OGC Testbed-16

Data Access and Processing API Engineering Report

Publication Date: 2021-01-13

Approval Date: 2020-12-14

Submission Date: 2020-11-20

Reference number of this document: 20-025r1

Reference URL for this document: http://www.opengis.net/doc/PER/t16-D026

Category: OGC Public Engineering Report

Editor: Luis Bermudez

Title: OGC Testbed-16: Data Access and Processing API Engineering Report

OGC Public Engineering Report

COPYRIGHT

WARNING

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Public Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Table of Contents

Chapter 1. Subject

This OGC Testbed 16 Engineering Report documents the advancement of an OGC Data Access and Processing API (DAPA).

Chapter 2. Executive Summary

Many data access mechanisms are built using generic web services that require complicated queries. Some web services are designed based on the data sources and not on a user centric view. This approach requires the user to download various data files and perform local processing. Testbed 16 explored the development of an Application Programming Interface (API), based on an end-user centric perspective, to improve data retrieval and processing.

The development and use of a user centric API has significant business value. By using a client the user can save valuable time if the client is making function calls instead of accessing multiple generic web services or local files. For example, function calls to calculate the minimum temperature value for an area of interest would look similar even if implemented on different products regardless of the data location, be it a local file, an in-memory structure (e.g. an xarray), or a remote data set stored in the cloud. This user-centric view reduces the number of lines of code while enhancing the end user experience with more powerful options.

This Testbed 16 work addressed the following research questions:

- What does a resource model look like that binds specific functions to specific data?
- How to realize an end user optimized Data Access and Processing API (DAPA) that looks like local function calls?
- What data encoding formats work best in which data retrieval situation for the use cases defined below?

The requirement for the DAPA was to enable an end user to access geospatial data for a specific area (or point), as a single point in time or time series, performing statistical computations (e.g. minimum, maximum, and average) via a single call.

DAPA was tested on point data, gridded data, and on Discrete Global Grid System (DGGS) based data. The Testbed 16 participants mostly agreed on one common API. By the end of the Testbed, two flavors of the API flourished: The first flavor focused more on the process and the second mostly focused on the output type.

The document "OGC 20-016 Data Access and Processing Engineering Report" {**Will need link prior to publication**} provides more details about the user/client experiences resulting from using DAPA.

Future OGC Innovation Program activities to further harmonize the two flavors of the APIs is recommended.

2.1. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Contacts

Name	Organization	Role
Luis Bermudez	GeoSolutions	Editor

Name	Organization	Role
Clemens Portele	interactive instruments GmbH	Contributor
Pedro Gonçalves	Terradue	Contributor
Fabrice Brito	Terradue	Contributor
Andrea Aime	GeoSolutions	Contributor
Stephan Meißl	EOX	Contributor
Fabian Schindler	EOX	Contributor

2.2. Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

Chapter 3. Overview

This report mainly captures the DAPA definitions developed in Testbed 16.

Section 4 documents the DAPA design considerations.

Chapter 5 documents the DAPA definition based on the OpenAPI specification.

Chapter 6 provides the DAPA definition for accessing a Discrete Global Grid System (DGGS) structured data source.

Chapter 7 documents the second DAPA definition that is more focused on the output geometry.

Chapter 4. DAPA API Design Considerations

The primary goal for developing DAPA is to make it easier for end users to interact with services that access data and processing services. The application design needed to shift from a provider-centric view to a user-centric view. The expectation is that when the end user interacts with the system this would be similar to calling a local function calls on a dataset in memory. For example, for an end user function calls to calculate the minimum temperature value for a target area should look similar, even if implemented on different products, regardless of the data location, be it a local file, an in-memory structure, or a remote data set stored in the cloud.

Testbed-16 participants explored this end-user centric perspective and developed the API described in this ER. This API definition can serve as the basis for future standardization work in the context of end-user centric data retrieval and processing APIs.

Two main design approaches were developed. The first design focused more on the process and the second design mostly focused on the output type. These APIs are presented in chapter 5,6 and 7 of this ER.

4.1. General Considerations

The following are the general considerations explored:

- The Data Access and Processing API (DAPA) is a set of OGC API building blocks that operate on a Collection with observation data.
- Each observation is associated with a point in space (2D or 3D) and time.
- The spatial and temporal aspects may be regular (e.g. gridded) or scattered.
- The same can be said of the observed properties. The same observed properties may be available at each spatiotemporal point or they may vary from point to point.
- All DAPA requests retrieve a subset of the observations. Data retrieval using DAPA is based on the selection of the observations by a spatial sampling geometry, the temporal sampling geometry, and the observed properties.
- In addition, DAPA requests offer a capability to aggregate the selected observations for each selected spatial point (aggregate over time), for each selected temporal point (aggregate over space), or for all selected observations (aggregate over space and time).
- The Testbed-16 DAPA design follows a similar approach to the design advanced by the Environmental Data Retrieval API (EDR API SWG [https://www.ogc.org/projects/groups/edr-apiswg]) approach of providing a separate resource, in this case "DAPA". Endpoints related to DAPA vary depending on the type of the spatial sampling geometry and the temporal sampling geometry. The observed properties are included in the query parameters.
- The goal is to simplify the use of the data for end-users, for example, using Jupyter Notebooks [https://jupyter.org/]. Using libraries like GeoPandas [https://geopandas.org/] implies that the data is returned as tabular data to simplify processing of the data in Jupyter notebooks.

4.2. Linking to the DAPA resources and DAPA metadata

The DAPA resources are sub-resources under the draft OGC API Collection [https://htmlpreview.github.io/?https://github.com/opengeospatial/oapi_common/blob/master/20-024.html] resource definition specified in part 2 of the draft OGC API Common [https://htmlpreview.github.io/?https://github.com/opengeospatial/oapi_common/blob/master/19-072.html] (OGC 19-072). Each DAPA endpoint should be accessible by following links from the Collection resource. For Testbed-16 a harmonized approach was not required since the APIs were used based on inspection of the documentation and not by generic clients. That said, in general, the way DAPA resources are linked needs to be specified so that generic clients navigating the API could be supported.

Links are used by a client to traverse information. They can be implemented in different ways. The two alternatives for using the link relation types consistently are as follows:

- Use the `collections/{collectionId}/processes` resource to link to each DAPA endpoint (link relation type, e.g., `http://www.opengis.net/def/rel/ogc/1.0/dapa-endpoint` or even `http://www.opengis.net/def/rel/ogc/1.0/process`) and include a single link to that resource from the Collection (link relation type, e.g., `http://www.opengis.net/def/rel/ogc/1.0/processes`).
- Link to all DAPA endpoints directly from the Collection (link relation type, e.g., `http://www.opengis.net/def/rel/ogc/1.0/dapa-endpoint`).

One of the challenges for any client (generic client or a human expert) is to understand how to use the API. For a human expert, this information can be provided through a good documentation of the DAPA endpoints in an OpenAPI definition document (specification for REST APIs) and/or in separate documentation. To support generic clients (in the future), there needs to be a well-defined pattern that can be analyzed by code. For example, through a consistent naming pattern for DAPA endpoint resources.

In order to support clients generating requests, the client also needs to know the spatial and temporal extent as well as the available observed properties. The spatial and temporal extents are provided by the Collection resource, but additional information on the available observed properties is needed. The properties should be documented in the OpenAPI definition of the API. However, this information should also be available as part of the API.

One option is to define another resource that lists the available observed properties (link relation type, e.g., `http://www.opengis.net/def/rel/ogc/1.0/dapa-variables`). For each observed property there should be at least the following information: The id to be used in the API and a human readable title/description including the unit of measurement of the values.

Proposal:

- Testbed-16 requirements did not specify whether there needs to be a separate root resource or which link relation types are to be used.
- The DAPA endpoints will be available at `/collections/{collectionId}/processes/{dapaProcessId}`. More on the `dataProcessId`s and the endpoints below.
- The observed properties will be available at `/collections/{collectionId}/variables`.

The following pattern was followed in the JSON response:

```
{
  "variables" : [ {
    "id" : "PRCP",
    "title" : "Precipitation",
    "uom" : "0.1 mm"
  }, {
    "id" : "TMAX",
    "title" : "Maximum temperature",
    "uom" : "0.1 °C"
  }, {
    "id" : "TMIN",
    "title" : "Minimum temperature",
    "uom" : "0.1 °C"
  } ]
}
```

4.3. Data access

The DAPA selection pattern is based on the knowledge that the set of all observations can be viewed as a spatio-temporal data cube. Data cubes are multidimensional array of values that enable efficient access of data. The data is selected based on three aspects:

- The spatial sampling geometry (2d point, 2d polygon including a bounding box, and/or 2d grid),
- The temporal sampling geometry (instant, interval, timeseries), and
- The observed properties / variables to retrieve.

Support for all sampling geometry types is optional. The available options can be determined from the OpenAPI definition and separate documentation. If moving forward with a DAPA specification, the optional elements should be moved to a separate conformance class.

Note that in Testbed-16 a mechanism about how to determine the available query parameters in the API was not specified. The query parameters are based on existing parameters available, following the OGC API Features [https://www.ogc.org/standards/ogcapi-features] (OGC 17-069r3) standard, the OGC API Common (OGC 19-072) candidate standard, and the OGC API - Environmental Data Retrieval [https://portal.ogc.org/files/?artifact_id=94444&version=1&utm_source=phplist759&utm_medium=email&utm_content=HTML&utm_campaign=OGC+seeks+public+comment+on+new+OGC+API+for+Environmental+Data+Retrieval] candidate standard (OGC 19-086r0).

