PYGEOAPI

API FOR GEOSPATIAL DATA

@tomkralidis @JMendesDeJesus @normanbarker @perrygeo @justb4 @pvgenuchten

version: June 19, 2019

I. IN THE BEGINNING THERE WAS... WFS V1/V2

CHARACTERISTICS

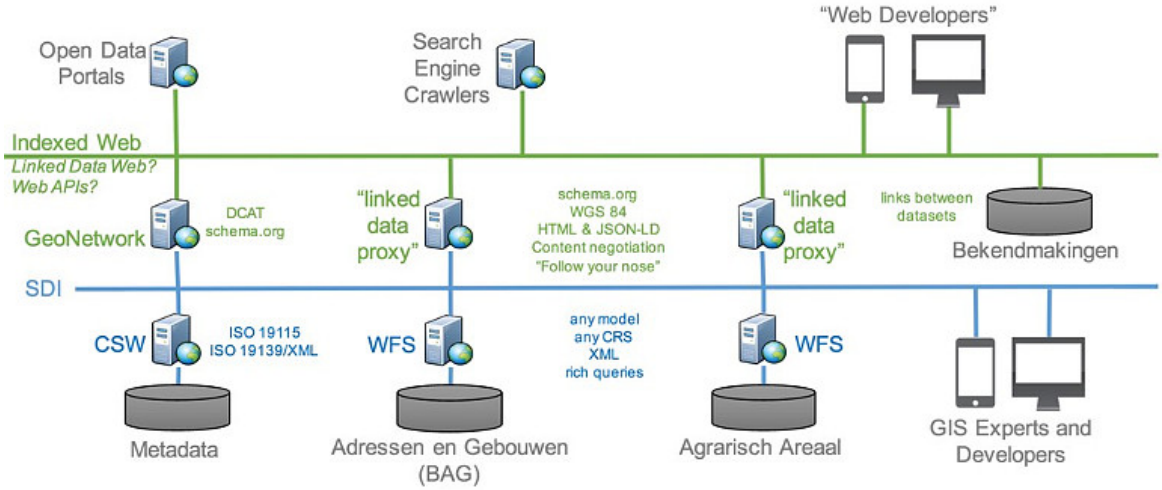
- Key Value Parameters (KVP) or XML encodings
- Requests using HTTP GET, HTTP POST and SOAP
- Responses as XML (GML)
- Error handling ExceptionReport in response

```
<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>[..]
        </gml:featureMember>[..]
</wfs:FeatureCollection>
```

ISSUES WITH WFS V1/2

- Highly verbose
- Highly dependent on XML technologies
- Verbose documentation (WFS 2.0 has 239 pages)
- OGC document from 2010
- No unique urls for features
- Features hard to index by search engine
- "In-band" error reporting

MORE ISSUES



- ldproxy for services to be crawlable
- OGC world is NOT W3C-/Web- friendly

II. A NEW ERA IS UPON US

RESTFUL

- REST is becoming a MUST
- Fast standards' development
- Opinion:
  Standards (-development) should be more developer-friendly

WHAT IS REST ???

- REpresentational State Transfer (REST)
- Uses HTTP's verbs (GET/POST/DELETE)
- Uses HTTP's codes (200, 404, 201, etc)
- Uses URI to identify resources
- Content negotiation (ask for media-type)
- Stateless

- Very popular among developers
- Easy to understand
- JSON is king in REST webservices
- No rigid standards (good or bad)

