

mapmap.js: A High-Level Client-Side API for Interactive Thematic Maps

Florian Ledermann¹

¹ Technische Universität Wien, Vienna, Austria

Abstract

Creating interactive thematic maps using web technology is still a complex task. Web mapping APIs can help us by hiding some of the details of the technical platform and offering high-level building blocks - existing open source solutions like the Leaflet API (Agafonkin, 2010) made available a greatly simplified way to display tile-based maps in browsers. From a cartographer's perspective, such tile-based mapping APIs offer little excitement by themselves, as the cartographic quality lies mostly in the tiles and the tools used to produce them (on the server), and not in the methods provided by the API. Furthermore, tile-based maps are not always well suited for fully interactive & animated mapping applications.

In recent years, interest in the possibility to render maps in the web client entirely, instead of downloading pre-rendered tiles, has increased, and solutions that render interactive maps using web technologies like Canvas, SVG or WebGL are currently actively investigated. Low-Level APIs like D3.js (Bostock, Ogievetsky, & Heer, 2011), which is not specifically a web mapping API but rather an abstraction of how to interact with document content and connect it to data, have been successfully used to create interactive maps and other spatial information visualizations. However, as cartographers, we primarily deal with map elements, symbolization, metadata and user interaction, but usually when creating web maps we have to think a lot more about technical details like asynchronous requests, event callbacks or the document object model. The necessity to cater for all those technical aspects to be able to produce a state-of-the art online map is a profound obstacle in creating and teaching interactive web maps, and obscures cartographic decisions and processes in the source code.

With mapmap.js¹, we present our approach to implementing a high-level, open source API for creating interactive thematic maps in the browser. mapmap.js supports geometry, data and metadata loading and transformation, selection and symbolization of map elements as well as specifying interaction through built-in or user-provided methods. Internally, mapmap.js uses state-of-the-art JavaScript features like promise chains or functional inheritance to allow us to expose a clear, simple and concise API to the user. Despite its simplicity, the mapmap.js API aims for transparency, meaning simple things should be simple,

¹ <https://github.com/floledermann/mapmap.js>

but every aspect of the internal implementation should be open to modification or the provision of an alternative implementation by the user.

Figure 1 shows an example of a complete implementation of an interactive choropleth map, showing population data loaded from a csv file.

```
var map = mapmap('#mapEl')
  .geometry('austria_admin.topojson', 'iso')
  .data('unemployment_AT_2014.csv', 'code')
  .meta({
    'unemploym':{
      domain: [0,0.15],
      colors: colorbrewer.YlOrRd[5],
      numberFormat: '%',
      label: "Unemployment Rate"
    }
  })
  .identify(['iso', 'name'])
  .choropleth('unemploym')
  .hoverInfo(['name', 'unemploym'])
  .highlight('Vienna')
  .on('click', mapmap.zoom());
```

Figure 1 - An interactive choropleth map, implemented using the mapmap.js API

The listing in Figure 1 shows only a trivial example - mapmap.js comes with sensible defaults, but every line can be further modified by adding parameters or injecting user-provided functions, and there are many more methods in the API. Mapmap.js is implemented using SVG (Ferraiolo & ed., 2001) and D3 internally, which means generated maps can be styled using CSS and extensions can be implemented using D3. The API works with imperfect data, comes with powerful data transformation tools, and has been successfully used with messy real-world datasets without the need for external preprocessing.

References

- ✓ Agafonkin, V. (2010). Leaflet — an open-source JavaScript library for interactive maps. Retrieved March 9, 2015, from <http://leafletjs.com/>
- ✓ Bostock, M., Ogievetsky, V., & Heer, J. (2011). D3: Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12), 2301–2309. <http://doi.org/10.1109/TVCG.2011.185>
- ✓ Ferraiolo, J., & ed. (2001). Scalable Vector Graphics (SVG) 1.0 Specification.