4.3.1. Spatial sampling geometry

4.3.1.1. 2d Point

Observations at a requested location (region of interest) are retrieved or interpolated. In general, the result is a time series for each available variable.

The location of the point can be provided in one of two ways:

- Query parameter coords: A Well Known Text (WKT) point geometry. Example: POINT(6.9299617 50.000008).
- Query parameter coordsRef: A URI that returns a GeoJSON feature, such as a request to an API that implements OGC API - Features. If the feature geometry is not a point, the centroid should be used.

NOTE There are some security concerns with respect to passing a URI to a server that the server has to resolve.

If point sampling is supported, support for coords is mandatory.

4.3.1.2. 2d polygon or bounding box

Observations that intersect the sampling geometry are retrieved. The result is a time series for each variable. For gridded data, the result is a time series per grid cell and for scattered data the result is a time series for each location (e.g. each point).

The polygon of the area can be provided in one of three ways:

- Query parameter bbox: Consisting of four numbers, separated by commas: Minimum longitude, minimum latitude, maximum longitude, maximum latitude. Example: 6,48.5,8,50.5.
- Query parameter coords: A Well Known Text (WKT) polygon or multi-polygon geometry.
- Query parameter coordRef: A URI that returns a GeoJSON feature, such as a request to an API that implements OGC API - Features. If the feature geometry is not a polygon or multi-polygon, the API will determine the area. For example, adding a buffer and using the bounding box of the resulting geometry.

The use of bbox in DAPA is consistent with OGC API - Features.

If polygonal sampling is supported, support for bbox is mandatory.

4.3.1.3. 2d grid

Observations in the grid area are retrieved and resampled to the grid. The result is a time series for each variable and each grid cell.

The grid can be specified provided as follows:

- Query parameter bbox: Same as above.
- Query parameters width and height: Number of grid cells along each coordinate axis. For gridded observation data, the default would be to use the same grid structure as the source data (no-resampling).

4.3.2. Temporal sampling geometry

4.3.2.1. Instant or interval

The temporal sampling geometry is always provided in the parameter `datetime`. Use timestamps or dates. Intervals are expressed according to ISO 8601. Open intervals can be supported, too.

The use of `datetime` in the DAPA API is consistent with OGC API - Features. See the OGC API - Features standard for details about open intervals.

4.3.2.2. Time series

NOTE Time series is not currently implemented in the DAPA endpoints.

In addition to interval, the information about the time steps is required. This could be provided in an additional `timestep` parameter.

4.3.3. Observed properties

One or more observed properties / variables can be retrieved using DAPA. The values are provided as a comma-separated list in parameter `variables`.

Note that EDR uses `parametername`, but that name was not considered intuitive. The use of the term `parameter'' can be confusing in the context and the` name'' suffix is unnecessary/unusual.

In some cases, the variables are related to each other, such as ordered spectral bands, which could be used for subsetting purposes similar to the spatial and temporal aspects. For Testbed-16, no relationships between the variables were considered and such capabilities were considered out of scope.

4.4. Data aggregation and processing

This step is optional. If selected, the observations that have been retrieved in the data access phase are aggregated for each variable.

- Over time (for point, polygon or grid sampling geometries),
- Over the whole area (polygon or grid sampling geometry), or
- Both (polygon or grid sampling geometry).

If the temporal sampling geometry is an instant, aggregation over time operates on a single value.

The statistical functions to use on the set of values for each variable are provided as a comma-separated list in a query parameter `functions`.

The following identifiers are used for a core set of statistical functions:

- Minimum value (`min`)
- Maximum value (`max`)
- Average/mean value (`mean`)
- Standard deviation (`std-dev`)

- Number of values (`count`)

4.5. DAPA endpoints

All path names are relative to `/collections/{collectionId}/processes/`.

> **NOTE** If this pattern would create issues with the current direction of OGC API Processes [https://github.com/opengeospatial/ogcapi-processes], this would be recorded in the ER(s).

- `position:retrieve`: Retrieve observation data at a location (point sampling) for a time instant, interval or series
- `position:aggregate-time`: Retrieve observation data at a location (point sampling) for a time interval or series and aggregate over time
- `area:retrieve`: Retrieve observation data in an area (polygonal sampling) for a time instant, interval or series
- `area:aggregate-time`: Retrieve observation data in an area (polygonal sampling) for a time interval or series and aggregate over time
- `area:aggregate-space`: Retrieve observation data in an area (polygonal sampling) for a time instant, interval or series and aggregate over space
- `area:aggregate-space-time`: Retrieve observation data in an area (polygonal sampling) for a time instant, interval or series and aggregate over space and time.
- Use `grid` instead of `area`, if the spatial sampling geometry is a grid.

4.6. DAPA endpoint responses

A challenge is harmonizing the data formats and their details as provided by different DAPA endpoints.

The Interactive Instruments DAPA implementation supports GeoJSON (with simple values, i.e., tabular data in `properties`) and CSV for all endpoints and GeoTIFF for `grid:aggregate-time` (See DAPA Design 1).

The Terradue DAPA implementation supports JSON representations based on xarray (See DAPA API Design 2).

The current requirement is that each DAPA endpoint specifies the data formats and data structures that clients will receive for each DAPA endpoint.

Harmonization of the response formats and structures was out of scope for Testbed-16 and it is unlikely these two can be fully harmonized. Future work might involve investigating what could be possible to be specified in a standard.

Chapter 5. DAPA Design 1

This section provides the details of the DAPA design more focused on the output type.

To demonstrate an implementation of the proposed DAPA, examples are provided that access the National Oceanic and Atmospheric Administration (NOAA)'s Global Historical Climatology Network Daily data.

5.1. About the NOAA Global Historical Climatology Network Daily (GHCN-D)

Global Historical Climatology Network - Daily [https://www.ncdc.noaa.gov/ghcn-daily-description] is a dataset from NOAA that contains daily observations over global land areas. The dataset contains station-based measurements from land-based stations worldwide, about two thirds of which are for precipitation measurement only. Other meteorological elements include, but are not limited to, daily maximum and minimum temperature, snowfall and snow depth. The dataset is a composite of climate records from numerous sources that were merged together and subjected to a common suite of quality assurance reviews.

The DAPA imlpementation provided access to 2019 data contained in the GHCN-D datasets. These data consist of 115,082 weather stations and 34,093,913 observations.

The DAPA implementation provided access to each weather station and observation via the OGC API Features [https://www.ogc.org/standards/ogcapi-features] implementation endpoint.

In addition, the concept of a Data Access and Processing API (DAPA) [https://portal.ogc.org/files/91644#PartDAPA] that was explored in OGC Testbed-16 provided additional options to retrieve the data with the goal of simplifying the use of the data for end-users, such as using Jupyter Notebooks. Data retrieval via a DAPA endpoint is based on the selection of the observations by the spatial sampling geometry (point, area, or grid), the temporal sampling geometry (instant or interval), and the observed properties followed by optional post-processing to aggregate observations by space and/or time.

5.2. DAPA Overview

Capabilities

- `GET /` landing page
- `GET /conformance` conformance declaration

Discover data collections

- `GET /collections` feature collections in the dataset 'NOAA Global Historical Climatology Network Daily (GHCN-D)'
- `GET /collections/observation` feature collection 'Observations'
- `GET /collections/station` feature collection 'Stations'

- `GET /collections/observation/queryables` retrieve the queryables of the feature collection 'observation'
- `GET /collections/station/queryables` retrieve the queryables of the feature collection 'station'

Access data

- `GET /collections/observation/items` retrieve features in the feature collection 'observation'
- `GET /collections/observation/items/{featureId}` retrieve a feature in the feature collection 'observation'
- `GET /collections/station/items` retrieve features in the feature collection 'station'
- `GET /collections/station/items/{featureId}` retrieve a feature in the feature collection 'station'

DAPA

- `GET /collections/observation/processes` list the available data retrieval patterns
- `GET /collections/observation/variables` fetch the observable properties included in this observation collection
- `GET /collections/observation/processes/area:aggregate-space` retrieve a time series for selected variables for each station in an area and apply functions on the values of each time step
- `GET /collections/observation/processes/area:aggregate-space-time` retrieve a time series for selected variables for each station in an area and apply functions on all values
- `GET /collections/observation/processes/area:aggregate-time` retrieve a time series for selected variables for each station in an area and apply functions on the values of each time series
- `GET /collections/observation/processes/area:retrieve` retrieve a time series for selected variables for each station in an area
- `GET /collections/observation/processes/grid:aggregate-time` retrieve a time series for selected variables for each station in an area, resample the observations to a time series in a 2D grid and apply functions on the values of each time series
- `GET /collections/observation/processes/grid:retrieve` retrieve a time series for selected variables for each station in an area and resample the observations to a time series for each cell in a 2D grid
- `GET /collections/observation/processes/position:aggregate-time` retrieve a time series for selected variables at a position
- `GET /collections/observation/processes/position:retrieve` retrieve a time series for selected variables at a position

5.3. Capabilities

5.3.1. GET /

Landing page.

5.3.1.1. Description

The landing page provides links to the API definition (link relations `service-desc` and `service-doc`), the Conformance declaration (path `/conformance`, link relation `conformance`), and other resources in the API.

