

```

1  @model Sabio.Web.Models.ViewModels.ItemViewModel<int?>
2
3  @{
4      ViewBag.Title = "IndexTheme";
5      Layout = "~/Views/Shared/_LayoutAlternate.cshtml";
6  }
7
8  <div class="container submit-cont">
9      <div class="row">
10         <form id="placesForm">
11             <div class="col-lg-8 col-md-offset-2">
12                 <div class="submit_form">
13                     <div class="form-group has-feedback">
14                         <h2 class="text-center">Submit Place</h2>
15                     </div>
16                     <div class="form-group has-feedback">
17                         <input id="geocomplete" type="text" placeholder="Enter Place" />
18                         <button id="findMyLocation" class="btn btn-danger btn-sm pull-right" type="button">Find My Location</button>
19                         <div id="map" class="map_canvas"></div>
20                     </div>
21                     <div class="form-group has-feedback">
22                         <label for="placesName">Name:</label>
23                         <input type="text" class="form-control" id="placesName" placeholder="Name" name="name" required />
24
25                         <input type="hidden" id="extPlaceId" name="place_id" />
26                         <input type="hidden" id="latitude" name="lat" />
27                         <input type="hidden" id="longitude" name="lng" />
28                         <input type="hidden" id="locationId" name="locationID" />
29                         <input type="hidden" id="googleImg" name="googleImg" />
30                     </div>
31                     <div class="form-group has-feedback">
32                         <label class='edited' id='edited_categoryaddplaces'>Category:</label>
33                         <select name='Category' class='form-control' id='category'></select>
34                     </div>
35                     <div class="form-group has-feedback">
36                         <label class='edited' id='edited_tagsaddplaces'>Tags:</label>
37                         <select name='Tags' class='form-control' id='tagsSelect' multiple></select>
38                     </div>
39                     <div class="form-group has-feedback">
40                         <label for="placesAddress">Street Number:</label>
41                         <input type="text" class="form-control" id="placesStreetNumber" placeholder="Street Number" name="street_number" required />
42                     </div>
43                     <div class="form-group has-feedback">
44                         <label for="placesAddress">Address:</label>

```

```
44         <input type="text" class="form-control"
           id="placesAddress" placeholder="Address" name="route"
           required />
45     </div>
46     <div class="form-group has-feedback">
47         <label for="placesCity">City:</label>
48         <input type="text" class="form-control" id="placesCity"
           placeholder="City" name="locality" required />
49     </div>
50     <div class="form-group has-feedback">
51         <label for="placesState">State:</label>
52         <input type="text" class="form-control" id="placesState"
           placeholder="State" name="administrative_area_level_1_short"
           required />
53     </div>
54     <div class="form-group has-feedback">
55         <label for="placesZipCode">Zip Code:</label>
56         <input type="text" class="form-control"
           id="placesZipCode" placeholder="Zip Code" name="postal_code"
           required />
57     </div>
58     <div class="form-group has-feedback">
59         <label for="placesPhoneNumber">Phone Number:</label>
60         <input type="text" class="form-control"
           id="placesPhoneNumber" placeholder="Phone Number"
           name="formatted_phone_number" required />
61     </div>
62     <div class="form-group has-feedback">
63         <label for="placesOperatingHours">Operating Hours:</
           label>
64         <textarea type="text" class="form-control" rows="3"
           id="placesOperatingHours" placeholder="Operating Hours"
           name="opening_hours" required></textarea>
65     </div>
66     <div class="form-group has-feedback">
67         <label for="placesWebsite">Website:</label>
68         <input type="text" class="form-control"
           id="placesWebsite" placeholder="Website" name="website"
           required />
69     </div>
70     <div class="form-group has-feedback">
71         <label for="placesDescription">Description:</label>
72         <textarea class="form-control" rows="3"
           id="placesDescription" placeholder="Description"
           name="Description" required></textarea>
73     </div>
74     <div class="form-group has-feedback">
75         <label for="slug">URL Slug:</label>
76         <input type="text" class="form-control" id="slug"
           placeholder="URL Slug" name="slug" required />
77     </div>
78     <div>
```

```
79         <input type="submit" value="Submit Place" class="btn btn-
            danger col-md-3 pull-right" id="submitPlaces">
80     </div>
81     <br>
82     <br>
83 </div>
84     <input type="hidden" id="editId" value="@Model.Item" />
85 </div>
86 </form>
87 </div>
88 </div>
89
90
91
92
93 <link href="~/Assets/bower_components/chosen/chosen.css" rel="stylesheet" />
94 <link href="~/Assets/bower_components/geocomplete/examples/styles.css"
    rel="stylesheet" />
95
96
97 @section scripts {
98     @Scripts.Render("~/bundles/jqueryval")
99     <script src="~/Scripts/sabio.services.places.js"></script>
100     <script src="~/Scripts/sabio.tag.services.js"></script>
101     <script src="~/Scripts/sabio.services.addresses.js"></script>
102     <script src="~/Scripts/sabio.services.util.js"></script>
103     <script src="~/Assets/bower_components/chosen/chosen.jquery.js"></script>
104     <script src="https://maps.googleapis.com/maps/api/js?
