

ASSOCIATION OF CANADIAN MAP LIBRARIES AND ARCHIVES  
BULLETIN

## Geospatial Data and Software Reviews

Meg Miller  
University of Manitoba

### Mapbox.js: an engaging open-source web mapping tool for teaching data visualization theory

#### Abstract

This review provides an overview of web mapping tools and discusses why and how Mapbox.js has been an effective tool in the delivery of one-shot instructional sessions on data literacy and visualization theory in an academic library context.

#### Tags

Web map, JavaScript, data literacy, visualization, academic library, instruction

#### Introduction

The following will examine the *Mapbox.js*. This open JavaScript API is used to make interactive online maps. The tool is very flexible with a modular flow that lends itself well to on-the-fly modification in a classroom context. In the following use case, the tool is used throughout a session to illustrate the theoretical concepts of data visualization that are being discussed in a hands-on way.

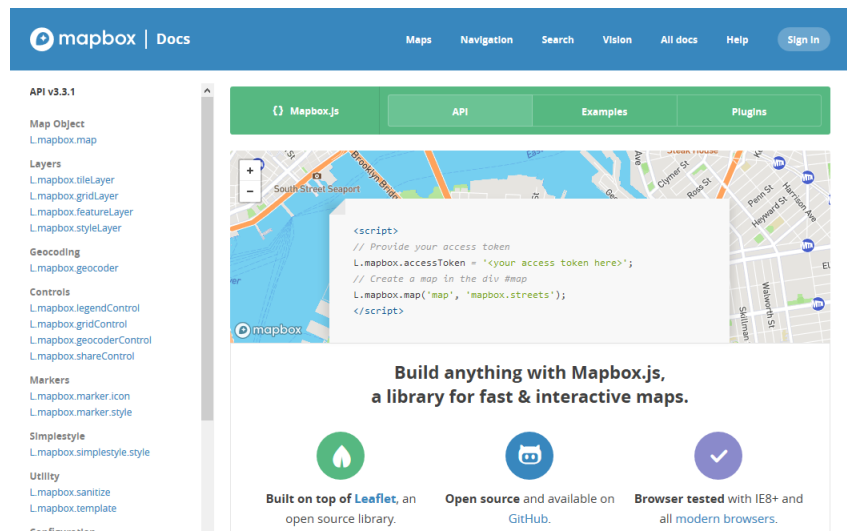


Figure 1: Mapbox API documentation landing

## Background

As a response to user feedback requesting more sessions on theory and less of a focus on software instruction this style of session was developed as a way to give users an engaging look at data visualization theory. Elements such as data types, colour, line weight and type and symbology are discussed in relation to the user experience. At the beginning of the session a scenario of making a map for a particular client is introduced so that there is a central example to work from. Through the session, users make adjustments to the HTML and JavaScript code of a live web map to see how they can change its feel.

This style of session is also conducted using ESRI's ArcGIS Online for those who are new to manipulating any kind of data or GIS. Using Mapbox, Leaflet or any web mapping tool where coding exploration will take place is best suited to those who have experience with JavaScript or who are comfortable with technology and have some level of coding experience. As was previously mentioned, the purpose of the sessions are to explore data visualization theory, not learn a new tool.

## Tool details

*Mapbox.js* is an enhanced JavaScript library built on top of Leaflet. The reason for selecting Mapbox over Leaflet or OpenLayers was its small footprint, modular, simple to understand code structure and numerous modifiable elements found in the core API. There are also many plugins to further extend projects, robust documentation and a healthy support community found on Stack Exchange.

To start out, users need to create a free Mapbox account and generate a key, this opens up the Mapbox API for the user to use and modify the available objects and classes. Major categories are listed below, with ones that lend themselves well to simple modification highlighted in italics.

## Format support

Formats accepted for import are grouped into four different categories: *TileLayers*, *GridLayers*, *FeatureLayers* and *StyleLayers*. These types of layers can be embedded in the code (as lat/long pairs) or loaded from a URL as a WMS, WFS or geoJSON format.

This is a useful place to discuss data conversions, residency, field types, and the importance of planning ones project. The *StyleLayer* class is used to import the Mapbox hosted base maps for the project, the *FeatureLayer* class is used to load a geoJSON file from a local URL.

## Features

Many different parameters can be set for features loaded into the map. Appropriate selection *symbology* in terms of colour, size, line weights, and symbol selection can be easily explored here. In the figure below the modular structure of the code is highlighted. A fair bit of time can be taken for exploration here.