5.3.1.2. Query Parameters

Name	Description	Required	Default
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply, i.e., the accept header will be used to determine the format.	-	null

5.3.1.3. Return Type

LandingPage

5.3.1.4. Content Type

- application/json
- text/html

5.3.1.5. Responses

Table 1. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	LandingPage
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.3.1.6. Example data

Content-Type: application/json

```
{
  "extent" : {
    "spatial" : {
      "crs" : "crs",
      "bbox" : [ [ 0.8008281904610115, 0.8008281904610115 ], [ 0.8008281904610115, 0.8008281904610115 ] ]
    },
    "temporal" : {
      "trs" : "trs",
      "interval" : [ [ "interval", "interval" ], [ "interval", "interval" ] ]
    }
  },
  "description" : "description",
  "links" : [ {
    "hreflang" : "en",
    "rel" : "alternate",
    "length" : 6,
    "href" : "http://data.example.com/buildings/123",
    "type" : "application/geo+json",
    "title" : "Trierer Strasse 70, 53115 Bonn"
  }, {
    "hreflang" : "en",
    "rel" : "alternate",
    "length" : 6,
    "href" : "http://data.example.com/buildings/123",
    "type" : "application/geo+json",
    "title" : "Trierer Strasse 70, 53115 Bonn"
  } ],
  "externalDocs" : {
    "description" : "description",
    "url" : "url"
  },
  "title" : "title"
}
```

5.3.2. GET /conformance

Conformance declaration.

5.3.2.1. Description

The URIs of all conformance classes supported by the server. This information is provided to support 'generic' clients that want to access multiple OGC API implementations - and not 'just' a specific API. For clients accessing only a single API, this information is in general not relevant and the OpenAPI definition details the required information about the API.

5.3.2.2. Query Parameters

Name	Description	Required	Default
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply: The accept header will be used to determine the format.	-	null

5.3.2.3. Return Type

ConformanceDeclaration

5.3.2.4. Content Type

- application/json
- text/html

5.3.2.5. Responses

Table 2. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	ConformanceDeclaration
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.3.2.6. Example data

Content-Type: application/json

```
{
  "description" : "description",
  "links" : [ {
    "hreflang" : "en",
    "rel" : "alternate",
    "length" : 6,
    "href" : "http://data.example.com/buildings/123",
    "type" : "application/geo+json",
    "title" : "Trierer Strasse 70, 53115 Bonn"
  }, {
    "hreflang" : "en",
    "rel" : "alternate",
    "length" : 6,
    "href" : "http://data.example.com/buildings/123",
    "type" : "application/geo+json",
    "title" : "Trierer Strasse 70, 53115 Bonn"
  } ],
  "conformsTo" : [ "conformsTo", "conformsTo" ],
  "title" : "title"
}
```

5.4. Discover data collections

5.4.1. GET /collections

Feature collections in the dataset 'NOAA Global Historical Climatology Network Daily (GHCN-D)'.

5.4.1.1. Description

The dataset is organized in feature collections. This resource provides information about and access to the feature collections. The response contains the list of collections. For each collection, a link to the items in the collection (path `/collections/{collectionId}/items`, link relation `items`) as well as key information about the collection. This information includes:

- A local identifier for the collection that is unique for the dataset;
- A list of coordinate reference systems (CRS) in which geometries may be returned by the server. The first CRS is the default coordinate reference system (the default is always WGS 84 with axis order longitude/latitude);
- An optional title and description for the collection;
- An optional extent that can be used to provide an indication of the spatial and temporal extent of the collection - typically derived from the data;
- An optional indicator about the type of the items in the collection (the default value, if the indicator is not provided, is 'feature').

5.4.1.2. Query Parameters

Name	Description	Required	Default
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply: The accept header will be used to determine the format.	null	

5.4.1.3. Content Type

- application/json
- text/html

5.4.1.4. Responses

Table 3. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	collection
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.4.1.5. Example data

Content-Type: application/json

```
{
  "extent" : {
    "spatial" : {
      "crs" : "crs",
      "bbox" : [ [ 0.8008281904610115, 0.8008281904610115 ], [ 0.8008281904610115,
0.8008281904610115 ] ]
    },
    "temporal" : {
      "trs" : "trs",
      "interval" : [ [ "interval", "interval" ], [ "interval", "interval" ] ]
    }
  },
  "storageCrs" : "storageCrs",
  "storageCrsCoordinateEpoch" : 0.8008281904610115,
  "nativeCrsCoordinateEpoch" : 6.027456183070403,
  "itemType" : "itemType",
  "crs" : [ "crs", "crs" ],
  "description" : "description",
  "nativeCrs" : "nativeCrs",
  "links" : [ {
    "hreflang" : "en",
    "rel" : "alternate",
    "length" : 6,
    "href" : "http://data.example.com/buildings/123",
    "type" : "application/geo+json",
    "title" : "Trierer Strasse 70, 53115 Bonn"
  }, {
    "hreflang" : "en",
    "rel" : "alternate",
    "length" : 6,
    "href" : "http://data.example.com/buildings/123",
    "type" : "application/geo+json",
    "title" : "Trierer Strasse 70, 53115 Bonn"
  } ],
  "id" : "id",
  "title" : "title"
}
```

5.4.2. GET /collections/observation

Feature collection 'Observations'.

5.4.2.1. Description

Information about the feature collection with id 'observation'. The response contains a link to the items in the collection (path `/collections/{collectionId}/items`,link relation `items`) as well as key information about the collection. This information includes:

- A local identifier for the collection that is unique for the dataset;

- A list of coordinate reference systems (CRS) in which geometries may be returned by the server. The first CRS is the default coordinate reference system (the default is always WGS 84 with axis order longitude/latitude);
- An optional title and description for the collection;
- An optional extent that can be used to provide an indication of the spatial and temporal extent of the collection - typically derived from the data;
- An optional indicator about the type of the items in the collection (the default value, if the indicator is not provided, is 'feature').

5.4.2.2. Query Parameters

Name	Description	Required	Default
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply: The accept header will be used to determine the format.	-	null

5.4.2.3. Return Type

collection

5.4.2.4. Content Type

- application/json
- text/html

5.4.2.5. Responses

Table 4. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	collection
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.4.3. GET /collections/station

Feature collection 'Stations'.

5.4.3.1. Description

Information about the feature collection with id 'station'. The response contains a link to the items in the collection (path `/collections/{collectionId}/items`,link relation `items`) as well as key information about the collection. This information includes:

- A local identifier for the collection that is unique for the dataset;
- A list of coordinate reference systems (CRS) in which geometries may be returned by the server. The first CRS is the default coordinate reference system (the default is always WGS 84 with axis order longitude/latitude);
- An optional title and description for the collection;
- An optional extent that can be used to provide an indication of the spatial and temporal extent of the collection - typically derived from the data;
- An optional indicator about the type of the items in the collection (the default value, if the indicator is not provided, is 'feature').

5.4.3.2. Query Parameters

Name	Description	Required	Default
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply: The accept header will be used to determine the format.	-	null

5.4.3.3. Return Type

collection

5.4.3.4. Content Type

- application/json
- text/html

5.4.3.5. Responses

Table 5. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	collection
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.4.3.6. Example data

Content-Type: application/json

```
{
  "extent" : {
    "spatial" : {
      "crs" : "crs",
      "bbox" : [ [ 0.8008281904610115, 0.8008281904610115 ], [ 0.8008281904610115,
0.8008281904610115 ] ]
    },
    "temporal" : {
      "trs" : "trs",
      "interval" : [ [ "interval", "interval" ], [ "interval", "interval" ] ]
    }
  },
  "storageCrs" : "storageCrs",
  "storageCrsCoordinateEpoch" : 0.8008281904610115,
  "nativeCrsCoordinateEpoch" : 6.027456183070403,
  "itemType" : "itemType",
  "crs" : [ "crs", "crs" ],
  "description" : "description",
  "nativeCrs" : "nativeCrs",
  "links" : [ {
    "hreflang" : "en",
    "rel" : "alternate",
    "length" : 6,
    "href" : "http://data.example.com/buildings/123",
    "type" : "application/geo+json",
    "title" : "Trierer Strasse 70, 53115 Bonn"
  }, {
    "hreflang" : "en",
    "rel" : "alternate",
    "length" : 6,
    "href" : "http://data.example.com/buildings/123",
    "type" : "application/geo+json",
    "title" : "Trierer Strasse 70, 53115 Bonn"
  } ],
  "id" : "id",
  "title" : "title"
}
```

5.4.4. GET /collections/observation/queryables

Retrieve the queryables of the feature collection 'observation'.

5.4.4.1. Query Parameters

Name	Description	Required	Default
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply: The accept header will be used to determine the format.	-	null

5.4.4.2. Return Type

Queryables

5.4.4.3. Content Type

- application/json
- text/html

5.4.4.4. Responses

Table 6. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	Queryables
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.4.4.5. Example Data

Content-Type: application/json

```
{
  "description" : "description",
  "links" : [ {
    "hreflang" : "en",
    "rel" : "alternate",
    "length" : 6,
    "href" : "http://data.example.com/buildings/123",
    "type" : "application/geo+json",
    "title" : "Trierer Strasse 70, 53115 Bonn"
  }, {
    "hreflang" : "en",
    "rel" : "alternate",
    "length" : 6,
    "href" : "http://data.example.com/buildings/123",
    "type" : "application/geo+json",
    "title" : "Trierer Strasse 70, 53115 Bonn"
  } ],
  "title" : "title",
  "queryables" : [ {
    "mediaTypes" : [ "mediaTypes", "mediaTypes" ],
    "min" : { },
    "max" : { },
    "values" : [ "values", "values" ],
    "pattern" : "pattern",
    "description" : "description",
    "language" : "language",
    "id" : "id",
    "title" : "title",
    "type" : "type",
    "required" : true
  }, {
    "mediaTypes" : [ "mediaTypes", "mediaTypes" ],
    "min" : { },
    "max" : { },
    "values" : [ "values", "values" ],
    "pattern" : "pattern",
    "description" : "description",
    "language" : "language",
    "id" : "id",
    "title" : "title",
    "type" : "type",
    "required" : true
  } ]
}
```

5.4.5. GET /collections/station/queryables

Retrieve the queryables of the feature collection 'station'.

