

Lecture 7.4 – OpenGIS Consortium Web Services

Luca Morandini

Data Architect – AURIN Project

University of Melbourne

luca.morandini@unimelb.edu.au

Outline of the Lecture

Part 1: The OGC stack

- Design principles
- CSW
- WMS
- WFS
- WPS

Part 2: Examples of WFS Use From the CLI

Part 3: Use of AURIN OpenAPI with QGIS

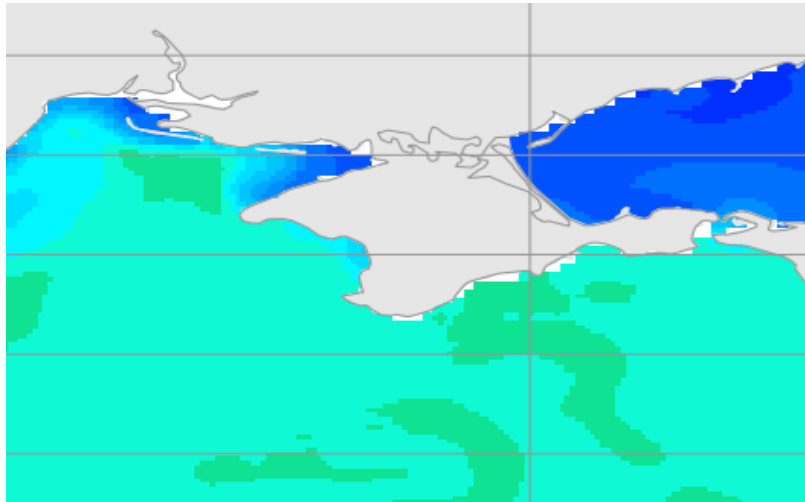
Part 1: The OGC Stack

OGC Services in a Nutshell

The *OpenGIS Consortium* (OGC) is a non-profit group of organizations (companies, universities, state agencies, etc.) that share the common goal of defining standards for all things geo-spatial (i.e. data with geographic information included!)

More to the point, OGC has defined, at the end of the 20th Century, a set of standards that define a family of SOAP-based web-services to support access to geo-spatial data.

Geo-spatial Data Come in Many Flavours



```
▼<gml:featureMember>
  ▼<aurin:ACARA_acara_school_points fid="acara_school_points.1">
    ▼<gml:boundedBy>
      ▼<gml:Box srsName="http://www.opengis.net/gml/srs/epsg.xml#4283">
        <gml:coordinates xmlns:gml="http://www.opengis.net/gml" decimal="." cs="," ts=" ">147.372,-42.871
          147.372,-42.871</gml:coordinates>
      </gml:Box>
    </gml:boundedBy>
    ▼<aurin:the_geom>
      ▼<gml:Point srsName="http://www.opengis.net/gml/srs/epsg.xml#4283">
        <gml:coordinates xmlns:gml="http://www.opengis.net/gml" decimal="." cs="," ts=" ">147.372,-42.871</gml:coordinates>
      </gml:Point>
    </aurin:the_geom>
    <aurin:Year>2008.0</aurin:Year>
    <aurin:ACARA_ID>40000.0</aurin:ACARA_ID>
    <aurin:SchoolName>Corpus Christi Catholic School</aurin:SchoolName>
    <aurin:Suburb>BELLERIVE</aurin:Suburb>
    <aurin:State>TAS</aurin:State>
    <aurin:PostCode>7018.0</aurin:PostCode>
    <aurin:SectorCode>C</aurin:SectorCode>
    <aurin:SchoolType>Primary</aurin:SchoolType>
    <aurin:SectorDesc>Non-Government</aurin:SectorDesc>
    <aurin:Latitude>-42.87097</aurin:Latitude>
    <aurin:Longitude>147.37199</aurin:Longitude>
  </aurin:ACARA_acara_school_points>
</gml:featureMember>
```

Three-letter Services

The OGC web-services cover different ways of interacting with geo-spatial data:

- Finished maps (as the ones you see on Google Maps)
- Vector data (just the geometry, in practice a collection of points)
- Raster data (a mathematical matrix)
- Metadata (information about the geo-spatial data available)

For each of this ways, there is one or more OGC Services:

- Finished maps: WMS, WMTS, SLD
- Vector data: WFS, FE
- Raster data: WCS
- Metadata: CSW
- Processing (remote execution/computation on data): WPS

Part 2: Examples of WFS Use

Vector Data

The relevant service for dealing with vector data is WFS (Web Feature Service), which allows the selection of geo-spatial data using their location and/or their contents.

In OGC-speak, a *feature* is a vector description of a geographic object and associated data (say: the location of a restaurant, its name, address, type, etc). A set of homogeneous features is a *feature type* (say, all the restaurants).

A typical interaction with a WFS data source, entails the following steps:

- Retrieving the list of feature types available from that service
- Retrieving information (metadata) about a feature type
- Retrieving the actual data

Vector Data Selection in WFS

WFS is a SOAP-style of Web Service, and it makes heavy use of XML to request services and to represent returned data (although returned data can be expressed in other format, such as JSON).