OGC AND W3C

https://www.w3.org/TR/sdw-bp/

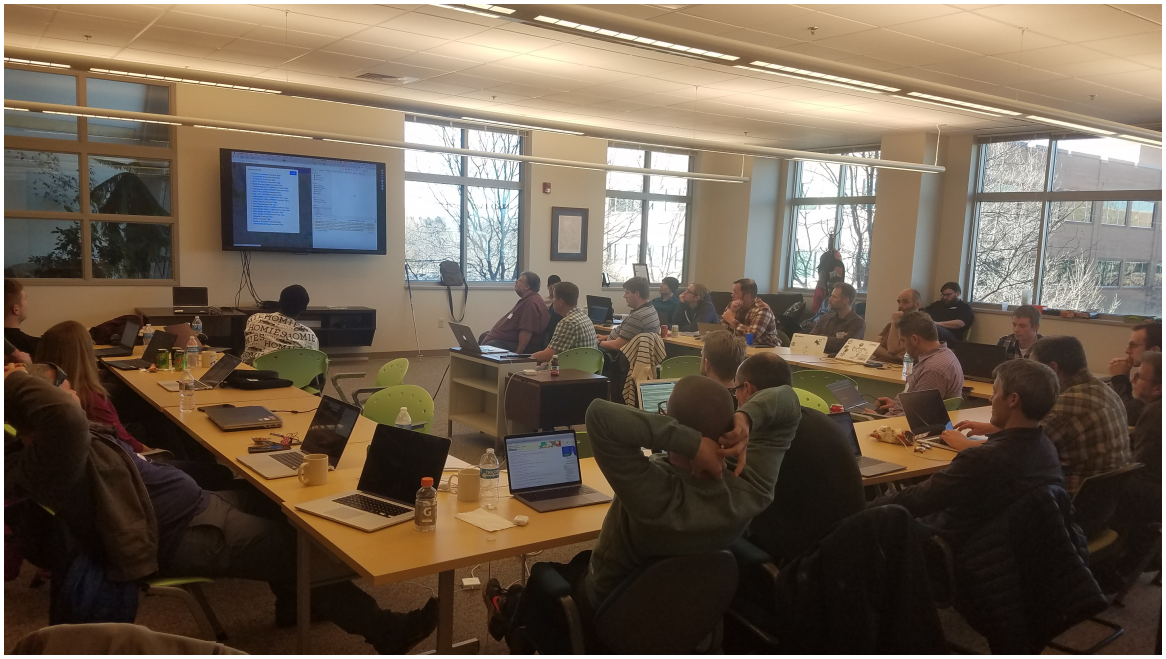
MUST READ DOCUMENT

III. KICKSTARTING WFS 3.0

WFS 3.0 HACKATHON

- March 2018 - **Fort Collins, CO, USA**
- OGC-sponsored
- Develop WFS 3.0 based on Swagger/OpenAPI (v3)
- Write specification using GitHub
- First implementations
- **pygeoapi was born on Valentine 2018!**


USGS
science for a changing world
Core Science Systems
John Wesley Powell Center
for Analysis and Synthesis



Matthew Douglas Wiggett aka Jivan Amara

1975 - 2018

(Go-lang client/server for WFS3.0)

IV. OPENAPI / SWAGGER

OPENAPI / SWAGGER

- Specification on how to describe a REST web service
- Originally was referred as Swagger
- Language agnostic


http://swagger.io

- OpenAPI is based on YAML Syntax
- Describes REST endpoints
- HTTP Verbs to use
- Format of Responses

Example yaml

```
openapi: 3.0.1
info:
title: OGC Web Feature Service standard API
version: 0.0.1
description: 'This is a sample OpenAPI definition for OGC'
contact:
name: La oreja de van Gogh
```

```
paths:
      /:
       get:
        summary: landing page of this API
        description: 'Description of dataset'
        operationId: getLandingPage
        tags:
        - Capabilities
responses:
      '200':
        description: links to the API capabilities
        content:
         - application/json:
```

V. THE WFS 3.0 SPEC

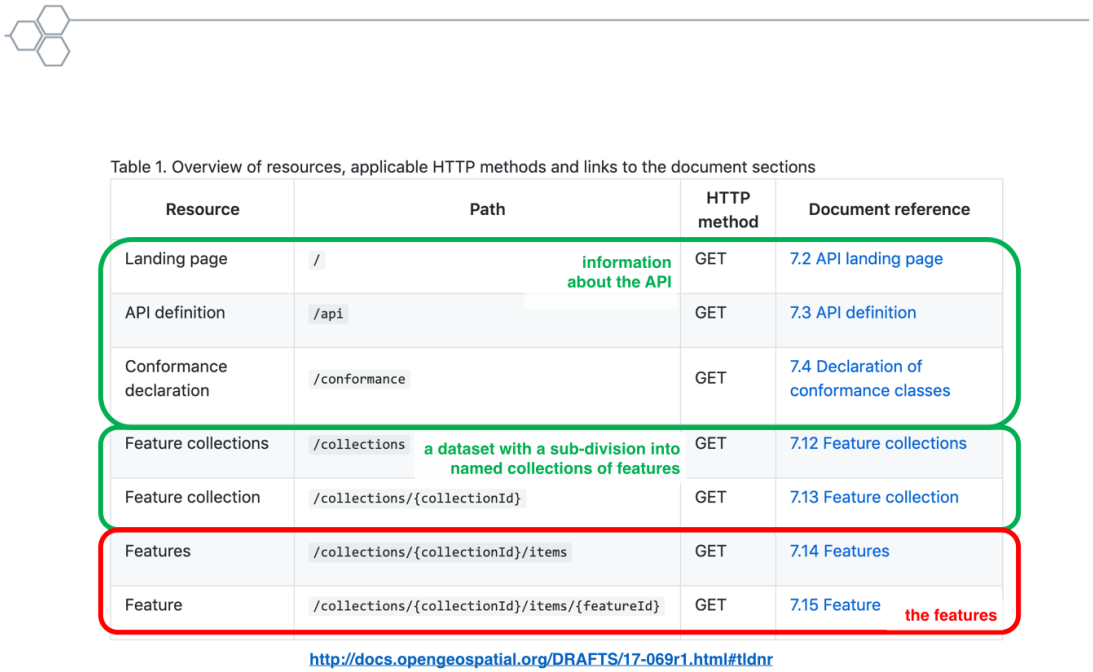
NB: renamed as: **"OGC API - Features"**

