# Ben Sainsbury

## GIS Developer

Email: redtrails@gmail.com
Github: http://github.com/sainsb
Phone: 360.721.6198
Portland, OR

**PORTFOLIO**

Listed below are four software projects that represent my approach to GIS development, user interfaces, and preferred platforms.

1) **RLIS API**

   The RLIS API was created in order to meet the needs of both internal (Metro) and regional developers in need of simple address lookup and query.
   http://gis.oregonmetro.gov/rlisapi2/#jsapi

   The API does not utilize ArcGIS Server, performs very fast, and exposes an autosuggest method to improve the success rate of address searches.
   http://gis.oregonmetro.gov/rlisapi2/#jsapi/autoSuggest

   The query method responds with over 100 properties ranging from watershed to taxlot ID to residential garbage hauler.
   http://gis.oregonmetro.gov/rlisapi2/#jsapi/queryPoint

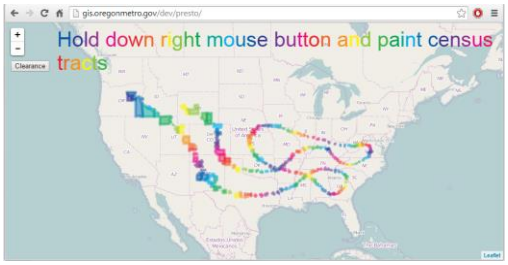
   The RLIS API is built with ASP.NET MVC, ArcObjects, and Oracle(PL/SQL) and has been in production for several years. It is my opinion that APIs such as this represent a promising step toward the ideal of Open Government.

2) **Presto**

   Presto is a very simple demonstration of a new .NET library combined with a custom C# R-Tree index. The code creates a collection of spatial primitives (point, line, polygon) from well-known text (WKT) in the web server's memory (2010 Census Tracts). Each of the features in this collection expose classic methods based on the OGC Simple Feature Specification (e.g. intersect, buffer, contains etc.). As the collection is populated, the extent rectangles are inserted into an R-Tree index, also in memory, allowing for expedited access. This "caching" of features along with a spatial index constitutes an extremely responsive web service, specifically when servicing identify requests to return geometry. Benchmarks have shown speeds of <5ms to obtain geometry when querying

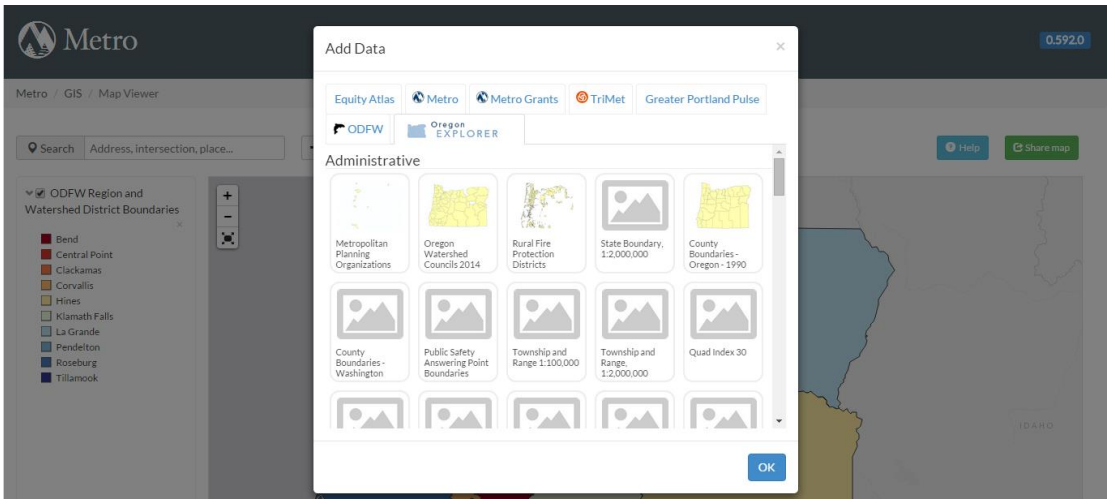
the RLIS Taxlots (>650,000 features) dataset from the service. No database nor GIS web server is required in this unconventional, but effective scenario.
http://gis.oregonmetro.gov/dev/presto/

(Allow the web site to recache the features on load (shouldn't take too long; this is my agency's production web server so I can't leave a bunch of my own stuff in the web server memory). ☺

