27-29 May 2024

imdis

International conference on Marine Data and Information Systems

SeaDataNet · Havforskningsinstituttet · Ifremer · MARIS · MEDIN · National Oceanography Centre · INGV · iMagine · IODE · eosc Blue-Cloud2026

52north exploring horizons

OGC API - Connected Systems

IMDIS - Bergen (Norway) - 27-29 May 2024

52°North - in a nutshell

exploring horizons

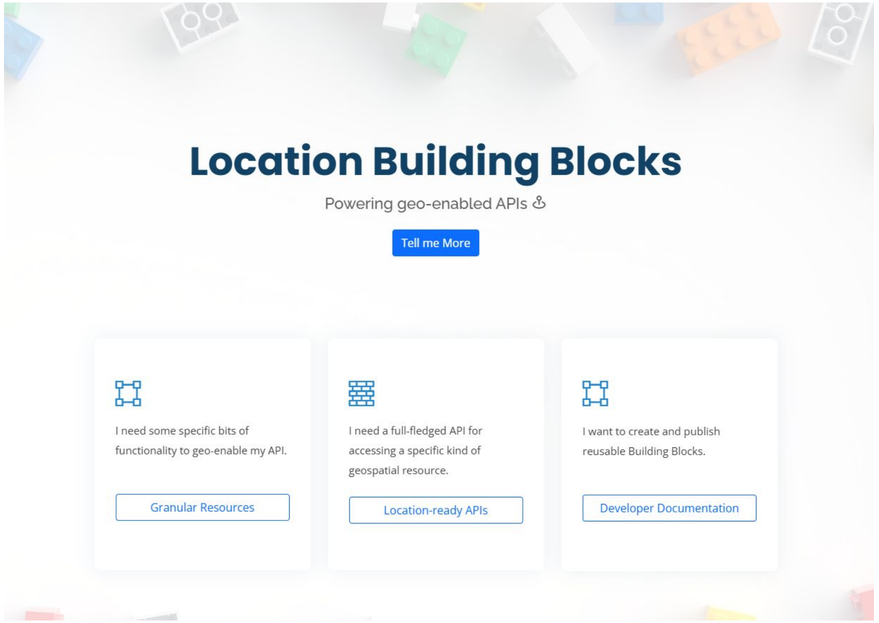
- Research and innovation company in the field of geoinformatics (non-profit)
- 25+ Employees ...
  - Research Software Engineers
  - Data Scientists
  - Administration
- Main activities are **applied research** and **knowledge transfer**
- Results of joint R&D innovation activities are published as **open source software**
- 52°North has revenues from national and international **R&D projects** and **professional services** (software/solution development, consulting, training)
- founded as a limited liability company in September 2006, re-formed in May 2021
- Shareholders: Universität Münster, Twente Universiteit, con terra GmbH, Esri Inc

OGC API - Connected Systems

- Attempt to specify a true successor to
  - Sensor Observation Service (SOS)
  - Sensor Planning Service (SPS)
- Based on and in alignment with the OGC API family of standards
- API for management of systems (sensors, platforms, actuators, etc) their observations and datastreams, commands and control streams
- 52°North is involved in the standards working group and is currently developing an implementation

OGC APIs

- Resource-oriented HTTP APIs
- “REST”
- JSON & JSON Schema
- OpenAPI specifications
- Schemas, data types, parameters, APIs are considered building blocks
- https://blocks.ogc.org/
- → reusability
- → composition



Location Building Blocks

Powering geo-enabled APIs

Tell me More

I need some specific bits of functionality to geo-enable my API.

Granular Resources

I need a full-fledged API for accessing a specific kind of geospatial resource.

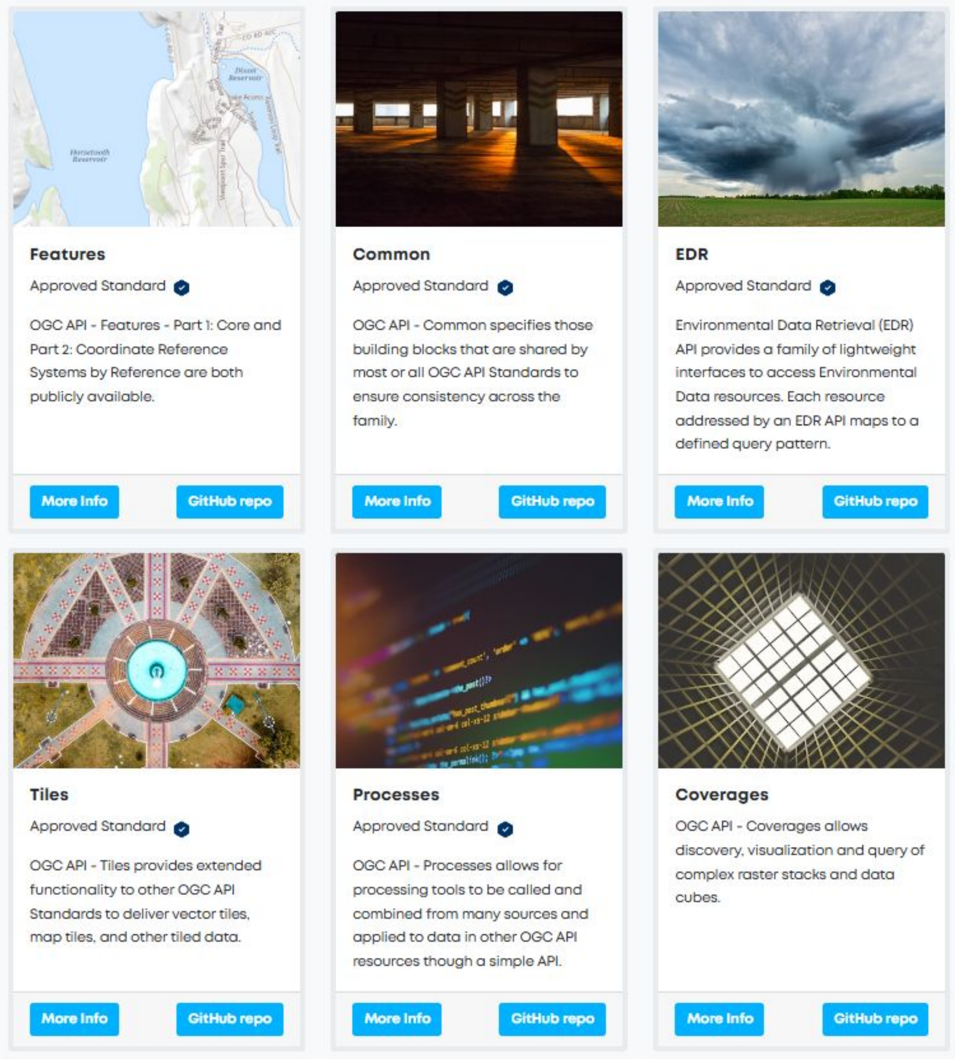
Location-ready APIs

I want to create and publish reusable Building Blocks.

