Store filtered Open Street Map in PostgreSQL/PostGIS and serve as web service

Introduction

This tutorial describes the steps to carry out to be able to serve specific items filtered from Open Street Map as web service.

Requirements

- · open street map tile
- osm2pgsql (stand alone tool osm2pgsql)
- geoserver
- PostgreSQL/PostGIS

If you do not have a Geoserver and/or PostgreSQL/PostGIS server then install them locally using the installation instruction provided by the specific websites

Setup database

Create a new database and give it a representative name (e.g. osm). Select the new database in the tree and open the query tool (Ctrl+e, or select the query tool from the <tools> menu). Paste in the following four lines of code.

```
CREATE EXTENSION postgis;
CREATE EXTENSION postgis_topology;
```

This will set up the necessary schemes in the database.

Store data in the database

Next, the osm2pgsql tool will be used to get the data from the OSM file into the database. This file can be downloaded from http://customdebug.com/osm/osm2pgsql.zip. Extract this file somewhere on your hard drive.

Also download the style file from https://raw.github.com/openstreetmap/osm2pgsql/master/default.style and place it in the same location as osm2pgsql. This file is necessary to prevent the osm2pgsql tool from looking for the file in a UNIX based location.

Finally, download the OSM data. As an example we will use Mexico as our subset of data. The file we used was downloaded from http://download.geofabrik.de/central-america/mexico-latest.osm.bz2. Extract the contents to a location on the hard drive.

There is an alternative way. OSM can be filtered before downloading, please check http://wiki.openstreetmap.org/wiki/Overpass_API. For Coastlines the following query can be used in the so called Overpass Turbo OSM Api.

```
/*
This has been generated by the overpass-turbo wizard.
The original search was:
"Coastline"

*/
[out:json][timeout:25];
// gather results
(
    // query part for: "Coastline"
    way["natural"="coastline"]({{bbox}});
);
// print results
out body;
>;
out skel qt;
```

The following two commands will perform the database operations necessary for creating the tables in the database that was just created.

```
set PGPASSWORD=password
```

osm2pgsgl.exe -c -k -S default.style -U username -H localhost -P 5432 -d osm location\to\mexico-latest.osm

Replace password, username and location\to\mexico-latest.osm with the correct values. localhost is the server of the database, 5432 the port which the database listens to (default values). Also, it is assumed that default.style is in the same directory as osm2pgsql.exe.

Setup geoserver

First, download the following two style files. These will be used to give the data its looks.

https://raw.github.com/FlorentDotMe/archives/master/2013.03.23-OSM-Styles-for-GeoServer/sld/style_osm_line.sld

https://raw.github.com/FlorentDotMe/archives/master/2013.03.23-OSM-Styles-for-GeoServer/sld/style_osm_polygon.sld

Navigate to the GeoServer administration console and log in. From the left menu, select Services>WMS. Scroll down to the **Max rendering memory (KB)** options and set this value to 262144.

Next, go to Data>Styles from the left menu and select Add a new style.

Styles



Fill in the properties and use the contents of the style_osm_line.sld file.

New style

Type a new SLD definition, or use an existing one as a template, or upload a ready made style from your file system. The editor can provide syntax highlight and be brought to full screen. Click on the "validate" button to verify the style is a valid SLD document.



Next, go to Data>Styles from the left menu and select Add a new style, but now fill in the properties and use the contents of the style_osm_polygon.sld file.

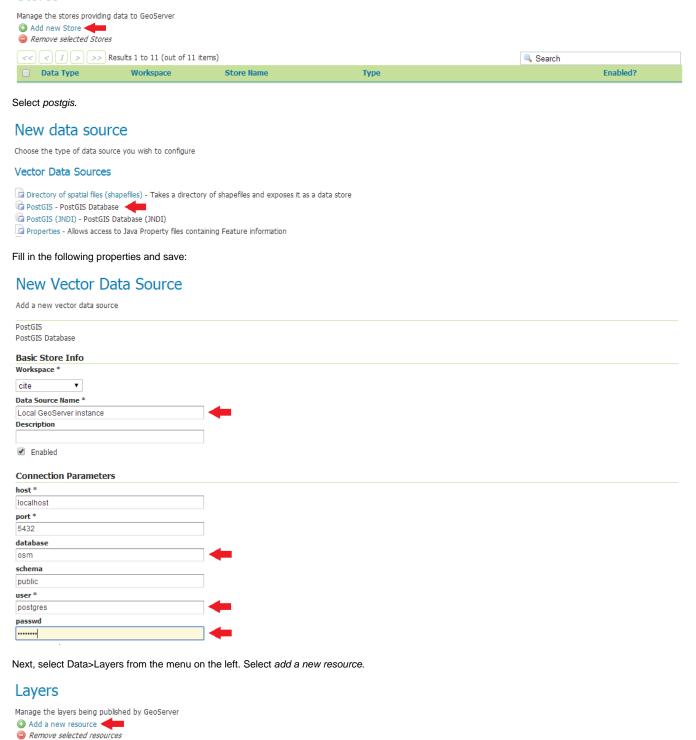
New style

Type a new SLD definition, or use an existing one as a template, or upload a ready made style from your file system. The editor can provide syntax highlight and be brought to full screen. Click on the "validate" button to verify the style is a valid SLD document.