5.4.5.1. Query Parameters

Name	Description	Required	Default	Pattern
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply: The accept header will be used to determine the format.	-	null	

5.4.5.2. Return Type

Queryables

5.4.5.3. Content Type

- application/json
- text/html

5.4.5.4. Responses

Table 7. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	Queryables
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.4.5.5. Example data

Content-Type: application/json

```
{
  "description" : "description",
  "links" : [ {
    "hreflang" : "en",
    "rel" : "alternate",
    "length" : 6,
    "href" : "http://data.example.com/buildings/123",
    "type" : "application/geo+json",
    "title" : "Trierer Strasse 70, 53115 Bonn"
  }, {
    "hreflang" : "en",
    "rel" : "alternate",
    "length" : 6,
    "href" : "http://data.example.com/buildings/123",
    "type" : "application/geo+json",
    "title" : "Trierer Strasse 70, 53115 Bonn"
  } ],
  "title" : "title",
  "queryables" : [ {
    "mediaTypes" : [ "mediaTypes", "mediaTypes" ],
    "min" : { },
    "max" : { },
    "values" : [ "values", "values" ],
    "pattern" : "pattern",
    "description" : "description",
    "language" : "language",
    "id" : "id",
    "title" : "title",
    "type" : "type",
    "required" : true
  }, {
    "mediaTypes" : [ "mediaTypes", "mediaTypes" ],
    "min" : { },
    "max" : { },
    "values" : [ "values", "values" ],
    "pattern" : "pattern",
    "description" : "description",
    "language" : "language",
    "id" : "id",
    "title" : "title",
    "type" : "type",
    "required" : true
  } ]
}
```

5.5. Access data

5.5.1. GET /collections/observation/items

Retrieve features in the feature collection 'observation'.

5.5.1.1. Description

The response is a document consisting of features in the collection. The features included in the response are determined by the server based on the query parameters of the request. To support access to larger collections without overloading the client, the API supports paged access with links to the next page, if the number of features selected is greater than the number that can be presented on a single page. The `bbox` and `datetime` parameter can be used to select only a subset of the features in the collection (the features that are in the bounding box or time interval). The `bbox` parameter matches all features in the collection that are not associated with a location, too. The `datetime` parameter matches all features in the collection that are not associated with a time stamp or interval, too. The `limit` parameter may be used to control the subset of the selected features that should be returned in the response, the page size. Each page may include information about the number of selected and returned features (`numberMatched` and `numberReturned`) as well as links to support paging (link relation `next`).

5.5.1.2. Query Parameters

Name	Description	Required	Default
bbox	Only features that have a geometry that intersects the bounding box are selected. The bounding box is provided as four or six numbers, depending on whether the coordinate reference system (CRS) includes a vertical axis (height or depth): • Lower left corner, coordinate axis 1 • Lower left corner, coordinate axis 2 • Minimum value, coordinate axis 3 (optional) • Upper right corner, coordinate axis 1 * Upper right corner, coordinate axis 2 • Maximum value, coordinate axis 3 (optional) The CRS of the values is WGS 84 longitude/latitude (http://www.opengis.net/def/crs/OGC/1.3/CRS84) unless a different CRS is specified in the parameter `bbox-crs`. For WGS 84 longitude/latitude the values are, in most cases, the sequence of minimum longitude, minimum latitude, maximum longitude and maximum latitude. However, in cases where the box spans the antimeridian the first value (west-most box edge) is larger than the third value (east-most box edge). If the vertical axis is included, the third and the sixth number are the bottom and the top of the 3-dimensional bounding box. If a feature has multiple spatial geometry properties, it is the decision of the server whether only a single spatial geometry property is used to determine the extent or all relevant geometries.	-	null
bbox-crs	The CRS of the 'bbox' parameter. Default is WGS84 longitude/latitude (http://www.opengis.net/def/crs/OGC/1.3/CRS84).	-	http://www.opengis.net/def/crs/OGC/1.3/CRS84
crs	The CRS of the response geometries. Default is WGS84 longitude/latitude (http://www.opengis.net/def/crs/OGC/1.3/CRS84).	-	http://www.opengis.net/def/crs/OGC/1.3/CRS84

Name	Description	Required	Default
datetime	Either a local date, a date-time with offsets or an open or closed interval. Date and time expressions adhere to RFC 3339. Examples: • A date-time: '2018-02-12T23:20:50Z' • A closed interval: '2018-02-12T00:00:00Z/2018-03-18T12:31:12Z' • Open intervals: '2018-02-12T00:00:00Z/..' or '../2018-03-18T12:31:12Z' Selects features that have a temporal property that intersects the value of the parameter.	-	null
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply: The accept header will be used to determine the format.	-	null
filter	Filter features in the collection using the query expression in the parameter value.	-	null
filter-lang	Language of the query expression in the 'filter' parameter.	-	cql-text
limit	The optional limit parameter limits the number of items that are presented in the response document. Only items are counted that are on the first level of the collection in the response document. Nested objects contained within the explicitly requested items shall not be counted.	-	10
offset	The optional offset parameter identifies the index of the first feature in the response in the overall result set.	-	0
properties	The properties that should be included for each feature. The parameter value is a comma-separated list of property names. String	-	null
skipGeometry	This option can be used to skip response geometries for each feature.	-	false
locationCode	Filter the collection by property 'locationCode'	-	null
locationName	Filter the collection by property 'locationName'	-	null
observedProperty	Filter the collection by property 'observedProperty'	-	null
result	Filter the collection by property 'result'	-	null

5.5.1.3. Return Type

featureCollectionGeoJson_observation

5.5.1.4. Content Type

- application/geo+json
- text/html
- application/json

5.5.1.5. Responses

Table 8. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	featureCollectionGeoJson_observation
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.5.1.6. Example Data

Content-Type: application/json

```
{
  "timeStamp" : "2017-08-17T08:05:32Z",
  "features" : [ {
    "geometry" : {
      "coordinates" : [ 0.8008281904610115, 0.8008281904610115 ],
      "type" : "Point"
    },
    "links" : [ {
      "hreflang" : "en",
      "rel" : "alternate",
      "length" : 6,
      "href" : "http://data.example.com/buildings/123",
      "type" : "application/geo+json",
      "title" : "Trierer Strasse 70, 53115 Bonn"
    }, {
      "hreflang" : "en",
      "rel" : "alternate",
      "length" : 6,
      "href" : "http://data.example.com/buildings/123",
      "type" : "application/geo+json",
      "title" : "Trierer Strasse 70, 53115 Bonn"
    } ],
    "id" : "id",
    "type" : "Feature",
    "properties" : {
      "unitOfMeasurement" : "unitOfMeasurement",
      "result" : 1,
      "locationName" : "locationName",
      "phenomenonTime" : "2000-01-23T04:56:07.000+00:00",
      "observedProperty" : "observedProperty",
      "id" : 6,
      "locationCode" : "locationCode",
      "locationLink" : "locationLink"
    }
  }, {
    "geometry" : {
      "coordinates" : [ 0.8008281904610115, 0.8008281904610115 ],
      "type" : "Point"
    },
    "links" : [ {
      "hreflang" : "en",
      "rel" : "alternate",
      "length" : 6,
      "href" : "http://data.example.com/buildings/123",
      "type" : "application/geo+json",
      "title" : "Trierer Strasse 70, 53115 Bonn"
    }, {
      "hreflang" : "en",
      "rel" : "alternate",
      "length" : 6,
      "href" : "http://data.example.com/buildings/123",
      "type" : "application/geo+json",
      "title" : "Trierer Strasse 70, 53115 Bonn"
    } ],
    "id" : "id",
    "type" : "Feature",
    "properties" : {
      "unitOfMeasurement" : "unitOfMeasurement",
      "result" : 1,
      "locationName" : "locationName",
      "phenomenonTime" : "2000-01-23T04:56:07.000+00:00",
      "observedProperty" : "observedProperty",
      "id" : 6,
      "locationCode" : "locationCode",
      "locationLink" : "locationLink"
    }
  } ],
  "numberReturned" : 10,
  "links" : [ null, null ],
  "type" : "FeatureCollection",
  "numberMatched" : 127
}
```

5.5.2. GET /collections/observation/items/{featureId}

Retrieve a feature in the feature collection 'observation'

5.5.2.1. Description

Fetch the feature with id `{featureId}`.

