

## Content

1. Open Layers.....	1
<b>1.1 Base Structure</b> .....	1
<b>1.2 OpenLayers and GeoJSON</b> .....	2
1.3 OpenLayers and XML.....	3
1.4 OpenLayers and WMS as a Service .....	3
1.5 OpenLayers + PosGreSQL + Geoserver.....	4
2. Leaflet .....	5
2.1 Estrutura Base .....	5
2.2 Leaflet and GeoJSON .....	8
2.3. Leaflet with XML.....	9
2.4 Leaflet and WMS as a service.....	9
2.5 PostGIS + GeoServer + Leaflet.....	11

## Scope

Exploration of georeferenced information viewers using Open Layers and Leaflet

## 1. Open Layers

### 1.1 Base Structure

**index.html**

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>OpenLayers</title>
    <style>
      html, body, #map-container {
        margin: 0;
        height: 100%;
        width: 100%;
        font-family: sans-serif;
      }
    </style>
  </head>
  <body>
    <div id="map-container"></div>
  </body>
</html>

```

+

**main.js**

```

import 'ol/ol.css';
import {Map, View} from 'ol';
import TileLayer from 'ol/layer/Tile';
import XYZSource from 'ol/source/XYZ';
import {fromLonLat} from 'ol/proj';

new Map({
  target: 'map-container',
  layers: [
    new TileLayer({
      source: new XYZSource({
        url: 'http://tile.stamen.com/terrain/{z}/{x}/{y}.jpg'
      })
    })
  ],
  view: new View({
    center: fromLonLat([0, 0]),
    zoom: 2
  })
});

```

Result:

## Tutorial

Visualization of Geographic Information – Open Layers and Leaflet

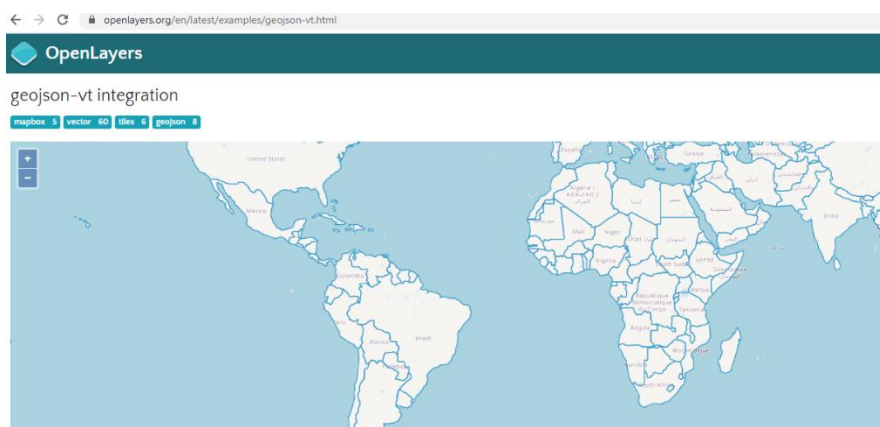
(versão 1)



### 1.2 OpenLayers and GeoJSON

Consider the example of the tutorial available at:

- Countries: <https://openlayers.org/en/latest/examples/geojson-vt.html>



- Workshop: <https://openlayers.org/workshop/en/vector/geojson.html>


OpenLayers Workshop
Introduction
Basics
Vector Data
Rendering GeoJSON
Drag and drop
Modifying features
Drawing new features
Snapping
Downloading features
Making it look nice
Mobile Maps and Sensors
A mobile map
Geolocation
Compass
WebGL Rendering
Map setup
Rendering points
Animating points
Vector Tiles and Mapbox Styles
The VectorTile layer