Developer Documentation

OGC APIs

- Web Feature Service (WFS)
  → OGC API - Features
- Catalogue Service for the Web (CSW)
  → OGC API - Records
- Web Processing Service (WPS)
  → OGC API - Processes
- Web Coverage Service (WCS)
  → OGC API - Coverages
- Web Map Service (WMS)
  → OGC API - Maps
- Web Map Tile Service (WMTS)
  → OGC API - Tiles
- …



Features

Approved Standard

OGC API - Features - Part 1: Core and Part 2: Coordinate Reference Systems by Reference are both publicly available.

More Info GitHub repo

Common

Approved Standard

OGC API - Common specifies those building blocks that are shared by most or all OGC API Standards to ensure consistency across the family.

More Info GitHub repo

EDR

Approved Standard

Environmental Data Retrieval (EDR) API provides a family of lightweight interfaces to access Environmental Data resources. Each resource addressed by an EDR API maps to a defined query pattern.

More Info GitHub repo

Tiles

Approved Standard

OGC API - Tiles provides extended functionality to other OGC API Standards to deliver vector tiles, map tiles, and other tiled data.

More Info GitHub repo

Processes

Approved Standard

OGC API - Processes allows for processing tools to be called and combined from many sources and applied to data in other OGC API resources though a simple API.

More Info GitHub repo

Coverages

OGC API - Coverages allows discovery, visualization and query of complex raster stacks and data cubes.

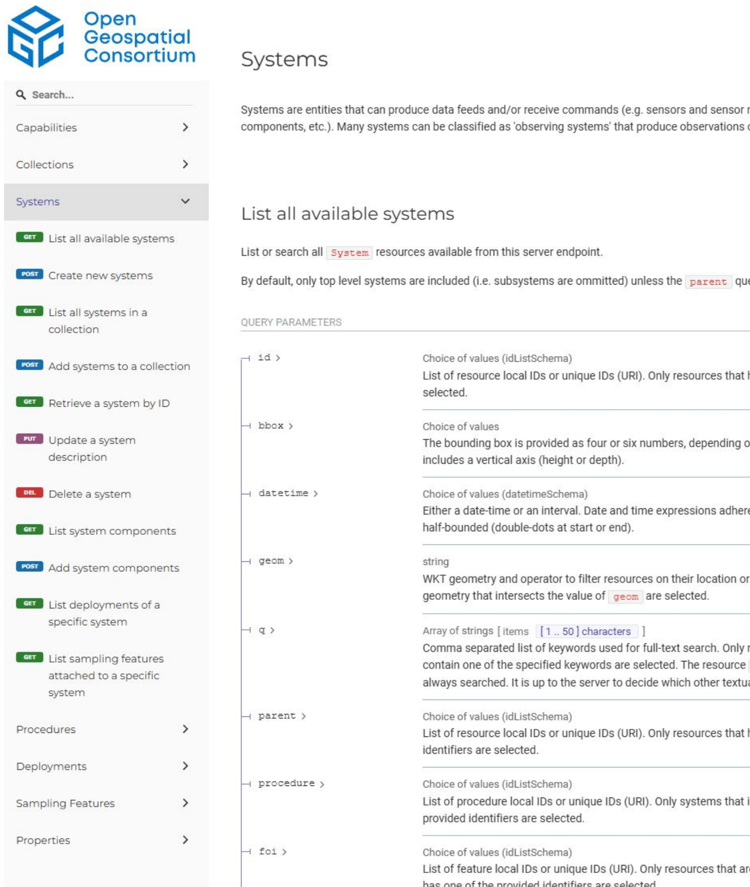
More Info GitHub repo

OGC API - Connected Systems

- Why not OGC SensorThings API?
  - It’s based on “OData - the best way to REST”
  - Very hard to put on top of existing systems
  - No support for detailed sensor system descriptions
- Preparatory work: SWE Common 3.0 and SensorML 3.0
  - JSON encoding and deprecation of XML encoding
  - Deployments of Systems
- *Currently* a five part specification of the API

Part 1 - Feature Resources

- Current Draft Preview,
- OpenAPI Docs
- Based on the OGC API - Features
- Systems
  - metadata of sensors
  - actuators
  - platforms
  - simulations
- Procedures
  - metadata of procedures implemented by system
  - automated system specs/datasheets


Open Geospatial Consortium
Search...
Capabilities
Collections
Systems
GET List all available systems
POST Create new systems
GET List all systems in a collection
POST Add systems to a collection
GET Retrieve a system by ID
PUT Update a system description
DEL Delete a system
GET List system components
POST Add system components
GET List deployments of a specific system
GET List sampling features attached to a specific system
Procedures
Deployments
Sampling Features
Properties

Systems

Systems are entities that can produce data feeds and/or receive commands (e.g. sensors and sensor r
components, etc.). Many systems can be classified as 'observing systems' that produce observations o

List all available systems

List or search all `System` resources available from this server endpoint.

