

## InteractionsApiController

---

```
1 using MGM.Web.Domain;
2 using MGM.Web.Models.Requests.Interactions;
3 using MGM.Web.Models.Responses;
4 using MGM.Web.Services;
5 using System;
6 using System.Collections.Generic;
7 using System.Linq;
8 using System.Net;
9 using System.Net.Http;
10 using System.Web.Http;
11
12 namespace MGM.Web.Controllers.Api
13 {
14     [RoutePrefix("api/interactions")]
15     public class InteractionsApiController : ApiController
16     {
17         IInteractionService _interactionsService = null;
18         IUserService _userService = null;
19         IClientStateService _clientStateService = null;
20
21         public InteractionsApiController(IInteractionService interactionsService, ➤
22             IUserService userService, IClientStateService clientStateService)
23         {
24             _interactionsService = interactionsService;
25             _userService = userService;
26             _clientStateService = clientStateService;
27         }
28
29         [Route, HttpPost]
30         public HttpResponseMessage AddInteraction(AddInteractionRequest model)
31         {
32             if (!ModelState.IsValid)
33             {
34                 return Request.CreateErrorResponse(HttpStatusCode.BadRequest, ➤
35                     ModelState);
36             }
37
38             string userId = _userService.GetCurrentUserId();
39             int accountId = _userService.GetCurrentAccountId();
40             ItemResponse<Int32> response = new ItemResponse<Int32>();
41
42             response.Item = _interactionsService.Add(model, userId, accountId);
43             return Request.CreateResponse(response);
44         }
45
46         [Route("client/{interactionId}"), HttpPost]
47         public HttpResponseMessage AddInteractionByClient(int interactionId)
48         {
49             if (!ModelState.IsValid)
50             {
51                 return Request.CreateErrorResponse(HttpStatusCode.BadRequest, ➤
52                     ModelState);
53             }
54         }
55     }
56 }
```

## InteractionsApiController

---

```
50     }
51
52     string clientId = _userService.GetCurrentUserId();
53     SuccessResponse response = new SuccessResponse();
54     _interactionsService.AddByClient(interactionId, clientId);
55
56     return Request.CreateResponse(response);
57 }
58
59 [Route("{Id:int}"), HttpPut]
60 public HttpResponseMessage EditContact(UpdateInteractionRequest model)
61 {
62     if (!ModelState.IsValid)
63     {
64         return Request.CreateErrorResponse(HttpStatusCode.BadRequest,
65                                             ModelState);
66     }
67
68     string userId = _userService.GetCurrentUserId();
69     int accountId = _userService.GetCurrentAccountId();
70
71     SuccessResponse response = new SuccessResponse();
72     _interactionsService.UpdateInteraction(model, userId, accountId);
73     return Request.CreateResponse(response);
74 }
75
76 [Route, HttpGet]
77 public HttpResponseMessage Get()
78 {
79     ItemsResponse<Interaction> response = new ItemsResponse<Interaction>
80     ();
81     response.Items = _interactionsService.Get();
82     return Request.CreateResponse(response);
83 }
84
85 [Route("account"), HttpGet]
86 public HttpResponseMessage GetByAccountId()
87 {
88     ItemsResponse<Interaction> response = new ItemsResponse<Interaction>
89     ();
90     int accountId = _userService.GetCurrentAccountId();
91
92     response.Items = _interactionsService.Get(accountId);
93     return Request.CreateResponse(response);
94 }
95
96 [Route("{Id:int}"), HttpDelete]
97 public HttpResponseMessage DeleteContact(int id)
98 {
99     SuccessResponse response = new SuccessResponse();
100     _interactionsService.DeleteContact(id);
101 }
```

## InteractionsApiController

---

```
99         return Request.CreateResponse(response);
100     }
101
102     [Route("client"), HttpGet]
103     public HttpResponseMessage GetClientInteractionsByClient()
104     {
105         ItemsResponse<ClientInteractionsByClient> response = new ItemsResponse<ClientInteractionsByClient>(); ➤
106         string clientId = _clientStateService.Read ➤
107             (ClientStateKey.CurrentClientId, Request);
108         response.Items = _interactionsService.GetAClientInteractions ➤
109             (clientId);
110         return Request.CreateResponse(response);
111     }
112
113     [Route("ping/test"), HttpPost]
114     public HttpResponseMessage InsertClientAppInteraction ➤
115         (InsertClientAppInteractionRequest model)
116     {
117         if (!ModelState.IsValid)
118         {
119             return Request.CreateErrorResponse(HttpStatusCode.BadRequest, ➤
120                 ModelState);
121         }
122         string userId = _userService.GetCurrentUserId();
123         SuccessResponse response = new SuccessResponse();
124         _interactionsService.AddClientAppInteraction(model, userId);
125         return Request.CreateResponse(response);
126     }
127
128     [Route("ping/report"), HttpGet]
129     public HttpResponseMessage GetClientAppInteractionReport()
130     {
131         int accountId = _userService.GetCurrentAccountId();
132         ItemsResponse<InteractionUserReportData> response = new ItemsResponse<InteractionUserReportData>(); ➤
133         response.Items = _interactionsService.GetClientAppInteractionReport ➤
134             (accountId);
135         return Request.CreateResponse(response);
136     }
137 }
138
139
140
141
142 }
143
```