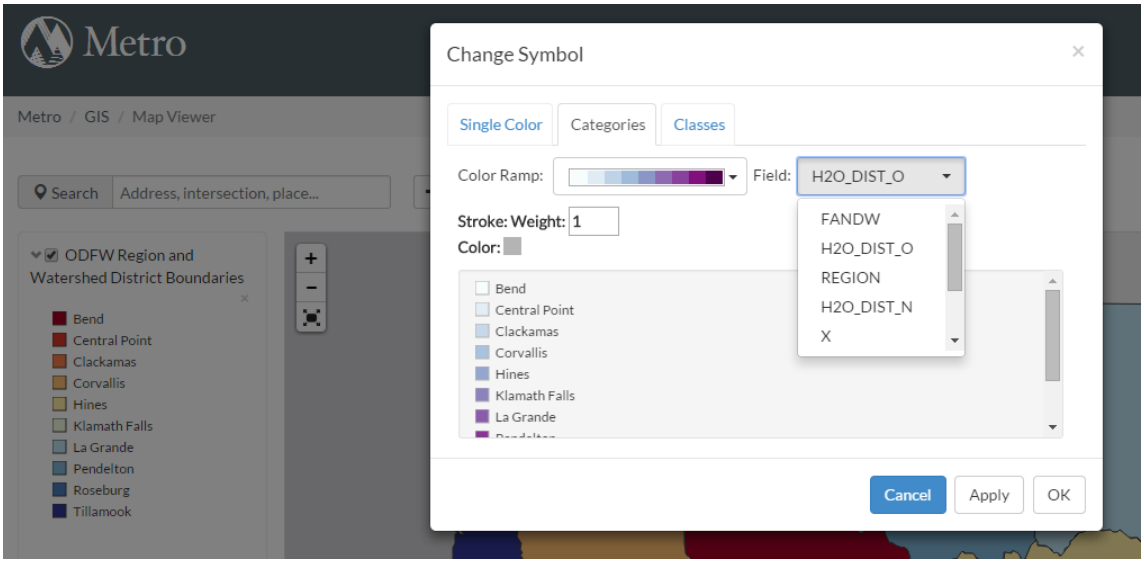


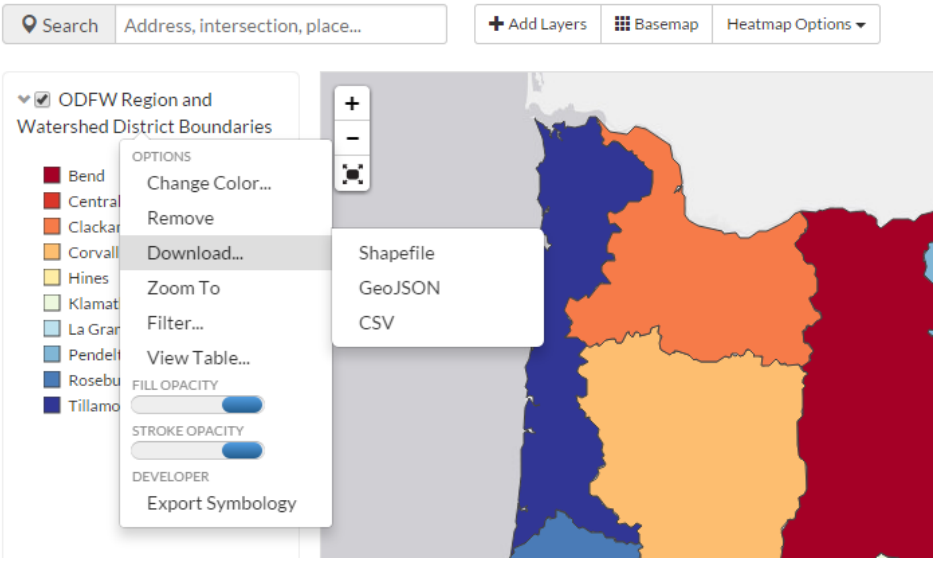

### 3) Allegro

Allegro is a full-featured map viewer that utilizes a variety of JavaScript mapping libraries (Leaflet) as well as some recent innovations in client-side binary file parsing (shapefiles). The web application exposes a multitude of layers, with the unique fact that none of them are map services, but rather Internet-accessible zipped shapefiles on a variety of hosts (Metro, TriMet, Oregon Explorer etc.).




Once brought into the application, the shapefiles are parsed, reprojected, and made available for filtering, resymbolization, table viewing, and export to GeoJSON and CSV.




The app provides a variety of other features including: dragging and dropping a small to medium sized zipped shapefile onto it, reweighting heatmaps a la Metro's Context Tool, reordering layers, full screen mode and more. While very functional, Allegro is alpha.

http://gis.oregonmetro.gov/dev/allegro/

http://gis.oregonmetro.gov/dev/allegro/#7/44.135/-119.833/ODFW-Region-and-Watershed-District-Boundaries

https://github.com/sainsb/allegro

**4) Tilecannon**

Tilecannon is a framework that allows for the deployment of tile services based on the MBtiles file specification as generated from either Tilemill or the more recent Mapbox Studio. The framework includes a REST-style interface for exploring tile services and additionally supports UTF Grids, TileJSON, and the WMTS specification. Tilecannon is used in production at Oregon Metro.

http://gis.oregonmetro.gov/services/
https://github.com/sainsb/tilecannon