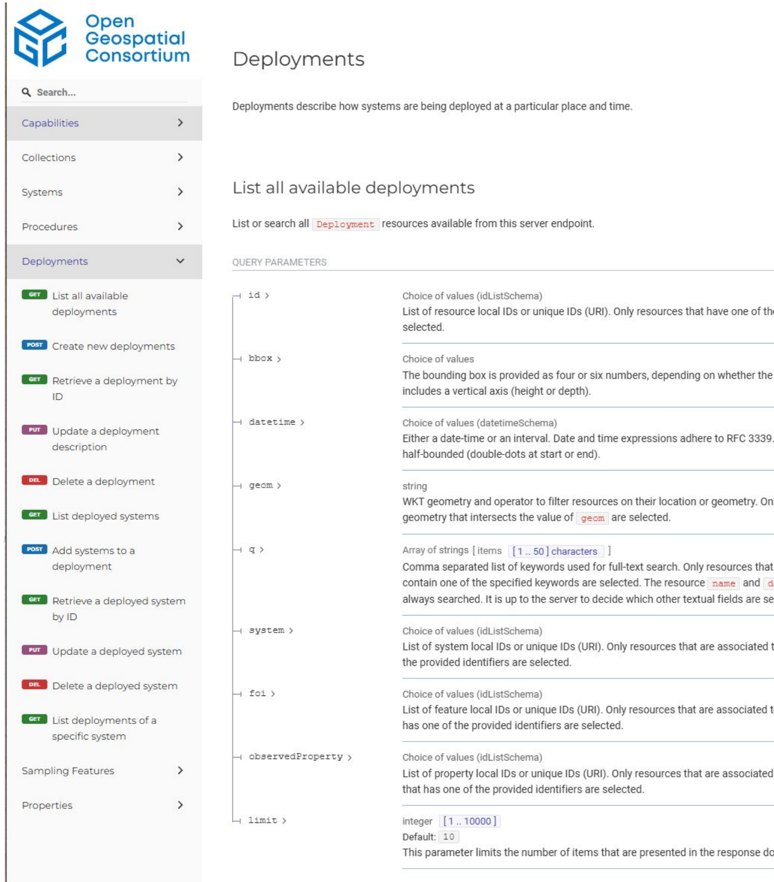
By default, only top level systems are included (i.e. subsystems are ommitted) unless the `parent` que

QUERY PARAMETERS

`id`	Choice of values (idListSchema) List of resource local IDs or unique IDs (URI). Only resources that h selected.
`bbox`	Choice of values The bounding box is provided as four or six numbers, depending o includes a vertical axis (height or depth).
`datetime`	Choice of values (datetimeSchema) Either a date-time or an interval. Date and time expressions adhere half-bounded (double-dots at start or end).
`geom`	string WKT geometry and operator to filter resources on their location or geometry that intersects the value of `geom` are selected.
`q`	Array of strings [items [1 .. 50] characters] Comma separated list of keywords used for full-text search. Only r contain one of the specified keywords are selected. The resource always searched. It is up to the server to decide which other textua
`parent`	Choice of values (idListSchema) List of resource local IDs or unique IDs (URI). Only resources that h identifiers are selected.
`procedure`	Choice of values (idListSchema) List of procedure local IDs or unique IDs (URI). Only systems that i provided identifiers are selected.
`foi`	Choice of values (idListSchema) List of feature local IDs or unique IDs (URI). Only resources that ar has one of the provided identifiers are selected.

Part 1 - Feature Resources

- Deployments:
  - metadata of system deployments
- Sampling Features
  - metadata about sampling geometries/methodologies used by observing systems
- Subsystems / Components
- Property Definitions
  - integrated option to derive specialised properties


Open Geospatial Consortium

Search...

Capabilities
Collections
Systems
Procedures
Deployments
GET List all available deployments
POST Create new deployments
GET Retrieve a deployment by ID
PUT Update a deployment description
DEL Delete a deployment
GET List deployed systems
POST Add systems to a deployment
GET Retrieve a deployed system by ID
PUT Update a deployed system
DEL Delete a deployed system
GET List deployments of a specific system
Sampling Features
Properties

Deployments

Deployments describe how systems are being deployed at a particular place and time.

List all available deployments

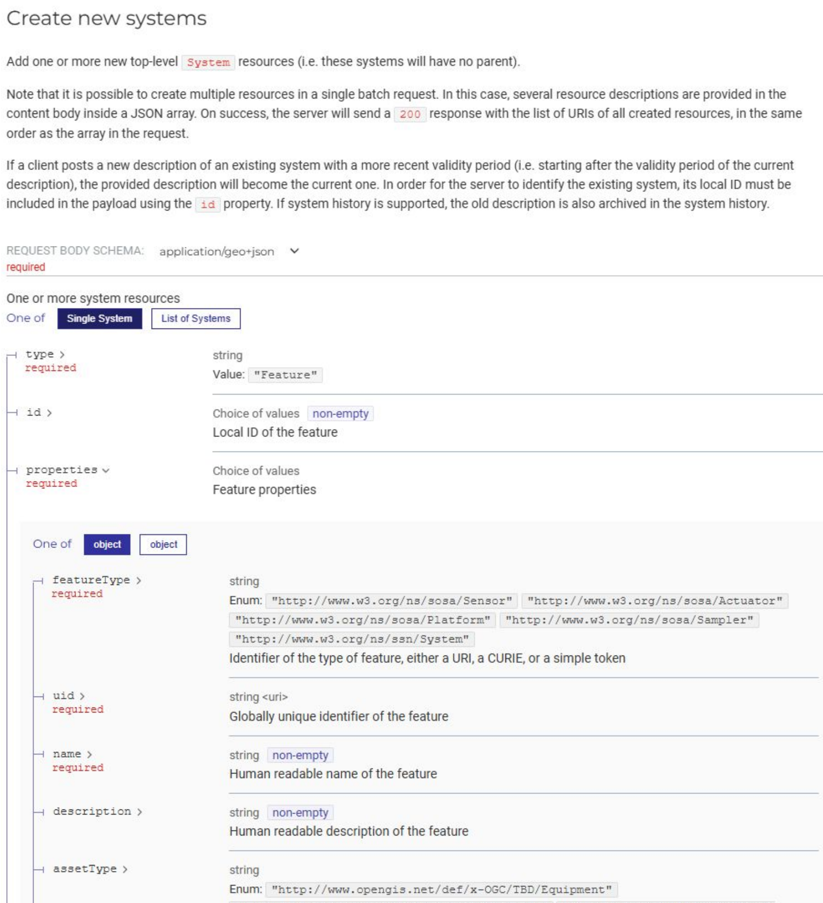
List or search all `Deployment` resources available from this server endpoint.

QUERY PARAMETERS

`id`	Choice of values (idListSchema) List of resource local IDs or unique IDs (URI). Only resources that have one of th selected.
`bbox`	Choice of values The bounding box is provided as four or six numbers, depending on whether the includes a vertical axis (height or depth).
`datetime`	Choice of values (datetimeSchema) Either a date-time or an interval. Date and time expressions adhere to RFC 3339. half-bounded (double-dots at start or end).
`geom`	string WKT geometry and operator to filter resources on their location or geometry. On geometry that intersects the value of `geom` are selected.
`q`	Array of strings [items [1 .. 50] characters] Comma separated list of keywords used for full-text search. Only resources that contain one of the specified keywords are selected. The resource `name` and `d` always searched. It is up to the server to decide which other textual fields are se
`system`	Choice of values (idListSchema) List of system local IDs or unique IDs (URI). Only resources that are associated t the provided identifiers are selected.
`foi`	Choice of values (idListSchema) List of feature local IDs or unique IDs (URI). Only resources that are associated t has one of the provided identifiers are selected.
`observedProperty`	Choice of values (idListSchema) List of property local IDs or unique IDs (URI). Only resources that are associated that has one of the provided identifiers are selected.
`limit`	integer [1 .. 10000] Default: `10` This parameter limits the number of items that are presented in the response do

Part 1 - Feature Resources

- Currently two JSON encodings
  - Complete metadata → SensorML 3.0
  - More lightweight → GeoJSON



Create new systems

Add one or more new top-level `System` resources (i.e. these systems will have no parent).

