Implementing an EMODnet borehole Web Feature Service with GeoServer

Version 2.0
May 2018

Date: 24/05/2018

Prepared by: Geological Survey of Slovenia

1. Introduction

This guidance describes how to set up a web service server using free and open source Geoserver software to provide an EMODnet/EPOS-lite ISO/OGC Web Feature Service (WFS) version 2.0.

This guidance is in great extent adopted from "How To Serve a GeoSciML Version 4.1 Web Feature Service (WFS) Using GeoServer" cookbook available at http://onegeology.org/docs/technical/OneGeologyWFSCookbook_v1.4.pdf.

2. Pre-requisites / System Requirements

The cookbook is technical and some assumptions are made about the reader's background knowledge:

- The reader is a, or is working closely with an expert in the data model and schema for the particular INSPIRE theme for which they are planning to supply data.
- The reader has some familiarity with setting up web servers and preferably Java servlet containers.
- The reader is able to install software on their machine and can follow the appropriate installation instructions for PostgreSQL, PostGIS and GeoServer documented on the websites for that software.

To set up a production WFS Download service you will need server equipment to run your database and GeoServer. Estimating the level of hardware resources required to support a responsive service is a complex task depending on the amount of your data, its complexity and the demand that will be placed on your service by users. It is out of the scope of this document to give advice on these issues but you can find some assistance from the GeoServer web site and mailing list. To test setting up a service using the example in this guide a modern PC with Intel Core or equivalent processor and 4Gb RAM should certainly be adequate and you can probably get away with less.

The software required includes the PostgreSQL database with PostGIS spatial extensions, Java, a Java servlet container such as Apache Tomcat, and GeoServer itself. If you use one of the GeoServer packages that include Jetty then you won't need to install a servlet container separately. There are a lot of different versions and all these software packages are continuously updated. You may have conditions specific to your site which make particular versions preferable. For example, you may already have your data in a different database system such as Oracle Spatial rather than PostGIS. It isn't possible to cover all possible set ups here. These are the specific software versions that have been used to test the contents of this cookbook: Windows Server 2008, PostgreSQL 9.4 and PostGIS 2.1, Oracle Java 8 and and GeoServer 2.8.3.
The guidance in this cookbook should also work with other versions of the software possibly with minor modifications. You should use GeoServer v2.7 or later to get support for WFS 2.0 features such as paging. v2.4.5 was the earliest version we have successfully used without significant bugs in the complex feature support. Specific GeoServer versions have specific requirements of the version of Java that they will work with. You should check http://docs.geoserver.org/stable/en/user/production/java.html#production-java for the current recommendations. For GeoServer 2.7 at least Java 7 (a.k.a. 1.7) is required and Java 8 seems to work without problems.

3. Installation of Software

The software used here all has extensive documentation and support forums and mailing lists. Here we will just point you to the appropriate places to download the software and get installation instructions.

We assume:

- You have a basic familiarity with relational databases
- You are able to install applications such as PostGIS, Apache Tomcat, GeoServer etc. on a server with your chosen operating system using their project documentation.

Database

If you already have your data in a relational database system you will want to check the Working with Databases section of the GeoServer manual to see if your database is supported by GeoServer and follow the instructions for installation of any extensions that may be needed for this support. PostGIS support is built into the core GeoServer download.

If you want to try the example data or don't have a supported database system you should install PostGIS. The PostGIS Installation page contains instructions on how to do this for different operating systems. For the purposes of testing this cookbook setup we used the Enterprise DB Windows Installer which installs the base PostgreSQL database, has an option to install the PostGIS extension into this and also includes the pgAdmin graphical database administration tool.

You may use a database on the same machine as GeoServer or you can have it on a separate machine which is accessible over a network from your GeoServer machine.

GeoServer

GeoServer has extensive documentation which you should refer to in addition to this cookbook. References will be made to relevant parts rather than repeating too much of what is already there. There is also a the geoserver-users mailing list.

Assumptions

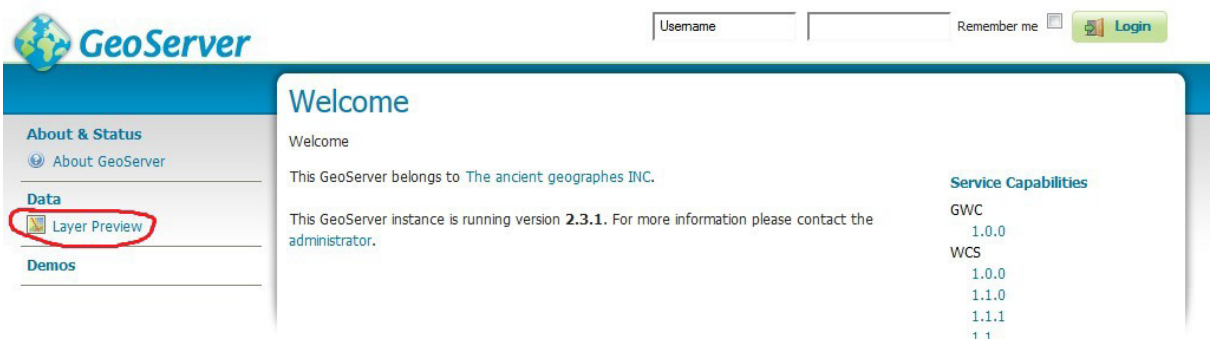
You will need to have Java installed on the machine on which you are going to install GeoServer. The