        key=AIzaSyB0yN-78Gmbvb1fdVJ1mwBCjaAMRK-ZFDY&libraries=places"></script>
105     <script src="~/Assets/bower_components/geocomplete/jquery.geocomplete.js"></
        script>
106
107     <script type="text/template" class="categoryTemplate">
108         <option class="tagsOption"></option>
109     </script>
110
111     <script type="text/template" class="tagsTemplate">
112         <option class="tagsSelectOption"></option>
113     </script>
114
115     <script type="text/javascript">
116
117         sabio.page.placesIntId = null; //initialize a property to hold the id
118
119         sabio.page.startUp = function () {
120             console.log("Startup is running");
121
122             sabio.page.placesIntId = $("#editId").val();
123
124             if (sabio.page.placesIntId && sabio.page.placesIntId.length > 0) {
125                 console.log("EDIT mode - go get the data");
126
127
128                 sabio.services.places.getById(sabio.page.placesIntId,
```

```
sabio.page.handlers.getIdSuccess); //Performing a GET ajax call and sticking the data into the form on success
130     }
131     else {
132
133         //Ajax Call to get tags for categories dropdown
134         sabio.tag.services.apiSearchTypeById(1, sabio.page.handlers.getTagIdSuccess);
135
136         //Ajax Call to get tags for tags dropdown
137         sabio.tag.services.apiSearchTypeById(2, sabio.page.handlers.getTagSelectSuccess);
138
139         //Geocomplete Object
140         var options = {
141             map: ".map_canvas",
142             mapOptions: {
143                 center: { lat: 40.7128, lng: -74.0059 },
144                 zoom: 9
145             },
146             details: "form",
147             types: ["geocode", "establishment"],
148
149         };
150
151         //Trigger Geocomplete Object
152         $("#geocomplete").geocomplete(options).bind("geocode:result", sabio.page.handlers.geocompleteResultCreate ); //handler to pull operating hours from google api
153
154         console.log("CREATE mode - EditId is empty");
155     };
156
157     //call a separate function to set up validation rules
158     sabio.page.initializeValidation();
159
160     //Submit Form Binding
161     $("#submitPlaces").on("click", sabio.page.handlers.submitPlacesForm);
162
163     //Tag Id Array
164     sabio.page.tagIds = [];
165
166     //Initalizes Google Maps API Geocoder
167     sabio.page.geocoder = new google.maps.Geocoder();
168
169     //Binding for GeoLocation
170     $("#findMyLocation").on("click", sabio.page.handlers.findMyLocation);
171
172
173     };
174
175
176
177
178     //jquery client side validation
179     sabio.page.initializeValidation = function () {
```

```
180         //set defaults for the entire plugin
181         jQuery.validator.setDefaults({
182             debug: true
183         });
184
185         //set rules and error messages
186         $('#placesForm').validate({
187             rules: {
188                 "Name": {
189                     required: true
190                 },
191                 "Description": {
192                     required: true
193                 },
194                 "Tags": {
195                     required: false
196                 }
197             },
198         });
199     }
200
201
202
203     //Geocomplete Result Hanlder for Creating a new place
204     sabio.page.handlers.geocompleteResultCreate = function (event, result) {
205         console.log("Google Place Result:", result);
206
207
208         var largestPicWidth = 0;
209
210         //Loop through pics finding the largest width
211         for (var i = 0; i < result.photos.length; i++) {
212
213             if (result.photos[i].width > largestPicWidth) {
214                 largestPicWidth = result.photos[i].width;
215                 sabio.page.largestPic = result.photos[i].getUrl({ 'maxWidth': ↗
216                     largestPicWidth });
217             }
218             console.log("Width:", largestPicWidth, "Largest Width Pic:", ↗
219                 sabio.page.largestPic);
220
221             //pull value from name field
222             var slugName = $('#placesName').val();
223
224             //Use created utility service to convert name to slug
225             var slug = sabio.services.util.toSlug(slugName);
226
227             //Input converted slug into slug form field
228             $('#slug').val(slug);
229
230             //Input operating hours from google places api result
231             $('#placesForm').find("#placesOperatingHours").val
232                 (result.opening_hours.weekday_text);
233
234             $('#placesForm').find("#googleImg").val(sabio.page.largestPic);
```

```
233     };
234
235
236
237     //Submit Places Form Handler
238     sabio.page.handlers.submitPlacesForm = function () {