Note that it is possible to create multiple resources in a single batch request. In this case, several resource descriptions are provided in the content body inside a JSON array. On success, the server will send a `200` response with the list of URIs of all created resources, in the same order as the array in the request.

If a client posts a new description of an existing system with a more recent validity period (i.e. starting after the validity period of the current description), the provided description will become the current one. In order for the server to identify the existing system, its local ID must be included in the payload using the `id` property. If system history is supported, the old description is also archived in the system history.

REQUEST BODY SCHEMA: application/geo+json
required

One or more system resources

One of: Single System | List of Systems

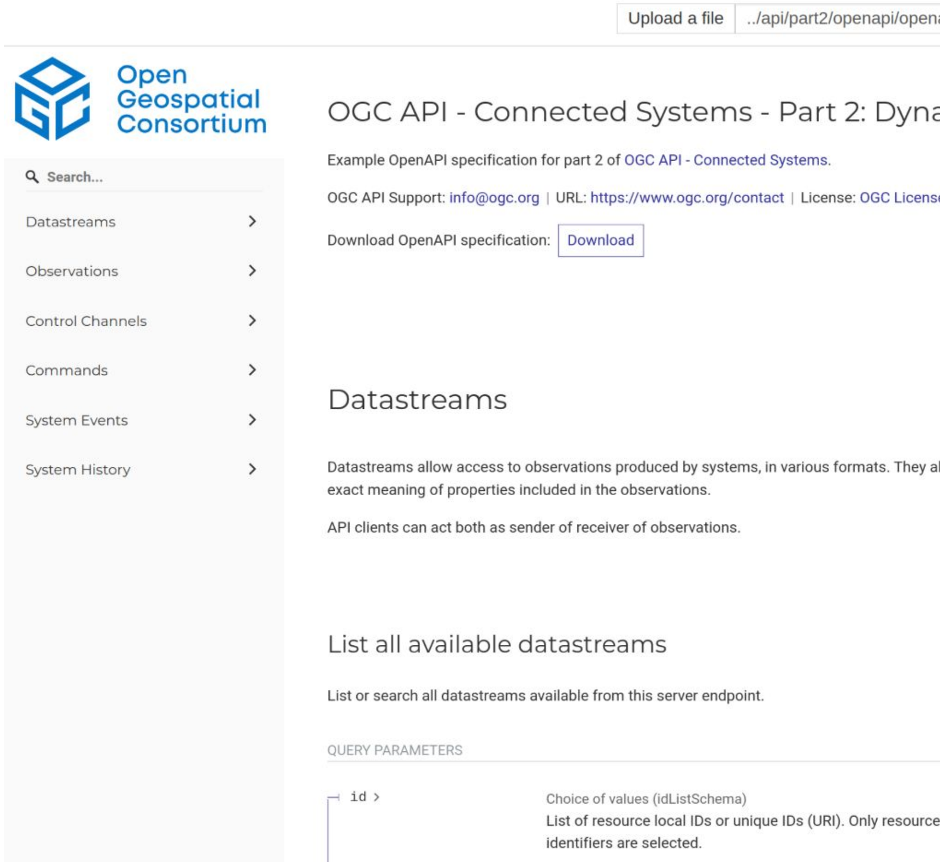
`type` required	string Value: `"Feature"`
`id`	Choice of values non-empty Local ID of the feature
`properties` required	Choice of values Feature properties

One of: object | object

`featureType` required	string Enum: `"http://www.w3.org/ns/sosa/Sensor"` `"http://www.w3.org/ns/sosa/Actuator"` `"http://www.w3.org/ns/sosa/Platform"` `"http://www.w3.org/ns/sosa/Sampler"` `"http://www.w3.org/ns/ssn/System"` Identifier of the type of feature, either a URI, a CURIE, or a simple token
`uid` required	string <uri> Globally unique identifier of the feature
`name` required	string non-empty Human readable name of the feature
`description`	string non-empty Human readable description of the feature
`assetType`	string Enum: `"http://www.opengis.net/def/x-OGC/TBD/Equipment"`

Part 2 - Dynamic Data

- Current Draft Preview
- OpenAPI Docs
- Dynamic Feature Properties
  - Data Streams
  - Observations
  - Control Streams
  - Commands and Status
  - System Events
  - System History



Upload a file ../api/part2/openapi/opena

Open Geospatial Consortium

Search...

Datastreams
Observations
Control Channels
Commands
System Events
System History

OGC API - Connected Systems - Part 2: Dyna

Example OpenAPI specification for part 2 of OGC API - Connected Systems.

OGC API Support: info@ogc.org | URL: https://www.ogc.org/contact | License: OGC Licens

Download OpenAPI specification: Download

Datastreams

Datastreams allow access to observations produced by systems, in various formats. They al
exact meaning of properties included in the observations.

API clients can act both as sender of receiver of observations.

