

Proposal for GSoC 2018

Implement Geospatial Data Viewer in pgAdmin4 for PostGIS

Contact Details

- Name: Xuri Gong
- Nickname: Suri
- Country: China
- Email: xurigoong@gmail.com
- Phone: +86 18811785238
- Github: <https://github.com/Goong>
- Personal blog: <https://gooong.cn>
- LinkedIn: <https://www.linkedin.com/in/xurigong>

Project

Project Name

Implement Geospatial Data Viewer in pgAdmin4 for PostGIS

Project Description

pgAdmin is the GUI management tool for PostgreSQL. The latest version pgAdmin4 is web-based and written in Python and jQuery with Bootstrap, using the Flask framework. But currently there is no geospatial data viewer in pgAdmin4, and external applications, such as QGIS are required if we want to browser the data in a map. It's not convenient for users to view geometries directly in pgAdmin4 "Data Output" table panel.

I propose to implement a geospatial data viewer in pgAdmin4 that will allow users to view the tables in a spatial database and the results of queries executed.

With the viewer I intend to develop, users can directly view geospatial data on a map within the pgAdmin4 GUI. Refer to the [project wiki](#) to get more detail.

For future development, the viewer can be expanded for selecting and editing geospatial data directly in the GUI.

Pre-design

Interface Design: pgAdmin4 web interface uses several panels to display the result of queries. As a preliminary design, I propose to develop an additional panel to display the map within (see Figure 1). The browser renders both the grid panel to show table and the viewer panel to show map after each query. Then users can browse the geometries by zooming and panning in the map.

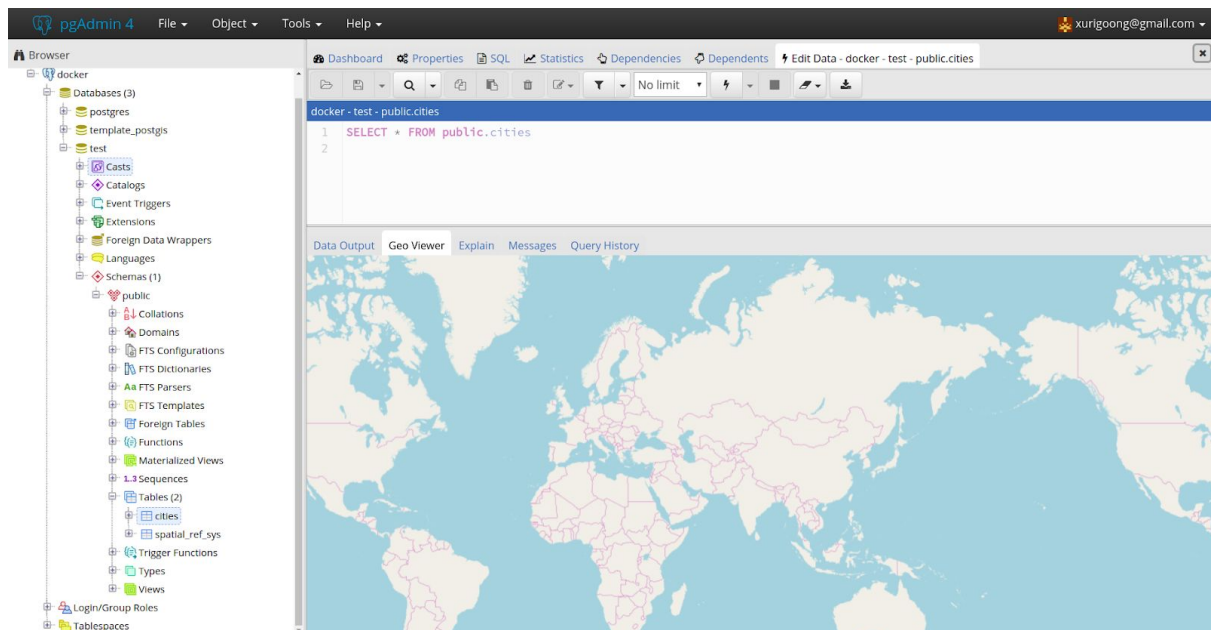


Figure 1: Geo Viewer Panel

Program Flow Design: After each user query, the browser receives the output data if the query succeeds. Then the “Data Output” renders table to show output data. If the data contains geometry data such as point, polyline and polygon, then the program will parse the data (maybe in different format such as WKB, WKT, GeoJSON etc.) and render “Geo Viewer” panel to show map (see Figure 2).