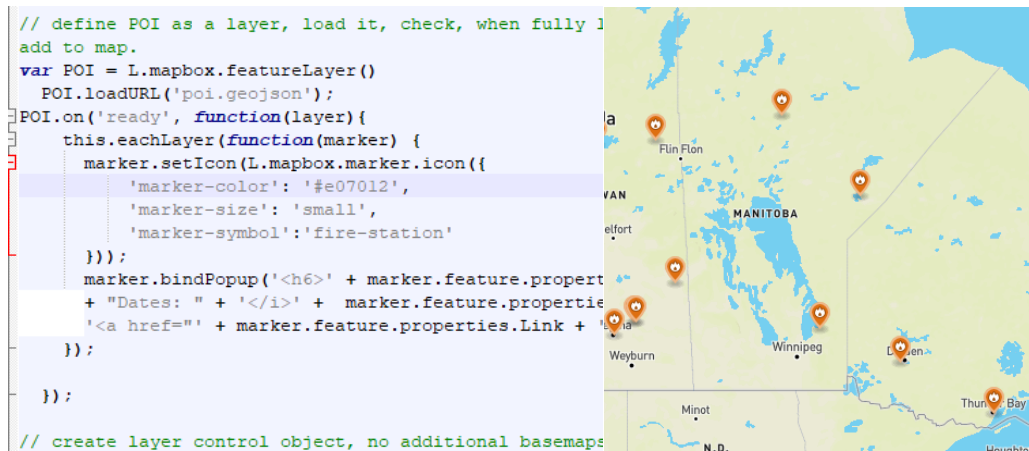


Figure 2: Left- Code block to add and symbolize POIs to a map; Right- Point symbols displayed on a map

## Vector Overlays

In addition to simply displaying features on a map, base data can be imported and displayed on the fly as a choropleth, animated based on a field or clustered to improve the readability of the map. This section is discussed in terms of the purpose behind visualizing data (to clarify data's themes for the audience) and that just because all of these tools are available to the developer, it doesn't mean that they need to be used.

## User Interface

This category is a catch all for everything from panning and zooming actions to pop-ups, embedded media and base-maps, legends and labels. Discussion here focuses on the purpose behind creating the visualization. Who is the audience? What will be helpful for them?

### Base maps

In setting the base tiles for the map many different options are available. Depending on the purpose of the project different bases will be more appropriate than others. Radio buttons can be used as a way to have multiple base map options in one map document. Different options are explored by the students from preselected links on tile sets available for low-bandwidth scenarios, imagery, topographic and artistic renderings.

### Explanatory text

There are many different ways to add explanatory text in Mapbox from basic labels, to legend blocks, tool-tips and pop-ups with embedded media. Creating a hierarchy in text by modifying styles of a pop-up window allows the participants to see how small changes can make the information much clearer for the end user.

### Map Actions

Pan and zoom controls are added to the map face by default are a simple way for the user to have a more intuitive experience. *Filter controls* can also easily be added to the map to keep the interface as uncluttered as possible for the user. Further controls can be added by constraining the map

window to the *bounding box* of a specific feature layer. These parameters are explored in a class setting if there is adequate time.

## Plugins

Countless plugins have been developed for Mapbox by their own team and external developers. They include functions such as Heatmap, Geocode, Directions, Mini-map, Omnivore (importer for GPS tracks, KML, TopoJSON, WKT), Turf GIS and many more. In the classroom context no time is spent actively exploring these plugins, but instead discussion is centered on seeking help in discussion forums, not reinventing the wheel, and knowing when to step back.

Many examples from each of these categories are available on their site with the code heavily commented to explain what is going on. A link to the documentation can be found at <https://docs.mapbox.com/mapbox.js/api/v3.3.1/>.

