# Adding Cluster Map

====553 Adding Earthquake cluster Map=====

<https://docs.mapbox.com/mapbox-gl-js/example/cluster/>

## views/campgrounds/index.ejs

<div id=”map” style=”width: 100%; height: 500px”></div>

<script>

const mapBoxToken = ‘<%- process.env.MAPBOX\_TOKEN %>’

const campgrounds = <%- {features: JSON.stringify(campgrounds)} %>

</script>

<script scr=”/javascript/clusterMap.js></script>

## public/javascript/clusterMap.js

~~<!DOCTYPE html>~~

~~<html>~~

~~<head>~~

~~<meta charset="utf-8">~~

~~<title>Create and style clusters</title>~~

~~<meta name="viewport" content="initial-scale=1,maximum-scale=1,user-scalable=no">~~

~~<link href="https://api.mapbox.com/mapbox-gl-js/v2.8.2/mapbox-gl.css" rel="stylesheet">~~

~~<script src="https://api.mapbox.com/mapbox-gl-js/v2.8.2/mapbox-gl.js"></script>~~

~~<style>~~

~~body { margin: 0; padding: 0; }~~

~~#map { position: absolute; top: 0; bottom: 0; width: 100%; }~~

~~</style>~~

~~</head>~~

~~<body>~~

~~<div id="map"></div>~~

~~<script>~~

mapboxgl.accessToken = mapBoxTopen

const map = new mapboxgl.Map({

container: 'map',

style: 'mapbox://styles/mapbox/dark-v10',

center: [-103.5917, 40.6699],

zoom: 3

});

map.on('load', () => {

console.log(‘Map Loaded’);

// Add a new source from our GeoJSON data and

// set the 'cluster' option to true. GL-JS will

// add the point\_count property to your source data.

map.addSource('~~earthquake~~campgrounds', {

type: 'geojson',

// Point to GeoJSON data. This example visualizes all M1.0+ earthquakes

// from 12/22/15 to 1/21/16 as logged by USGS' Earthquake hazards program.

data: ~~'https://docs.mapbox.com/mapbox-gl-js/assets/earthquakes.geojson'~~,

cluster: true,

clusterMaxZoom: 14, // Max zoom to cluster points on

clusterRadius: 50 // Radius of each cluster when clustering points (defaults to 50)

});

map.addLayer({

id: 'clusters',

type: 'circle',

source: '~~earthquake~~campgrounds',

filter: ['has', 'point\_count'],

paint: {

// Use step expressions (https://docs.mapbox.com/mapbox-gl-js/style-spec/#expressions-step)

// with three steps to implement three types of circles:

// \* Blue, 20px circles when point count is less than 100

// \* Yellow, 30px circles when point count is between 100 and 750

// \* Pink, 40px circles when point count is greater than or equal to 750

'circle-color': [

'step',

['get', 'point\_count'],

'#51bbd6',

100,

'#f1f075',

750,

'#f28cb1'

],

'circle-radius': [

'step',

['get', 'point\_count'],

20,

100,

30,

750,

40

]

}

});

map.addLayer({

id: 'cluster-count',

type: 'symbol',

source: '~~earthquake~~campgrounds',

filter: ['has', 'point\_count'],

layout: {

'text-field': '{point\_count\_abbreviated}',

'text-font': ['DIN Offc Pro Medium', 'Arial Unicode MS Bold'],

'text-size': 12

}

});

map.addLayer({

id: 'unclustered-point',

type: 'circle',

source: '~~earthquake~~campgrounds',

filter: ['!', ['has', 'point\_count']],

paint: {

'circle-color': '#11b4da',

'circle-radius': 4,

'circle-stroke-width': 1,

'circle-stroke-color': '#fff'

}

});

// inspect a cluster on click

map.on('click', 'clusters', (e) => {

console.log(‘Click on clusters’);

const features = map.queryRenderedFeatures(e.point, {

layers: ['clusters']

});

const clusterId = features[0].properties.cluster\_id;

map.getSource('~~earthquake~~campgrounds').getClusterExpansionZoom(

clusterId,

(err, zoom) => {

if (err) return;

map.easeTo({

center: features[0].geometry.coordinates,

zoom: zoom

});

}

);

});

// When a click event occurs on a feature in

// the unclustered-point layer, open a popup at

// the location of the feature, with

// description HTML from its properties.

map.on('click', 'unclustered-point', function (e) => {

const {popUpMarkup} = e.features[0].properties;

~~console.log(‘Uncluster click’);~~

const coordinates = e.features[0].geometry.coordinates.slice();

~~const mag = e.features[0].properties.mag;~~

~~const tsunami =~~

~~e.features[0].properties.tsunami === 1 ? 'yes' : 'no';~~

// 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;

}

new mapboxgl.Popup()

.setLngLat(coordinates)

.setHTML(popUpMarkup

`~~magnitude: ${mag}<br>Was there a tsunami?: ${tsunami}`~~

~~‘<h3>Campgrounds</h3>’~~

)

.addTo(map);

});

map.on('mouseenter', 'clusters', () => {

console.log(‘Mousing over cluster’);

map.getCanvas().style.cursor = 'pointer';

});

map.on('mouseleave', 'clusters', () => {

map.getCanvas().style.cursor = '';

});

});

~~</script>~~

~~</body>~~

~~</html>~~

====555 basic campground cluster =====

====556 Tweeting cluster code =====

====558 Adding custom popup =====

## model/campground

const opts = {toJson: {virtuals : true}};

campgroundSchema.virtual(‘properties.popUpMarkup’).get(function(){

return ‘<strong><a href=”/campgrounds/${this.\_id}”>${this.title}</a></strong>

<p>${this.description}.substring(0,20)</p>’