

```
1  (function () {
2      "use strict";
3
4
5      angular.module(APPNAME)
6          .controller('geoController', GeoController);
7
8      GeoController.$inject = ['$scope', '$baseController', '$cityService',
9          '$utilityService'];
10
11      function GeoController(
12          $scope // these three arguments are dependencies which are specified below
13              and passed into the controller by ng
14          , $baseController
15          , $cityService
16          , $utilityService
17          ) {
18
19          var vm = this; //this points to a new {}
20
21          vm.showCityErrors = false;
22
23          // inherit from $baseController
24          $.extend(vm, $baseController);
25
26          //this is a wrapper for our small dependency on $scope
27          vm.notify = $cityService.getNotifier($scope);
28
29          // hoist our public function(s)
30          vm.submitCityForm = _submitCityForm;
31          vm.slugify = _slugify;
32
33          //initialize our data and other properties for this page
34
35          vm.city = { //ngModel object
36              radius: 10,
37          };
38
39          vm.$cityService = $cityService;
40          vm.$utilityService = $utilityService;
41          vm.$scope = $scope;
42          vm.postItems = _postItems;
43          vm.updateItems = _updateItems;
44          vm.getItem = _getItem;
45
46          vm.cityIntId = null;
47          vm.newSlugName = null;
48
49          //invoke the initialize function to setup the map for this page
50          _initialize() /* Startup Function */
51
52          function _initialize() {
53              console.log("initialize is rendering");
54          }
```

```
55     vm.cityIntId = $("#editId").val(); //Grab value for view model id
56
57     if (vm.cityIntId && vm.cityIntId.length > 0) { // check if there is an id
58         console.log("EDIT mode - go get the data");
59
60         vm.$cityService.getById(vm.cityIntId, vm.getItem) //Performing a GET ajax call and sticking the data into the form on success
61     }
62
63     else {
64
65         console.log("CREATE mode - EditId is empty");
66
67         //Geocode default location on Create Screen
68         vm.geocoder = new google.maps.Geocoder();
69         var latlng = new google.maps.LatLng(40.7128, -74.0059);
70         var mapOptions = {
71             zoom: 10,
72             center: latlng
73         }
74         vm.map = new google.maps.Map($('#map-canvas')[0], mapOptions); // Initialize Map
75
76     };
77
78     $(".error").css("opacity", "1");
79 }
80
81 //Get By ID Ajax Success Handler
82 function _getItem(data) {
83
84     console.log("Got the data:", data.item);
85
86     //VM Notify puts the data back into the form fields
87     vm.notify(function () {
88         vm.city = data.item;
89     });
90
91     //Geocode saved lat/long of city for edit screen
92     vm.geocoder = new google.maps.Geocoder();
93     var latlng = { lat: data.item.latitude, lng: data.item.longitude };
94     var mapOptions = {
95         zoom: 10,
96         center: latlng
97     }
98     vm.map = new google.maps.Map($('#map-canvas')[0], mapOptions); // initialize the map
99
100     //Drop marker on lat/lng location
101     var marker = new google.maps.Marker({
102         map: vm.map,
103         position: latlng
104     });
105
106 }
```

```
107
108
109     //Ng-Submit Form Hanlder
110     function _submitCityForm() {
111
112         var addressString = vm.city.address;
113
114         _codeAddress(addressString);
115
116         vm.showCityErrors = true;
117     }
118
119     //Create a slug on blur
120     function _slugify() {
121
122         //Create Slug from Name Input via toSlug Utility
123         vm.city.slug = $utilityService.toSlug(vm.city.name);
124     }
125
126     //Function to geocode city
127     function _codeAddress(address) {
128
129         vm.geocoder.geocode({ 'address': address }, _onCodeAddress); //
130         Geocode address
131     }
132
133     //Geocode Success Handler
134     function _onCodeAddress(results, status) {
135
136         if (status === google.maps.GeocoderStatus.OK) {
137
138             var geometry = results[0].geometry;
139             var loc = geometry.location;
140
141             console.log("Got location data from Google API", loc);
142
143             //center map to geocoded location
144             vm.map.setCenter(loc);
145
146             //Drop marker on geocoded location
147             var marker = new google.maps.Marker({
148                 map: vm.map,
149                 position: loc
150             });
151
152             if (geometry.viewport)
153                 vm.map.fitBounds(geometry.viewport);
154
155
156             var lat = loc.lat();
157             var lon = loc.lng();
158
159             console.log("Got the coordinates for city -> (%s, %s)", lat,
160                 lon);
161
162             //Declare lat/long to controller
```

```
162         vm.city.latitude = lat;
163         vm.city.longitude = lon;
164
165         _saveCity();
166
167     } else {
168
169         vm.$alertService.error('Geocode was not successful for the
           following reason: ' + status); //alerts built with the angular-
           toastr library.
170     }
171 }
172
173
174
175 function _saveCity() {
176
177     var cityId = vm.city.id;
178
179     if (vm.cityForm.$valid) { //Check to see if form fields are valid
180         console.log("data is valid! go save this object -> ", vm.city);
181
182         if (cityId && cityId.toString().length > 0) { //If there is a
           city Id we update
183             console.log("UPDATE City Data", vm.city);
184
185             vm.$cityService.update(vm.city.id, vm.city,
           vm.updateItems); //Ajax Update Call
186         }
187         else { // if not we post
188             console.log("CREATE City Data", vm.city);
189
190             vm.$cityService.insert(vm.city, vm.postItems); //Ajax Post
           Call
191         }
192     }
193
194     else {
195
196         console.log("form submitted with invalid data")
197     }
198 }
199
200 }
201
202 //Post Success Handler
203 function _postItems(data) {
204
205
206     vm.$alertService.success(data.item.name + " has been created!",
           "Create Success"); //alerts built with the angular-toastr library.
207
208     console.log("Post City Success", data.item.name);
209
210 }
211
212 //Update Success Handler
```

```
213     function _updateItems(data) {
214
215         vm.$alertService.success(vm.city.name + " has been updated!", "Update ⚡
           Success"); //alerts built with the angular-toastr library.
216
217         console.log("Update City Success");
218
219     }
220
221 }
222
223 }());
224
225
```