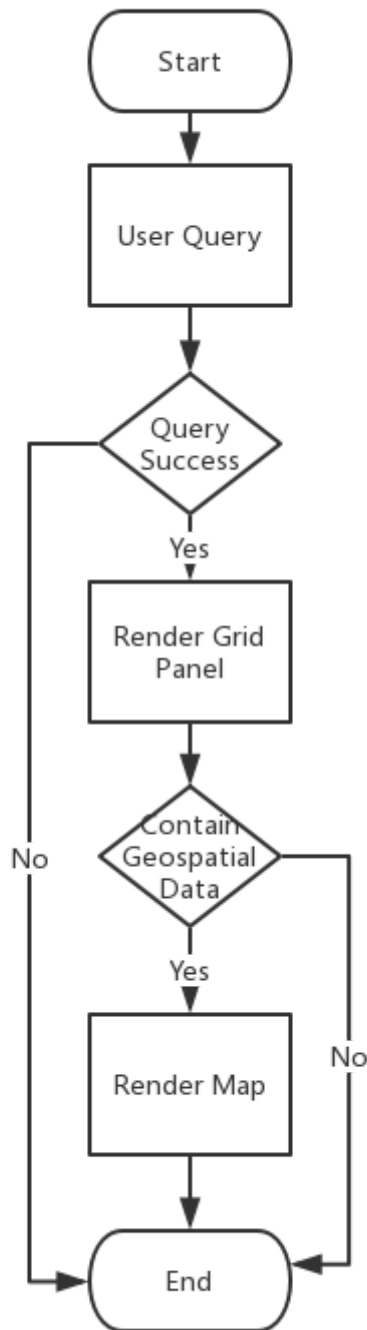


Figure 2: Flow Chart

Technology Roadmap: Since pgAdmin4 is web based, I will need to use JavaScript library for web mapping. Between the two main web mapping libraries: OpenLayers and Leaflet, I prefer the former because OpenLayers is a lot more mature than Leaflet and support various formats of geometry, which will make it easier to parse the query result. Then only minor changes need to be

made on the server side and the main part of the coding work is to implement some specific functions to parse data and render map in “sqleditor.js” file.

More to Consider:

- **Geometry and Geography.** PostGIS supports both Geometry (based on the OGC Simple Features) and Geography (spatial features represented on "geographic" coordinates) data type. The viewer should be able to handle data in both types.
- **2d, 3dm, 3dz and 4d geometries.** OGC “Simple Features” format only support 2d geometries and PostGIS EWKT/EWKB extends the format by adding 3dm, 3dz, 4d coordinates. I will decide if the viewer should support 3d and 4d geospatial data. Since there is no 3d rendering api in OpenLayers currently, I may use [OL-Cesium](#) library if 3dm, 3dz and 4d geometries are required to be supported.
- **Coordinate system.** PostGIS supports different kinds of geographic coordinate system and projection system and there are over 3000 known spatial reference systems in spatial_ref_sys table. It’s not elegant if the viewer just renders data ignoring the coordinate system. So the program is supposed to transform or reproject the coordinates before rendering web map. The JavaScript library [Proj4js](#) can transform coordinates from one coordinate system to another, including datum transformations. In addition, it can be easily integrated into OpenLayers to transform coordinates. The example can be found [here](#).
- **Multi geospatial columns.** Sometimes there are two or more geospatial columns in the query result. Moreover, the columns can be different geo types and in different coordinate system. For example, points in Mercator projection and polylines in UTM projection. Thus the program can’t render all the columns in one single map because of different projection systems. My solution is to render only one column in one map if the coordinate system conflict with each other. And the viewer will offer column options so that users can choose which column to display.
- **Testing.** The regression framework in pgAdmin4 is designed to execute unit tests for server-side code. It’s required to place test cases under the */tests* directory of the respective module for tests. For JavaScript code, I will use [Karma](#) to run tests.

Timeline

Community Bonding Period (April 24 - May 14)

- Introduce myself and this project in pgAdmin4, PostGIS and SoC mailing list.
- Remain constant touch with my mentors using Google Hangouts. Set up user requirements and discuss the design details with mentors. Settle the final design according to user requirements and feasibility.
- Discuss with mentors about the implementation plan.
- Try to fix bugs to get further understanding for pgAdmin4 source code and require.js lib.
- Set up dev environment and my wiki page for TODO list and weekly report.

Official Coding Period (May 15 - August 16)

Week 1 (May 15 - 20)

- Define server side (Flask API) and client side (JavaScript) classes and functions according to the final design.
- Create static web UI for viewing map.

