|  |
| --- |
| <!DOCTYPE html> |
|  | <html> |
|  |  |
|  | <head> |
|  | <meta charset='utf-8' /> |
|  | <title></title> |
|  | <meta name='viewport' content='initial-scale=1,maximum-scale=1,user-scalable=no' /> |
|  | <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.0/jquery.min.js"></script> |
|  | <script src="https://api.mapbox.com/mapbox-gl-js/v1.9.1/mapbox-gl.js"></script> |
|  | <link href="https://api.mapbox.com/mapbox-gl-js/v1.9.1/mapbox-gl.css" rel="stylesheet" /> |
|  | <script src='https://npmcdn.com/csv2geojson@latest/csv2geojson.js'></script> |
|  | <script src='https://npmcdn.com/@turf/turf/turf.min.js'></script> |
|  | <style> |
|  | body { |
|  | margin: 0; |
|  | padding: 0; |
|  | } |
|  |  |
|  | #map { |
|  | position: absolute; |
|  | top: 0; |
|  | bottom: 0; |
|  | width: 100%; |
|  | } |
|  |  |
|  | /\* Popup styling \*/ |
|  |  |
|  | .mapboxgl-popup { |
|  | padding-bottom: 5px; |
|  | } |
|  |  |
|  | .mapboxgl-popup-close-button { |
|  | display: none; |
|  | } |
|  |  |
|  | .mapboxgl-popup-content { |
|  | font: 400 15px/22px 'Source Sans Pro', 'Helvetica Neue', Sans-serif; |
|  | padding: 0; |
|  | width: 250px; |
|  | } |
|  |  |
|  | .mapboxgl-popup-content-wrapper { |
|  | padding: 1%; |
|  | } |
|  |  |
|  | .mapboxgl-popup-content h3 { |
|  | background: rgb(61, 59, 59); |
|  | text-align: center; |
|  | color: #fff; |
|  | margin: 0; |
|  | display: block; |
|  | padding: 15px; |
|  | font-weight: 700; |
|  | margin-top: -5px; |
|  | } |
|  |  |
|  | .mapboxgl-popup-content h4 { |
|  | margin: 0; |
|  | display: block; |
|  | padding: 10px 3px 10px 10px; |
|  | font-weight: 400; |
|  | } |
|  |  |
|  | .mapboxgl-container { |
|  | cursor: pointer; |
|  | } |
|  |  |
|  | .mapboxgl-popup-anchor-top>.mapboxgl-popup-content { |
|  | margin-top: 3px; |
|  | } |
|  |  |
|  | .mapboxgl-popup-anchor-top>.mapboxgl-popup-tip { |
|  | border-bottom-color: rgb(61, 59, 59); |
|  | } |
|  | </style> |
|  | </head> |
|  |  |
|  | <body> |
|  |  |
|  | <div id='map'></div> |
|  | <script> |
|  |  |
|  | var transformRequest = (url, resourceType) => { |
|  | var isMapboxRequest = |
|  | url.slice(8, 22) === "api.mapbox.com" || |
|  | url.slice(10, 26) === "tiles.mapbox.com"; |
|  | return { |
|  | url: isMapboxRequest |
|  | ? url.replace("?", "?pluginName=sheetMapper&") |
|  | : url |
|  | }; |
|  | }; |
|  | //YOUR TURN: add your Mapbox token |
|  | mapboxgl.accessToken = '<Add your Mapbox token >'; //Mapbox token |
|  | var map = new mapboxgl.Map({ |
|  | container: 'map', // container id |
|  | style: 'mapbox://styles/mapbox/streets-v11', //stylesheet location |
|  | center: [-122.411464, 37.7852299], // starting position |
|  | zoom: 10,// starting zoom |
|  | transformRequest: transformRequest |
|  | }); |
|  |  |
|  | $(document).ready(function () { |
|  | $.ajax({ |
|  | type: "GET", |
|  | //YOUR TURN: Replace with csv export link |
|  | url: '<https://docs.google.com/spreadsheets/d/1KUX1m\_ns-GLW4oiEkDr6ueadZ-QACuQpMYTeE53KhBg/gviz/tq?tqx=out:csv&sheet=Sheet1>', |
|  | dataType: "text", |
|  | success: function (csvData) { makeGeoJSON(csvData); } |
|  | }); |
|  |  |
|  |  |
|  |  |
|  | function makeGeoJSON(csvData) { |
|  | csv2geojson.csv2geojson(csvData, { |
|  | latfield: 'Latitude', |
|  | lonfield: 'Longitude', |
|  | delimiter: ',' |
|  | }, function (err, data) { |
|  | map.on('load', function () { |
|  |  |
|  | //Add the the layer to the map |
|  | map.addLayer({ |
|  | 'id': 'csvData', |
|  | 'type': 'circle', |
|  | 'source': { |
|  | 'type': 'geojson', |
|  | 'data': data |
|  | }, |
|  | 'paint': { |
|  | 'circle-radius': 5, |
|  | 'circle-color': "purple" |
|  | } |
|  | }); |
|  |  |
|  |  |
|  | // When a click event occurs on a feature in the csvData layer, open a popup at the |
|  | // location of the feature, with description HTML from its properties. |
|  | map.on('click', 'csvData', function (e) { |
|  | var coordinates = e.features[0].geometry.coordinates.slice(); |
|  |  |
|  | //set popup text |
|  | //You can adjust the values of the popup to match the headers of your CSV. |
|  | // For example: e.features[0].properties.Name is retrieving information from the field Name in the original CSV. |
|  | var description = `<h3>` + e.features[0].properties.Name + `</h3>` + `<h4>` + `<b>` + `Address: ` + `</b>` + e.features[0].properties.Address + `</h4>` + `<h4>` + `<b>` + `Phone: ` + `</b>` + e.features[0].properties.Phone + `</h4>`; |
|  |  |
|  | // Ensure that if the map is zoomed out such that multiple |
|  | // copies of the feature are visible, the popup appears |
|  | // over the copy being pointed to. |
|  | while (Math.abs(e.lngLat.lng - coordinates[0]) > 180) { |
|  | coordinates[0] += e.lngLat.lng > coordinates[0] ? 360 : -360; |
|  | } |
|  |  |
|  | //add Popup to map |
|  |  |
|  | new mapboxgl.Popup() |
|  | .setLngLat(coordinates) |
|  | .setHTML(description) |
|  | .addTo(map); |
|  | }); |
|  |  |
|  | // Change the cursor to a pointer when the mouse is over the places layer. |
|  | map.on('mouseenter', 'csvData', function () { |
|  | map.getCanvas().style.cursor = 'pointer'; |
|  | }); |
|  |  |
|  | // Change it back to a pointer when it leaves. |
|  | map.on('mouseleave', 'places', function () { |
|  | map.getCanvas().style.cursor = ''; |
|  | }); |
|  |  |
|  | var bbox = turf.bbox(data); |
|  | map.fitBounds(bbox, { padding: 50 }); |
|  |  |
|  | }); |
|  |  |
|  | }); |
|  | }; |
|  | }); |
|  |  |
|  |  |
|  |  |
|  |  |
|  | </script> |
|  |  |
|  | </body> |
|  |  |
|  | </html> |