List all available datastreams

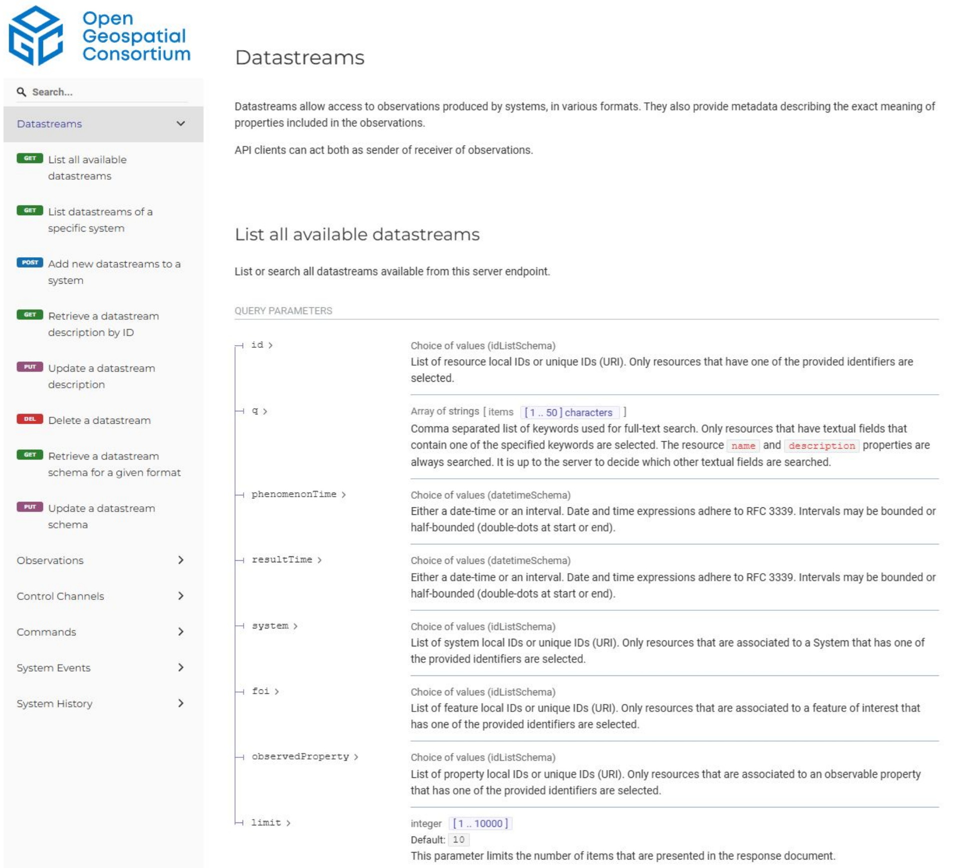
List or search all datastreams available from this server endpoint.

QUERY PARAMETERS

id ›	Choice of values (idListSchema) List of resource local IDs or unique IDs (URI). Only resource identifiers are selected.

Part 2 - Dynamic Data

- Data Streams
  - access to observations produced by systems
  - aggregation of observation metadata
  - holds the schema of the actual observations
- Observations
  - Free in the choice of formats
  - Linking to external service, e.g. OGC API - Coverages, ERDDAP, etc


Open Geospatial Consortium
Search...
Datastreams
GET List all available datastreams
GET List datastreams of a specific system
POST Add new datastreams to a system
GET Retrieve a datastream description by ID
PUT Update a datastream description
DEL Delete a datastream
GET Retrieve a datastream schema for a given format
PUT Update a datastream schema
Observations
Control Channels
Commands
System Events
System History

Datastreams

Datastreams allow access to observations produced by systems, in various formats. They also provide metadata describing the exact meaning of properties included in the observations.

API clients can act both as sender of receiver of observations.

List all available datastreams

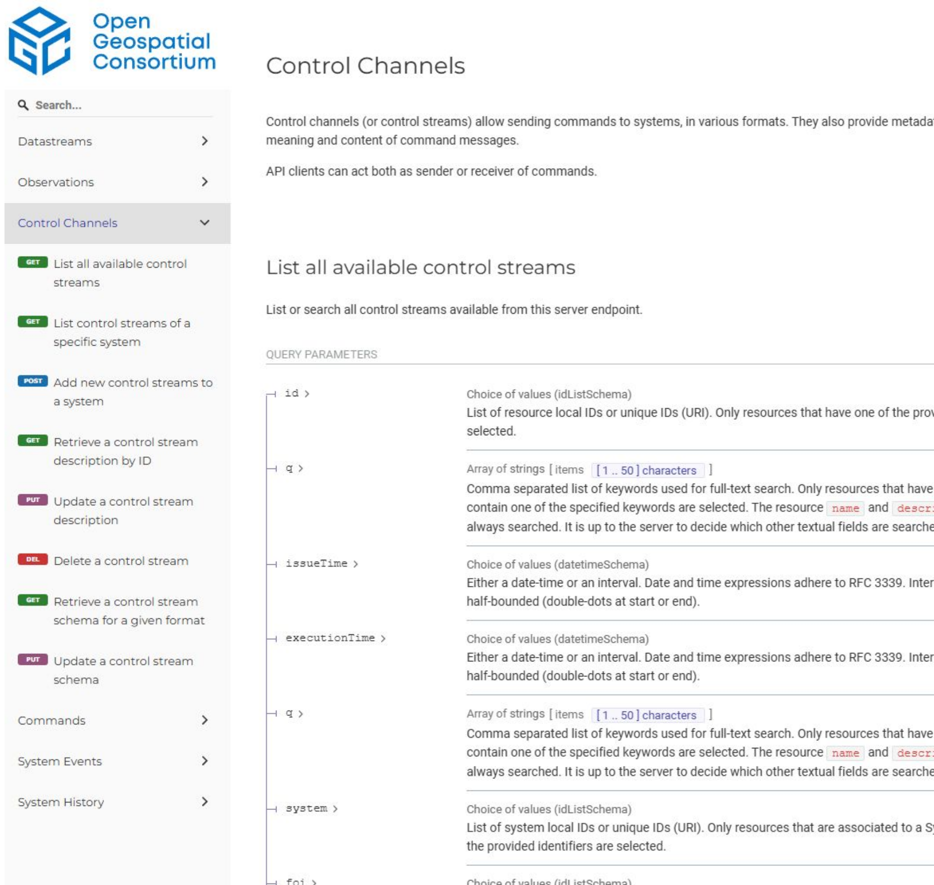
List or search all datastreams available from this server endpoint.

QUERY PARAMETERS

id ›	Choice of values (idListSchema) List of resource local IDs or unique IDs (URI). Only resources that have one of the provided identifiers are selected.
q ›	Array of strings [items [1 .. 50] characters] Comma separated list of keywords used for full-text search. Only resources that have textual fields that contain one of the specified keywords are selected. The resource `name` and `description` properties are always searched. It is up to the server to decide which other textual fields are searched.
phenomenonTime ›	Choice of values (datetimeSchema) Either a date-time or an interval. Date and time expressions adhere to RFC 3339. Intervals may be bounded or half-bounded (double-dots at start or end).
resultTime ›	Choice of values (datetimeSchema) Either a date-time or an interval. Date and time expressions adhere to RFC 3339. Intervals may be bounded or half-bounded (double-dots at start or end).
system ›	Choice of values (idListSchema) List of system local IDs or unique IDs (URI). Only resources that are associated to a System that has one of the provided identifiers are selected.
foi ›	Choice of values (idListSchema) List of feature local IDs or unique IDs (URI). Only resources that are associated to a feature of interest that has one of the provided identifiers are selected.
observedProperty ›	Choice of values (idListSchema) List of property local IDs or unique IDs (URI). Only resources that are associated to an observable property that has one of the provided identifiers are selected.
limit ›	integer [1 .. 10000] Default: 10 This parameter limits the number of items that are presented in the response document.