Week 2 - 3 (May 21- June 3)

- Implement *parse_geo_data* function to parse the query result data in different formats.
- Write document and test the *parse_geo_data* function.
- Create basic map that can render the features with WGS84 Web Mercator coordinates.

Week 4 (June 4 - June 10)

- Implement convert projection function using [Proj4js](#) lib. Support features with coordinate system that is defined in *spatial_ref_sys* table.

- Write document and test the function.

*** First evaluation period (June 11 - 15)**

- Deliver the *parse_geo_data* function and test result.
- The viewer can parse and render geospatial data in different formats and coordinate systems.

Week 5 - 6 (June 11 - June 24)

- Implement the column options that allow users to choose which geospatial data column to render.
- Update web UI and add column options.

Week 7 - 8 (June 25 - July 8)

- Create and execute unit tests in the regression frame for all modules to help catch regressions in the code.
- Check the code style to meet the [pgAdmin coding standards](#).
- Write documentation for the viewer.

*** Second evaluation period (July 9 - 13):**

- Deliver the documentation and tests for the viewer.
- The viewer is nearing completion and can support options for users to choose which data column to render.

Week 9 (July 9 - 15)

- Fix bugs and update document.

Week 10 - 12 (July 16 - August 5)

- Pull request for code review and merge.
- Buffer time for unexpected delay.

*** Final evaluation period (August 7 - 14):**

- Deliver the working implementation of geospatial data viewer.
- Wrap up the project and submit the final evaluation of my mentors.

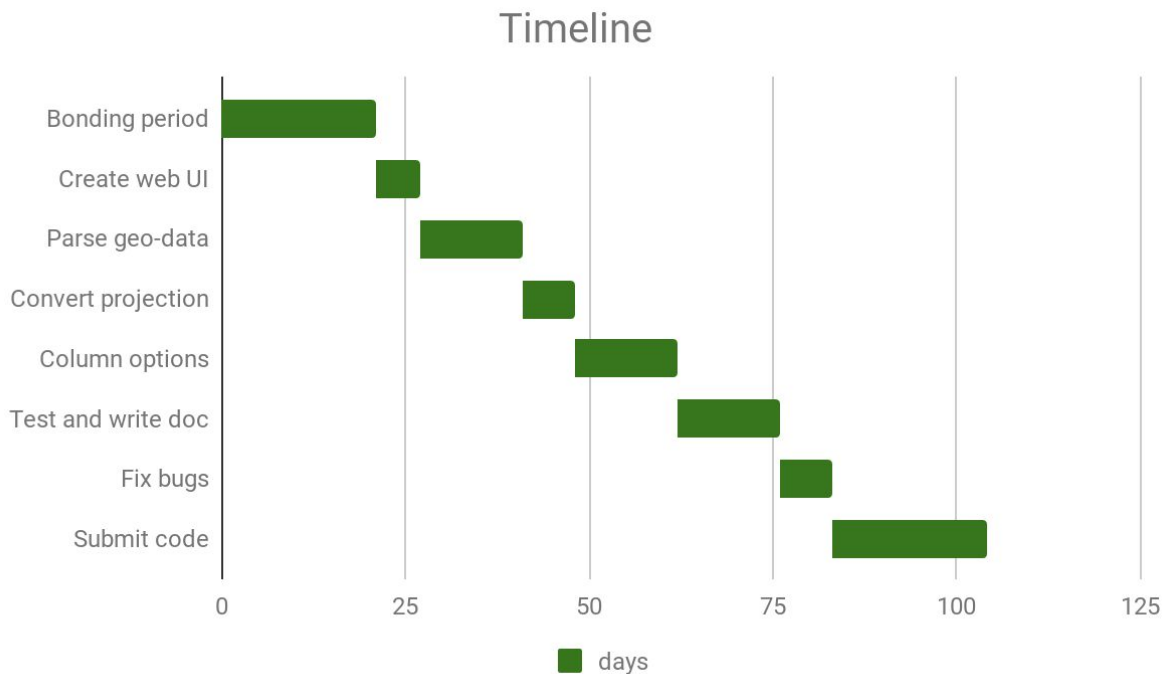


Figure 3: Gantt Chart

Do you understand this is a serious commitment, equivalent to a full-time paid summer internship or summer job?

> Yes, I understand. I will commit my full time in this project and can spend at least 40 hours per week. I have great passion in coding and implementing the data viewer. I am willing to pay my efforts in making the tool more convenient to use.