## Session workflow (1-1.5hrs)

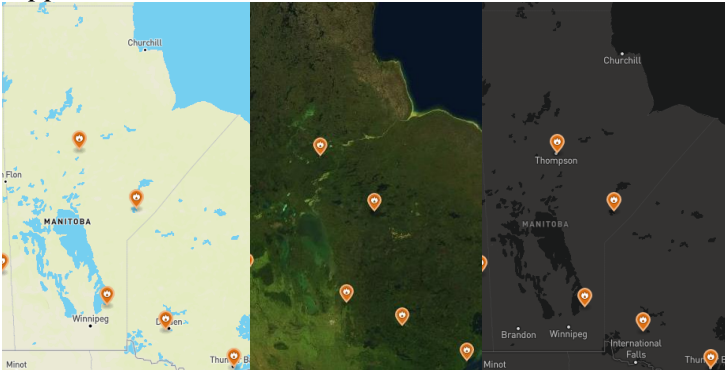
In terms of actual implementation setting up a meeting with the faculty member and their TA beforehand is important so that learning objectives can be discussed. This helps to guide the flow of the class which is based around a scenario that flows through the duration of the session.

### The week before:

1. Students were instructed to create a Mapbox account and generate their API keys and have them ready;
2. Learning materials (folder with .pdf walk through, .geojson files and .html map document) are saved to the class shared space.

### The session:

| <i>Topic</i>   | <i>Action</i>  |
|--|--|
| <b>Session introduction</b>  | <i>Prompt to download materials to their desktop</i>   |
| <b>Define data visualization</b>   |  |
| <b>Introduce the example scenario:</b>   | <i>Participants had been hired on at a forest fire consulting firm to make a web-map depicting old fire locations for their survey crew.</i>           |
| <b>Overview of topics:</b> <ol style="list-style-type: none"> <li>1. Data format</li> <li>2. Symbology</li> <li>3. Audience/ User Experience</li> </ol>                              |  |
| <b>1. Data format</b> <ul style="list-style-type: none"> <li>• Data cleaning</li> <li>• File types</li> <li>• Metadata</li> <li>• Discuss importance of knowing your data</li> </ul> | <i>Participants examine the contents of the data folder and open the POI.geojson file in Notepad++.</i>  |
| <b>2. Symbology</b> <ul style="list-style-type: none"> <li>• Size/ weight</li> <li>• Hue/Value</li> </ul>  | <i>Participants modify the marker code block from a generic blue dot to an orange symbol with a fire symbol in it (see red text in example below):</i> |

|  |  |
|--|--|
| <ul style="list-style-type: none"> <li>• Clarity</li> </ul>  | <pre>marker.setIcon(L.mapbox.marker.icon({   'marker-color': '#e07012',   'marker-size': 'small',   'marker-symbol': 'fire-station'</pre>  |
| <p><b>3. Audience/ User experience</b></p> <ul style="list-style-type: none"> <li>• Visualization type</li> <li>• Explanatory Text</li> <li>• User Experience</li> </ul> | <p><i>Participants modify the base-map code block to contain the layers they feel will be most useful to our audience. Additional data sources are discussed here as a way to supplement base data.</i></p>  <p><i>Figure 3: Examples of stock basemaps from mapbox.js</i></p> <p><i>If there is time they also constrain the map window to the bounding box of the POI layer on load for a better user experience for the client.</i></p> |
| <p><b>4. Conclusion/ Plug for library services</b></p>   | <p><i>Questions from audience</i></p>  |

**Future**

Engineering capstone students, STEM grad students and computer science students have all left positive feedback about feeling engaged with the materials and the level of expertise required. Moving forward, there are two major changes to work through in the coming semesters: a move to an online instructional environment in response to the COVID pandemic and a major version shift by Mapbox.

*Online Instruction:* Up until this point, this style of session has only been conducted in person in the lab environment with code modifications being done in Notepad ++ and previewed in the default web browser. Having everyone working off the exact same set-up means less troubleshooting personal machines settings during the session for the instructor, and being able to circulate and assess where students are at comfort-wise. Two potential solutions for this are to a) shift to a Jupyter Notebook Hub model or b) to use some kind of a coding sandbox like JS Bin or Plunker.

*New software version:* Mapbox.js is no longer in active development and they are focusing all of their efforts on Mapbox GL JS and are marketing it as a modern alternative. The base library is still open and free, but now there are additional pay tiers of usage and support. All support forums

and documentation is being left up, so for my simplistic purposes, do I keep my teaching materials in the old version, migrate to the new one, or shift to core Leaflet?

Note: For some context I am a single person offering support for Data Visualization services, my position is new so I had no learning materials to start with. Theory on its own can be dry, so adding interactive elements helps keep people engaged. Hopefully by discussing the ways which this topic has been approached allows for an easier entry point in adopting a new resource.

Meg Miller  
GIS & Data Visualization Librarian, University of Manitoba  
meg.miller@umanitoba.ca