239         event.preventDefault();
240
241         if ($('#placesForm').valid()) {
242
243             var apiPlacesName = $("#placesName").val();
244             var apiPlacesDescription = $("#placesDescription").val();
245             var apiPlacesOperatingHours = $("#placesOperatingHours").val();
246             var apiPhoneNumber = $("#placesPhoneNumber").val();
247             var apiWebsite = $("#placesWebsite").val();
248             var apiCategory = $("#category").val();
249             var apiTagIds = $("#tagsSelect").val();
250             sabio.page.apiSlug = $("#slug").val();
251             var apiLocationId = $("#locationId").val();
252             var apiExtPlaceId = $("#extPlaceId").val();
253             var apiStreetNumber = $("#placesStreetNumber").val();
254             var apiStreetAddress = $("#placesAddress").val();
255             var apiCity = $("#placesCity").val();
256             var apiState = $("#placesState").val();
257             var apiZipCode = $("#placesZipCode").val();
258             var apiLat = $("#latitude").val();
259             var apiLng = $("#longitude").val();
260             var apiUrl = $("#googleImg").val();
261
262
263             var placesObject = {
264                 Name: apiPlacesName,
265                 Description: apiPlacesDescription,
266                 OperatingHours: apiPlacesOperatingHours,
267                 PhoneNumber: apiPhoneNumber,
268                 Website: apiWebsite,
269                 CategoryId: apiCategory,
270                 TagIds: apiTagIds,
271                 Slug: sabio.page.apiSlug,
272                 ExtPlaceId: apiExtPlaceId,
273                 LocationId: apiLocationId,
274                 Address1: apiStreetNumber + " " + apiStreetAddress,
275                 City: apiCity,
276                 State: apiState,
277                 ZipCode: apiZipCode,
278                 Latitude: apiLat,
279                 Longitude: apiLng,
280                 Url: apiUrl,
281
282             };
283
284
285
286             //Create an if statement to update existing posted Place
```

```
287         if (sabio.page.placesIntId) {
288             //Create a namespaced variable for Places Id
289             placesObject.id = sabio.page.placesIntId;
290             //Ajax Update Call
291             sabio.services.places.update(sabio.page.placesIntId,
292                                     placesObject, sabio.page.handlers.updateSuccess);
293         } else {
294             //Ajax Post Call
295             sabio.services.places.insert(placesObject,
296                                     sabio.page.handlers.placesPostSuccess);
297         };
298     }
299     else { }
300 }
301
302 };
303
304 sabio.page.handlers.addressPostSuccess = function (data) {
305     console.log("Post Address Success", data)
306 };
307
308 sabio.page.handlers.addressUpdateSuccess = function (data) {
309     console.log("Update Address Success", data);
310 };
311
312 //Post Success Handler
313 sabio.page.handlers.placesPostSuccess = function (data) {
314     console.log("POST Places SUCCESS", data);
315
316     sabio.page.placesId = data.item.id;
317
318     //Reroutes to the singlePlace page of the place we just created
319     window.location.href = "/" + "places/singleplace/" + data.item.slug;
320 };
321
322 //Update Success Handler
323 sabio.page.handlers.updateSuccess = function (data) {
324     console.log("update places success");
325
326     //Reroutes to the singlePlace page of the place we just created
327     window.location.href = "/" + "places/singleplace/" + data.item.slug;
328 };
329
330 //Get Success Handler sticking the data into the form
331 sabio.page.handlers.getIdSuccess = function (data) {
332     console.log("got the id data for update:", data);
333
334     $("#placesForm").find("#placesName").val(data.name);
335     $("#placesForm").find("#placesOperatingHours").val
336         (data.operatingHours);
```

```
341     $("#placesForm").find("#placesDescription").val(data.description);
342     $("#placesForm").find("#category").val(data.categoryId);
343     $("#placesForm").find("#placesPhoneNumber").val(data.phoneNumber);
344     $("#placesForm").find("#placesWebsite").val(data.website);
345     $("#placesForm").find("#slug").val(data.slug);
346     $("#placesForm").find("#locationId").val(data.locationId);
347     $("#placesForm").find("#googleImg").val(data.media.url);
348
349
350     sabio.page.formattedAddress = (data.name + " " +
        data.address.address1 + ", " + data.address.city + ", " +
        data.address.state + " " + data.address.zipCode);
351
352     var options = {
353         map: ".map_canvas",
354         location: sabio.page.formattedAddress,
355         types: ["geocode", "establishment"],
356     };
357
358
359     $("#geocomplete").geocomplete(options).bind("geocode:result",
