

## Mapping API's: Leaflet - GeoJSON Layer

Welcome to the Essential ArcGIS Task Sheet Series. This series supplements the Iowa State University GIS Geospatial Technology Training Program short course series. The task sheets are designed to provide quick, easy instructions for performing mapping tasks.

In a previous task sheet, *Shapefile to GeoJSON: Conversion and Visualization* **PM2082-15g** you learned how to convert a point data file into a GeoJSON file format and visualize it with geojson.io and GitHub. In this task sheet you will learn how to add a GeoJSON data file to a basic leaflet then filter the data to display. The code for this task sheet (**geojsonLeaflet.html**) and the previous task sheets can be found on the ISU Geospatial Technology Program GitHub page at <https://github.com/ISUEOGTP/GISTaskSheets>.

### 1. Introduction

- First, you will need a basic leaflet map setup. Reference the task sheet: *Mapping API's: Leaflet - Getting Started* **PM2082-14r** to learn how to get this set up, or get the starter code from our GitHub page at <https://github.com/ISUEOGTP/GISTaskSheets/blob/master/Leaflet-Tutorials/helloLeaflet.htm>.
- Two additions to the basic Leaflet code are necessary in order to display the GeoJSON layer and filter the data. Add the following lines of code within the HTML head tags, shown on the right.
  - Link to a jQuery JavaScript library.

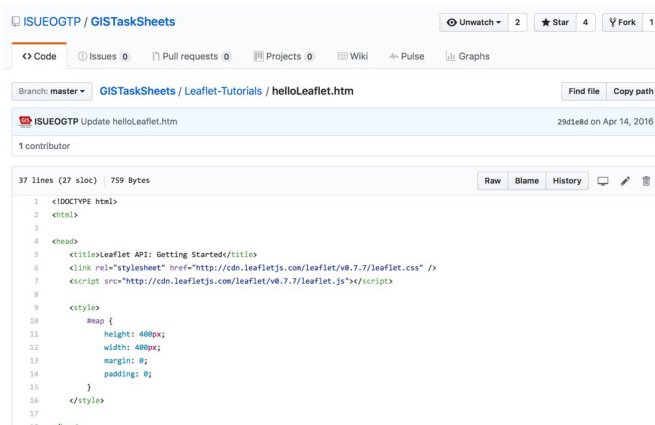
```
<script src="https://ajax.googleapis.com/ajax/
libs/jquery/1.11.1/jquery.mis.js">
</script>
```

- Link to the GeoJSON data file. *Hint: the file extension for your GeoJSON file could be .js, .json, or .geojson depending on how your web server is setup to support MIME types.*

```
<script src="airports.js"></script>
```

### 2. Add GeoJSON Data to the Map

- Typically, it is best practice to store GeoJSON data in an external file. While it is possible to link to data stored on other servers, this involves working with cross-origin resource sharing (CORS). To keep things simple, the GeoJSON file used in this task sheet will need to be stored in the same directory as the HTML map file.
- Open the **airports.json** file you created in the previous task sheet, *Shapefile to GeoJSON: Conversion and Visualization* **PM2082-15g**. If you don't have the file, copy it from our GitHub page: <https://github.com/ISUEOGTP/GeoJSON/blob/master/IowaAirports.json> and



