13.5.4 Add Multiple Maps

Before Sadhana shows you how to map GeoJSON LineString data, she's going to walk you through adding another map so that you can toggle between maps to visualize the data. One of the features you'll be working on for the earthquake data is different map styles, which will allow individuals to choose how they want to visualize the data.

Adding another map to showcase the data is a nice feature for a map. Most mobile apps with a mapping service use two different styles, or layers: a satellite layer and a dark layer. This makes the app visually appealing and brings functionality to the map.

To create two map choices, we'll edit the `logic.js` file for mapping the major airports without the popup markers. We'll move some code to make it more readable, and we'll add more code to the `logic.js` file.

To add another map, we'll use the Leaflet Layers Control (see the **documentation (https://leafletjs.com/examples/layers-control/)**). The Layers Control allows us to control which layers, or styles, we'll see on our map. For this task, we'll work with the streets and dark layers.

1. First, we'll add another `tileLayer()` to create a dark map. If you don't have the `tileLayer()` code for the dark map, add the following code block below the code for the streets map.

```
// We create the dark view tile layer that will be an option for our map
let dark = L.tileLayer('https://api.mapbox.com/styles/v1/mapbox/dark-v10
attribution: 'Map data © <a href="https://www.openstreetmap.org/">OpenSt
    maxZoom: 18,
```

```
    accessToken: API_KEY
});
```

2. Don't add the `addTo(map)` at the end of your streets or dark `tileLayer()` code. We'll add it later.

3. At this point, your `logic.js` file should look like the following:

```
// We create the street view tile layer that will be the default background of our map.
let streets = L.tileLayer('https://api.mapbox.com/styles/v1/mapbox/streets-v11/tiles/{z}/{x}/{y}?access_token={accessToken}', {
    attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a> contributors, <a href="https://creati
    maxZoom: 18,
    accessToken: API_KEY
});

// We create the dark view tile layer that will be an option for our map.
let dark = L.tileLayer('https://api.mapbox.com/styles/v1/mapbox/dark-v10/tiles/{z}/{x}/{y}?access_token={accessToken}', {
    attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a> contributors, <a href="https://creati
    maxZoom: 18,
    accessToken: API_KEY
});
```

4. Next, we'll add both map variables to a new variable, `baseMaps`. This variable will be used as our base layer, which we'll reference later.

5. After the code for the dark map, add the following variable to reference the base layer:

```
// Create a base layer that holds both maps.
let baseMaps = {
  Street: streets,
  Dark: dark
};
```

6. In the base layer code, the Street and Dark keys set the text, which we'll see in the `index.html` file, while the corresponding values reference the tile layers. Street and Dark can be used to toggle between styles in the `index.html` file and will look like the following:

7. Modify the `map` object to change the center and zoom level, and add the base layer with the default map. For the `map` object, we won't use the `setView()` method; instead, we'll apply the alternative method that we used earlier in this module.

REWIND

An alternative to using the `setView()` method is to modify each attribute in the map object using the curly braces notation as follows:

```
// Create the map object with a center and zoom level.
let map = L.map("mapid", {
  center: [40.7, -94.5],
  zoom: 4
});
```

8. Add the following code after the base layer code:

```
// Create the map object with center, zoom level and default layer.
let map = L.map('mapid', {
    center: [30, 30],
    zoom: 2,
    layers: [streets]
})
```

9. To complete the code for the map layers, use the Leaflet `control.layers`, which will control the layers we'll see on the map. Add the following code below the `map` object:

```
// Pass our map layers into our layers control and add the layers contro
L.control.layers(baseMaps).addTo(map);
```

10. When creating the Layers Control, the argument passed, `baseMaps`, is the base layer object, which will allow the two different map styles to be shown on the `index.html` file. The Layers Control will look like the following before it is clicked to show the Street and Dark options:

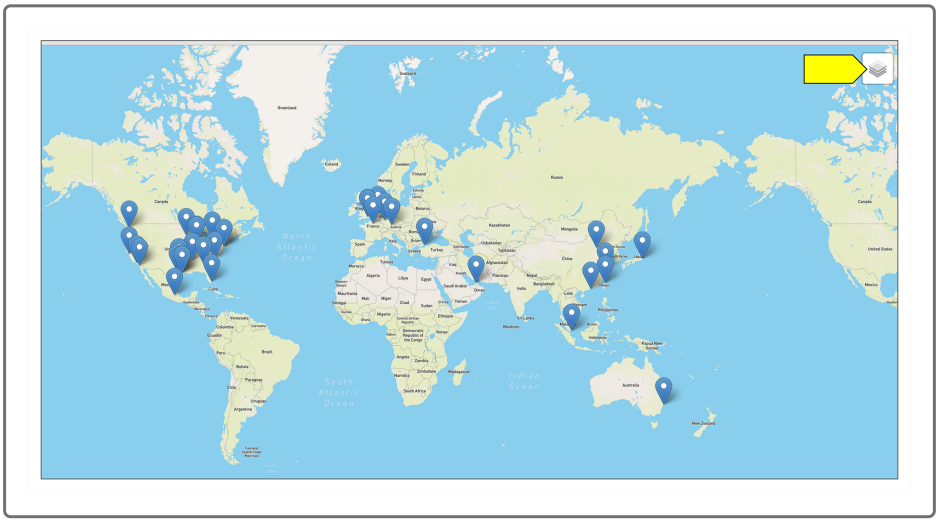
11. Your `logic.js` file should look like the following:

```
// We create the street view tile layer that will be the default background of our map.
let streets = L.tileLayer('https://api.mapbox.com/styles/v1/mapbox/streets-v11/tiles/{z}/{x}/{y}?access_token={accessToken}', {
  attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a> contributors, <a href="https://creat
  maxZoom: 18,
  accessToken: API_KEY
});

// We create the dark view tile layer that will be an option for our map.
let dark = L.tileLayer('https://api.mapbox.com/styles/v1/mapbox/dark-v10/tiles/{z}/{x}/{y}?access_token={accessToken}', {
  attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a> contributors, <a href="https://creat
  maxZoom: 18,
  accessToken: API_KEY
});

// Create a base layer that holds both maps.
let baseMaps = {
  Street: streets,
  Dark: dark
};

// Create the map object with center, zoom level and default layer.
let map = L.map('mapid', {
  center: [30, 30],
  zoom: 2,
  layers: [streets]
});

// Pass our map layers into our layer control and add the layer control to the map.
L.control.layers(baseMaps).addTo(map);
```

