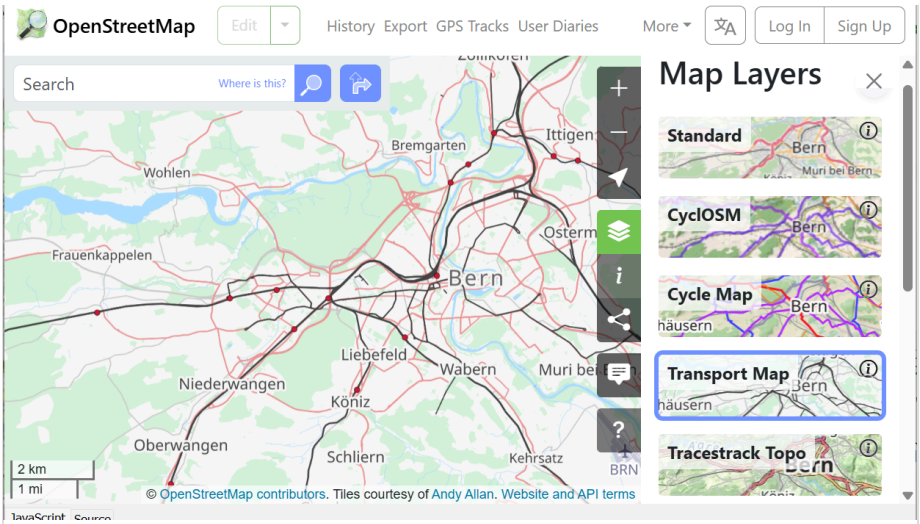
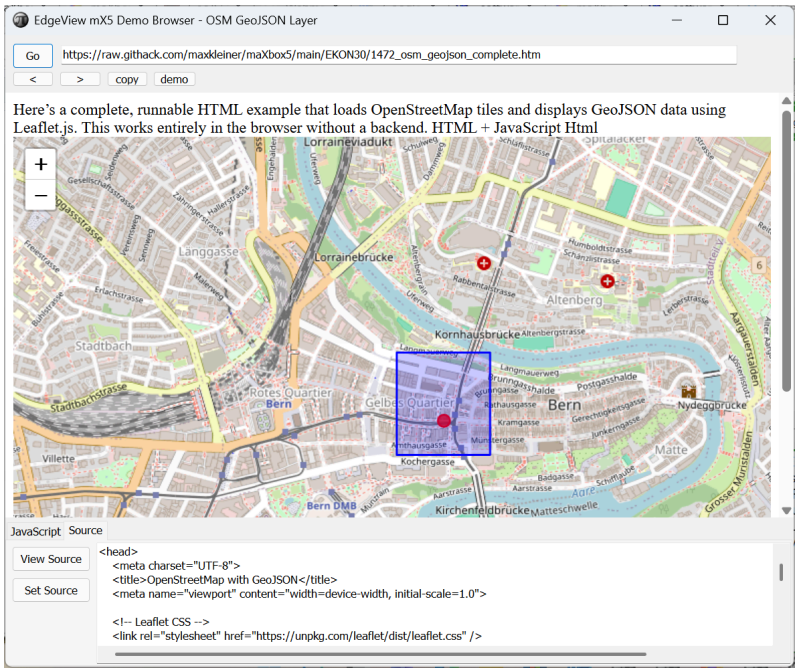


GeoJSON in OSM

March 17, 2026

Here's a complete, runnable HTML example that loads OpenStreetMap tiles and displays GeoJSON data using Leaflet.js. and EdgeView runtime. This works entirely in the browser without a backend. We set a point and a polygon on the map!



https://sourceforge.net/projects/maxbox5/files/examples/1472_API_GeoJSON64_5_OSM_maps2edge3.txt/download

The GeoJSON layer is part of the script which we load on the edgeview.

How it works

1. Leaflet.js loads OpenStreetMap tiles.
2. A GeoJSON object is defined inline (you can also load it from a `.geojson` file via `fetch()`).
3. `L.geoJSON()` renders the features with custom styles and popups.
4. Supports points, lines, and polygons.