Name
osm-polygon
Workspace
cite ▼
Copy from existing style
Choose One ▼ Copy
〕 ⓒ □ ≡ 12pt ▼
75U <sld!maxscalelenominator>75UUUUU</sld!maxscalelenominator>
751 <sld:polygonsymbolizer><sld:fill></sld:fill></sld:polygonsymbolizer>
752 <sld:cssparameter name="fill">#ededed</sld:cssparameter>
753
754
755
756 <sld:featuretypestyle></sld:featuretypestyle>
757 <sld:rule></sld:rule>
758 <pre></pre>
759 <pre>cogc:PropertyIsEqualTo>cgc:PropertyName>natural</pre> ropertyName>cgc:Literal>wetland/ogc:PropertyIsEqualTo>
760
761 <sld:maxscaledenominator>400000</sld:maxscaledenominator>
762 <sld:polygonsymbolizer×sld:fill>sld:GraphicFill>sld:GraphicFill>sld:GraphicFill>sld:ExternalGraphic>sld:DilneResource xmlns:xlink="http://www.w3.org/1999/xlink" ></sld:polygonsymbolizer×sld:fill>

Select Data>Stores from the menu. Select add a new store from the top.

Stores



Select the Local GeoServer instance from the dropdown list and select publish from the action column for the planet_osm_line layer.

Layer Name

Search

Enabled?

Native SRS

<< | < | 1 | > >> Results 1 to 21 (out of 21 items)

Store

Workspace

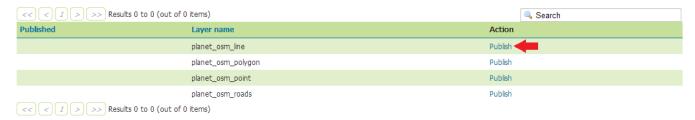
Type

New Layer

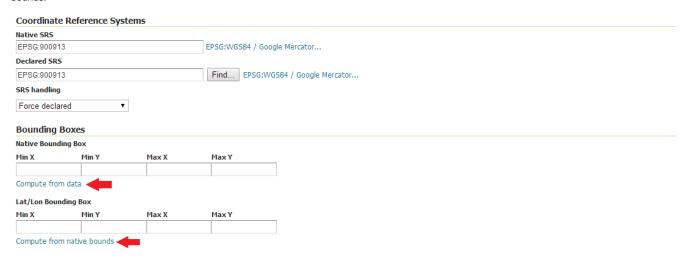
Add a new layer

Add layer from cite:Local GeoServer instance ▼

You can create a new feature type by manually configuring the attribute names and types. **Create new feature type...**On databases you can also create a new feature type by configuring a native SQL statement. **Configure new SQL view...**Here is a list of resources contained in the store 'Local GeoServer instance'. Click on the layer you wish to configure



In the first tab (Data), make sure Native SRS and Declared SRS are both set to EPSG:900913, then select compute from data and compute from native bounds.



In the second tab (Publishing), select the osm-line style. Then scroll down and press the save button.



Perform the same actions to add the polygon layer.

Step 1

New Layer Add a new layer Add layer from cite:Local GeoServer instance ▼ You can create a new feature type by manually configuring the attribute names and types. Create new feature type... On databases you can also create a new feature type by configuring a native SQL statement. Configure new SQL view... Here is a list of resources contained in the store 'Local GeoServer instance'. Click on the layer you wish to configure Search Published Layer name Action Publish planet_osm_line planet_osm_polygon Publish Publish planet_osm_point planet_osm_roads Publish << | < | 1 | > >> Results 0 to 0 (out of 0 items) Step 2 **Coordinate Reference Systems** Native SRS EPSG:WGS84 / Google Mercator... EPSG:900913 Declared SRS Find... EPSG:WGS84 / Google Mercator... EPSG:900913 SRS handling Force declared **Bounding Boxes** Native Bounding Box Min Y Min X Max X Compute from data Lat/Lon Bounding Box Min X Min Y Max X Max Y Compute from native bounds -Step 3 WFS Settings Per-Request Feature Limit Maximum number of decimals

Testing the geoserver

WMS Settings

Queryable

Default Style

osm-polygon

Everything should be setup now and the server can be used to retrieve geospatial data from the database. To test whether the layers are correct, it is possible to create a layer preview. This can be performed by selecting Data>Layer Preview. From the Common Formats column select *OpenLayers*. Other formats can also be tested by selecting a format from the dropdown menu from the *All formats* column.