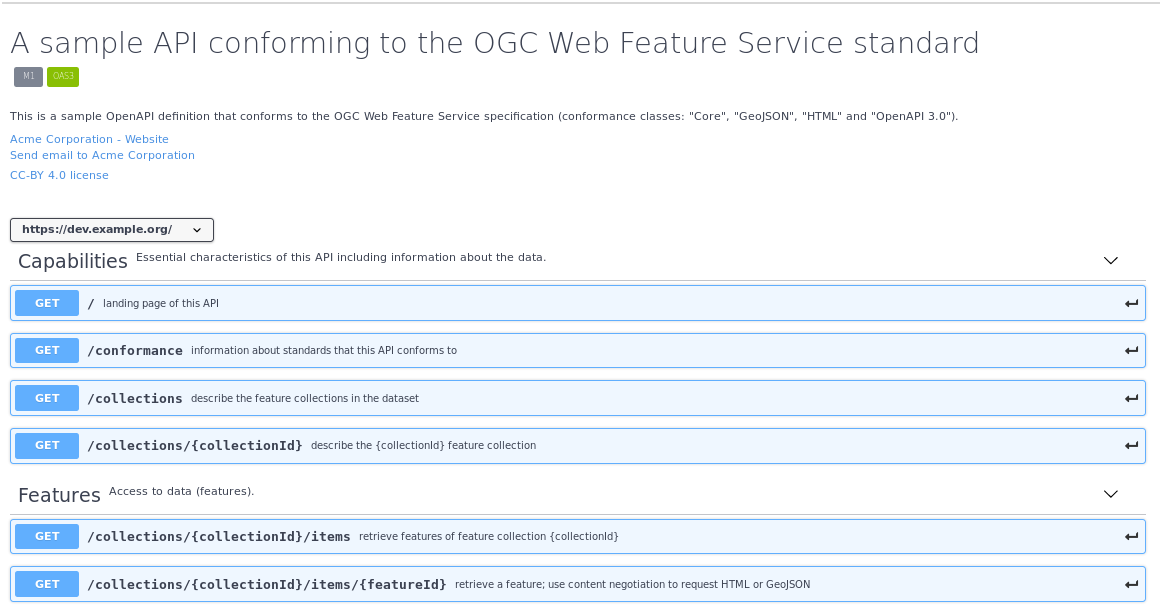
SPEC

github.com/opengeospatial/WFS_FES

Maintained in GitHub

By C. Portele - **link**





Check it in SwaggerHub

EXAMPLE

```
/collections

/collections/countries

/collections/countries/items

/collections/countries/items/The_Netherlands
```

Data is structured and each Feature is a REST representation

OGC API - FEATURES

LINKS

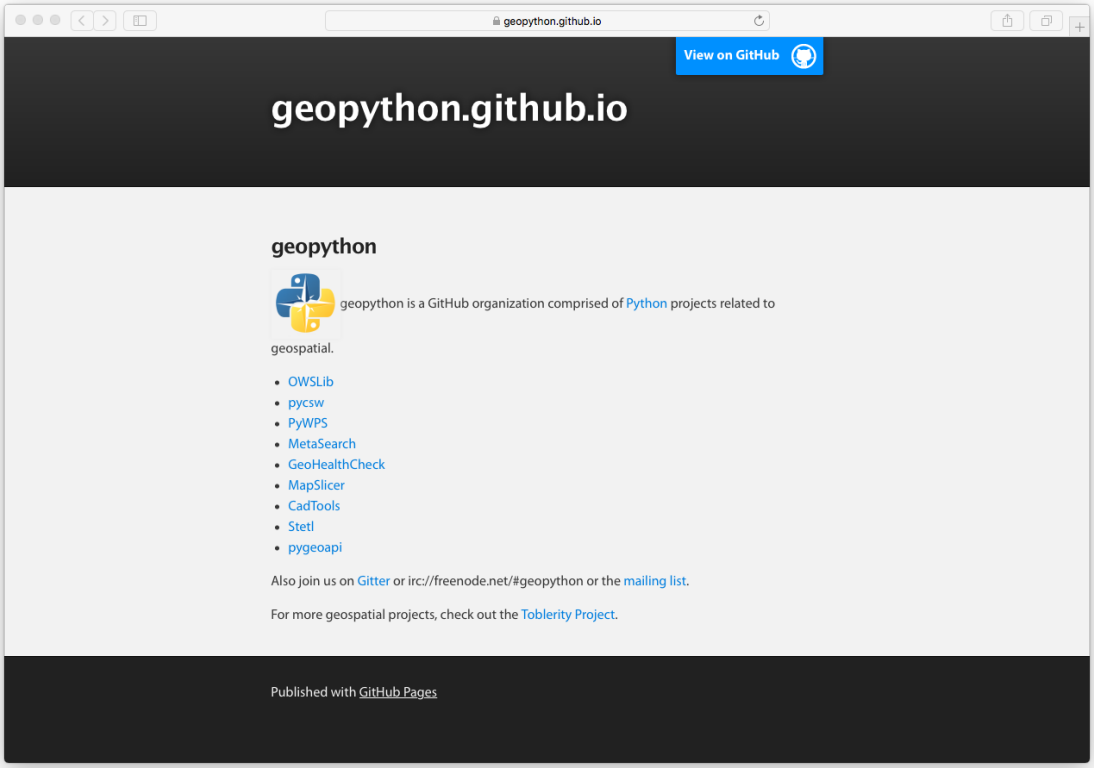
- GitHub Repository
- OpenAPI Schemas
- Draft 17-069.html
- Gitter Chat
- OGC-API-Hackathon-2019