✅ Tip:

If you have a `.geojson` file, replace the inline `geojsonData` with:

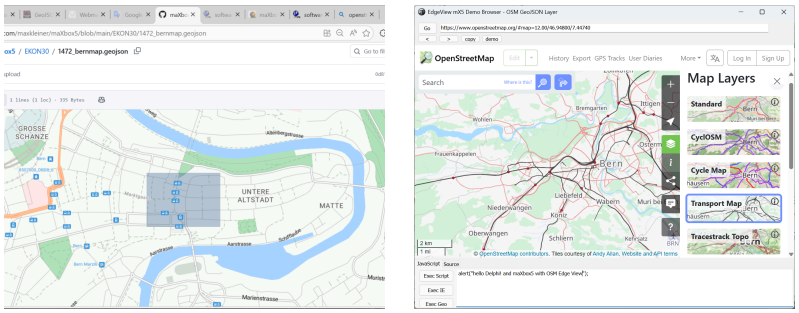
```
fetch('data.geojson')
  .then(res => res.json())
  .then(data => L.geoJSON(data).addTo(map));
```

If you want, you can also extend this example to load live GeoJSON data directly from OpenStreetMap using Overpass API so you can query real map features dynamically. The main part in the script with maXbox goes like this:

```
procedure letGeoJson_EdgeOSM_GeoJSON(zoom, lon,lat: double);
var htmlgeojs: string;
 begin
   with TEdgeViewForm.create(self) do begin
     height:= 1000; width:= 1200;
     caption:= caption+ ' - OSM GeoJSON Layer'
     htmlgeojs:= loadfile(exepath+'examples\1472_osm_geojson_
     pagecontrol1.height:= 160;
     edit1.text:= TESTURL5; //OSMDIRECT; //'https://wttr.in/B
     //edit1.text:= Format(OSM_URLFormat,[zoom, lon,lat]); //
     //btngoclick(self);
     sleep(1500);
     memoHTML.Text:= utf8toansi(htmlgeojs); ////GEOJSON_Layer
     btnSetSourceClick(self);  //}
     showmodal
     free;
   end;
 end;
```

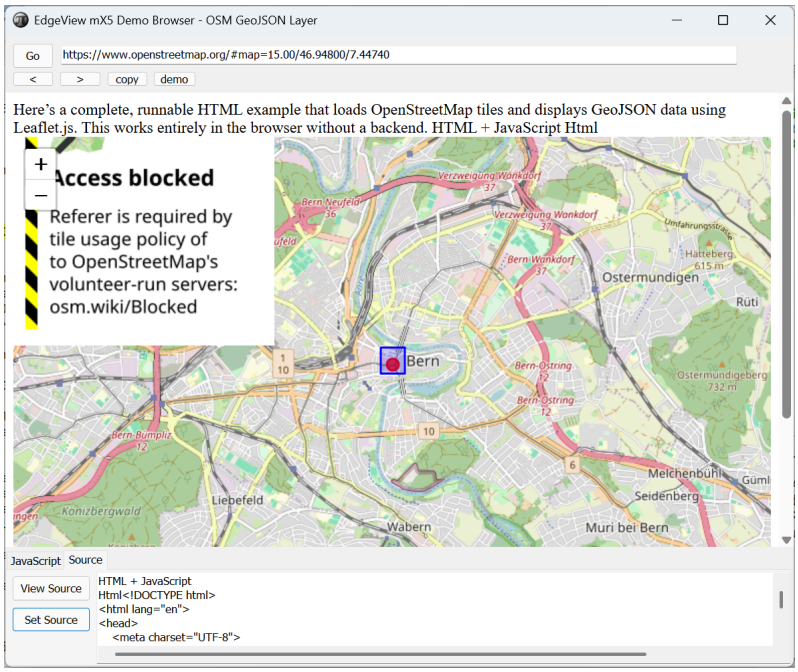
You can load the script direct from a host or as an file locally. The point and the rectangular polygon ist the GeoJSON part.

A position is an array of numbers. There MUST be two or more elements. The first two elements are longitude and latitude, or easting and northing, precisely in that order and using decimal numbers. Altitude or elevation MAY be included as an optional third element.



Comparison Google Maps and OpenStreetMaps

While using open streets overlay, a repeated error message window pops up which covers the map below. it says 403r Access Blocked: Referer is required by title usage policy of (to) OpenSteetMap's volunteer run servers: ism.wiki/Blocked. The overlay map is functioning but covered by the repeated messages.