GeoServer manual recommends Oracle JRE 7 for GeoServer 2.7 at the time of writing but the mailing lists and our own experience indicate that JRE 8 is also fine. See http://docs.geoserver.org/stable/en/user/production/java.html#production-java for the current recommendations. If you want to try other versions of Java you may need to do extra work and be prepared to enter into technical discussions on the geoserver-users email list in order to get it working with these.

Ideally you will be familiar with deploying applications in a servlet container application such as Apache Tomcat or Jetty. If not, however, you can use one of the all-in-one installers to install a stand-alone version of GeoServer. Also, note that Tomcat and Jetty are the two most tested servlet containers used with GeoServer so, although you may well be able to use others, there may be extra work involved and you may need to get advice from the geoserver-users email list.

Basic Installation

- Download and install GeoServer following whichever one of the installation paths in the GeoServer manual suits your situation best. If you already have a servlet container application such as Apache Tomcat set up or you are familiar with how to set one up then you will probably wish to download the web archive (WAR) and deploy that in your servlet container. If you are not comfortable with configuring a servlet container then you will probably wish to use one of the installer programs.
- Run GeoServer at least once to check the installation has worked. If, for example, you are testing in a local instance of Tomcat on your own machine with default settings you should be able to visit http://localhost:8080/geoserver and click on the *Layer Preview* link to check that the example services shipped with GeoServer work properly. You will have to modify the preceding URL appropriately if you have deployed GeoServer on a different machine or port.



App-schema Plugin Installation

- Download and install the app-schema plugin geoserver-*-app-schema- plugin.zip following the application schema installation instructions. Note the WFS service settings.
- Check you can restart GeoServer without any errors being thrown.

INSPIRE Plugin Installation

If you are providing an INSPIRE download service you will need to provide the extra INSPIRE mandated metadata in the WFS GetCapabilities response. This is optional for EMODnet.

- Download and install the INSPIRE plugin geoserver-*-inspire-plugin.zip following the INSPIRE plugin installation instructions.

4. Example Data and Configuration

This tutorial chapter will show you how to set up an EMODnet borehole WFS following EPOS lite schema using the Open Source GeoServer WFS and PostGIS spatial database. It enables you to get some initial experience setting up a service. Your own services may be set up by customising it or you may use it to get some understanding of what is involved when setting up your own service.

The tutorial assumes you are familiar with representing complex features in GML applications.

Loading example PostGIS data

It is assumed that you have installed the PostGIS software and that you have a spatially enabled database (the default installation will create one called postgis). The following will all be working within this spatially enabled database.

This example is following the guidelines of implementing an “EPOS WFS App Schema for BoreholeView” that is available at https://forge.brgm.fr/projects/epos/wiki.

Create a separate schema for the example data. The schema and table names in the EPOS guidelines are used in the database creation scripts and Geoserver’s application schema configuration files so, if you change it from epos and borehole_eu you will need to edit the scripts and application schema configuration files accordingly.

```
CREATE SCHEMA epos AUTHORIZATION postgres;
```

App schema provided in EPOS SVN https://forge.brgm.fr/svnrepository/epos/trunk/tools/geoserver/ is actually set up for one table called "borehole_eu" with those columns:

```
CREATE TABLE harvest.emodnet_geology_borehole_index
(
 id character varying(255) NOT NULL,
 description character varying(7000),
 name character varying(255),
 gmlidentifier character varying(255),
 gsmlpidentifier character varying(255),
 purpose_href character varying(2000),
 purpose character varying(255),
 status_href character varying(2000),
 status character varying(255),
```

```
  drillingmethod_href character varying(2000),
  drillingmethod character varying(255),
  operator character varying(255),
  driller character varying(255),
  drillstartdate date,
  drillenddate date,
  startpoint_href character varying(2000),
  startpoint character varying(255),
  inclinationtype_href character varying(2000),
  inclinationtype character varying(255),
  boreholematerialcustodian character varying(255),
  boreholelength_m real,
  elevation_m real,
  elevation_srs character varying(2000),
  positionalaccuracy character varying(30),
  source character varying(2000),
  parentborehole_uri character varying(2000),
  metadata_uri character varying(2000),
  genericsymbolizer character varying(255),
  cored boolean,
  accesstophysicaldrillcore boolean,
  boreholeuse_href character varying(2000),
  boreholeuse character varying(255),
  detaileddescription character varying(2000),
  geophysicallogs_href character varying(2000),
  geophysicallogs character varying(2000),
  geologicaldescription_href character varying(2000),
  geologicaldescription character varying(2000),
  groundwaterlevel_href character varying(2000),
  groundwaterlevel character varying(2000),
  rockgeochemistry_href character varying(2000),
  rockgeochemistry character varying(2000),
  poregaschemistry_href character varying(2000),
  poregaschemistry character varying(2000),
  geotechnicalinfo_href character varying(2000),
  geotechnicalinfo character varying(2000),
  groundwaterchemistry_href character varying(2000),
  groundwaterchemistry character varying(2000),
  location_wgs84 geometry(Geometry,4326),
  detaileddescription_href character varying(255),
  CONSTRAINT borehole_id_key UNIQUE (id)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE harvest.emodnet_geology_borehole_index
  OWNER TO harvester;
```