ISUEOGTP / GISTaskSheets

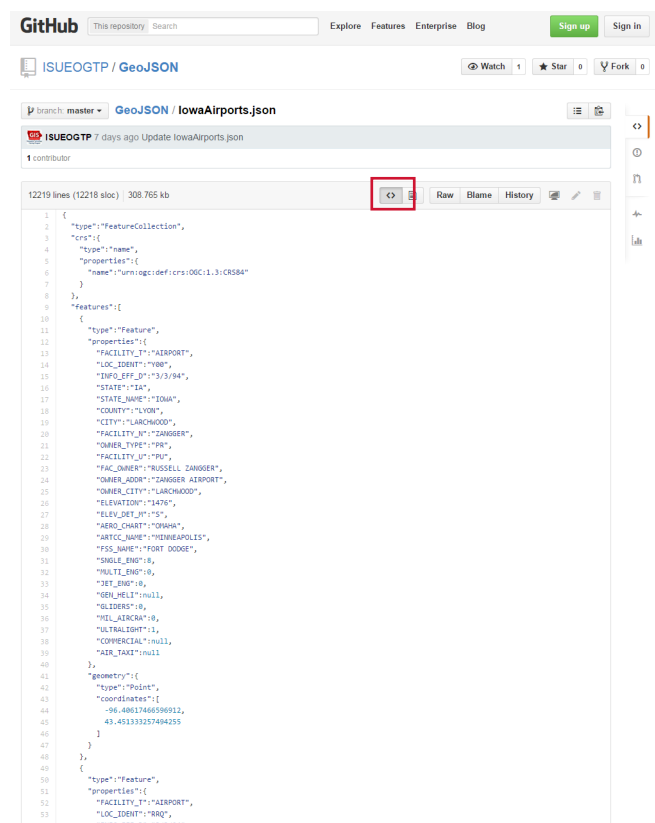
Branch: master

GISTaskSheets / Leaflet-Tutorials / helloLeaflet.htm

1 contributor

37 lines (27 sloc) 759 Bytes

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>Leaflet API: Getting Started</title>
6   <link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet/v0.7.7/leaflet.css" />
7   <script src="http://cdn.leafletjs.com/leaflet/v0.7.7/leaflet.js"></script>
8
9   <style>
10     #map {
11       height: 400px;
12       width: 400px;
13       margin: 0;
14       padding: 0;
15     }
16   </style>
17 </head>
```



GitHub

ISUEOGTP / GeoJSON

branch: master

GeoJSON / IowaAirports.json

1 contributor

12219 lines (12218 sloc) 308.765 kb

```
1 {
2   "type": "FeatureCollection",
3   "crs": {
4     "type": "name",
5     "name": "urn:ogc:def:crs:OGC:1.3:CRS84"
6   },
7   "features": [
8     {
9       "type": "Feature",
10      "properties": {
11        "FACILITY": "AIRPORT",
12        "LOC_IDENT": "HOB",
13        "INFO_EFF_DT": "3/3/84",
14        "STATE": "IA",
15        "STATE_NAME": "IOWA",
16        "COUNTRY": "USA",
17        "CITY": "LARCHMONT",
18        "FACILITY": "ZANGGER",
19        "OWNER": "PUB",
20        "FACILITY": "PUB",
21        "FACILITY": "RUSSELL ZANGGER",
22        "OWNER": "ZANGGER AIRPORT",
23        "COUNTRY": "LARCHMONT",
24        "ELEVATION": "1476",
25        "ELEV_DET": "15",
26        "AERO_CHART": "OMAW",
27        "ARTIC_NAME": "MINNEAPOLIS",
28        "FSS_NAME": "FORT DODGE",
29        "SMALL_ENG": "8",
30        "MULTI_ENG": "8",
31        "JET_ENG": "8",
32        "GEI_HELI": "null",
33        "GLIDERS": "8",
34        "HIL_JERSEA": "8",
35        "ULTRALIGHT": "1",
36        "COMMERCIAL": "null",
37        "AER_TAXI": "null"
38      },
39      "geometry": {
40        "type": "Point",
41        "coordinates": [
42          -96.4861746596912,
43          43.45133257494255
44        ]
45      }
46    },
47    {
48      "type": "Feature",
49      "properties": {
50        "FACILITY": "AIRPORT",
51        "LOC_IDENT": "HOB",
52        "INFO_EFF_DT": "3/3/84",
53        "STATE": "IA",
54        "STATE_NAME": "IOWA",
55        "COUNTRY": "USA",
56        "CITY": "LARCHMONT",
57        "FACILITY": "ZANGGER",
58        "OWNER": "PUB",
59        "FACILITY": "PUB",
60        "FACILITY": "RUSSELL ZANGGER",
61        "OWNER": "ZANGGER AIRPORT",
62        "COUNTRY": "LARCHMONT",
63        "ELEVATION": "1476",
64        "ELEV_DET": "15",
65        "AERO_CHART": "OMAW",
66        "ARTIC_NAME": "MINNEAPOLIS",
67        "FSS_NAME": "FORT DODGE",
68        "SMALL_ENG": "8",
69        "MULTI_ENG": "8",
70        "JET_ENG": "8",
71        "GEI_HELI": "null",
72        "GLIDERS": "8",
73        "HIL_JERSEA": "8",
74        "ULTRALIGHT": "1",
75        "COMMERCIAL": "null",
76        "AER_TAXI": "null"
77      },
78      "geometry": {
79        "type": "Point",
80        "coordinates": [
81          -96.4861746596912,
82          43.45133257494255
83        ]
84      }
85    }
86  ]
87 }
```

save it in the same directory as the **geojson.html**. *Hint: click the icon to display the source blob to view the GeoJSON.*

- c. In the **airports.json** file add **var airports =** before the first curly brace and save the file. This will put the data into a variable called airports.
- d. To add the data to the map, add the following line of code after the map constructor. Note that in this line of code the variable **airports** is used because that is the variable name used in the GeoJSON file.