Part 2 - Dynamic Data

- Control Streams/Channels
  - allows sending commands to systems
  - describes the exact meaning of commands
  - holds the schema of the actual commands
- Commands
  - Various formats: SWE Common, Protobuf, …
- Status of command executions



Open Geospatial Consortium
Search...
Datastreams
Observations
Control Channels
GET List all available control streams
GET List control streams of a specific system
POST Add new control streams to a system
GET Retrieve a control stream description by ID
PUT Update a control stream description
DEL Delete a control stream
GET Retrieve a control stream schema for a given format
PUT Update a control stream schema
Commands
System Events
System History

Control Channels

Control channels (or control streams) allow sending commands to systems, in various formats. They also provide metada meaning and content of command messages.

API clients can act both as sender or receiver of commands.

List all available control streams

List or search all control streams available from this server endpoint.

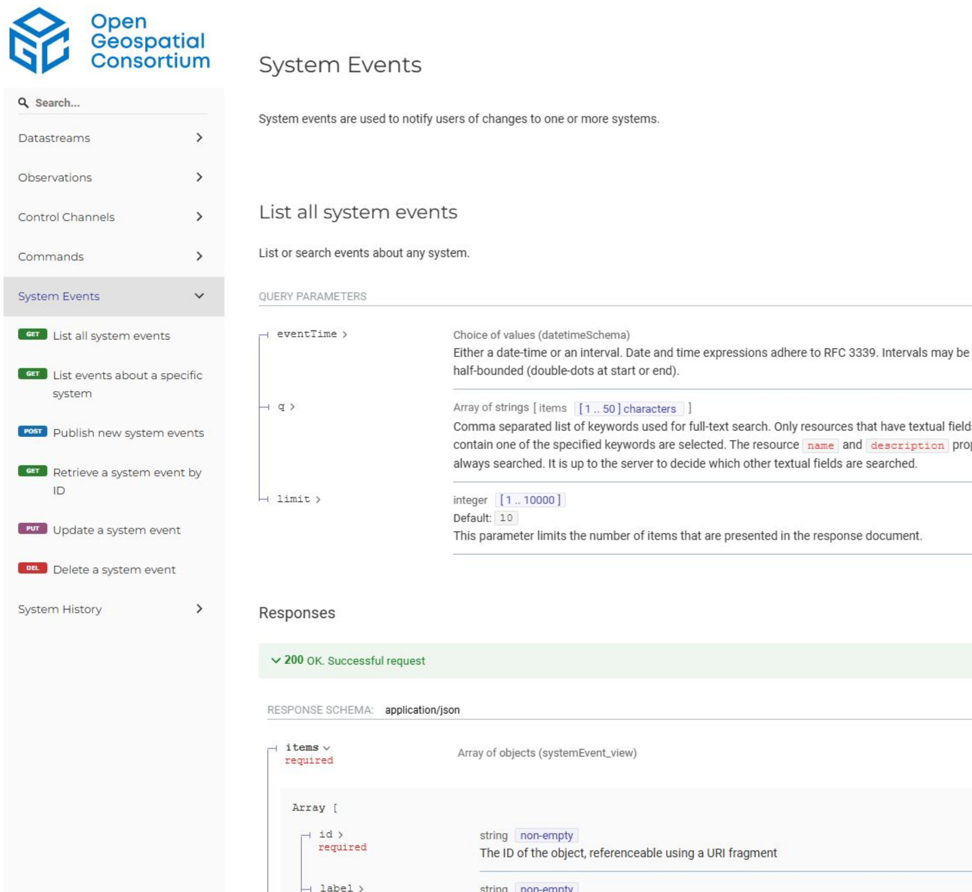
QUERY PARAMETERS

`id` ›	Choice of values (idListSchema) List of resource local IDs or unique IDs (URI). Only resources that have one of the prov selected.
`q` ›	Array of strings [items [1 .. 50] characters] Comma separated list of keywords used for full-text search. Only resources that have contain one of the specified keywords are selected. The resource `name` and `descr` always searched. It is up to the server to decide which other textual fields are searche
`issueTime` ›	Choice of values (datetimeSchema) Either a date-time or an interval. Date and time expressions adhere to RFC 3339. Inter half-bounded (double-dots at start or end).
`executionTime` ›	Choice of values (datetimeSchema) Either a date-time or an interval. Date and time expressions adhere to RFC 3339. Inter half-bounded (double-dots at start or end).
`q` ›	Array of strings [items [1 .. 50] characters] Comma separated list of keywords used for full-text search. Only resources that have contain one of the specified keywords are selected. The resource `name` and `descr` always searched. It is up to the server to decide which other textual fields are searche
`system` ›	Choice of values (idListSchema) List of system local IDs or unique IDs (URI). Only resources that are associated to a S the provided identifiers are selected.
`foi` ›	Choice of values (idListSchema)

Part 2 - Dynamic Data

Connected Systems

- System Events
  - No longer part of the system description
  - Pagination, Search
- System History
  - Historical archive of system descriptions



Open Geospatial Consortium

Search...

- Datastreams
- Observations
- Control Channels
- Commands
- System Events
  - GET List all system events
  - GET List events about a specific system
  - POST Publish new system events
  - GET Retrieve a system event by ID
  - PUT Update a system event
  - DEL Delete a system event
- System History

System Events

System events are used to notify users of changes to one or more systems.

List all system events

List or search events about any system.

QUERY PARAMETERS

eventTime ›	Choice of values (datetimeSchema) Either a date-time or an interval. Date and time expressions adhere to RFC 3339. Intervals may be half-bounded (double-dots at start or end).
q ›	Array of strings [items [1 .. 50] characters] Comma separated list of keywords used for full-text search. Only resources that have textual field contain one of the specified keywords are selected. The resource `name` and `description` prop always searched. It is up to the server to decide which other textual fields are searched.
limit ›	integer [1 .. 10000] Default: 10 This parameter limits the number of items that are presented in the response document.