Here is how an interaction translate into WFS requests:

- Retrieving the list of feature types available:

[GetCapabilities](#)

- Retrieving information about a feature type:

[DescribeFeatureType](#)

- Retrieving the actual data:

[GetFeature](#)

GetCapabilities

Let's retrieve Melbourne urban trees from the AURIN WFS server... starting with a list of available feature types:

```
curl -X GET
```

```
"https://geoserver.aurin.org.au/vic_lgovt_hume/ows?  
service=WFS&version=1.1.0&request=GetCapabilities"
```

This request returns a list of the feature types that are accessible from that service, and more other information beside that.

DescribeFeatureType

Attributes and general information about a specific feature type:

```
curl -X GET
```

```
"https://geoserver.aurin.org.au/vic_lgovt_hume/ows?  
request=DescribeFeatureType&  
version=1.1.0&  
service=WFS&  
typeName=vic_lgovt_hume:trees_july2015"
```

GetFeature

We are now ready to retrieve the actual tree data (capped to the first three sites).

```
curl -X GET
```

```
"https://geoserver.aurin.org.au/vic_lgovt_hume/ows?  
request=GetFeature&version=1.1.0&service=WFS&typeName=vic_lgovt_hume:trees_july2015&maxFeatures=3"
```

GetFeature Selected by Spatial Extent

Retrieval of waste sites in a given rectangle (between 37.7 and 37.6 latitude south, and between 144 and 145 longitude east), capped to the first 10 features.

```
curl -X GET
```

```
"https://geoserver.aurin.org.au/vic_lgovt_hume/ows?  
request=GetFeature&version=1.1.0&service=WFS&typeName=  
vic_lgovt_hume:trees_july2015&maxFeatures=10&  
&bbox=-37.7,144,-37.6,145"
```

GetFeature Selected by Expression, GET

Let's retrieve all the trees of a given type.

```
curl -X GET
```

```
"https://geoserver.aurin.org.au/vic_lgovt_hume/ows?  
request=GetFeature&version=1.1.0&service=WFS&typeName=vic_lgovt_hume:trees_july2015&filter=<Filter><PropertyIsEqualTo><PropertyName>COMMON_NAME</  
PropertyName><Literal>Sydney Apple Gum</Literal></  
PropertyIsEqualTo></Filter>&maxFeatures=10"
```

GetFeature Selected by Expression, POST

Let's retrieve all the trees of a given type.

```
curl -X POST "https://geoserver.aurin.org.au/vic_lgovt_hume/ows/wfs" --data @./selectByExpr.xml --header "Content-Type:text/xml"
```

Where selectByExpr.xml is:

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:GetFeature xmlns:wfs="http://www.opengis.net/wfs" xmlns:gml="
http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc" service="WFS"
version="1.1.0" outputFormat="text/xml; subtype=gml/3.1.1" maxFeatures="2">
  <wfs:Query srsName="urn:ogc:def:crs:EPSG::3067"
typeName="vic_lgovt_hume:trees_july2015">
    <ogc:Filter>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>COMMON_NAM</ogc:PropertyName>
        <ogc:Literal>Sydney Apple Gum</ogc:Literal>
      </ogc:PropertyIsEqualTo>
    </ogc:Filter>
  </wfs:Query>
</wfs:GetFeature>
```

GetFeature Returning JSON

Let's retrieve the first three trees in JSON.

```
curl -X GET
```

```
"https://geoserver.aurin.org.au/vic_lgovt_hume/ows?  
request=GetFeature&version=1.1.0&service=WFS&typeNam  
e=vic_lgovt_hume:trees_july2015&maxFeatures=3&outputF  
ormat=json"
```


Error handling #1

In a departure from the HTTP way of handling errors, that is, by returning error status code, WFS returns 200 even when an exception occurs, but with an XML describing the error, like in:

```
<ows:ExceptionReport xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ows="http://www.opengis.net/ows" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" version="1.0.0" xsi:schemaLocation="http://
www.opengis.net/ows https://geoserver.aurin.org.au:443/schemas/ows/1.0.0/
owsExceptionReport.xsd">
  <ows:Exception exceptionCode="InvalidParameterValue">
    <ows:ExceptionText>Illegal property name: COMMON_NAME for feature type
vic_lgovt_hume:trees_july2015</ows:ExceptionText>
  </ows:Exception>
</ows:ExceptionReport>
```

The AURIN OpenAPI

The AURIN project has released a draft version of its API, which is a composition of OGC Services (CSW, WFS and WPS).

So far around 400 layers have been released, but its number will increase.

The WFS endpoint is:

<http://openapi.aurin.org.au/wfs>

Credentials are: [student / dj78dfGF](#)

QGIS as WFS Client

- Install QGIS (an open-source desktop GIS)
- Add the AURIN OpenAPI WFS endpoint the list of WFS servers
- Insert credentials
- Choose a layer to be displayed
- Under the hood, QGIS is sending WFS request to the WFS server, as it is shown in the Kibana log

References

- *OGC Standards*

<http://www.opengeospatial.org/docs/is>

- *WFS Reference*

<http://docs.geoserver.org/latest/en/user/services/wfs/reference.html>

- *QG/S*

<http://http://qgis.org>