5.5.2.2. Parameters

5.5.2.2.1. Path Parameters

Name	Description	Required	Default	Pattern
featureId	The local identifier of a feature, unique within the feature collection.	X	null	/[\\w\\-\\.]+/

5.5.2.2.2. Query Parameters

Name	Description	Required	Default
crs	The coordinate reference system of the response geometries. Default is WGS84 longitude/latitude (http://www.opengis.net/def/crs/OGC/1.3/CRS84).	-	http://www.opengis.net/def/crs/OGC/1.3/CRS84
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply: The accept header is used to determine the format. The Accept request-header field can be used to specify certain media types which are acceptable for the response.	-	null
properties	The properties that should be included for each feature. The parameter value is a comma-separated list of property names.	-	null
skipGeometry	This option can be used to skip response geometries for each feature.	-	false

5.5.2.3. Return Type

Observations

5.5.2.4. Content Type

- application/geo+json
- text/html
- application/json

5.5.2.5. Responses

Table 9. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	Observations
400	A query parameter has an invalid value.	exception
404	The requested resource does not exist on the server. For example, a path parameter had an incorrect value.	Variables_variables
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.5.2.6. Example data

Content-Type: application/json

```
{
  "geometry" : {
    "coordinates" : [ 0.8008281904610115, 0.8008281904610115 ],
    "type" : "Point"
  },
  "links" : [ {
    "hreflang" : "en",
    "rel" : "alternate",
    "length" : 6,
    "href" : "http://data.example.com/buildings/123",
    "type" : "application/geo+json",
    "title" : "Trierer Strasse 70, 53115 Bonn"
  }, {
    "hreflang" : "en",
    "rel" : "alternate",
    "length" : 6,
    "href" : "http://data.example.com/buildings/123",
    "type" : "application/geo+json",
    "title" : "Trierer Strasse 70, 53115 Bonn"
  } ],
  "id" : "id",
  "type" : "Feature",
  "properties" : {
    "unitOfMeasurement" : "unitOfMeasurement",
    "result" : 1,
    "locationName" : "locationName",
    "phenomenonTime" : "2000-01-23T04:56:07.000+00:00",
    "observedProperty" : "observedProperty",
    "id" : 6,
    "locationCode" : "locationCode",
    "locationLink" : "locationLink"
  }
}
```

5.5.3. GET /collections/station/items

Retrieve features in the feature collection 'station'.

5.5.3.1. Description

The response is a document consisting of features in the collection. The features included in the response are determined by the server based on the query parameters of the request. To support access to larger collections without overloading the client, the API supports paged access with links to the next page, if the number of features selected is greater than the number that can be presented on a single page. The `bbox` and `datetime` parameter can be used to select only a subset of the features in the collection (the features that are in the bounding box or time interval). The `bbox` parameter matches all features in the collection that are not associated with a location, too. The `datetime` parameter matches all features in the collection that are not associated with a time stamp or interval, too. The `limit` parameter may be used to control the subset of the selected features that should be returned in the response, i.e. the page size. Each page may include information about the number of selected and returned features (`numberMatched` and `numberReturned`) as well as links to support paging (link relation `next`).

5.5.3.2. Query Parameters

Name	Description	Required	Default
bbox	Only features that have a geometry that intersects the bounding box are selected. The bounding box is provided as four or six numbers, depending on whether the coordinate reference system includes a vertical axis (height or depth): • Lower left corner, coordinate axis 1 • Lower left corner, coordinate axis 2 • Minimum value, coordinate axis 3 (optional) • Upper right corner, coordinate axis 1 • Upper right corner, coordinate axis 2 • Maximum value, coordinate axis 3 (optional) The CRS of the values is WGS 84 longitude/latitude (http://www.opengis.net/def/crs/OGC/1.3/CRS84) unless a different CRS is specified in the parameter `bbox-crs`. For WGS 84 longitude/latitude the values are in most cases the sequence of minimum longitude, minimum latitude, maximum longitude and maximum latitude. However, in cases where the box spans the antimeridian the first value (west-most box edge) is larger than the third value (east-most box edge). If the vertical axis is included, the third and the sixth number are the bottom and the top of the 3-dimensional bounding box. If a feature has multiple spatial geometry properties, it is the decision of the server whether only a single spatial geometry property is used to determine the extent or all relevant geometries. [Double]	-	null

Name	Description	Required	Default
bbox-crs	The CRS of the 'bbox' parameter. Default is WGS84 longitude/latitude (http://www.opengis.net/def/crs/OGC/1.3/CRS84).	-	http://www.opengis.net/def/crs/OGC/1.3/CRS84
crs	The CRS of the response geometries. Default is WGS84 longitude/latitude (http://www.opengis.net/def/crs/OGC/1.3/CRS84).	-	http://www.opengis.net/def/crs/OGC/1.3/CRS84
datetime	Either a local date, a date-time with offsets or an open or closed interval. Date and time expressions adhere to RFC 3339. Examples: • A date-time: '2018-02-12T23:20:50Z' • A closed interval: '2018-02-12T00:00:00Z/2018-03-18T12:31:12Z' • Open intervals: '2018-02-12T00:00:00Z/..' or '../2018-03-18T12:31:12Z' Selects features that have a temporal property that intersects the value of the parameter.	-	null
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply, i.e., the accept header will be used to determine the format.	-	null
filter	Filter features in the collection using the query expression in the parameter value.	-	null
filter-lang	Language of the query expression in the 'filter' parameter.	-	cql-text
limit	The optional limit parameter limits the number of items that are presented in the response document. Only items are counted that are on the first level of the collection in the response document. Nested objects contained within the explicitly requested items shall not be counted.	-	10
offset	The optional offset parameter identifies the index of the first feature in the response in the overall result set.	-	0

Name	Description	Required	Default
properties	The properties that should be included for each feature. The parameter value is a comma-separated list of property names. String	-	null
skipGeometry	This option can be used to skip response geometries for each feature.	-	false
id	Filter the collection by property 'id'	-	null
name	Filter the collection by property 'name'	-	null
elevation	Filter the collection by property 'elevation'	-	null

5.5.3.3. Return Type

featureCollectionGeoJson_station

5.5.3.4. Content Type

- application/geo+json
- text/html
- application/json

5.5.3.5. Responses

Table 10. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	featureCollectionGeoJson_station
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.5.3.6. Example Data

Content-Type: application/json

```
{
  "timeStamp" : "2017-08-17T08:05:32Z",
  "features" : [ {
    "geometry" : {
      "coordinates" : [ 0.8008281904610115, 0.8008281904610115 ],
      "type" : "Point"
    },
    "links" : [ {
      "hreflang" : "en",
      "rel" : "alternate",
      "length" : 6,
      "href" : "http://data.example.com/buildings/123",
      "type" : "application/geo+json",
      "title" : "Trierer Strasse 70, 53115 Bonn"
    }, {
      "hreflang" : "en",
      "rel" : "alternate",
      "length" : 6,
      "href" : "http://data.example.com/buildings/123",
      "type" : "application/geo+json",
      "title" : "Trierer Strasse 70, 53115 Bonn"
    } ],
    "id" : "id",
    "type" : "Feature",
    "properties" : {
      "elevation" : 0.8008281904610115,
      "wmoId" : "wmoId",
      "observations" : "observations",
      "name" : "name",
      "id" : "id"
    }
  }, {
    "geometry" : {
      "coordinates" : [ 0.8008281904610115, 0.8008281904610115 ],
      "type" : "Point"
    },
    "links" : [ {
      "hreflang" : "en",
      "rel" : "alternate",
      "length" : 6,
      "href" : "http://data.example.com/buildings/123",
      "type" : "application/geo+json",
      "title" : "Trierer Strasse 70, 53115 Bonn"
    }, {
      "hreflang" : "en",
      "rel" : "alternate",
      "length" : 6,
      "href" : "http://data.example.com/buildings/123",
      "type" : "application/geo+json",
      "title" : "Trierer Strasse 70, 53115 Bonn"
    } ],
    "id" : "id",
    "type" : "Feature",
    "properties" : {
      "elevation" : 0.8008281904610115,
      "wmoId" : "wmoId",
      "observations" : "observations",
      "name" : "name",
      "id" : "id"
    }
  } ],
  "numberReturned" : 10,
  "links" : [ null, null ],
  "type" : "FeatureCollection",
  "numberMatched" : 127
}
```

5.5.4. GET /collections/station/items/{featureId}

retrieve a feature in the feature collection 'station'

5.5.4.1. Description

Fetch the feature with id `{featureId}`.

5.5.4.2. Path Parameters

Name	Description	Required	Default
featureId	The local identifier of a feature, unique within the feature collection.	X	null

5.5.4.2.1. Query Parameters

Name	Description	Required	Default
crs	The CRS of the response geometries. Default is WGS84 longitude/latitude (http://www.opengis.net/def/crs/OGC/1.3/CRS84).	-	http://www.opengis.net/def/crs/OGC/1.3/CRS84
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply: The accept header will be used to determine the format.	-	null
properties	The properties that should be included for each feature. The parameter value is a comma-separated list of property names.	-	null
skipGeometry	This option can be used to skip response geometries for each feature.	-	false

5.5.4.3. Return Type

Stations

5.5.4.4. Content Type

- application/geo+json
- text/html
- application/json

5.5.4.5. Responses

Table 11. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	Stations
400	A query parameter has an invalid value.	exception
404	The requested resource does not exist on the server. For example, a path parameter had an incorrect value.	Variables_variables
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.5.4.6. Example Data

Content-Type: application/json

```
{
  "geometry" : {
    "coordinates" : [ 0.8008281904610115, 0.8008281904610115 ],
    "type" : "Point"
  },
  "links" : [ {
    "hreflang" : "en",
    "rel" : "alternate",
    "length" : 6,
    "href" : "http://data.example.com/buildings/123",
    "type" : "application/geo+json",
    "title" : "Trierer Strasse 70, 53115 Bonn"
  }, {
    "hreflang" : "en",
    "rel" : "alternate",
    "length" : 6,
    "href" : "http://data.example.com/buildings/123",
    "type" : "application/geo+json",
    "title" : "Trierer Strasse 70, 53115 Bonn"
  } ],
  "id" : "id",
  "type" : "Feature",
  "properties" : {
    "elevation" : 0.8008281904610115,
    "wmoId" : "wmoId",
    "observations" : "observations",
    "name" : "name",
    "id" : "id"
  }
}
```

5.6. DAPA

5.6.1. GET /collections/observation/processes

List the available data retrieval patterns.