Responses

200 OK. Successful request

RESPONSE SCHEMA: application/json

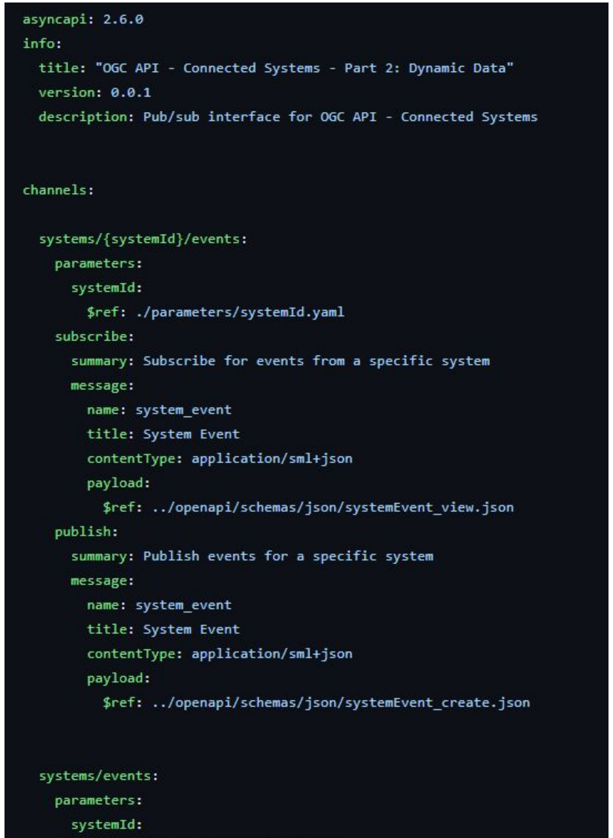
items ⌄ required	Array of objects (systemEvent_view)

Array [

id › required	string non-empty The ID of the object, referenceable using a URI fragment
label ›	string non-empty

Part 3 - Pub/Sub

- AsyncAPI Docs
- Topic structure based on HTTP paths
- Events
- Commands
- Observations
- MQTT
- AMQP
- …



```
asyncapi: 2.6.0
info:
  title: "OGC API - Connected Systems - Part 2: Dynamic Data"
  version: 0.0.1
  description: Pub/sub interface for OGC API - Connected Systems


channels:

  systems/{systemId}/events:
    parameters:
      systemId:
        $ref: ./parameters/systemId.yaml
    subscribe:
      summary: Subscribe for events from a specific system
      message:
        name: system_event
        title: System Event
        contentType: application/sml+json
        payload:
          $ref: ../openapi/schemas/json/systemEvent_view.json
    publish:
      summary: Publish events for a specific system
      message:
        name: system_event
        title: System Event
        contentType: application/sml+json
        payload:
          $ref: ../openapi/schemas/json/systemEvent_create.json


  systems/events:
    parameters:
      systemId:
```

Part 4 - Sampling Feature Types

- Integration of OMS Sample Types
  - SpatialSample
  - StatisticalSample
  - MaterialSample
- Feature Parts
- Parametric Sampling Features
  - Relative Sampling Point
  - Sampling Sphere (or Ellipsoid)
  - Sampling Profile
  - Viewing Frustum
  - Viewing Sector

Part 5 - Binary Encoding Formats

- For features:
  - FlatGeobuf
- For observations and commands:
  - FlatBuffers
  - Protobuf
  - …
- Common Video Formats
- …

The OGC API - Connected Systems

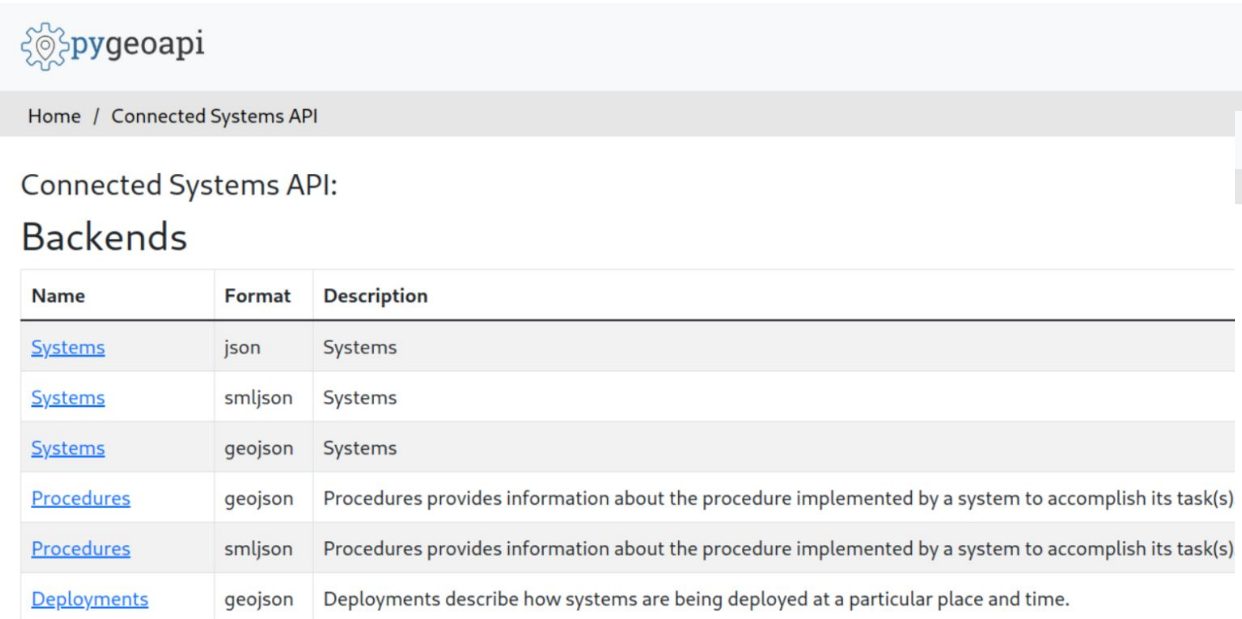
- It was planned to vote on the Part 1 and 2 on the next OGC TC Meeting in June
- Delayed due to a dependency on OGC API Common
- Hopefully we will get a chance in autumn
- The next parts are planned to follow shortly after that

pygeoapi Implementation

- https://pygeoapi.io/
- Python based server implementation of the OGC API suite of standards
    - Features
    - Coverages
    - Maps
    - Tiles
    - Processes
    - Records
    - Environmental Data Retrieval
    - SpatioTemporal Asset Catalog

pygeoapi Implementation

- The past year we've worked on our own implementation of the OGC API - Connected Systems based on pygeoapi
- We're hoping to contribute it sooner or later to upstream



