AN OGC API TO GEOSPATIAL DATA

pygeoapi.io

This presentation available at https://pygeoapi.io/presentations/default
TABLE OF CONTENTS

- Geospatial API evolution
- REST/JSON/OpenAPI/Swagger
- OGC activity
- Say hello to pygeoapi
- Core capabilities
- Deploying
- Production instances
- Roadmap
- Support
GEOSPATIAL API
EVOLUTION
1990s

- Spirit of XML-RPC/CORBA
- SOAP/WSDL/UDDI
- Service Oriented Architecture (SOA)
- Strong concept of RDBMS as the backend
- OGC WMS (1999)
2000s

- Web 2.0
- JavaScript/AJAX/Google Maps
- Slippy maps, tiles
REALITIES OF OGC WEB SERVICES

- Key Value Parameters (KVP) or XML encodings
- Requests using HTTP GET, HTTP POST and SOAP
- Responses as XML (GML)
- Error handling as custom *ExceptionReport* within an HTTP 200 response
- Exhaustive specifications/requirements
<wfs:FeatureCollection
 xmlns:ms="http://mapserver.gis.umn.edu/mapserver"
 xmlns:gml="http://www.opengis.net/gml"[..]>
 <gml:boundedBy>
   <gml:Envelope srsName="EPSG:32661">[..]</gml:Envelope>
 </gml:boundedBy>
 <gml:featureMember>
   <ms:north_poles_wfs gml:id="north_poles_wfs.0">
     <gml:boundedBy>[..]</gml:boundedBy>
     <ms:msGeometry>[..]</ms:msGeometry>
     <ms:Id>0</ms:Id>[..]
   </ms:north_poles_wfs>[..]
 </gml:featureMember>[..]
</wfs:FeatureCollection>
MORE REALITIES
Complex machinery/architectures (e.g. ldproxy) for services to be crawlable
Challenging for web developers to implement
Challenging for mainstream web integration (search indexing)

Geonovum testbed showed OGC world can easily be crawlable using the crafted LDP.
REST

- REpresentational State Transfer (REST)
- HTTP verbs (GET/PUT/POST/DELETE)
- HTTP codes (200, 201, 404, etc.)
- Uses URI to identify resources
- Content negotiation (media types)
- Stateless
JSON

- JavaScript Object Notation
- Easy to write and understand
- Very popular among web developers
- JSON is first class in RESTful web services
- No rigid standards (for better or worse)
OPENAPI

- Specification on how to describe a REST API
- Defines endpoints, request parameters, response
- Programming language agnostic
- Formerly Swagger
"info": {
    "contact": {
        "email": "you@example.org",
        "name": "pygeoapi Development Team",
        "url": "https://pygeoapi.io"
    },
    "description": "pygeoapi provides an API to geospatial data",
    "license": {
        "name": "CC-BY 4.0 license",
        "url": "https://creativecommons.org/licenses/by/4.0/"
    },
    "termsOfService": "None",
    "title": "pygeoapi Demo instance - running latest GitHub version",
    "version": "3.0.2",
    "x-keywords": [
        "geospatial",
        "data",
        "api"
    ]
},
"openapi": "3.0.2"
SWAGGER

- OpenAPI client tooling
- Robust, web developer focused
- Automated documentation, testing, code generation
OGC ACTIVITY

https://ogcapi.ogc.org
W3C SPATIAL DATA ON THE WEB BEST PRACTICES (2017)

- Recommendations on formats, identifiers, access, licensing, provenance
- Being webby
- https://www.w3.org/TR/sdw-bp
DESIGN PATTERNS

- Being webby (humans, search engines)
- Developer friendly
- Lightweight specification development
- Removing HTTP use as a tunnel
  - /ows?
    - request=GetFeature&typename=roads&featureid=5
  - /api/collections/roads/items/5
- Modular specification development
  - Core and extensions
TIMELINE