```

    })
  },
  view: new View({
    center: [0, 0],
    zoom: 2
  })
});

```

You should now be able to see a map with country borders at <http://localhost:3000/>.




GeoJSON features

Since we'll be reloading the page a lot, it would be nice if the map stayed where we left it in a reload. We can bring in the `ol-hashed` package to make this work. This package is already installed as part of the workshop dependencies. If it were not already included, you could install it with `npm install ol-hashed`.

## 1.3 OpenLayers and XML

Consider the example in: <https://openlayers.org/en/latest/examples/vector-osm.html>

OpenLayers
OSM XML
vector 60 osmxml 1 loading 6 server 4 strategy 1 bbox 2 maptiler 24



OSM XML vector data is loaded dynamically from a the [Overpass API](#) using a `bbox` strategy. Note that panning and zooming will eventually lead to "Too many requests" errors from

main.js

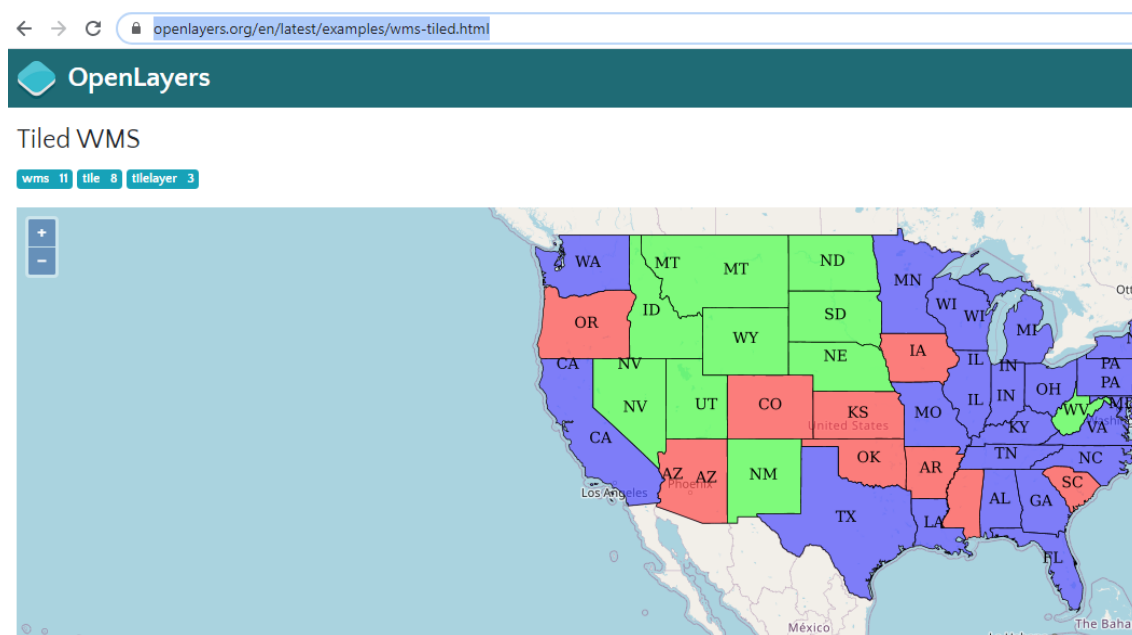
```