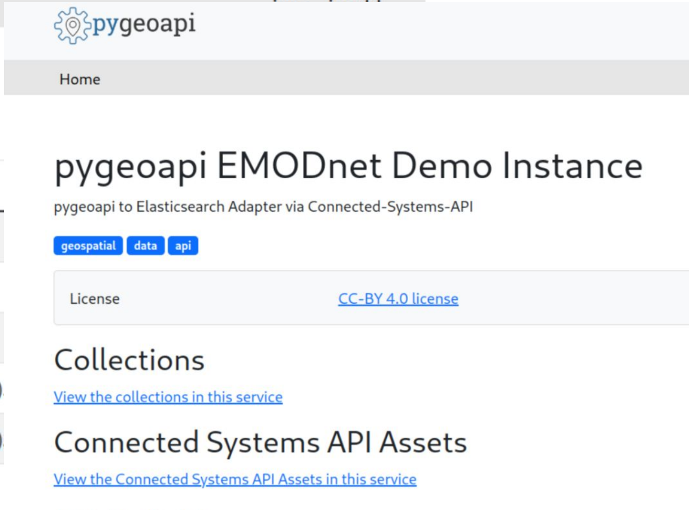
pygeoapi Contact

Home / Connected Systems API

Connected Systems API:

Backends

Name	Format	Description
Systems	json	Systems
Systems	smljson	Systems
Systems	geojson	Systems
Procedures	geojson	Procedures provides information about the procedure implemented by a system to accomplish its task(s)
Procedures	smljson	Procedures provides information about the procedure implemented by a system to accomplish its task(s)
Deployments	geojson	Deployments describe how systems are being deployed at a particular place and time.



pygeoapi

Home

pygeoapi EMODnet Demo Instance

pygeoapi to Elasticsearch Adapter via Connected-Systems-API

geospatial data api

License CC-BY 4.0 license

Collections

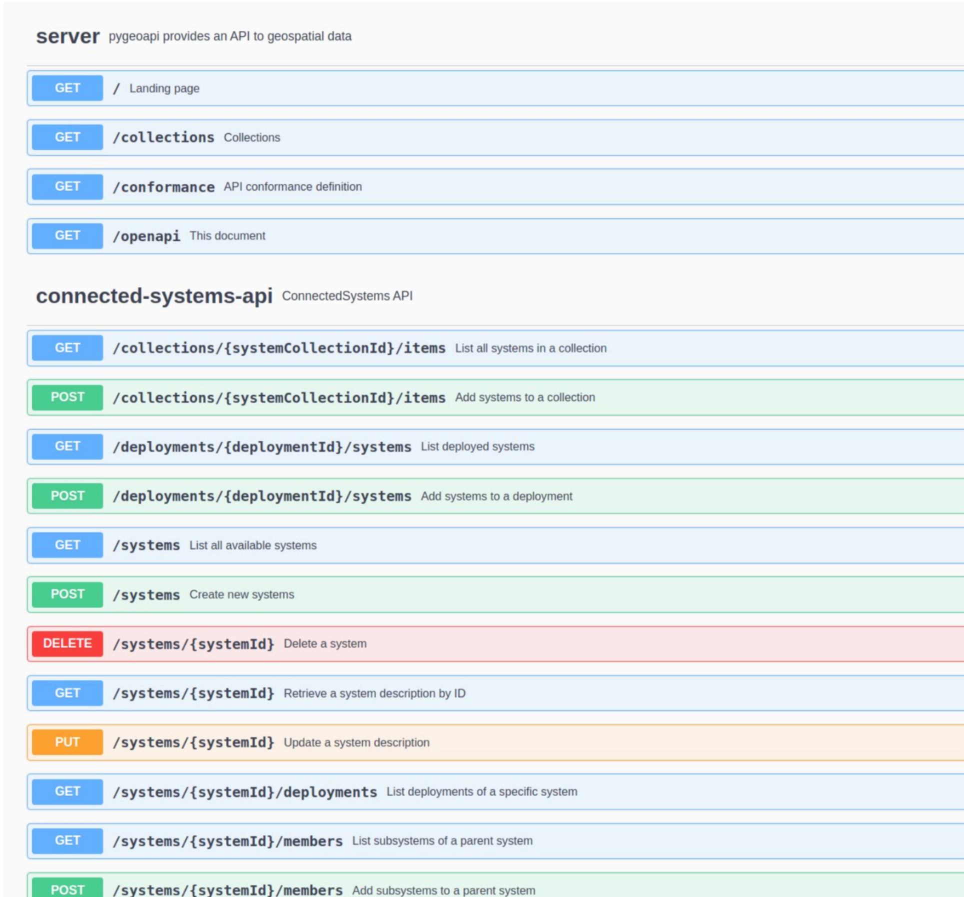
View the collections in this service

Connected Systems API Assets

View the Connected Systems API Assets in this service

pygeoapi Implementation

- Part 1 is done for the most part
  - Generic transactional backend based on ElasticSearch
  - Easy to implement custom backends on top of external databases, services, etc.
- Part 2 is currently under development
  - Idea is to separate metadata and data backends
  - currently evaluating timeseries database candidates, e.g. PostgreSQL/Timescale



server pygeoapi provides an API to geospatial data

GET / Landing page

GET /collections Collections

GET /conformance API conformance definition

GET /openapi This document

connected-systems-api ConnectedSystems API

GET /collections/{systemCollectionId}/items List all systems in a collection

POST /collections/{systemCollectionId}/items Add systems to a collection

GET /deployments/{deploymentId}/systems List deployed systems

POST /deployments/{deploymentId}/systems Add systems to a deployment

GET /systems List all available systems

POST /systems Create new systems

DELETE /systems/{systemId} Delete a system

GET /systems/{systemId} Retrieve a system description by ID

PUT /systems/{systemId} Update a system description

GET /systems/{systemId}/deployments List deployments of a specific system

GET /systems/{systemId}/members List subsystems of a parent system

POST /systems/{systemId}/members Add subsystems to a parent system

Future work

- Current software solutions do not yet support new APIs
- Integrations into existing clients
  → 52°North’s Helgoland
- Integration into spatial/research data infrastructures

Thank you for your attention

exploring horizons

Christian Autermann

Contact: c.autermann@52north.org

27-29 May 2024

imdis

International conference on Marine Data and Information Systems

SeaDataNet Havforskningsinstituttet Ifremer MARIS MEDIN National Oceanography Centre INGV iMagine IODE eosc Blue-Cloud2026