

```

1 <script type="text/ng-template" id="myModalContent.html">
2   <div class="sarah-modal-body">
3     <a class="evolbclose" ng-click="mc.cancel()">X</a>
4     <div class="text-center">
5       <h1 style="color:red">{{mc.modalContent.name}}</h1>
6       
7     </div>
8     <div class="text-center">
9       <h2 style="color:red">Congrats! You have discovered
10         {{mc.modalContent.name}}!</h2>
11     </div>
12     <div class="form-group has-feedback myModal">
13       <label class="edited pull-left"
14         id='edited_categoryaddplaces'>Category:</label>
15       <select name='Category' class='form-control' id='category'
16         data-ng-model="mc.selectedCategory"
17         ng-options="category as category.tagName for category in
18           mc.categoryContent" ng-change="mc.onCategoryChange()"
19         required></select>
20     </div>
21     <div class="form-group has-feedback myModal">
22       <label class="edited col-lg-12 myModal">Attribute:</label>
23       <select class="form-control col-lg-6" ng-
24         options="attrib.attributeName for attrib in mc.attributes track by
25         attrib.attributeName" ng-model="mc.attribute"></select>
26       <button type="button" class="btn btn-danger" ng-
27         click="mc.addAttribute()"><i class="fa fa-plus"></i> Add</button>
28     </div>
29     <div class="form-group has-feedback myModal" ng-repeat="attrib in
30       mc.addedAttributes track by attrib.attributeName">
31       <label class="edited">{{attrib.attributeName}}</label>
32       <input type="text" class="form-control" ng-if="attrib.attributeType
33         === 0" ng-model="attrib.value" />
34       <input type="number" class="form-control" ng-if="attrib.attributeType
35         === 1 || attrib.attributeType === 2" ng-model="attrib.value" />
36     </div>
37     <div class="form-group has-feedback myModal">
38       <label class="edited pull-left" id='edited_tagsaddplaces'>Tags:</
39       label>
40       <select name='Tags' class='form-control' id='tagsSelect'
41         chosen
42         multiple
43         data-ng-model="mc.selectedFreeTag"
44         ng-options="tag as tag.tagName for tag in
45           mc.freeTagContent"></select>
46     </div>
47   </div>
48   <div class="modal-footer">
49     <button type="button" class="btn btn-danger" ng-click="mc.save()">Save</
50     button>
51   </div>
52 </script>

```

```
40 <script>
41     (function () {
42         "use strict";
43
44         angular.module(APPNAME)
45             .controller('modalController', ModalController);
46
47         ModalController.$inject = ['$scope', '$baseController',
48                                     '$uibModalInstance', 'modalContent', 'pictureContent',
49                                     'categoryContent', 'freeTagContent', 'cityContent', '$placesService',
50                                     '$utilityService', '$entitiesService'];
51
52         function ModalController(
53             $scope
54             , $baseController
55             , $uibModalInstance
56             , modalContent
57             , pictureContent
58             , categoryContent
59             , freeTagContent
60             , cityContent
61             , $placesService
62             , $utilityService
63             , $entitiesService) {
64
65             var vm = this;
66
67             vm.$entitiesService = $entitiesService;
68
69             $baseController.merge(vm, $baseController);
70
71             vm.$scope = $scope;
72             vm.notify = $entitiesService.getNotifier($scope);
73             vm.$uibModalInstance = $uibModalInstance;
74
75             vm.selectedCategory = null;
76             vm.selectedFreeTag = null,
77
78             vm.modalContent = modalContent;
79             vm.pictureContent = pictureContent;
80             vm.categoryContent = categoryContent;
81             vm.freeTagContent = freeTagContent;
82             vm.cityContent = cityContent;
83             vm.onCategoryChange = _onCategoryChange;
84             vm.addAttribute = _addAttribute;
85
86             function _onCategoryChange() {
87                 vm.addedAttributes = [];
88                 vm.attributes = [];
89
90                 return vm.$entitiesService.getEntityAttributes
91                     (vm.selectedCategory.tagName).success(onGetEntityAttributes);
92             }
93         }
94     })
95 </script>
```

```
90         function onGetEntityAttributes(data) {
91             var items = data.items;
92
93             vm.notify(function () {
94                 vm.attributes = items;
95             });
96         }
97
98         function _addAttribute() {
99             if (vm.attribute && vm.addedAttributes.indexOf(vm.attribute) === -1) {
100                 vm.addedAttributes.push(vm.attribute);
101             }
102         }
103
104         vm.save = function () {
105
106             var address_components = vm.modalContent.address_components;
107             var components = {};
108
109             $.each(address_components, function (k, v1) {
110                 $.each(v1.types, function (k2, v2) {
111                     components[v2] = v1.short_name
112                 });
113             })
114
115             var tagIds = $.map(vm.selectedFreeTag || [], function (v) {
116                 return v.id;
117             });
118
119             var placeSlug = $utilityService.toSlug(vm.modalContent.name);
120
121             if (vm.modalContent.price_level == null) {
122                 vm.modalContent.price_level = 0;
123             } else { }
124
125             console.log("Components >>", components);
126
127             var placesObject = {
128                 Name: vm.modalContent.name,
129                 Description: vm.modalContent.name,
130                 OperatingHours: vm.modalContent.opening_hours ?
131                     vm.modalContent.opening_hours.weekday_text.join('\n') : "" || "Everyday",
132                 PhoneNumber: vm.modalContent.formatted_phone_number,
133                 Website: vm.modalContent.website,
134                 CategoryId: vm.selectedCategory.id,
135                 TagIds: tagIds,
136                 Slug: placeSlug + "-" + components.postal_code,
137                 ExtPlaceId: vm.modalContent.place_id,
138                 CityId: vm.cityContent || 33,
139                 Price: vm.modalContent.price_level,
```

```

143         Address1: components.street_number + ' ' + components.route,
144         City: components.locality || components.sublocality,
145         State: components.administrative_area_level_1,
146         ZipCode: components.postal_code,
147         Latitude: vm.modalContent.geometry.location.lat(),
148         Longitude: vm.modalContent.geometry.location.lng(),
149         Url: vm.pictureContent
150     };
151
152     console.log("PAYLOAD OBJECT>>>", placesObject);
153     $placesService.insert(placesObject, onSubmitPlace,           ↗
        onSubmitPlaceError);
154
155
156     };
157
158     function onSubmitPlace(data) {
159         console.log("Post Success:", data)
160
161         vm.existingId = data.item.id;
162         $entitiesService.addRecord(vm.selectedCategory.tagName,   ↗
            data.item.id).success(onAddRecord);
163         vm.$alertService.success(data.item.name + " has been     ↗
            discovered!", "New Place Added!");
164         vm.$uibModalInstance.close();
165     }
166
167     function onAddRecord() {
168         console.log("Record Added");
169
170         var values = $.map(vm.addedAttributes, function (v) {
171             return {
172                 RecordId: vm.existingId,
173                 AttributeId: v.attributeId,
174                 Entity: vm.selectedCategory.tagName,
175                 Value: v.value
176             };
177         });
178         $entitiesService.addAttributeValues(values);
179     }
180
181     function onSubmitPlaceError(data) {
182         console.log("ERROR");
183         vm.$alertService.error("Place may already exist or city is not   ↗
            selected!", "Error!");
184     }
185
186     vm.cancel = function () {
187         vm.$uibModalInstance.dismiss('cancel');
188     };
189 }
190 })();
191 </script>

```