1 import 'ol/ol.css';
2 import Map from 'ol/Map';
3 import OSMXML from 'ol/format/OSMXML';
4 import VectorSource from 'ol/source/Vector';

```

## 1.4 OpenLayers and WMS as a Service

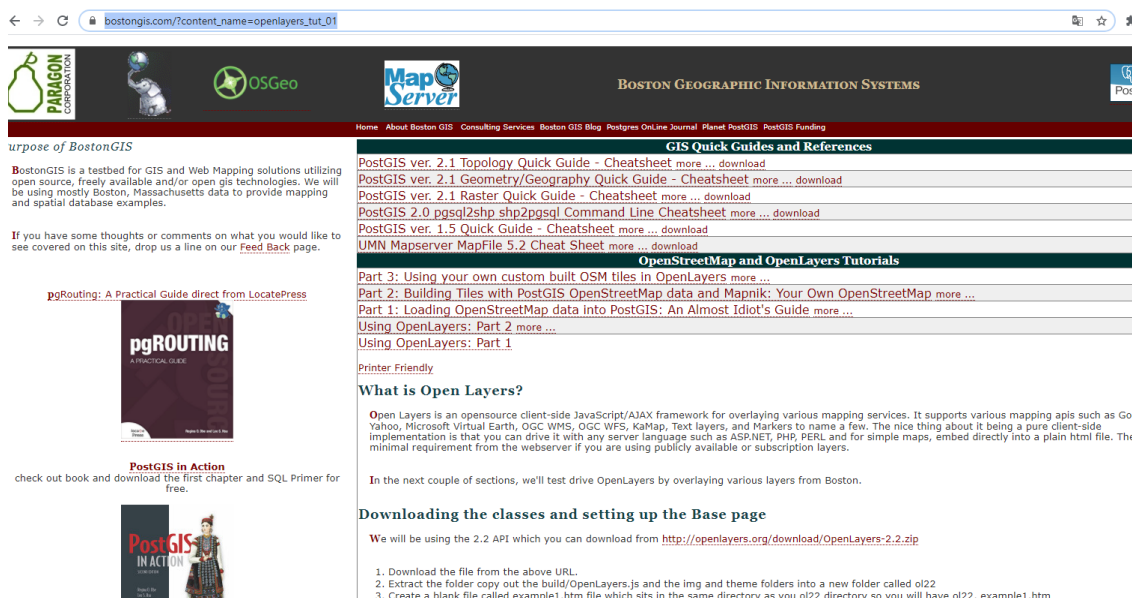
Consider the example in: <https://openlayers.org/en/latest/examples/wms-tiled.html>



## 1.5 OpenLayers + PosgreSQL + Geoserver

Consider the examples available at:

[https://www.bostongis.com/?content\\_name=openlayers\\_tut\\_01](https://www.bostongis.com/?content_name=openlayers_tut_01)



## 2. Leaflet

### 2.1 Estrutura Base

Consider the information available at the link: <https://leafletjs.com/examples.html> Seguindo these bases, hereinafter a set of examples of exploration of the functionalities of the Leaflet.

Scripts for applying the Leaflet library to in a file JavaScript:

```
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.3.4/dist/leaflet.css"
integrity="sha512-puBpdR07980ZvTTbP4A8Ix/l+A4dHDD0DGqYW6RQ+9jxkRfClaxxQb/SJAWZfWAkuyeQUytO7+7N4QKrDh+drA=="
crossorigin="" />

<script src="https://unpkg.com/leaflet@1.3.4/dist/leaflet.js"
integrity="sha512-nMMmRyTVoLYqjP9hrbed9S+FzjZHW5gY1TWCHA5ckwXZBadntCNS8kEqAWdrb907rxbCaA41KTIWjDXZxf10cA=="
crossorigin=""></script>
```

Stylization of the map to be presented in the browser:

```
<style>

  #mapid {
    height: 480px;
  }

</style>
```

Map declaration and layer creation (layer that presents the map):

```
var mymap = L.map('mapid').setView([51.505, -0.09], 13);

var token = "pk.eyJ1Ijoiam51bm9mZXJyZWlyYSIsImEiOiIjam5zMGdsb3owYjFqM2txcTA2bmN0OHZwIn0.U1hgHg316EPnNvALad0oqQ";

L.tileLayer('https://api.tiles.mapbox.com/v4/{id}/{z}/{x}/{y}.png?access_token={accessToken}', {
  attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a> contributors, <a href="https://c',
  maxZoom: 18,
  id: 'mapbox.streets',
  accessToken: token
}).addTo(mymap);
```

Creating a point, circle and polygon on the map:

```
var marker = L.marker([51.5, -0.09]).addTo(mymap);

var circle = L.circle([51.508, -0.11], {
  color: 'red',
  fillColor: '#f03',
  fillOpacity: 0.5,
  radius: 500
}).addTo(mymap);

var polygon = L.polygon([
  [51.509, -0.08],
  [51.503, -0.06],
  [51.51, -0.047]
]).addTo(mymap);
```





## Tutorial

Creation of the map and assign the layers that will be presented in the first instance:

```
//Pomos o Mapa (já preparado com as layers acima definidas) associado ao div com o id "mapid"
var mymap = L.map('mapid', {
  center: [41.50, -7.73],
  zoom: 9,
  layers: [streets, carros] //aqui definimos as layers que queremos que estejam visíveis numa primeira instancia
});
```

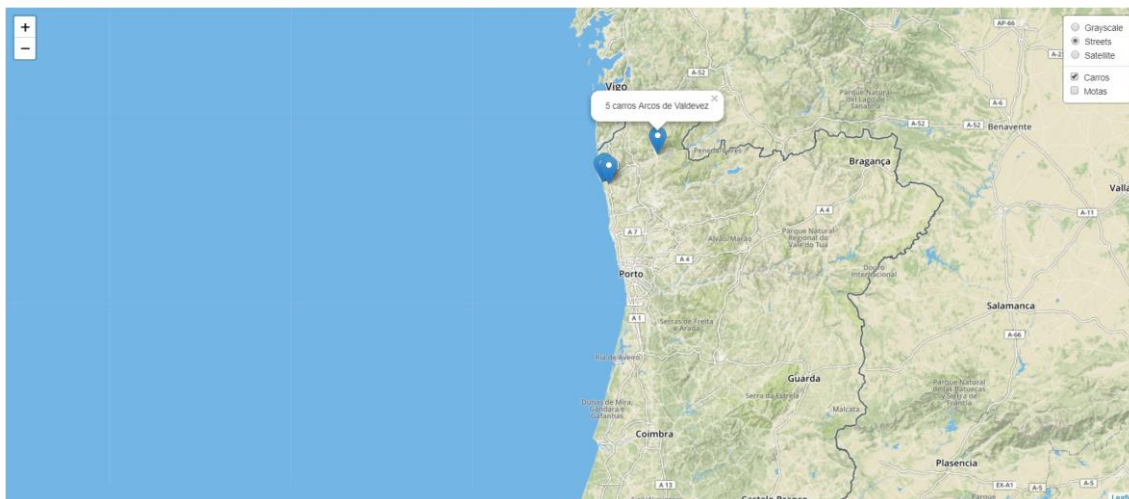
Creation of layers that can be presented simultaneously (overlay):