VI. PYGEOAPI

PYGEOAPI

Part of the amazing Geopython community in GitHub

geopython.github.io

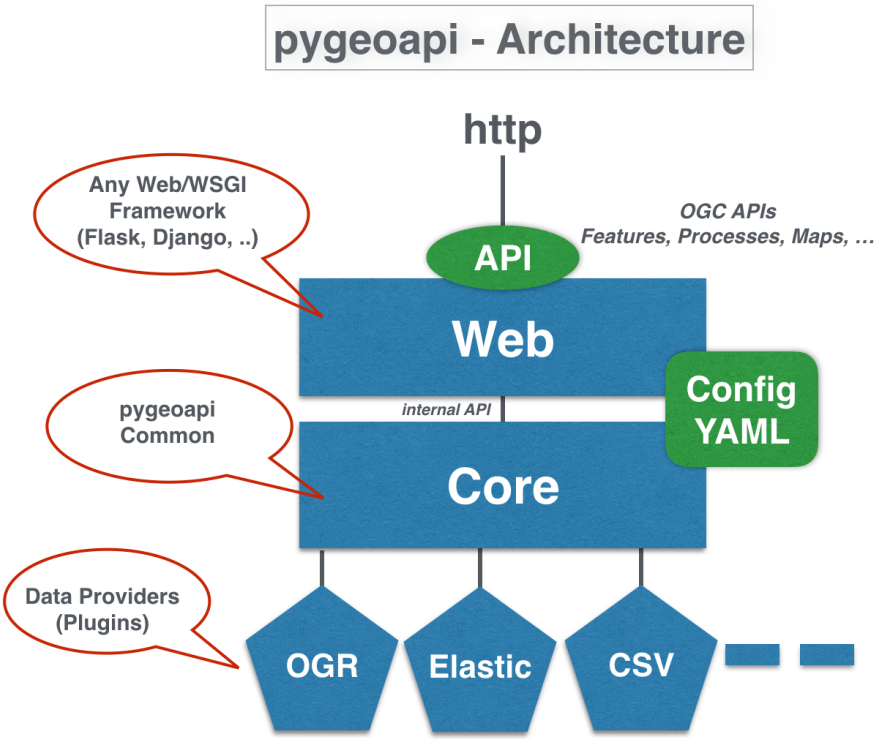

geopython.github.io
View on GitHub
geopython.github.io
geopython
geopython is a GitHub organization comprised of Python projects related to geospatial.
OWSLib
pycsw
PyWPS
MetaSearch
GeoHealthCheck
MapSlicer
CadTools
Stetl
pygeoapi
Also join us on Gitter or irc://freenode.net/#geopython or the mailing list.
For more geospatial projects, check out the Toblerity Project.
Published with GitHub Pages

PYGEOAPI

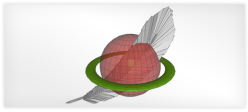
- More than just a WFS v3 implementation
- Access geospatial data via OGC APIs
- OSGeo Community Project (in Motion)
- Powered (default) by:

PYGEOAPI - TECHNICAL

- Flask with REST support
- OpenAPI endpoint automatically generated
- Data provider agnostic (plugins)
- Docker Images


pygeoapi - Architecture
http
Any Web/WSGI Framework (Flask, Django, ..)
API
OGC APIs
Features, Processes, Maps, ...
Web
Config YAML
internal API
pygeoapi Common
Core
Data Providers (Plugins)
OGR
Elastic
CSV