Do you have any known time conflicts during the official coding period?

> No, I don't have any time conflicts with the official coding time.

Extra Information

Studies

I am an undergraduate student in Peking University. I major in GIS science and use PostGIS very frequently for data storage and analysis.

This project will definitely contribute to my study because it will not only make it more convenient for me to view geospatial data in PostGIS, but also help me improve skills in web mapping, coding, writing documentation and getting involved in open source community.

Programming and GIS

I am familiar with Python, C/C++, Java and have the basic knowledge of html, JavaScript and CSS. Also, I have the experience in using Flask framework and OpenLayers. I use Window 10 and Ubuntu 16.04 LTS for daily basis.

As a GIS student, I use GIS software and development tools almost every day. I use PostGIS, ArcMap, QGIS, ENVI to for spatial analysis and OpenLayers, GeoTools, Arcpy, GeoServer for developing. I have the knowledge of geospatial data (e.g. projection system, spatial analysis and mapping) as well as part of OGC standards (e.g. Simple Features Interface Standard, WMS, and WFS). Here are my two personal GIS projects:

- **Weather Map of China.** The project is to show the real-time weather in China. I develop a web crawler to get real-time weather data of each region in China (see Figure 4), and then display the result in web map using OpenLayers. The weather data and geojson data are published using Flask API. Users can browse each place by dragging, clicking, scrolling and searching. Please [click here](#) to view the Weather Map. The source code is available on [my github](#).



Figure 4: Weather Map

- **Light GIS.** A light GIS desktop software for showing, querying, editing and rendering map. It is written in C# from the scratch without any third-party library. The source code is available on [my github](#).

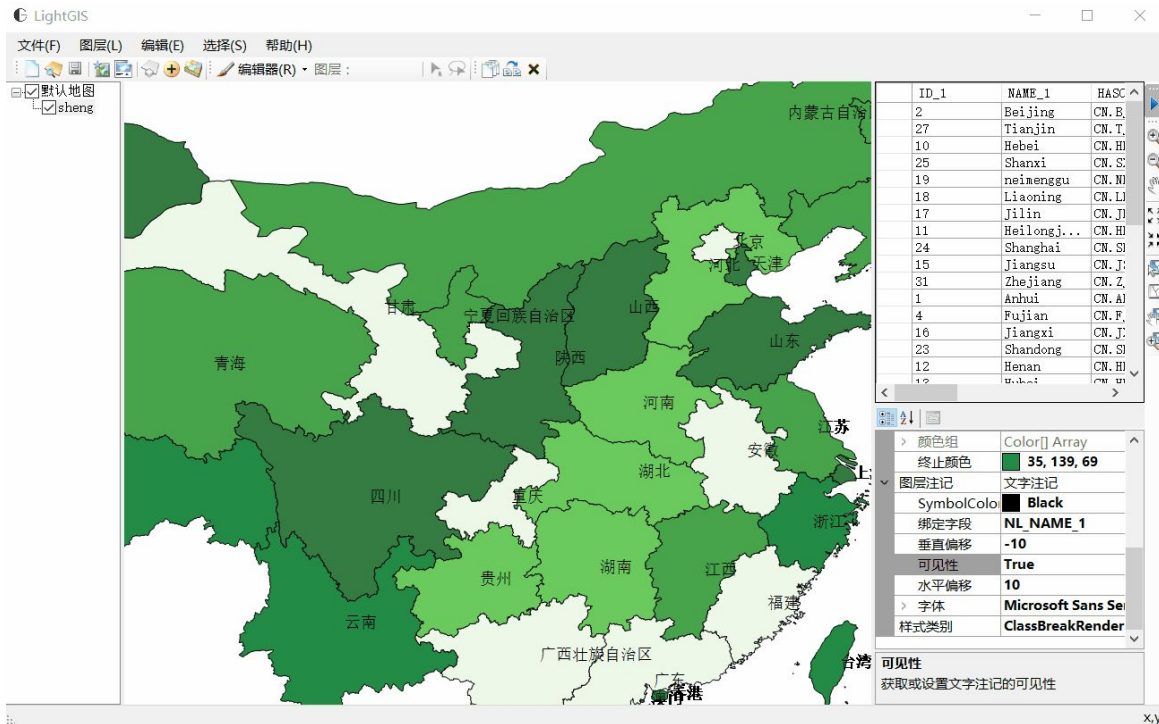


Figure 5: Light GIS

GSoC participation

This is my first time to apply for Google Summer of Code. I did not submit a proposal to any other organisation.