• 2017: W3C Spatial Data on the Web Best Practices
• 2017: OGC API Whitepaper
• 2018: WFS3 Hackathon, Weather on the Web API
• 2019-2020: OGC API Hackathons
• OGC API Roadmap
FUTURE OGC
STANDARDS
OGC ACTIVITY ON GITHUB
SAY HELLO TO PYGEOOAPI

A Geopython project
Created by Tom Kralidis in 2018 on Valentine's Day with ❤
OSGeo Community Project
PROJECT OVERVIEW

- Geospatial data API framework
- OGC Compliant (already!)
- OSGeo Community Project
- International team (Canada, Netherlands, Greece, Italy, New Zealand)
  - Numerous core contributors
  - 21000 upstream contributors (dependencies)
TECHNICAL OVERVIEW

- Publishing (leaves data pristine/alone)
  - Fits to the data (not the inverse)
- Leverages powerful ecosystem of Python packages
TECHNICAL OVERVIEW

- Core abstract API; web framework agnostic (default Flask)
- YAML configuration (metadata, dataset connections)
- Automated OpenAPI document generation (data binding)
- Robust plugin framework (data connections, formats, processing)
- Easy deployment (pip install, Docker, conda, UbuntuGIS)
- Minimal dependencies
FEATURE DATA PROVIDERS

- Elasticsearch
- CSV
- Spatialite*
- GeoJSON*

* also via GDAL-OGR Provider
FEATURE DATA PROVIDERS - GDAL/OGR

- Via OGR-Python bindings
- Unleashes ~100 Vector Formats! (WFS 1/2, GeoPackage, Shapefile, GeoJSON, PostGIS, ...)
- Paging, reprojection, spatial/attribute query
- Unlock existing WFS 1/2s!
COVERAGE DATA

PROVIDERS

- xarray
- rasterio

- NetCDF, Cloud Optimized GeoTIFF (COG)
TILE DATA PROVIDERS

- minio
- ZXY directory tree
  - MapBox Vector Tiles via tippecanoe
- Lower the barrier to data search
- HTML encoding is annotated as schema.org/Dataset to facilitate search engines
- Google dataset search locates pygeoapi datasets
LINKED DATA - JSON-LD

- JSON-LD provided as additional representation
- Default out of the box functionality for most pygeoapi resources / endpoints
- Additional configuration
pygeoapi Demo instance - running latest GitHub version

pygeoapi provides an API to geospatial data

Terms of service: https://creativecommons.org/licenses/by/4.0/
License: CC-BY 4.0 license

Collections

View the collections in this service

Processes

View the processes in this service

API Definition

Documentation
OpenAPI Document

Conformance

View the conformance classes of this service

Powered by pygeoapi 0.9.dev0
<table>
<thead>
<tr>
<th>Request URL</th>
<th><a href="https://demo.pygeoapi.io/master/collections?f=json">https://demo.pygeoapi.io/master/collections?f=json</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Details</td>
</tr>
<tr>
<td>200</td>
<td></td>
</tr>
</tbody>
</table>

**Response body**

```
{
  "collections": [
    {
      "links": [
        {
          "type": "text/csv",
          "rel": "canonical",
          "title": "data",
          "href": "https://github.com/mapserver/mapserver/blob/branch-7-0/maasotest/wxs/data/obs.csv",
          "hreflang": "en-US"
        },
        {
          "type": "text/csv",
          "rel": "alternate",
          "title": "data",
          "href": "https://raw.githubusercontent.com/mapserver/mapserver/branch-7-0/maasotest/wxs/data/obs.csv",
          "hreflang": "en-US"
        },
        {
          "type": "application/json",
          "rel": "self",
          "title": "This document as JSON",
          "href": "https://demo.pygeoapi.io/master/collections/obs?f=json"
        },
        {
          "type": "application/ld+json",
          "rel": "alternate",
          "title": "This document as RDF (JSON-LD)",
          "href": "https://demo.pygeoapi.io/master/collections/obs?f=jsonld"
        }
      ]
    }
  ]
}
```

**Response headers**

- access-control-allow-origin: *
- content-length: 48471
- content-type: application/json
- date: Tue, 08 Sep 2020 00:22:43 GMT
- server: gunicorn/20.0.4
- x-firefox-epay: h2
- x-powered-by: pygeoapi 0.9.dev0
**Large Lakes**

Items in this collection.