5.6.1.1. Query Parameters

Name	Description	Required	Default
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply: The accept header will be used to determine the format.	-	null

5.6.1.2. Return Type

[processes]

5.6.1.3. Content Type

- application/json
- text/html

5.6.1.4. Responses

Table 12. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	[processes]
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.6.1.5. Example Data

Content-Type: application/json

```
{
  "endpoints" : [ {
    "inputCollectionId" : "inputCollectionId",
    "mediaTypes" : [ "mediaTypes", "mediaTypes" ],
    "description" : "description",
    "links" : [ {
      "hreflang" : "en",
      "rel" : "alternate",
      "length" : 6,
      "href" : "http://data.example.com/buildings/123",
      "type" : "application/geo+json",
      "title" : "Trierer Strasse 70, 53115 Bonn"
    }, {
      "hreflang" : "en",
      "rel" : "alternate",
      "length" : 6,
      "href" : "http://data.example.com/buildings/123",
      "type" : "application/geo+json",
      "title" : "Trierer Strasse 70, 53115 Bonn"
    } ],
    "externalDocs" : {
      "description" : "description",
      "url" : "url"
    },
    "id" : "id",
    "title" : "title"
  }, {
    "inputCollectionId" : "inputCollectionId",
    "mediaTypes" : [ "mediaTypes", "mediaTypes" ],
    "description" : "description",
    "links" : [ {
      "hreflang" : "en",
      "rel" : "alternate",
      "length" : 6,
      "href" : "http://data.example.com/buildings/123",
      "type" : "application/geo+json",
      "title" : "Trierer Strasse 70, 53115 Bonn"
    }, {
      "hreflang" : "en",
      "rel" : "alternate",
      "length" : 6,
      "href" : "http://data.example.com/buildings/123",
      "type" : "application/geo+json",
      "title" : "Trierer Strasse 70, 53115 Bonn"
    } ],
    "externalDocs" : {
      "description" : "description",
      "url" : "url"
    },
    "id" : "id",
    "title" : "title"
  } ],
  "description" : "description",
  "links" : [ null, null ],
  "title" : "title"
}
```

Content-Type: application/json

```
{"title":"Data retrieval patterns","description":"The following endpoints are
available to retrieve and process the observations in addition to the standard feature
queries. The endpoints are described in the API definition and the links point to the
specification of the operation in the OpenAPI definition with the available input
parameters and the response schema.","endpoints":[{"title":"retrieve a time series for
selected variables for each station in an area and apply functions on the values of
each time step","description":"This DAPA endpoint returns a time series for an area
(parameter `bbox`, `coords` or `coordsRef`) in the selected time interval or at the
selected time instant (parameter `datetime`). \n\nAll values in the area for each
requested variable (parameter `variables`) are aggregated for each time step and each
of the requested statistical functions (parameter `functions`) is applied to the
aggregated values.","links":[{"rel":"ogc-dapa-endpoint","title":"Execute the data
retrieval pattern with the default
parameters","href":"/collections/observation/processes/area:aggregate-
space"},{"rel":"ogc-dapa-endpoint-definition","title":"Detailed description in the API
definition (OpenAPI
3.0)","href":"/api?f=json#/paths/~1collections~1observation~1processes"},{"rel":"ogc-
dapa-endpoint-documentation","title":"Detailed description in the API
documentation","href":"/api?f=html#/DAPA/get_collections_observation_processes"}],"id"
:"area:aggregate-
space","inputCollectionId":"observation","mediaTypes":["application/geo+json","text/cs
v"],"externalDocs":{"description":"Example requests and
responses","url":"https://github.com/cportele/ogc-api-dev/blob/master/dapa/10-area-
agg-space.md"}},{"title":"retrieve a time series for selected variables for each
station in an area and apply functions on all values","description":"This DAPA
endpoint returns observation values for an area (parameter `bbox`, `coords` or
`coordsRef`) in the selected time interval or at the selected time instant (parameter
`datetime`). \n\nAll values for each requested variable (parameter `variables`) are
aggregated and each of the requested statistical functions (parameter `functions`) is
applied to the aggregated values.","links":[{"rel":"ogc-dapa-
endpoint","title":"Execute the data retrieval pattern with the default
parameters","href":"/collections/observation/processes/area:aggregate-space-
time"},{"rel":"ogc-dapa-endpoint-definition","title":"Detailed description in the API
definition (OpenAPI
3.0)","href":"/api?f=json#/paths/~1collections~1observation~1processes"},{"rel":"ogc-
dapa-endpoint-documentation","title":"Detailed description in the API
documentation","href":"/api?f=html#/DAPA/get_collections_observation_processes"}],"id"
:"area:aggregate-space-
time","inputCollectionId":"observation","mediaTypes":["application/geo+json","text/csv
"],"externalDocs":{"description":"Example requests and
responses","url":"https://github.com/cportele/ogc-api-dev/blob/master/dapa/11-area-
agg-space-time.md"}}]}
```

5.6.2. GET /collections/observation/variables

Fetch the observable properties included in this observation collection.

5.6.2.1. Query Parameters

Name	Description	Required	Default	Pattern
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply: The accept header will be used to determine the format.	-	null	

5.6.2.2. Return Type

Variables

5.6.2.3. Content Type

- application/json
- text/html

5.6.2.4. Responses

Table 13. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	Variables
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.6.2.5. Example Data

Content-Type: application/json

```
{
  "variables" : [ {
    "uom" : "uom",
    "description" : "description",
    "id" : "id",
    "title" : "title"
  }, {
    "uom" : "uom",
    "description" : "description",
    "id" : "id",
    "title" : "title"
  } ],
  "description" : "description",
  "links" : [ {
    "hreflang" : "en",
    "rel" : "alternate",
    "length" : 6,
    "href" : "http://data.example.com/buildings/123",
    "type" : "application/geo+json",
    "title" : "Trierer Strasse 70, 53115 Bonn"
  }, {
    "hreflang" : "en",
    "rel" : "alternate",
    "length" : 6,
    "href" : "http://data.example.com/buildings/123",
    "type" : "application/geo+json",
    "title" : "Trierer Strasse 70, 53115 Bonn"
  } ],
  "title" : "title"
}
```

Content-Type: application/json

```
{"variables":[{"id":"PRCP","title":"Precipitation","uom":"0.1
mm"},{"id":"SNOW","title":"Snowfall","uom":"mm"},{"id":"SNWD","title":"Snow
depth","uom":"mm"}]}
```

5.6.2.6. Return Type

Collections

5.6.2.7. Content Type

- application/json
- text/html

5.6.2.8. Responses

Table 14. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	Collections
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.6.2.9. Example Data

Content-Type: application/json

```
{
  "collections" : [ {
    "extent" : {
      "spatial" : {
        "crs" : "crs",
        "bbox" : [ [ 0.8008281904610115, 0.8008281904610115 ], [ 0.8008281904610115,
0.8008281904610115 ] ]
      },
      "temporal" : {
        "trs" : "trs",
        "interval" : [ [ "interval", "interval" ], [ "interval", "interval" ] ]
      }
    },
    "storageCrs" : "storageCrs",
    "storageCrsCoordinateEpoch" : 0.8008281904610115,
    "nativeCrsCoordinateEpoch" : 6.027456183070403,
    "nativeCrs" : "nativeCrs",
    "links" : [ {
      "hreflang" : "en",
      "rel" : "alternate",
      "length" : 6,
      "href" : "http://data.example.com/buildings/123",
      "type" : "application/geo+json",
      "title" : "Trierer Strasse 70, 53115 Bonn"
    }, {
      "hreflang" : "en",
      "rel" : "alternate",
      "length" : 6,
      "href" : "http://data.example.com/buildings/123",
      "type" : "application/geo+json",
      "title" : "Trierer Strasse 70, 53115 Bonn"
    } ],
    "id" : "id",
    "title" : "title"
  }, {
    "extent" : {
      "spatial" : {
        "crs" : "crs",
        "bbox" : [ [ 0.8008281904610115, 0.8008281904610115 ], [ 0.8008281904610115,
0.8008281904610115 ] ]
      },
      "temporal" : {
        "trs" : "trs",
        "interval" : [ [ "interval", "interval" ], [ "interval", "interval" ] ]
      }
    },
    "storageCrs" : "storageCrs",
    "storageCrsCoordinateEpoch" : 0.8008281904610115,
    "nativeCrsCoordinateEpoch" : 6.027456183070403,
    "itemType" : "itemType",
    "crs" : [ "crs", "crs" ],
    "description" : "description",
    "nativeCrs" : "nativeCrs",
    "links" : [ {
      "hreflang" : "en",
      "rel" : "alternate",
      "length" : 6,
      "href" : "http://data.example.com/buildings/123",
      "type" : "application/geo+json",
      "title" : "Trierer Strasse 70, 53115 Bonn"
    }, {
      "hreflang" : "en",
      "rel" : "alternate",
      "length" : 6,
      "href" : "http://data.example.com/buildings/123",
      "type" : "application/geo+json",
      "title" : "Trierer Strasse 70, 53115 Bonn"
    } ],
    "id" : "id",
    "title" : "title"
  } ],
  "crs" : [ "crs", "crs" ],
  "description" : "description",
  "links" : [ null, null ],
  "title" : "title"
}
```

"description" : "description",

5.6.3. GET /collections/observation/processes/area:aggregate-space

Retrieve a time series for selected variables for each station in an area and apply functions on the values of each time step.