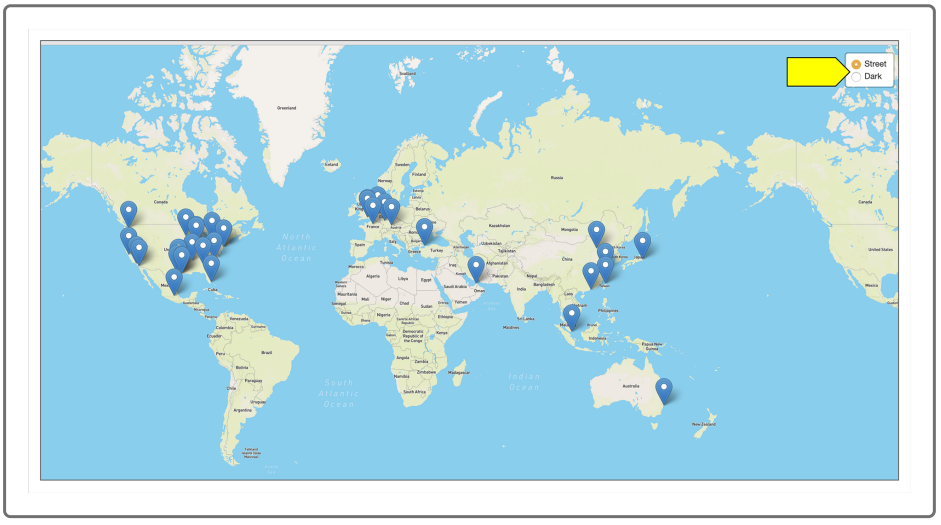
12. Finally, add the `airportData` variable to the `d3.json()` method as shown below:

```
// Accessing the airport GeoJSON URL
let airportData = "https://raw.githubusercontent.com/         /Mapping_Earthquakes/master/majorAirports.json";

// Grabbing our GeoJSON data.
d3.json(airportData).then(function(data) {
  console.log(data);
  // Creating a GeoJSON layer with the retrieved data.
  L.geoJson(data).addTo(map);
});
```

Save the `logic.js` file and open `index.html` in your browser. Your map should look like the following with a Layers Control in the top right corner of the map:



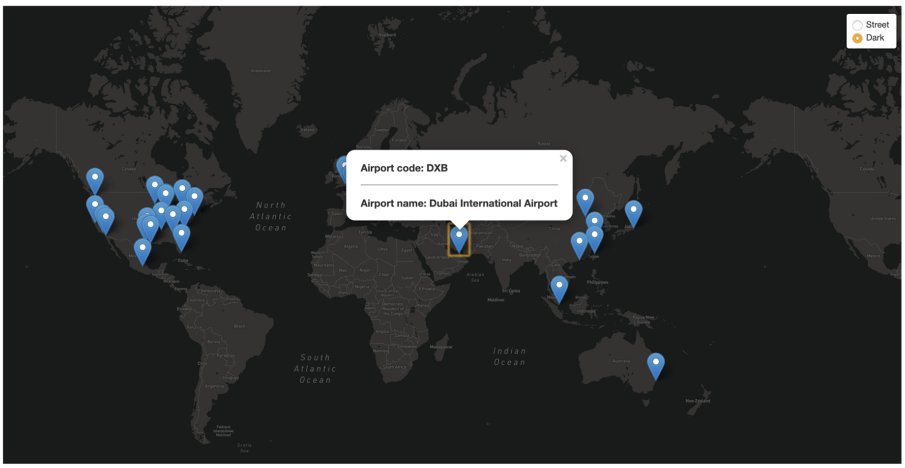
When we hover over the Layers Control, the map options are shown.



SKILL DRILL

Edit your `L.geoJson()` layer to add a popup marker that displays all airports' codes and names for both the Street and Dark layers.

Your map should look like the following:



Great job adding multiple-point type GeoJSON data to your map! Next, Sadhana will show you how to add map LineString type GeoJSON data.

ADD/COMMIT/PUSH

Add, commit, and push your changes to your Mapping_GeoJSON_Points branch. Don't delete the branch so that others can use it to learn how to map GeoJSON points.