```
//Aqui definimos as restantes layers (para além das default). Podemos mostrar várias ao mesmo tempo
var overlayMaps = {
  "Carros": carros,
  "Motas": motas
};
```

Addition of a controller on the page to view the created layers:

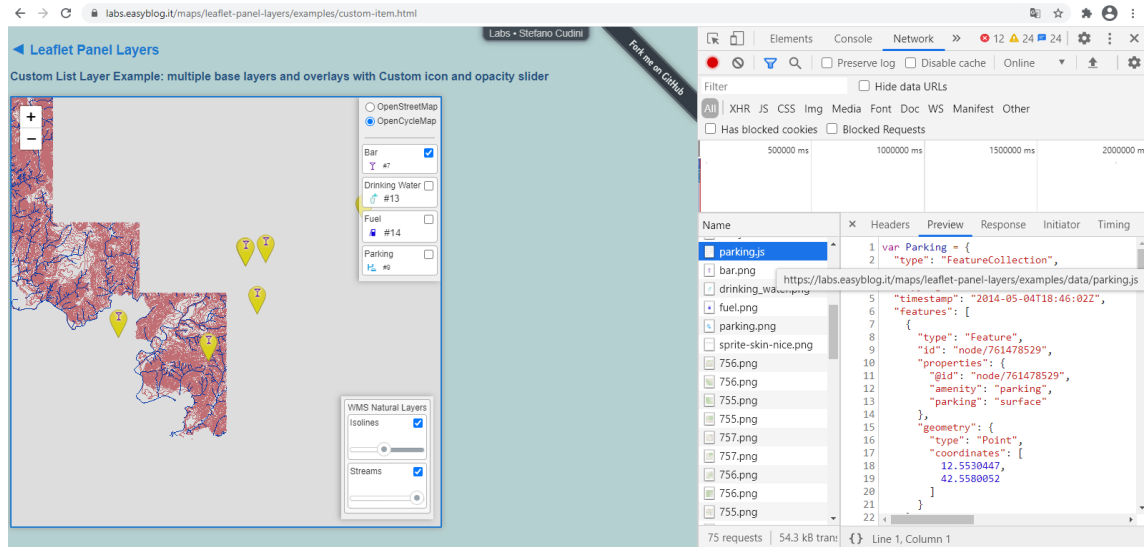
```
//No canto superior direito aparece um controlador para por as layers visíveis ou invisíveis
L.control.layers(baseMaps, overlayMaps).addTo(mymap);
```

Result:



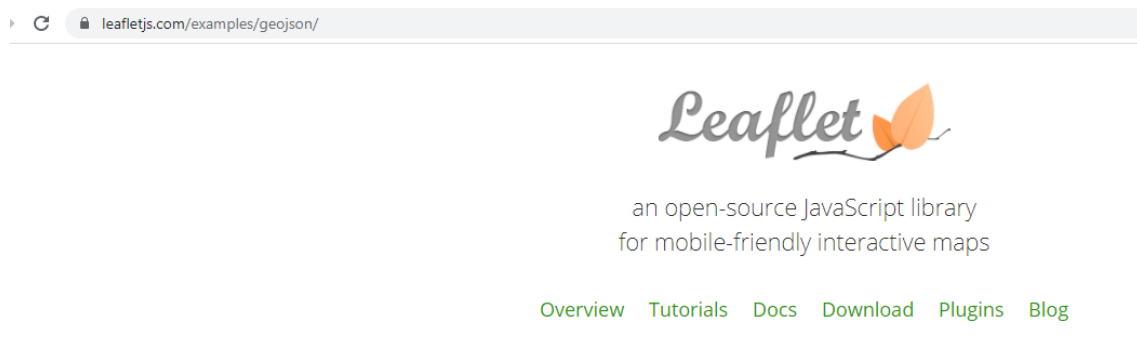
Consider the following examples, following the tutorials available at the following links:

<https://labs.easyblog.it/maps/leaflet-panel-layers/examples/custom-item.html>



## 2.2 Leaflet and GeoJSON

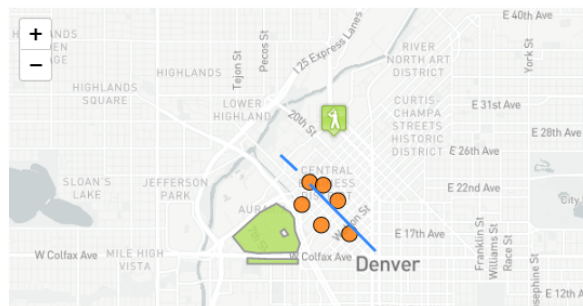
Consider the examples in: <https://leafletjs.com/examples/geojson/>



### [← Tutorials](#)

#### Using GeoJSON with Leaflet

GeoJSON is becoming a very popular data format among many GIS technologies and services — it's simple, lightweight, straightforward, and