Geostreams API

By enabling the Postgres plugin, you gain access to Clowder's Geostreams API. This module supports definition of Sensors, Streams and Datapoints in Clowder that allow for geospatial visualization and querying.

Installation & Initialization

1. Install Postgres

2. Create necessary database & roles in Postgres command-line
   - `createdb geostream`
   - `psql geostream`
   - `CREATE ROLE clowder;`
   - `ALTER ROLE clowder with LOGIN;`
   - `ALTER ROLE clowder with password 'clowder'`
   - `CREATE EXTENSION Postgis;`

3. To prepare database tables, execute geostreams.sql (available on GitHub)

   - `https://opensource.ncsa.illinois.edu/bitbucket/projects/CATS/repos/clowder/browse/geostream.sql`
   - `psql -d geostream -f ~/geostream.sql;`

4. By default, the SQL script will assign database/tables to owner 'clowder'. If you specify a different Postgres user in Clowder (see Customization below) you should make sure databases/tables are owned by that user.
   a. Or you can add the following in Clowder to custom/custom.conf:

      `postgres.user=clowder`

      `postgres.password=clowder`

5. Enable Postgres plugin in Clowder by adding the following to `clowder/custom/play.plugins`:
   - `10502:services.PostgresPlugin`

Once installed you should see a "Sensors" entry in the top menu of Clowder, alongside "Help".

Customization

The following values can be changed in clowder/custom/custom.conf.

Postgres user configuration:

- **postgres.user**=clowder
- **postgres.password**=postgresPassword

Location where data for geostream API calls will be cached:

- **geostream.cache**=/tmp/medici
- **geostream.dashboard.url**=""

`These properties are used when the geostreaming service returns data as type CSV instead of JSON:`

- **json2csv.ignore**="type,geometry|type"
- **json2csv.fixgeometry**=true
- **json2csv.seperator**=|
- **json2csv.hideprefix**=true

Geography vs. Geometry

PostGIS offers two spatial types, "geometry" and "geography".

- **Geometry** refers to Cartesian data, i.e. specifying coordinates on a flat plane. Conceptually the spherical Earth is flattened and an origin point is assigned, and from that point geometric coordinates refer to distance from that origin and distances between two points can be easily measured. One unit of distance is the same no matter where you are on the Earth's surface.

  In PostGIS, geometric types can be any arbitrary projection/coordinate system.

- **Geography** refers to geographic data, i.e. specifying angular coordinates on a sphere. The angles describe the distance from a reference meridian (longitude) and the equator (latitude). Due to the distorted surface of the sphere, one unit of distance (degrees) will vary depending on how close you are to the poles - one degree at the equator is a larger distance than one degree at the North Pole.

  In PostGIS, geographic data must be projected into WGS '84 Geographic Coordinate System (EPSG:4326) lat/lon Calculated distances will be automatically returned in meters.

So why choose one over the other?

- Geography, being newer, does not have access to the all of the functions that Geometry does, even though you can dynamically cast between the two with some effort. Geography does allow for Intersection, Coverage and Buffer operations (in addition to basic Distance, Length, Area calculations).

- Geography calculations are slower, given they are operating on a sphere rather than a plane. It also requires data to be in EPSG 4326, which can be an impediment but can also act as a useful standardization tool for varied input data. Geography allows any (or no) projection/CRS.

- Geography tends to be more suitable for large (continental scale) analyses, while geometry is suitable for smaller (city scale) analyses where the curvature of the Earth is less impactful. An example from the first link below:

  Here, geography (red) shows the actual shortest great circle path between LAX and CDG, while geometry (purple) shows the shortest path on a flat plane that is incorrect.

- From "PostGIS in Action" (Obe, Hsu):
  - "When choosing between the geometry and geography type for data storage, you should consider what you'll be using it for. If all you do are simple measurements and relationship checks on your data, and your data covers a fairly large area, then most likely you'll be better off storing your data using the new geography type. Although the new geography data type can cover the globe, the geometry type is far from obsolete. The geometry type has a much richer set of functions than geography, relationship checks are generally faster, and it has wider support currently across desktop and web-mapping tools."

Further reading:

http://workshops.boundlessgeo.com/postgis-intro/geography.html

http://gis.stackexchange.com/questions/6681/what-are-the-pros-and-cons-of-postgis-geography-and-geometry-types

https://gis.stackexchange.com/questions/26082/what-is-the-difference-between-geometric-and-geographic-columns