        sabio.page.handlers.geocompleteResultUpdate); //Geocomplete result
        update handler to input converted address components into from
        fields for update
360
361
362     for (var i = 0; i < data.tags.length; i++) {
363
364         sabio.page.tagIds.push(data.tags[i].id);
365
366         console.log("tags for id:", data.tags[i].tagName);
367     }
368
369     //Ajax Call to get tags for categories dropdown
370     sabio.tag.services.apiSearchTypeById(1,
        sabio.page.handlers.getTagIdSuccess);
371
372     //Ajax Call to get tags for tags dropdown
373     sabio.tag.services.apiSearchTypeById(2,
        sabio.page.handlers.getTagSelectSuccess);
374
375
376     // namespaced variable for category ID
377     sabio.page.categoryId = data.categoryId;
378
379 };
380
381 //Geocomplete result handler to input converted address components into
    form fields for update
382 sabio.page.handlers.geocompleteResultUpdate = function (event, result) {
383     console.log("Update Result:" , result);
384
385     //Utility for address components decoder
386     var components = sabio.services.util.decodeAddress(result)
387
388     console.log("address components:" , components);
```



```
389
390     $("#placesForm").find("#placesStreetNumber").val
        (components.street_number);
391     $("#placesForm").find("#placesAddress").val(components.route);
392     $("#placesForm").find("#placesCity").val(components.locality);
393     $("#placesForm").find("#placesState").val
        (components.administrative_area_level_1);
394     $("#placesForm").find("#placesZipCode").val(components.postal_code);
395
396
397     $("#placesForm").find("#extPlaceId").val(result.place_id);
398     $("#placesForm").find("#latitude").val(result.geometry.location.lat);
399     $("#placesForm").find("#longitude").val
        (result.geometry.location.lng);
400
401
402 };
403
404
405 //Get success handler for CATEGORY names
406 sabio.page.handlers.getTagIdSuccess = function (data) {
407
408     //For loop to populate Categories in the dropdown for places form
409     for (var i = 0; i < data.length; i++) {
410
411         var category = $(".categoryTemplate").html();
412         var categoryClone = $(category).clone();
413
414         categoryClone.val(data[i].id).text(data[i].tagName);
415
416         if (data[i].id == sabio.page.categoryId) {
417             categoryClone.attr("selected", true);
418         }
419
420         categoryClone.appendTo("#category");
421     }
422 };
423
424
425 //Get success handler for TAG names
426 sabio.page.handlers.getTagSelectSuccess = function (data) {
427
428     //For loop to populate Tags in the dropdown for places form
429     for (var i = 0; i < data.length; i++) {
430
431         var tags = $(".tagsTemplate").html();
432         var tagsClone = $(tags).clone();
433
434         tagsClone.val(data[i].id).text(data[i].tagName);
435
436         if (sabio.page.tagIds.indexOf(data[i].id) > -1) {
437             tagsClone.attr("selected", true);
438         }
439
440         tagsClone.appendTo("#tagsSelect");
441     }
442 }
```

```
443
444     //Chosen jquery plugin for tags field
445     $("#tagsSelect").chosen();
446 };
447
448 //Reverse Geocode Success Handler
449 sabio.page.handlers.onReverseGeoSuccess = function (results, status) {
450     if (results && results.length) {
451         //Inititalize geocomplete with formatted address of current
452         //Latitude and Longitude
453         $("#geocomplete").val(results[0].formatted_address)
454             .trigger("geocode");
455     }
456 }
457
458 //Get Current Position Success Handler
459 sabio.page.handlers.onCurrentPosition = function (position) {
460     console.log("Show My Location:", position);
461
462     //Variable Grab Latitude and longitude of current position
463     var latlng = new google.maps.LatLng(position.coords.latitude,
464         position.coords.longitude);
465
466     //Google API Geocode lat and lng
467     sabio.page.geocoder.geocode({
468         'latLng': latlng
469     }, sabio.page.handlers.onReverseGeoSuccess);
470 }
471
472 //Get Current Position Error Handler
473 sabio.page.handlers.onCurrentPositionError = function (position) {
474     alert("Unable to load location from your browser - please enable your
475         location");
476 }
477
478 //Handler for geolocation / get current location
479 sabio.page.handlers.findMyLocation = function () {
480     if (navigator.geolocation) {
481         navigator.geolocation.getCurrentPosition(
482             sabio.page.handlers.onCurrentPosition,
483             sabio.page.handlers.onCurrentPositionError);
484     } else {
485         //client doesn't support the navigator object
486         alert("Unable to load location from your browser");
487     }
488 }
489
490 </script>
```