5.6.3.1. Description

This DAPA endpoint returns a time series for an area (parameter `bbox`, `coords` or `coordsRef`) in the selected time interval or at the selected time instant (parameter `datetime`). All values in the area for each requested variable (parameter `variables`) are aggregated for each time step and each of the requested statistical functions (parameter `functions`) is applied to the aggregated values.

5.6.3.2. Query Parameters

Name	Description	Required	Default
bbox	Only features that have a geometry that intersects the bounding box are selected. The bounding box is provided as four numbers: • Lower left corner, coordinate axis 1 • Lower left corner, coordinate axis 2 • Upper right corner, coordinate axis 1 • Upper right corner, coordinate axis 2 The CRS of the values is WGS 84 longitude/latitude (http://www.opengis.net/def/crs/OGC/1.3/CRS84). [Double]	-	[2,15,1,3]
coords	A Well Known Text representation of a (MULTI)POLYGON geometry as defined in Simple Feature Access - Part 1: Common Architecture.	-	null
coordsRef	A URI that returns a GeoJSON feature. For a polygon or multi-polygon the geometry is used. For other geometries a buffer is added.	-	null
datetime	Either a local date, a date-time with offsets or a closed interval. Date and time expressions adhere to RFC 3339. Examples: • A date-time: '2018-02-12T23:20:50Z' • A closed interval: '2018-02-12T00:00:00Z/2018-03-18T12:31:12Z' Selects features that have a temporal property that intersects the value of the parameter.	-	2019-08-07/2019-08-11

Name	Description	Required	Default
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply, i.e., the accept header will be used to determine the format.		null
functions	The statistical function(s) to apply when aggregating multiple values to a single value. String	-	["max","mean","min"]
variables	A comma-separated list of values with names of observable properties that should be returned in the response. More information about the available properties can be retrieved from the '../variables' path. The default is to return all observed properties. String	-	["PRCP","SNOW","SNWD","TMAX","TMIN","TAVG"]

5.6.3.3. Return Type

String

5.6.3.4. Content Type

- text/csv
- application/geo+json
- application/json
- text/html

5.6.3.5. Responses

Table 15. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	String
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.6.3.6. Example Data

Content-Type: text/csv

```
"phenomenonTime,longitude,latitude,TMAX,TMIN,PRCP,SNWD\n2019-08-
11,6.9299617,50.000008,206.35806,103.4677,8.454727,0.0\n2019-08-
10,6.9299617,50.000008,229.26971,136.94968,0.19743806,0.0"
```

Content-Type: text/csv

```
phenomenonTime,PRCP_count,PRCP_max,PRCP_mean,PRCP_min,PRCP_std-
dev,TMAX_count,TMAX_max,TMAX_mean,TMAX_min,TMAX_std-
dev,TMIN_count,TMIN_max,TMIN_mean,TMIN_min,TMIN_std-dev
2019-08-
11,709,281.0,26.931225,0.0,45.49806,485,306.0,258.65854,115.0,27.334553,486,187.0,137.
17947,58.0,20.78836
2019-08-
10,709,229.0,12.449949,0.0,26.586481,485,305.0,245.46971,99.0,24.421343,486,206.0,157.
32607,61.0,20.63871
2019-08-
09,709,431.0,55.968887,0.0,56.121117,485,351.0,277.59433,134.0,34.76338,486,188.0,128.
55798,52.0,21.31758
2019-08-
08,709,439.0,5.9656353,0.0,25.032032,485,294.5,248.36009,67.0,23.795979,486,178.0,131.
79442,20.0,16.903353
2019-08-
07,709,490.0,63.96998,0.0,84.83652,485,285.0,225.75258,73.0,26.090208,486,177.53384399
414062,144.37961,27.0,16.778051
```

Content-Type: application/geo+json

```
{"type":"FeatureCollection","features":[{"type":"Feature","geometry":{"type":"Polygon"
,"coordinates":[[[5.8,47.2],[15.1,47.2],[15.1,55.1],[5.8,55.1],[5.8,47.2]]]},"properti
es":{"phenomenonTime":"2019-08-11","TMAX_mean":258.65854,"TMIN_std-
dev":20.78836,"TMIN_max":187,"TMAX_max":306,"PRCP_min":0,"TMIN_count":486,"TMAX_count"
:485,"PRCP_mean":26.931225,"TMAX_min":115,"PRCP_max":281,"PRCP_count":709,"TMIN_min":5
8,"PRCP_std-dev":45.49806,"TMAX_std-
dev":27.334553,"TMIN_mean":137.17947}},{"type":"Feature","geometry":{"type":"Polygon",
"coordinates":[[[5.8,47.2],[15.1,47.2],[15.1,55.1],[5.8,55.1],[5.8,47.2]]]},"propertie
s":{"phenomenonTime":"2019-08-10","TMAX_mean":245.46971,"TMIN_std-
dev":20.63871,"TMIN_max":206,"TMIN_count":486,"TMAX_count":485,"PRCP_mean":12.449949,"
TMAX_max":305,"PRCP_min":0,"TMAX_min":99,"PRCP_max":229,"TMIN_min":61,"PRCP_std-
dev":26.586481,"TMAX_std-
dev":24.421343,"TMIN_mean":157.32607}},{"type":"Feature","geometry":{"type":"Polygon",
"coordinates":[[[5.8,47.2],[15.1,47.2],[15.1,55.1],[5.8,55.1],[5.8,47.2]]]},"propertie
s":{"phenomenonTime":"2019-08-09","TMAX_mean":277.59433,"TMIN_std-
dev":21.31758,"TMIN_max":188,"TMIN_count":486,"TMAX_count":485,"PRCP_mean":55.968887,"
TMAX_max":351,"PRCP_min":0,"PRCP_count":709,"TMIN_min":52,"PRCP_std-
dev":56.121117,"TMAX_min":134,"PRCP_max":431,"TMAX_std-
dev":34.76338,"TMIN_mean":128.55798}},{"type":"Feature","geometry":{"type":"Polygon","
coordinates":[[[5.8,47.2],[15.1,47.2],[15.1,55.1],[5.8,55.1],[5.8,47.2]]]},"properties
":{"phenomenonTime":"2019-08-08","TMAX_mean":248.36009,"TMIN_std-
dev":16.903353,"TMIN_max":178,"TMIN_count":486,"TMAX_count":485,"PRCP_mean":5.9656353,
"TMAX_max":294.5,"PRCP_min":0,"TMAX_std-
dev":23.795979,"TMIN_mean":131.79442,"TMAX_min":67,"PRCP_max":439,"PRCP_count":709,"TM
IN_min":20,"PRCP_std-
dev":25.032032}},{"type":"Feature","geometry":{"type":"Polygon","coordinates":[[[5.8,4
7.2],[15.1,47.2],[15.1,55.1],[5.8,55.1],[5.8,47.2]]]},"properties":{"phenomenonTime":"
2019-08-07","TMAX_mean":225.75258,"TMIN_std-
dev":16.778051,"TMIN_max":177.53384,"TMIN_count":486,"TMAX_count":485,"PRCP_mean":63.9
6998,"TMAX_max":285,"PRCP_min":0,"PRCP_count":709,"TMIN_min":27,"PRCP_std-
dev":84.83652,"TMAX_min":73,"PRCP_max":490,"TMAX_std-
dev":26.090208,"TMIN_mean":144.37961}}]}
```

5.6.4. GET /collections/observation/processes/area:aggregate-space-time

Retrieve a time series for selected variables for each station in an area and apply functions on all values.

5.6.4.1. Description

This DAPA endpoint returns observation values for an area (parameter `bbox`, `coords` or `coordsRef`) in the selected time interval or at the selected time instant (parameter `datetime`). All values for each requested variable (parameter `variables`) are aggregated and each of the requested statistical functions (parameter `functions`) is applied to the aggregated values.