DATA PROVIDERS



Elasticsearch	CSV	Spatialite*	GeoJSON*

* also via GDAL-OGR Provider

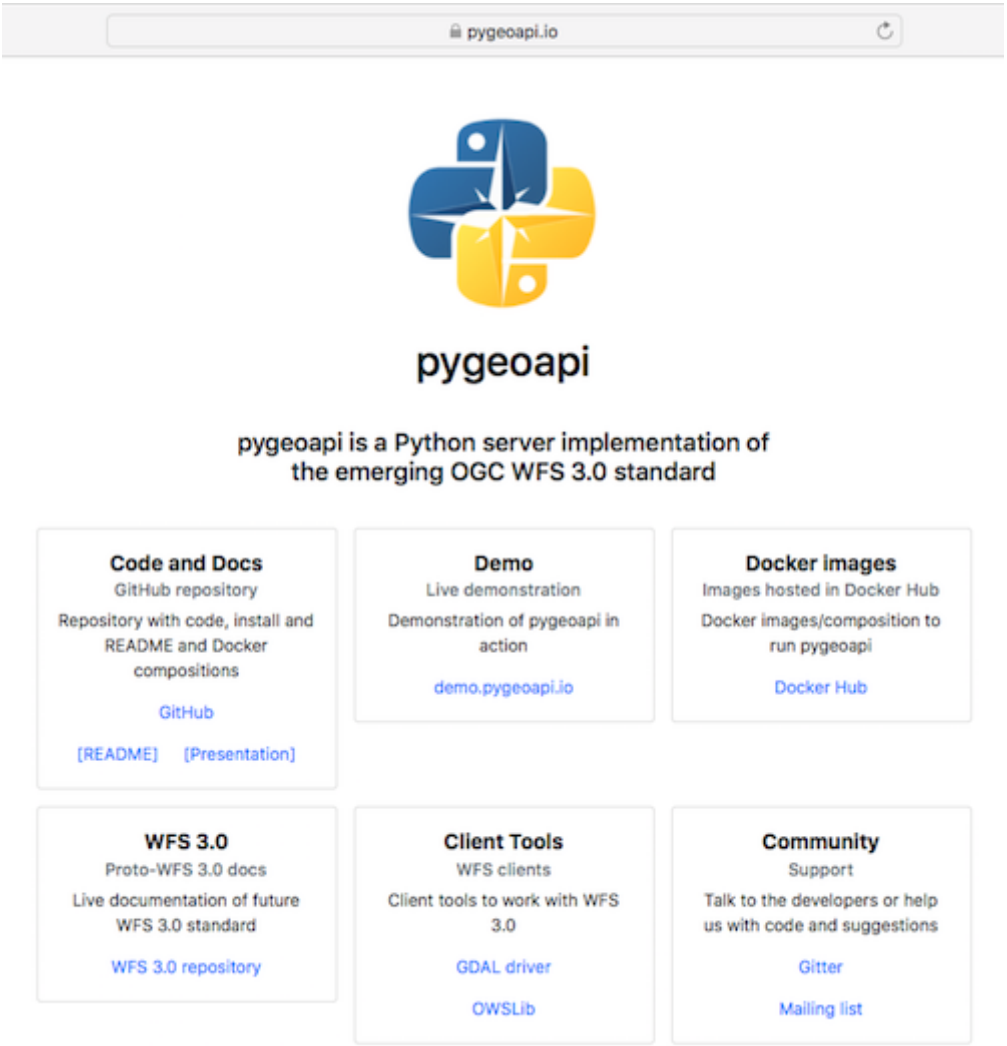
DATA PROVIDERS - CONT

GDAL-OGR

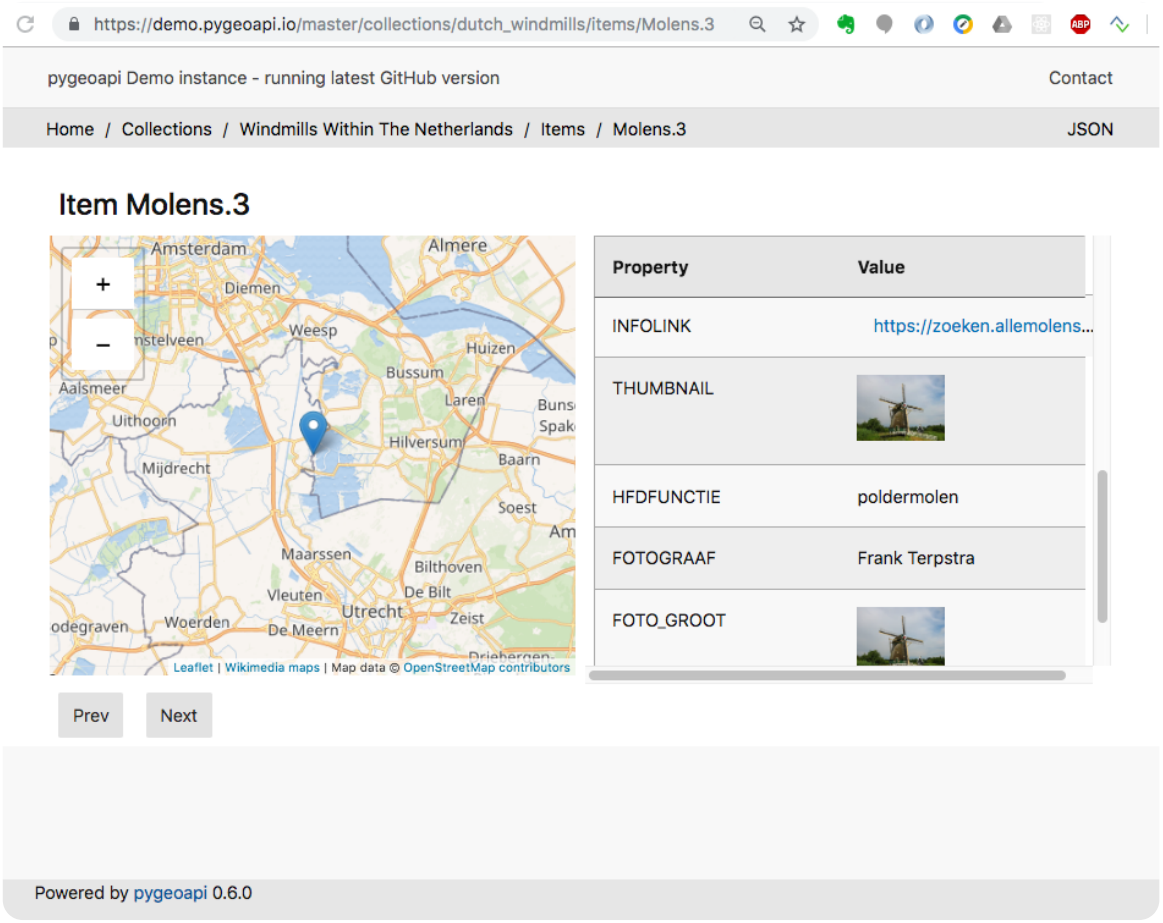
DATA PROVIDERS - GDAL-OGR

- via OGR-Python bindings
- in theory **~100 Vector Formats!**
- thus: WFS2, GeoPackage, Shapefile, GeoJSON, PostGIS, ...
- paging, reprojection, bbox/attr/id-query, auth
- ==> unlock existing WFS2s!

https://pygeoapi.io



DEMO


https://demo.pygeoapi.io/master/collections/dutch_windmills/items/Molens.3
pygeoapi Demo instance - running latest GitHub version
Contact
Home / Collections / Windmills Within The Netherlands / Items / Molens.3