![Map of Large Lakes with labels for Lake Baikal, Lake Winnipeg, Great Slave Lake, L. Ontario, L. Erie, Lake Superior, and Lake Huron.](map.png)

Warning: Higher limits not recommended!

Limit: 10 (default) →

<table>
<thead>
<tr>
<th>id</th>
<th>id</th>
<th>scalerank</th>
<th>name</th>
<th>name_alt</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Lake Baikal</td>
<td><a href="https://e">https://e</a>...</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Lake Winnipeg</td>
<td><a href="https://e">https://e</a>...</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>Great Slave Lake</td>
<td><a href="https://e">https://e</a>...</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
<td>L. Ontario</td>
<td><a href="https://e">https://e</a>...</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>L. Erie</td>
<td><a href="https://e">https://e</a>...</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0</td>
<td>Lake Superior</td>
<td><a href="https://e">https://e</a>...</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>0</td>
<td>Lake Huron</td>
<td><a href="https://e">https://e</a>...</td>
</tr>
</tbody>
</table>
```json
1 {
2   "type": "Coverage",
3   "domain": {
4     "type": "Domain",
5     "domainType": "Grid",
6     "axes": {
7       "x": {
8         "start": -180.075,
9         "stop": 179.9250000000007,
10        "num": 2400
11       },
12       "y": {
13         "start": 90.075,
14         "stop": -90.075,
15         "num": 1201
16       }
17     },
18     "referencing": {
19       "coordinates": {
20         "x",
21         "y"
22       },
23       "system": {
24         "type": "GeographicCRS",
25         "id": "http://www.opengis.net/def/crs/OGC/1.3/"
26       }
27   }
```
Large Lakes
lakes of the world, public domain

Tiles
Tile Matrix Set
WorldCRS84Quad

Metadata
Tiles metadata in tilejson format

Map
Name: Reindeer Lake
Class: Lake
OGC API - PROCESSES

version: "0.1.0"
id: "hello-world"
title: "Hello World process"
description: "Hello World process"

keywords: "hello world"

links:
  0:
    type: "text/html"
    rel: "canonical"
    title: "information"
    href: "https://example.org/process"
    hreflang: "en-US"

inputs:
  0:
    id: "name"
    title: "name"
    input:
      literalDataDomain:
        dataType: "string"
        valueDefinition:
          anyValue: true
          minOccurs: 1
          maxOccurs: 1

outputs:
  0:
    id: "hello-world-response"
    title: "output hello world"
    output:
      formats:
        0:
          mimeType: "application/json"

example:
# Spatiotemporal Asset Catalog (STAC)

## Item CMC_glb_TMP_TGL_2_lation.15x.15_2020040112_P000

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>CMC_glb_TMP_TGL_2_lation.15x....</td>
</tr>
<tr>
<td>GRIB_COMMENT</td>
<td>Temperature [C]</td>
</tr>
<tr>
<td>GRIB_DISCIPLINE</td>
<td>0 (Meteorological)</td>
</tr>
<tr>
<td>GRIB_ELEMENT</td>
<td>TMP</td>
</tr>
<tr>
<td>GRIB_FORECAST_SECONDS</td>
<td>0 sec</td>
</tr>
<tr>
<td>GRIB_IDS</td>
<td>CENTER=54 (Montreal) SUBCENTER=0 MASTER_TABLE=4 LOCAL_TABLE=0</td>
</tr>
</tbody>
</table>

Assets

Leaflet | Wikimedia maps | Map data © OpenStreetMap contributors
DEPLOYING
PACKAGING

Numerous deployment options

- PyPI
- UbuntuGIS
- Docker
- Conda
- FreeBSD
- Coming to OSGeoLive
**DOCKER OVERVIEW**

