Using D3.js to Visualise Geospatial Data - Making data accessible online

At develop_for we specialize in creating data visualisations and making data easier to understand and more accessible.

I have had an interest in data and information for a long time and know how powerful it can be. If people can view the data and play around with it they can get a greater understanding of the story and complicated concepts.

Geospatial data is something we have been working with for a number of years. One of our early projects was to map murals around the world. We have worked with leaflet.js to produce maps for clients.

We use D3.js for our data visualisations. It is a javascript library so it will work well on most browsers and integrates well with other javascript libraries. Interactive data visualisations can be produced faster and distributed easily.

With D3.js representing and interacting with geographic information online has become easier. The data can be brought in and programmed with javascript and added to maps to create visualisations such as dot distribution maps, choropleth maps or proportional symbol maps. It is also easy to integrate the data into other types of data visualisations, such as bar or pie charts.

There are two parts to creating geospatial map data visualisations in D3.js. Creating the map and adding the data to the map.

The map is represented as a SVG line and the data for the map is stored as JSON, either GeoJSON or TopoJSON. Each path that will make up the map is a geoJSON feature and this feature is bound to the a SVG path element.

There are more and more geoJSON files available to create different maps, but they can also be created from shape files.

There are a number of different map projections built into D3.js such as Albers and Mercator and the projection is bound to the path that the geoJSON features will be bound to.

The data to be added to the map is bound to the geoJSON SVG map so longitude and latitude coordinates in the data map directly onto the SVG map.

The data can be supplied to D3.js in a number of formats including TSV, CSV and JSON.

The general update pattern that D3.js uses means that new data sets can be introduced or a current data set can be updated and there is a smooth transition between the data.

As D3.js is a javascript library you are also able to manipulate the html and css on the page. This is done by adding styles directly in the code or linking the data dynamically to different classes or id's that are referenced in your main css file.

D3.js is easy to integrate with application development frameworks such as Ruby on Rails. This gives real flexibility in storing and retrieving the data from a database. The data is searchable and different data sets can be linked together.

We have found that there are a number of key advantages to using D3.js. The visualisations are scalable and maintainable. The visualisations are available to a large number of people. The data can be searched and sorted. The user can view the area of information that they want and build up the visualisations that interest them. The visualisation can be made interactive. The visualisations can be made responsive for different devices. Using D3.js is an effective way to create visualisations to communicate data to a large number of people.

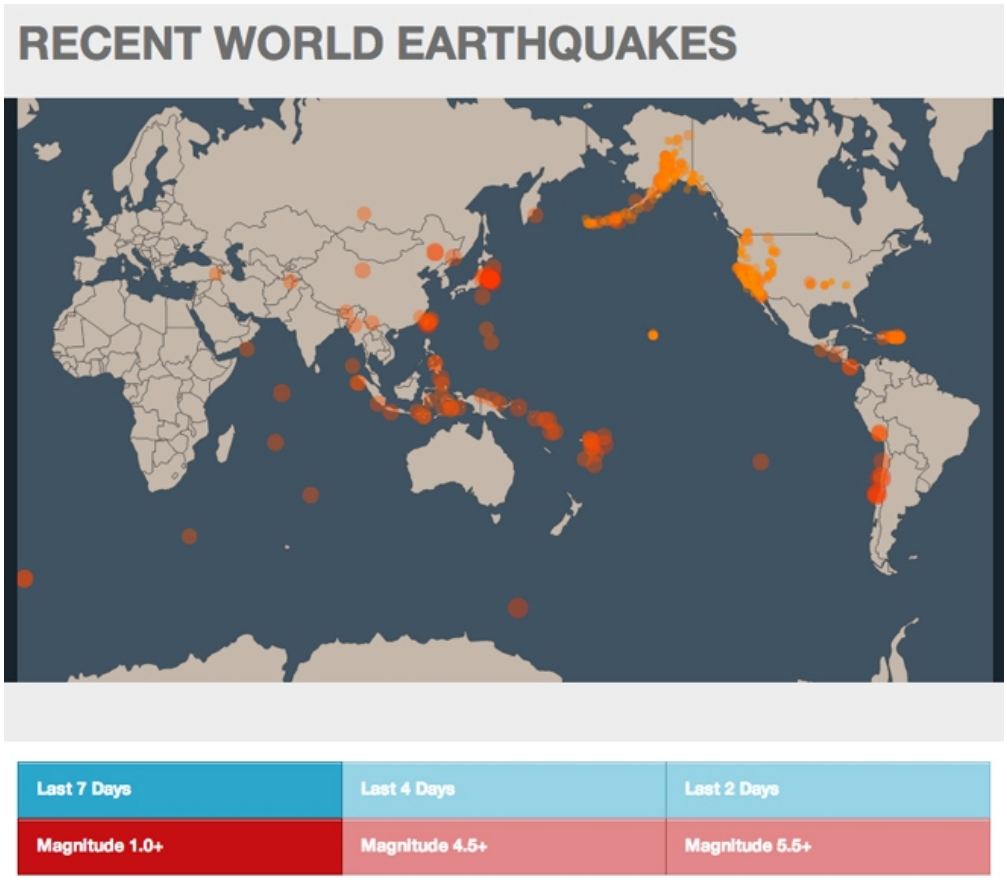
Our current geospatial project is Seismism which maps earthquakes around the world. We want the user to be able to look at current and historical earthquake data and create the map they want via search queries.

We get the data from the The U.S. Geological Survey website which provides the data in a number of formats including XML, CSV and JSON and is regularly updated.

We have taken this data into D3.js to create a map of the earthquakes and also a scatterplot diagram and histogram.

We are building the application with Ruby on Rails and using MongoDB as the database, which keeps the application easy to maintain and update.

Using these technologies we feel that the user gets the information in an easy to understand format that is up to date in an application that is easy to use.



Seismism - main interface