Bug: 403r Access Blocked error message · Issue #396 · derekhe/xplane-map-enhancement-release

Here's the complete, runnable HTML example that loads OpenStreetMap tiles and displays GeoJSON data using Leaflet.js. This works entirely in the browser without a backend.

```
HTML + JavaScript
Html<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>OpenStreetMap with GeoJSON</title>
    <meta name="viewport" content="width=device-width, initia
    <!-- Leaflet CSS -->
    <link rel="stylesheet" href="https://unpkg.com/leaflet/di
    <style>
        #map {
            height: 100vh; /* Full screen height */
            width: 100%;
        }
    </style>
</head>
<body>
<div id="map"></div>

<!-- Leaflet JS -->
<script src="https://unpkg.com/leaflet/dist/leaflet.js"></scr
<script>
    // Initialize the map
    const map = L.map('map').setView([46.948, 7.447], 15); //
    // Add OpenStreetMap tile layer
    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{
        attribution: '&copy; <a href="https://www.openstreetm
    }).addTo(map);
    // Example GeoJSON data (Point + Polygon)
    const geojsonData = {
        "type": "FeatureCollection",
        "features": [
            {
                "type": "Feature",
                "properties": { "name": "Sample Point" },
                "geometry": {
                    "type": "Point",
                    "coordinates": [7.447, 46.948] // lon, la
                }
            },
            {
                "type": "Feature",
                "properties": { "name": "Sample Polygon" },
                "geometry": {
                    "type": "Polygon",
                    "coordinates": [[
```

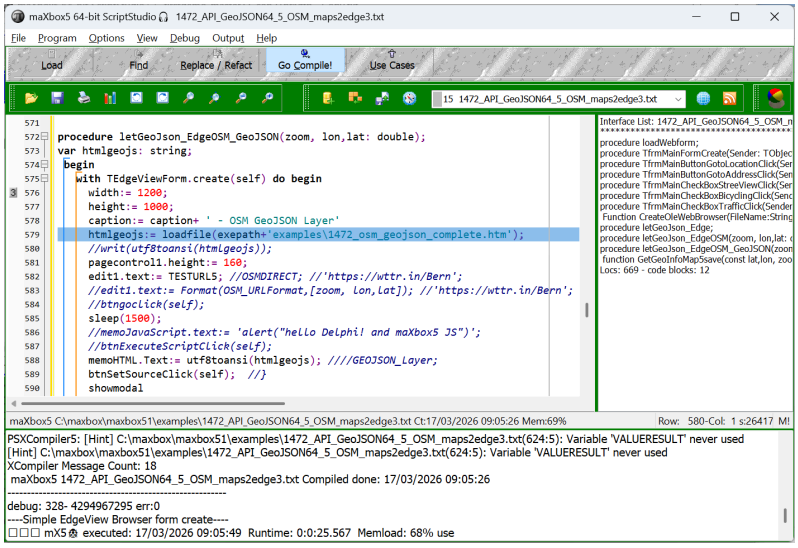
About
Champagnole
TEE Kraftwerk Concert
TEE Overview

recent posts

GeoJSON in OSM
GeoJSON in GMaps
GeoJSON
Barcode API
Railuino II
Cité du Train News 2025
Hello TEE World!

about

```
                        [7.445, 46.947],
                        [7.449, 46.947],
                        [7.449, 46.950],
                        [7.445, 46.950],
                        [7.445, 46.947]
                    ]]
                }
            }
        ]
    };
    // Add GeoJSON layer with styling and popup
    L.geoJSON(geojsonData, {
        style: feature => ({
            color: feature.properties.name === "Sample Polygo
            weight: 2
        }),
        pointToLayer: (feature, latlng) => {
            return L.circleMarker(latlng, {
                radius: 6,
                fillColor: "red",
                color: "#000",
                weight: 1,
                opacity: 1,
                fillOpacity: 0.8
            });
        },
        onEachFeature: (feature, layer) => {
            if (feature.properties && feature.properties.name
                layer.bindPopup(feature.properties.name);
            }
        }
    }).addTo(map);
</script>
</body>
</html>
```



https://sourceforge.net/projects/maxbox5/files/examples/1472_API_GeoJSON64_5_OSM_maps2edge3.txt/download

Conclusion

The provided web content describes a tutorial on using **Leaflet** with OpenStreetMap to create an interactive map of a city, including custom markers and popups for office locations based on GeoJSON data.

With EdgeView as a runnable and as a developer, you need to make well-

Windows-native platforms, components, and programming languages and the use of web technologies, such as HTML, CSS, and JavaScript.

Posted in Code, Geomap, Trains

One response to “GeoJSON in OSM”

maxkleiner
March 17, 2026 Edit

Leaflet is a great JavaScript library for creating map visualizations in web applications. Leaflet is created with that specific purpose in mind. It caters for most of the things you want in a map visualization, such as zoom and pan, markers, popups (on hover and / or click), geo coordinate aware events, multiple layers — of type raster, vector, SVG, image and video — and multiple projections.

★ Like

Reply

Leave a comment

← GeoJSON in GMaps