Leaflet is quite good at handling it. In this example, you'll learn how to interact with map vectors created from [GeoJSON](#) objects.



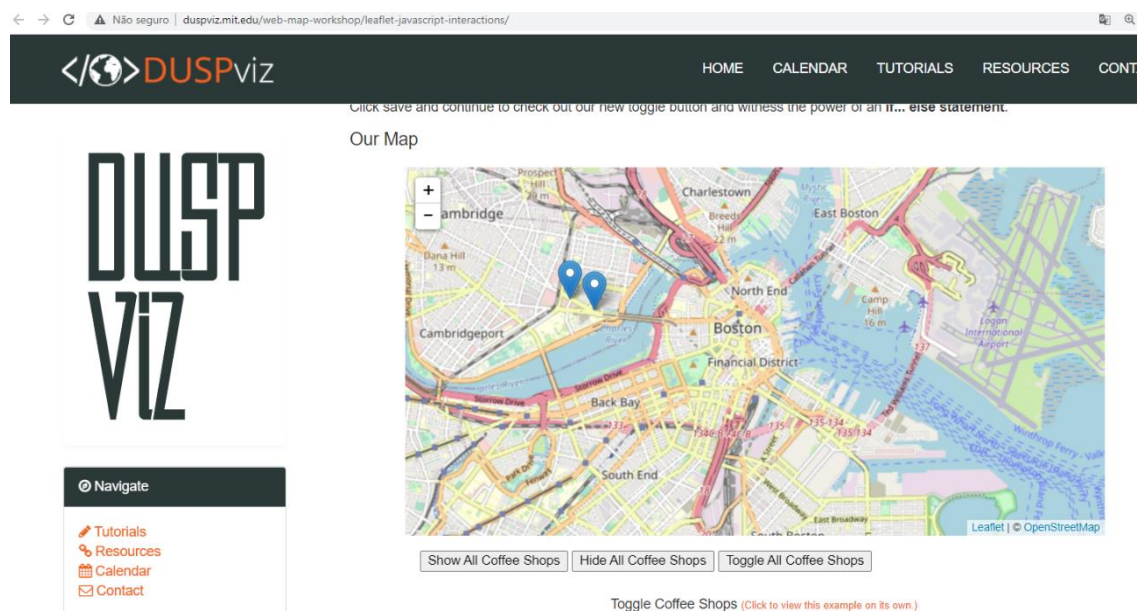


## Tutorial

Visualization of Geographic Information – Open Layers and Leaflet

(versão 1)

You can also run the following tutorial: <http://duspviz.mit.edu/web-map-workshop/leaflet-javascript-interactions/>



### 2.3. Leaflet with XML

Maps from a Database: Reading XML data into Leaflet: [http://erica.altschul.info/Tutorial\\_XML-to-Leaflet.pdf](http://erica.altschul.info/Tutorial_XML-to-Leaflet.pdf)

### 2.4 Leaflet and WMS as a service

Consider the example of the link: <https://leafletjs.com/examples/wms/wms.html>

## Tutorial

Visualization of Geographic Information – Open Layers and Leaflet

(versão 1)

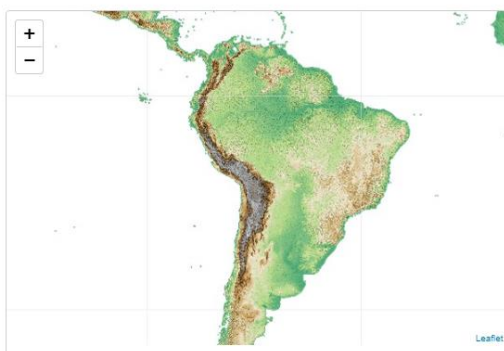
leafletjs.com/examples/wms/wms.html



[See this example stand-alone.](#)

Or we can try the SRTM30-Colored-Hillshade WMS layer:

```
var wmsLayer = L.tileLayer.wms('http://ows.mundialis.de/services/service?', {  
  layers: 'SRTM30-Colored-Hillshade'  
}).addTo(map);
```



[See this example stand-alone.](#)

Also consider the following example: <https://flexberry.github.io/Leaflet-WMS/examples/basic.html>

flexberry.github.io/Leaflet-WMS/examples/basic.html




Also consider the following example: <https://www.e-education.psu.edu/geog585/node/765>

## Tutorial


Visualization of Geographic Information – Open Layers and Leaflet

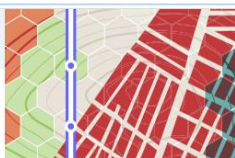
(versão 1)

← → ↻ e-education.psu.edu/geog585/node/765


**PennState**  
College of Earth  
and Mineral Sciences

DEPARTMENT OF  
**GEOGRAPHY**


**GEOG 585**  
**OPEN WEB MAPPING**



[HOME](#) [SYLLABUS](#) [ORIENTATION](#) [LESSONS](#) [CANVAS](#) [RESOURCES](#) [LOGIN](#)

### Walkthrough: Overlaying a WMS on a tiled map with Leaflet

[Print](#)

The goal of this walkthrough is to get some practice overlaying different kinds of web services in Leaflet. You will first publish a WMS showing farmers' markets in Philadelphia. You will then use Leaflet to place this layer on top of the Philadelphia basemap tiles you made with TileMill in the previous lesson. You'll also add code, so that a user of your application can click any farmers market and see some more information in a popup.

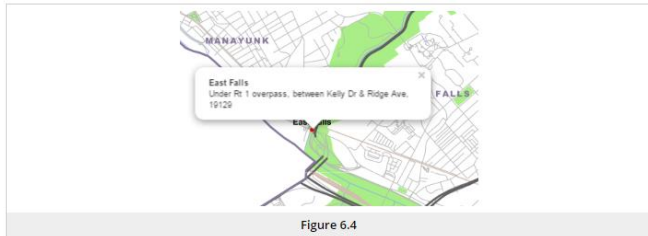


Figure 6.4

Setting up the farmers' markets WMS

**GEOG 585: Open Web Mapping**

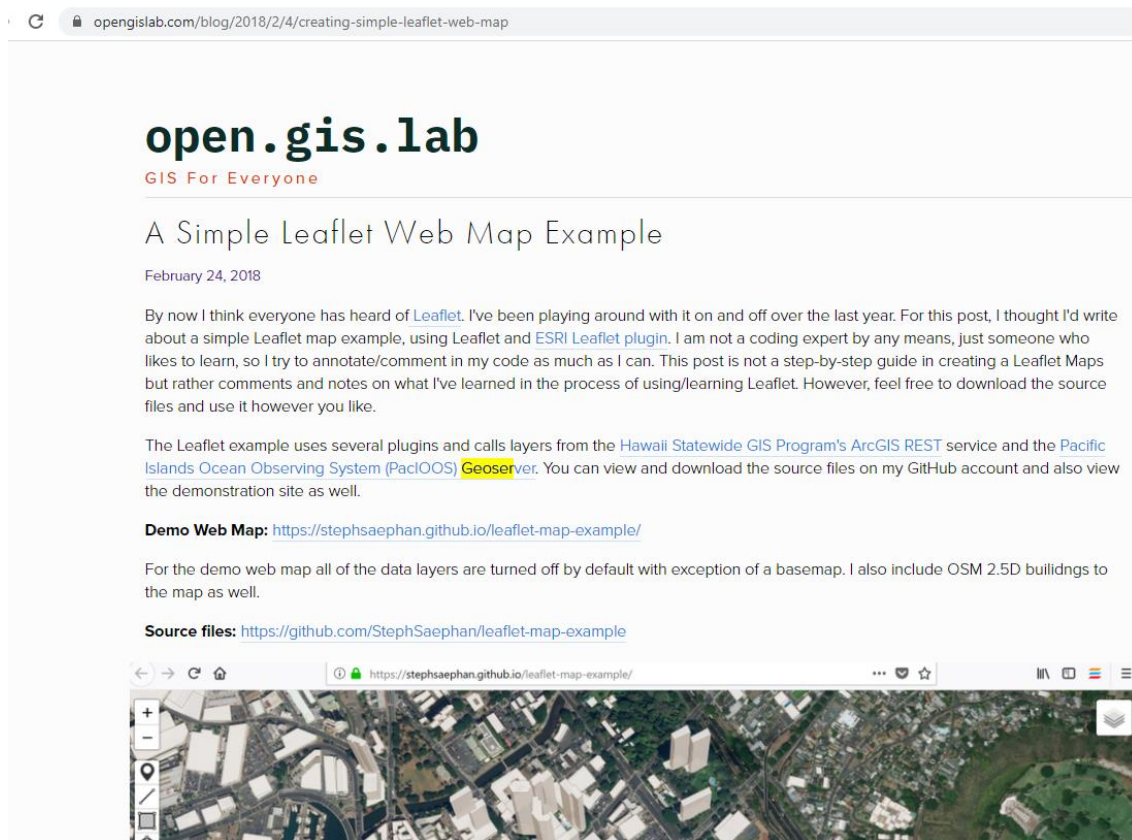
**Lessons**

- Welcome to GEOG 585 - Open Web Mapping
- ▶ Lesson 1: FOSS and its use in web mapping
- ▶ Lesson 2: Designing web services and web maps
- ▶ Lesson 3: Storing and processing spatial data with FOSS
- ▶ Lesson 4: Drawing and querying maps on the server using WMS
- ▶ Lesson 5: Building tiled maps with FOSS
- ▼ Lesson 6: Putting layers together with a web mapping API
  - Overview

## 2.5 PostGIS + GeoServer + Leaflet

Consider the example of exploring the Leaflet with Geoserver, available at:

<https://opengislab.com/blog/2018/2/4/creating-simple-leaflet-web-map>



Consider the example of exploring the Leaflet with Geoserver and PostgreSQL, available at:  
[https://www.earder.com/tutorials/postgis\\_geoserver\\_leaflet/](https://www.earder.com/tutorials/postgis_geoserver_leaflet/)