JSON
Item Molens.3
Amsterdam
Almere
Diemen
Weesp
Huizen
Bussum
Laren
Aalsmeer
Uithoorn
Hilversum
Baarn
Mijdrecht
Soest
Maarssen
Bilthoven
De Bilt
Vleuten
Utrecht
Zeist
Woerden
De Meern
Leaflet | Wikimedia maps | Map data © OpenStreetMap contributors
Property
Value
INFOLINK
https://zoeken.allemolens...
THUMBNAIL
HFDFUNCTIE
poldermolen
FOTOGRAAF
Frank Terpstra
FOTO_GROOT
Prev
Next
Powered by pygeoapi 0.6.0

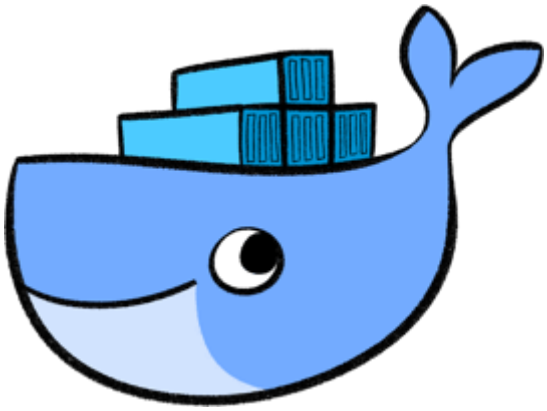
PYGEOAPI - ROADMAP

- More data providers
- Content negotiation (e.g. Response as GeoPackage, GML,...)
- Advanced filters (CQL)
- More OGC APIs: Maps, Tiles, Coverages, Processes

