

CS 6315 Project Checkpoint

Team: Housing Inspection Explorer Team

Title: Housing Inspection Explorer: A Tool For Viewing Inspection Scores Among Public And Multihousing Units

Project Type: Simple Project

Collaborators

- Phillips, Christopher Trevor [ctp180001]
- Nazneen, Amtul [axn180041]

Building LOGD Visualizations

The plan is to provide a visualization of the U.S Map with an option to drill down to the State -> City -> County and view the Inspection Scores for the chosen criteria. To support our listed features in the Project Proposal, we also intend to provide input fields that allow users to choose a 'City' or a 'Year' and view Inspection Scores accordingly. This is an abstract idea right now as we are in the middle of researching other potential ways to visualize our data.

Understanding LOGD Data

The selected datasets (1258,1259) are quite similar with the only distinction being the type of housing – Multifamily vs. Public Housing. So, understanding them was relatively easy. Both the datasets have the data regarding the Inspection Scores given to the Multifamily/Public Housing properties where the properties are stored in their complete address along with the location coordinates.

Understanding LOGD Metadata

Vital attributes in both the datasets include: Property Identification Number, Full Property Name, Property Address - (City, State Name, State Code, Zip), Property Location Coordinates (Latitude, Longitude), Inspection Score and Inspection Date. Dataset_1258 (for Public Housing) additionally includes - Public Housing Authority Code, Public Housing Authority Formal Name in the fields.

Mashing Up LOGD Data With SPARQL

As mentioned previously, both datasets are quite similar, and thus share many of the same attributes. In particular, both datasets share a “state_name” attribute, which may be used as a “link” between the datasets. Thus, this attribute (along with the other shared attributes, if needed) will provide a means to combine the datasets within the WHERE clause of our SPARQL queries.

How to Find Datasets Using the LOGD SPARQL Endpoint

Until about a week ago, we were able to utilize the LOGD SPARQL endpoint (<https://data-gov.tw.rpi.edu/sparql>) to access our chosen datasets. The datasets were obtained by using the GRAPH construct in SPARQL, which allows one to designate a specific named graph to query against. Although we have now transitioned to using Fuseki (discussed more in “Problems Encountered”), we still access the datasets using the same GRAPH construct, and the SPARQL queries themselves haven’t changed.

Any Other Research Performed

We began investigating the process of executing SPARQL queries via code (e.g. AJAX).

SPARQL Query Progress

Up to this point, we have focused on writing SPARQL queries that will allow us to return data in tabular form (similar to a relational database) to facilitate our ability to explore and understand it. As such, we have written two SPARQL queries that return all data for all attributes in datasets 1258 and 1259, where each row represents a single data item and each column represents a specific attribute of the data item.

Any Results So Far

Most of our work up to this point has been concentrated on writing SPARQL queries to retrieve data, setting up Fuseki, and investigating means of executing SPARQL queries via code, so we don't have specific results to share at this time. However, we are proceeding at a good pace and are looking forward to producing results as the project proceeds.

Problems Encountered

The SPARQL endpoint at <https://data-gov.tw.rpi.edu/sparql> (which provided SPARQL access to the LOGD data) unfortunately stopped working within the last week, and given the age of the website, it's unlikely it will ever be restored. Thus, we will have to utilize Fuseki in our project to allow us to perform SPARQL queries. We have already installed Fuseki and loaded our datasets into it, and initial work seems to indicate it performs our queries with a similar speed as the SPARQL endpoint previously did.

Changes in the Project Plan

The only major change is that we will now implement Fuseki for our data storage and SPARQL query needs. Otherwise, nothing has changed- that is, we are still using the same data sets, and we still expect the same results in our final project.