```
L.geoJson(airports).addTo(map);
```

- e. Now, all the airport locations should be drawn on the map with the default leaflet marker style.
- f. To add pop-ups to the map, comment out the line from **step 2d** and add the following lines.

```
//L.geoJson(airports).addTo(map);

function addPopups(feature, layer) {
  layer.bindPopup("Name: "+feature.
properties.FACILITY_N + "<BR> City:
"+feature.properties.CITY)
}

L.geoJson(airports, {
  onEach Feature:
    addPopups
}).addTo(map);
```

### 3. Filtering GeoJSON

- a. Data can be filtered based on desired attribute criteria. Add a comma after addPopups, from **step 2f** and the following filter function. This will filter the GeoJSON data and only show the features with the facility type equal to heliport.

If you want to filter for all public facilities instead of facility type, replace:

```
FACILITY_T == 'HELIPORT'
```

with: (now the map will show only public airports)

```
OWNER_TPE == 'PU'
```

### Contact:

Bailey Hanson bahanson@iastate.edu, 515-520-1436 or Professor Christopher J. Seeger, ASLA, GISP cjseeger@iastate.edu, 515-509-0651 for more information about the Geospatial Technology Program. This task sheet and more are available at [www.extension.iastate.edu/communities/gis](http://www.extension.iastate.edu/communities/gis)

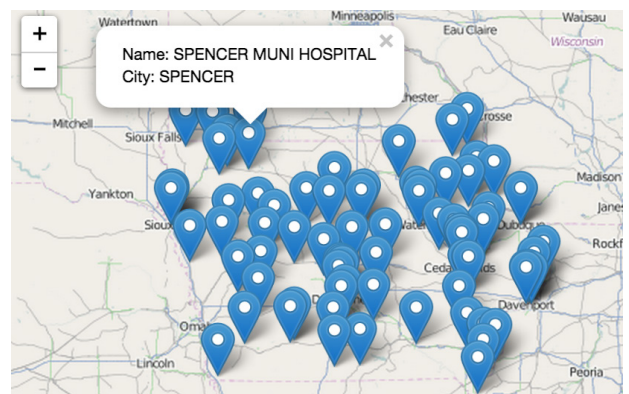
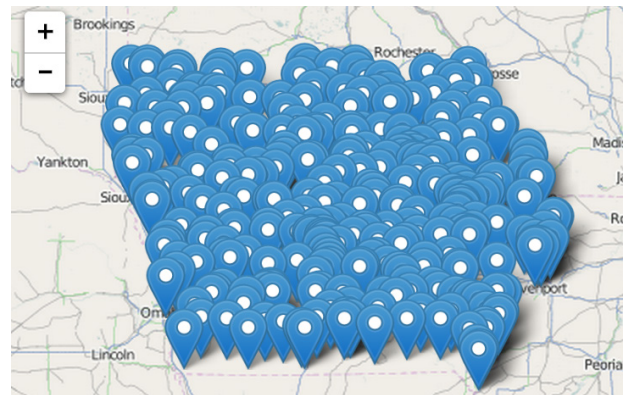
Iowa State University Extension and Outreach does not discriminate on the basis of age, disability, ethnicity, gender identity, genetic information, marital status, national origin, pregnancy, race, religion, sex, sexual orientation, socioeconomic status, or status as a U.S. veteran. (Not all prohibited bases apply to all programs.) Inquiries regarding non-discrimination policies may be directed to Ross Wilburn, Diversity Officer, 2150 Beardshear Hall, 515 Morrill Road, Ames, Iowa 50011, 515-294-1482, wilburn@iastate.edu.

```
var airports = { "type": "FeatureCollection", "c
"FACILITY_T": "AIRPORT", "LOC_IDENT": "Y00", "INF
"OWNER_TYPE": "PR", "FACILITY_U": "PU", "FAC_OWNE
"AERO_CHART": "OMAHA", "ARTCC_NAME": "MINNEAPOLIS
"ULTRALIGHT": 1, "COMMERCIAL": null, "AIR_TAXI": n
"FACTITIV M": "ATBDBB" "LOC_IDENT": "DBO" "T
```

```
var map = L.map('map').setView([43.4,
-93], 8);

//add an OpenStreetMap as base map
L.tileLayer('http://{s}.tile.osm.org/
{z}/{x}/{y}.png', {
  attribution: 'The map contributors',
  maxZoom: 18
}).addTo(map);

L.geoJson(airports).addTo(map);
```



```
addPopups,
filter: function(feature, layer) {
  if (feature.properties.FACILITY_T ==
'HELIPORT') {
    return 'true'
  }
}
```