string EntityIndex = "places";
20
21
22
            [Dependency]
            public IPlacesService _PlacesService { get; set; }
23
24
25
26
            //Post
            [Route(), HttpPost]
27
28
            [Authorize]
29
            public HttpResponseMessage Add(PlacesAdd Model)
30
31
                if (!ModelState.IsValid)
32
33
                    return Request.CreateErrorResponse(HttpStatusCode.BadRequest,
                      ModelState);
34
35
                Model.UserId = UserService.GetCurrentUserId();
36
37
                Places newplace = PlacesService.Insert(Model);
38
39
                ItemResponse<Places> Response = new ItemResponse<Places>();
40
41
42
                Response.Item = newplace;
43
                return Request.CreateResponse(HttpStatusCode.OK, Response);
44
45
46
            //Get By ID
47
48
            [Route("{id:int}"), HttpGet]
            [CacheOutput(ServerTimeSpan = 120)]
49
50
           public HttpResponseMessage Get(int id)
51
52
                if (!ModelState.IsValid)
53
                    return Request.CreateErrorResponse(HttpStatusCode.BadRequest,
54
```

```
C:\SF.Code\C23\Sabio.Web\Controllers\Api\PlacesApiController.cs
                                                                                         2
                       ModelState);
 55
 56
 57
                 Places newplace = _PlacesService.GetPlace(id);
 58
 59
                 ItemResponse<Places> Response = new ItemResponse<Places>();
 60
 61
                 Response.Item = newplace;
 62
 63
                 return Request.CreateResponse(HttpStatusCode.OK, Response.Item);
 64
 65
             //Get By Slug
 66
             [Route("{Slug}"), HttpGet]
 67
             //[CacheOutput(ServerTimeSpan = 120)]
 68
 69
             public HttpResponseMessage GetSlug(string Slug)
 70
                 if (!ModelState.IsValid)
 71
 72
                     return Request.CreateErrorResponse(HttpStatusCode.BadRequest,
 73
                       ModelState);
 74
 75
 76
                 Places newPlaceSlug = _PlacesService.GetPlaceSlug(Slug);
 77
 78
                 ItemResponse<Places> Response = new ItemResponse<Places>();
 79
 80
                 Response.Item = newPlaceSlug;
 81
                 return Request.CreateResponse(HttpStatusCode.OK, Response.Item);
 82
             }
 83
 84
             //Get By List
 85
             [Route(), HttpGet]
 86
 87
             public HttpResponseMessage GetAll([FromUri]PaginatedRequest model)
 88
 89
                 if (!ModelState.IsValid)
 90
 91
                     return Request.CreateErrorResponse(HttpStatusCode.BadRequest,
                       ModelState);
 92
 93
 94
                 int TotalCount = 0;
 95
                 List<Places> Listofplaces = _PlacesService.GetList(model, ref
 96
                   TotalCount);
 97
 98
                 PagedItemsResponse<Places> Response = new PagedItemsResponse<Places> →
                   ();
 99
                 Response.Items = Listofplaces;
100
101
                 Response.CurrentPage = model.CurrentPage;
102
                 Response.ItemsPerPage = model.ItemsPerPage;
                 Response.TotalItemCount = TotalCount;
103
104
```

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

```
107
108
             //Update By ID
109
             [Route("{id:int}"), HttpPut]
110
             public HttpResponseMessage Put(PlacesUpdate Model, int id)
111
                 if (!ModelState.IsValid)
112
113
                     return Request.CreateErrorResponse(HttpStatusCode.BadRequest,
114
                       ModelState);
115
116
                 _PlacesService.Update(Model, id);
117
118
                 ItemResponse<Places> Response = new ItemResponse<Places>();
119
120
121
                 Response.IsSuccessful = true;
122
123
124
                 return Request.CreateResponse(HttpStatusCode.OK, Response);
125
126
127
             //Delete By ID
128
             [Route("{id:int}"), HttpDelete]
             public BaseResponse Delete(int id)
129
130
131
                 var place = _PlacesService.GetPlace(id);
132
133
                 if (place == null)
134
                     return Fail($"Place with id {id} does not exist!");
135
                 _PlacesService.Delete(id);
136
137
                 return Success();
138
139
140
141
142
         }
143
144
```