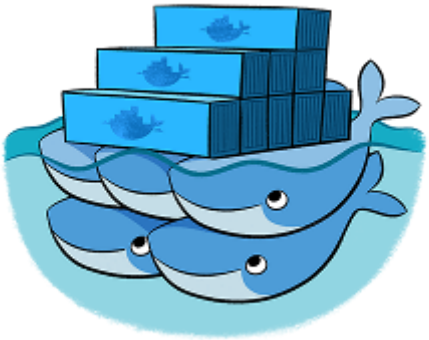
PYGEOAPI - LINKS

- Home: **https://pygeoapi.io**
- Demo: **https://demo.pygeoapi.io**
- GitHub: **https://github.com/geopython/pyge**
- Docker: **https://cloud.docker.com/u/geopython/rep**
- Chat: **https://gitter.im/geopython/pygeoapi**
- Mail: **https://lists.osgeo.org/mailman/listinf**

VII. DOCKER AND PYGEOAPI



- pygeoapi Docker Images on DockerHub: **Link**
- Images create linux containers running content
- Use Docker Images to test locally or deploy in server
- See READMEs



Docker Images for geopython/pygeoapi

UP AND RUNNING IN 2 MINUTES

Pull from DockerHub

```
docker pull geopython/pygeoapi:latest
```

Run (default config)

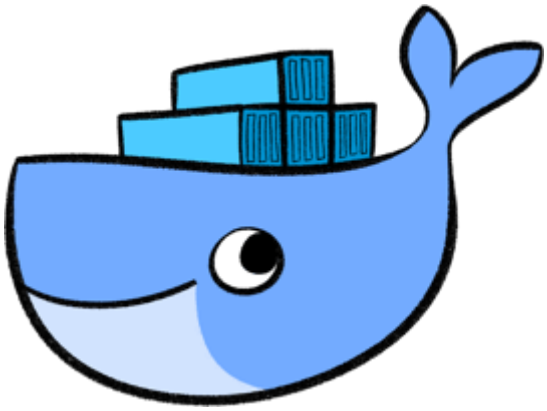
```
docker run -p 5000:80 -it geopython/pygeoapi:latest
```

Test:

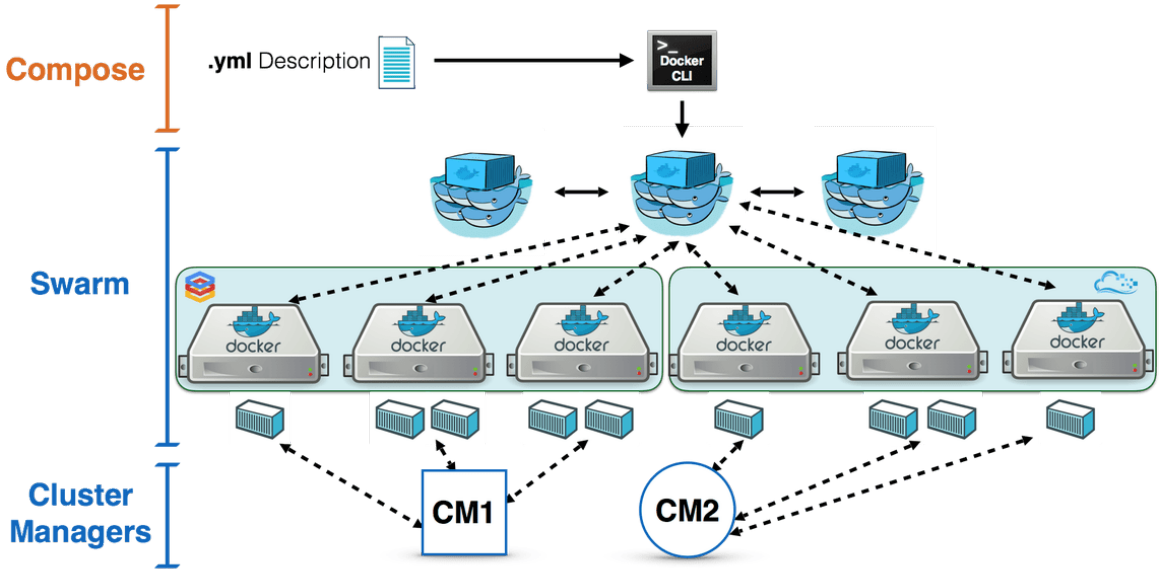
```
http://localhost:5000
http://localhost:5000/collections
http://localhost:5000/collections/lakes/items?limit=2
http://localhost:5000/collections/lakes/items?f=html
```

DOCKER - NEXT

- Custom config via Docker Volume Mapping
- Run with Docker Compose
- Subpath running via Flask "SCRIPT_NAME" env var
- Examples in **pygeoapi GitHub**