- pygeoapi Docker Images on DockerHub
- Images create Linux containers running content
- Use Docker Images to test locally or deploy in server
# Painless deployment
# pull from DockerHub

docker pull geopython/pygeoapi:latest

# run (default config)
docker run -p 5000:80 -it geopython/pygeoapi:latest

# test

curl http://localhost:5000

curl http://localhost:5000/collections


DOCKER OPTIONS

- Custom config via Docker Volume Mapping
- Run with Docker Compose
- Subpath running via Flask `SCRIPT_NAME` environment variable
- Examples
Cloud clustering with high availability
PRODUCTION INSTANCES
OGC web services platform of Canadian weather, climate and water data
- Realtime weather
- Numerical weather prediction
- Hydrometric and Climate archives
MSC PYGEOAPI
IMPLEMENTATION

- Climate services data portal
- climatedata.ca
- https://geo.weather.gc.ca/geomet/features
GLOSIS - GLOBAL SOIL INFORMATION

- Global soil information system under development by FAO and ISRIC
- Using pygeoads as an external API providing OGC API
- Features
- Testbed for JSON-LD for soil data distribution
- https://glosis.isric.org/ogc
ROADMAP

• More OGC APIs: Maps, Records
• OGC API Improvements: Processes
• More data providers
• Advanced filters (CQL)
• Transactions
• Content negotiation (e.g. Response as GeoPackage, GML,...)
• OSGeoLive
Need support for WFS3.0???

GeoCat can help you
Just Objects B.V. is the trading name for the professional activities of Just van den Broecke. You can find Just's CV here. Services include consultancy, training, development, hosting and maintenance, all in the domain of Open Geospatial Information Technology. Just Objects is a member of the Dutch OpenGeoGroup. Just is an OSGeo Charter member and chair of the OSGeoNL Foundation, the Dutch Language Chapter of OSGeo.org.

Latest Tweets

justobjects.nl
About GeoComvos

GeoComvos Ltd. is a Cyprus-based technical consultancy specializing in the development, deployment of Free and Open Source Software (FOSS) Geographic Information Systems (GIS) and technology, particularly in the development of Spatial Data Infrastructure (SDI) platforms and Geospatial Catalogues. The company specializes in promoting, developing and contributing Open Source and Open Data solutions to governments, organizations and the private sector.

Although GeoComvos Ltd is a new company, it carries years of experience in the geospatial domain: members of the company have implemented the official open data catalogue for the government of the United States of America as well as the open geospatial data catalogue for the Greek government.

GeoNode Development

GeoNode is a web-based application and platform for developing geospatial information systems (GIS) and for deploying spatial data infrastructures (SDI). We offer core development and customization services for your needs.
Geobeyond Srl

**Geospatial Solutions**
Spatial databases, WebGIS and Dashboard

We use the best breed of spatial open-source software to publish geographic information on the Web and making it searchable, easily sharable and standards compliant, particularly appealing on meaningful maps.

**Location Science**
Spatial Data Infrastructures, Open Data and IoT

Geospatial is a business mission to turn digital assets into an open data driven strategy. We build tailored open-source Spatial Data Infrastructures which facilitates the use of geographical information, Earth Observation and IoT data.

**Digital Identity**
Identity Access Management and Web API security

The implementation of up-to-date and tamper-proof digital identity systems is crucial. Our expertise achieves it by allowing continuous innovation and delivering efficient, secure, digital-based products.

**Digital Transformation**
API design, development and security

Web-based APIs are a key foundation of digital transformation. We help organizations to respond to the need to enable mobile apps and create integrated digital ecosystems which drive their businesses.
LINKS

- Home: https://pygeoapi.io
- Docs: https://docs.pygeoapi.io
- Demo: https://demo.pygeoapi.io
- GitHub: https://github.com/geopython/pygeoapi
- Docker: https://hub.docker.com/r/geopython/pygeoapi
- Chat: https://gitter.im/geopython/pygeoapi
- Mail: https://lists.osgeo.org/mailman/listinfo/pygeoapi
THANK YOU!

@tomkralidis @JMendesDeJesus @normanbarker
@perrygeo @justb4 @pvgenuchten @tzotsos @alpha-beta-soup @franccbartoli