Create a new table using the above SQL script.

You should create a database user with read-only access to these tables for the WFS software to use when accessing them. With more recent versions of PostgreSQL you can use the shorter syntax to grant access to all tables in the inspirewfs schema, with older versions you will have to grant access to each

table individually.

```
CREATE ROLE ows_reader LOGIN PASSWORD 'your_password'
 VALID UNTIL 'infinity';
COMMENT ON ROLE ows_reader
 IS 'A role with read only access to data used in web services.';
grant usage on schema epos to ows_reader;
```

```
grant select on table geometry_columns to ows_reader;
grant select on table spatial_ref_sys to ows_reader;
```

Short syntax for versions of PostgreSQL (works at least as far back as v9.1)

```
grant select on all tables in schema epos to ows_reader;
```

Separate grants for each table for older versions of PostgreSQL (known to be required for v8.4)

```
grant select on table epos.borehole_eu to ows_reader;
```

If you have problems with the above steps which are difficult to resolve you may find that setting log_statement=all in postgresql.conf, reloading the server and then monitoring the log file is helpful for debugging them.

You will now have to fill your database with appropriate data using the method that suits you the most.

Configuring GeoServer WFS

Download the example configuration files from https://forge.brgm.fr/svnrepository/epos/trunk/tools/geoserver/. Copy the files from this directory to the matching locations in your GeoServer data directory. The main configuration files are inside the workspaces directory. Copy its content to the data/workspaces directory.

Copy the file app-schema.properties to WEB-INF/classes/app-schema.properties and edit the database connection parameters appropriately for wherever your installation of PostgreSQL is running. If you want to use a JNDI data connection configured in your servlet container then you will also need to edit the appropriate places in the datastore.xml files described in a subsequent section. So it will be easier for initial testing just to enter the host localhost, port 5432, database postgis, schema epos, user ows_reader and password your_password parameters.

Perform any configuration required by your servlet container, and then start the servlet.
When you set up your own data use the web GUI to change the service metadata values or use the configuration files (global.xml, wfs.xml,...) at the top level of the data directory.

.

One configuration item you may need to change is to increase the memory available for Java. The method depends on how you have installed GeoServer but if you get `java.lang.OutOfMemoryError: Java heap space` errors with the request below you will need to increase the memory with a directive such as `-Xmx256M`. The details of tuning memory and other options of the Java Virtual Machine are complex and not dealt with in this cookbook. Some information is in the GeoServer User Manual under the Running in a Production Environment section.

If you have used the Windows Installer you can apply this by editing the file `C:\Program Files (x86)\GeoServer 2.4.5\wrapper\wrapper.conf` (The exact file location will depend on where you installed GeoServer and which version you are using.) Find the line `wrapper.java.maxmemory=128` and increase the value 128 (or whatever it happens to be) to something like 256.

If you are running in Apache Tomcat on Windows you can use the "Configure Tomcat" program that the Tomcat Windows installer provides. In the "Java" tab you can put a maximum memory value such as 256 (MB) in the Maximum Memory pool field.

- The first time GeoServer starts with the tutorial configuration, it will download all the schema (XSD) files it needs and store them in the `app-schema-cache` folder in the data directory. You must be connected to the internet for this to work.

Complex Feature Test requests

When GeoServer is running, test app-schema WFS in a web browser. You can query the feature types using these links. (Change `localhost:8080` in the examples below if you have deployed it at a different location.):

- http://localhost:8080/geoserver/wfs?service=wfs&version=2.0.0&request=GetFeature&TYPENAMES=gsmlp:BoreholeView&OUTPUTFORMAT=text/xml;%20subtype=gml/3.2&count=25

Note that the web interface does not yet support app-schema store for layer administration
App-schema cannot be configured using the web interface, you will need to edit the configuration files directly. You will see the configured workspaces and stores appear in the web interface but not the layers (features). The properties that can be edited in the web interface are very limited.

5. Complex Feature Configuration

For the more in depth explanation on configuring complex feature services using app schemas you can refer to the "How To Serve a GeoSciML Version 4.1 Web Feature Service (WFS) UsingGeoServer" cookbook.