- Dockerhub image without Elasticsearch
- Docker-compose for Elasticsearch in github
- Easy to scale pygeoapi with docker swarm



- Cloud clustering with high availability

VIII. PYGEOAPI IN PRODUCTION

(Some) Production instances of pygeoapi

Contact: **@tomkralidis**

http://bit.do/hydat

Canadian National Water Data Archive

numberMatched: 60 770 985

```
http://geo.weather.gc.ca/geomet-beta/features/collections/
hydrometric-daily-mean/items?resulttype=hits
```

100 Features

```
http://geo.weather.gc.ca/geomet-beta/features/collections/
hydrometric-daily-mean/items?limit=100
```

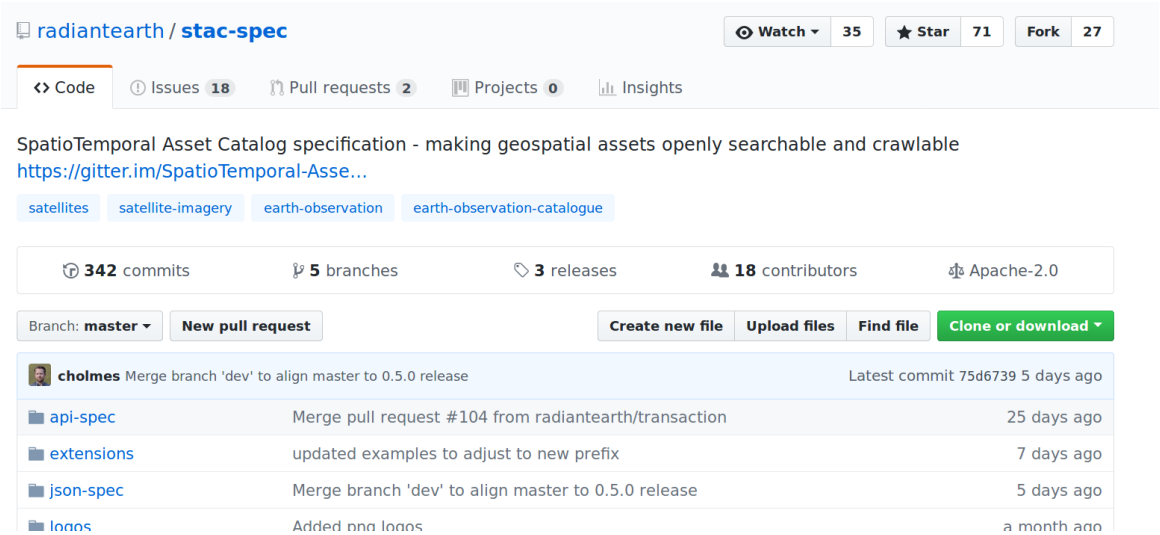
Fetching 1 feature (200ms)

```
http://geo.weather.gc.ca/geomet-beta/features/collections/
hydrometric-daily-mean/items/10SB001.1992-01-11
```

IX. STAC

SPATIOTEMPORAL ASSET CATALOG (STAC)

- Simple catalog of spatiotemporal items (JSON)
- (stac != csw || csw => stac)
- More oriented for sat imagery
- Parallel work to WFS 3.0
- Pull from storages like S3
- Geospatial assets openly searchable and crawlable



https://github.com/radiantearth/stac-spec

LANDSAT8 - STAC EXAMPLE

```
{
  "id": "LC81530252014153LGN00",
  "type": "Feature",
  "bbox": [ 49.16354,72.27502,51.36812,75.67662
        ],
  "geometry": {
    "type": "MultiPolygon",
"coordinates": [..]
```

LANDSAT8 - STAC EXAMPLE


```
"properties": {
    "datetime": "2014-06-02T09:22:02Z",
    "provider": "USGS",
    "eo:gsd": 30.0,
    "eo:cloud_cover" : 10,
    "eo:off_nadir" : 0.000,
    "eo:azimuth": 0.00,
    "eo:sun_azimuth" : 149.01607154,
    "eo:sun_elevation" : 59.21424700,
         }
```

LANDSAT8 - STAC EXAMPLE

```
  "assets" :{
    "thumbnail": {
          "href": "http://landsat[..]/LC81530252014153LGN00_thumb_large.jpg",
          "required": true,
          "type": "jpeg"
    },
    "metadata": {
      "href": "http://landsat[..]LC81530252014153LGN00_MTL.txt",
      "required": true,
      "type": "mtl"
},
          }
```

X. PYGEOAPI SUPPORT

Need support for WFS3.0???

@Geocat can help you

GEOCAT
GOVERNMENT GEOGRAPHIC DATA PUBLISHING

justobjects.nl