5.6.4.2. Query Parameters

Name	Description	Required	Default
bbox	Only features that have a geometry that intersects the bounding box are selected. The bounding box is provided as four numbers: • Lower left corner, coordinate axis 1 • Lower left corner, coordinate axis 2 • Upper right corner, coordinate axis 1 • Upper right corner, coordinate axis 2 The CRS of the values is WGS 84 longitude/latitude (http://www.opengis.net/def/crs/OGC/1.3/CRS84). Double	-	[6.0,48.5,8.0,50.5]
coords	A Well Known Text representation of a (MULTI)POLYGON geometry as defined in Simple Features Access - Part 1: Common Architecture.	-	null
coordsRef	A URI that returns a GeoJSON feature. For a polygon or multi-polygon the geometry is used. For other geometries a buffer is added.	-	null
datetime	Either a local date, a date-time with offsets or a closed interval. Date and time expressions adhere to RFC 3339. Examples: * A date-time: '2018-02-12T23:20:50Z' * A closed interval: '2018-02-12T00:00:00Z/2018-03-18T12:31:12Z' Selects features that have a temporal property that intersects the value of the parameter.	-	2019-08-07/2019-08-11
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply, i.e., the accept header will be used to determine the format.	-	null
functions	The statistical function(s) to apply when aggregating multiple values to a single value. String	-	["max","mean","min"]
variables	A comma-separated list of values with names of observable properties that should be returned in the response. More information about the available properties can be retrieved from the '../variables' path. The default is to return all observed properties. String	-	["PRCP","SNOW","SNWD", "TMAX","TMIN","TAVG"]

5.6.4.3. Return Type

String

5.6.4.4. Content Type

- text/csv
- application/geo+json
- application/json
- text/html

5.6.4.5. Responses

Table 16. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	String
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.6.4.6. Example Data

Content-Type: application/json

```
"phenomenonTime,longitude,latitude,TMAX,TMIN,PRCP,SNWD\n2019-08-
11,6.9299617,50.000008,206.35806,103.4677,8.454727,0.0\n2019-08-
10,6.9299617,50.000008,229.26971,136.94968,0.19743806,0.0"
```

Content-Type: text/csv

```
PRCP_count,PRCP_max,PRCP_mean,PRCP_min,PRCP_std-
dev,TMAX_count,TMAX_max,TMAX_mean,TMAX_min,TMAX_std-
dev,TMIN_count,TMIN_max,TMIN_mean,TMIN_min,TMIN_std-dev
3545,490.0,33.057137,0.0,57.29267,2425,351.0,251.16705,67.0,32.354607,2430,206.0,139.8
475,20.0,21.919683
```

Content-Type: application/geo+json

```
{"type":"FeatureCollection","features":[{"type":"Feature","geometry":{"type":"Polygon"
,"coordinates":[[[5.8,47.2],[15.1,47.2],[15.1,55.1],[5.8,55.1],[5.8,47.2]]]},"properti
es":{"phenomenonTime":"2019-08-07/2019-08-
11","TMIN_count":2430,"TMAX_count":2425,"PRCP_count":3545,"TMIN_min":20,"TMAX_mean":25
1.16705,"PRCP_std-dev":57.29267,"TMIN_std-
dev":21.919683,"TMIN_max":206,"TMAX_max":351,"PRCP_min":0,"PRCP_mean":33.057137,"TMAX_
min":67,"PRCP_max":490,"TMAX_std-dev":32.354607,"TMIN_mean":139.8475}}]}
```

5.6.5. GET /collections/observation/processes/area:aggregate-time

Retrieve a time series for selected variables for each station in an area and apply functions on the values of each time series.

5.6.5.1. Description

This DAPA endpoint returns a time series for each station in an area (parameter `box`, `coords` or `coordsRef`) in the selected time interval or at the selected time instant (parameter `datetime`). Each time series contains daily observation values for each selected variable (parameter `variables`) for which a value has been observed at the station during the time interval.

5.6.5.2. Query Parameters

Name	Description	Required	Default
bbox	Only features that have a geometry that intersects the bounding box are selected. The bounding box is provided as four numbers: • Lower left corner, coordinate axis 1 • Lower left corner, coordinate axis 2 • Upper right corner, coordinate axis 1 • Upper right corner, coordinate axis 2 The CRS of the values is WGS 84 longitude/latitude (http://www.opengis.net/def/crs/OGC/1.3/CRS84). [Double]	-	[6.0,48.5,8.0,50.5]
coords	A Well Known Text representation of a (MULTI)POLYGON geometry as defined in Simple Feature Access - Part 1: Common Architecture.	-	null
coordsRef	A URI that returns a GeoJSON feature. For a polygon or multi-polygon the geometry is used. For other geometries a buffer is added.	-	null

Name	Description	Required	Default
datetime	Either a local date, a date-time with offsets or a closed interval. Date and time expressions adhere to RFC 3339. Examples: * A date-time: '2018-02-12T23:20:50Z' * A closed interval: '2018-02-12T00:00:00Z/2018-03-18T12:31:12Z' Selects features that have a temporal property that intersects the value of the parameter.	-	2019-08-07/2019-08-11
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply: The accept header will be used to determine the format.	-	null
functions	The statistical function(s) to apply when aggregating multiple values to a single value. String	-	["max","mean","min"]
variables	A comma-separated list of values with names of observable properties that should be returned in the response. More information about the available properties can be retrieved from the '../variables' path. The default is to return all observed properties. String	-	["PRCP","SNOW","SNWD", "TMAX","TMIN","TAVG"]

5.6.5.3. Return Type

String

5.6.5.4. Content Type

- text/csv
- application/geo+json
- application/json
- text/html

5.6.5.5. Responses

Table 17. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	String
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables

Code	Message	Datatype
500	A server error occurred.	exception

5.6.5.6. Example data

Content-Type: application/json

```
"phenomenonTime,longitude,latitude,TMAX,TMIN,PRCP,SNWD\n2019-08-11,6.9299617,50.000008,206.35806,103.4677,8.454727,0.0\n2019-08-10,6.9299617,50.000008,229.26971,136.94968,0.19743806,0.0"
```

Content-Type: text/csv

```
longitude,latitude,locationCode,locationName,PRCP_count,PRCP_max,PRCP_mean,PRCP_min,PRCP_std-dev,TMAX_count,TMAX_max,TMAX_mean,TMAX_min,TMAX_std-dev,TMIN_count,TMIN_max,TMIN_mean,TMIN_min,TMIN_std-dev
11.785599708557129,50.729400634765625,GME00126070,SCHMIERITZ-WELTWITZ,5,59.0,11.8,0.0,26.385603,5,297.0,256.2,204.0,36.16905,5,170.0,144.8,127.0,17.626684
...
6.258900165557861,52.434200286865234,NLE00152479,HEINO,5,53.0,18.8,0.0,26.090227,5,235.0,229.2,221.0,5.6302752,5,170.0,135.0,108.0,26.1247
```

Content-Type: application/geo+json

```
{"type":"FeatureCollection","features":[{"type":"Feature","geometry":{"type":"Point","coordinates":[11.785599708557129,50.729400634765625]},"properties":{"locationCode":"GME00126070","locationName":"SCHMIERITZ-WELTWITZ","phenomenonTime":"2019-08-07/2019-08-11","TMAX_mean":256.2,"TMIN_std-dev":17.626684,"TMIN_max":170,"TMIN_count":5,"TMAX_count":5,"TMAX_max":297,"PRCP_mean":11.8,"PRCP_min":0,"PRCP_count":5,"TMIN_min":127,"PRCP_std-dev":26.385603,"TMAX_min":204,"PRCP_max":59,"TMAX_std-dev":36.16905,"TMIN_mean":144.8}},{"type":"Feature","geometry":{"type":"Point","coordinates":[6.258900165557861,52.434200286865234]},"properties":{"locationCode":"NLE00152479","locationName":"HEINO","phenomenonTime":"2019-08-07/2019-08-11","TMAX_mean":229.2,"TMIN_std-dev":26.1247,"TMIN_max":170,"TMIN_count":5,"TMAX_count":5,"PRCP_mean":18.8,"TMAX_max":235,"PRCP_min":0,"TMAX_min":221,"PRCP_max":53,"PRCP_count":5,"TMIN_min":108,"PRCP_std-dev":26.090227,"TMAX_std-dev":5.6302752,"TMIN_mean":135}}]}
```

5.6.6. GET /collections/observation/processes/area:retrieve

Retrieve a time series for selected variables for each station in an area.

5.6.6.1. Description

This DAPA endpoint returns observation values at the selected location (parameter `coords` or `coordsRef`) in the selected time interval or at the selected time instant (parameter `datetime`). All values in the time interval for each requested variable (parameter `variables`) are aggregated and each of the requested statistical functions (parameter `functions`) is applied to the aggregated values.

5.6.6.2. Query Parameters

Name	Description	Required	Default
bbox	Only features that have a geometry that intersects the bounding box are selected. The bounding box is provided as four numbers: • Lower left corner, coordinate axis 1 • Lower left corner, coordinate axis 2 • Upper right corner, coordinate axis 1 • Upper right corner, coordinate axis 2 The CRS of the values is WGS 84 longitude/latitude (http://www.opengis.net/def/crs/OGC/1.3/CRS84). [Double]	-	[6.0,48.5,8.0,50.5]
coords	A Well Known Text representation of a (MULTI)POLYGON geometry as defined in Simple Feature Access - Part 1: Common Architecture.	-	null
coordsRef	A URI that returns a GeoJSON feature. For a polygon or multi-polygon the geometry is used. For other geometries a buffer is added.	-	null
datetime	Either a local date, a date-time with offsets or a closed interval. Date and time expressions adhere to RFC 3339. Examples: * A date-time: '2018-02-12T23:20:50Z' * A closed interval: '2018-02-12T00:00:00Z/2018-03-18T12:31:12Z' Selects features that have a temporal property that intersects the value of the parameter.	-	2019-08-07/2019-08-11
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply: The accept header will be used to determine the format.	-	null

Name	Description	Required	Default
variables	A comma-separated list of values with names of observable properties that should be returned in the response. More information about the available properties can be retrieved from the '../variables' path. The default is to return all observed properties.	-	["PRCP","SNOW","SNWD", "TMAX","TMIN","TAVG"]

5.6.6.3. Return Type

String

5.6.6.4. Content Type

- text/csv
- application/geo+json
- application/json
- text/html

5.6.6.5. Responses

Table 18. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	String
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.6.6.6. Example Data

Content-Type: text/csv

```
longitude,latitude,locationCode,locationName,phenomenonTime,TMAX,TMIN,PRCP
7.888899803161621,51.57780075073242,GMM00010424,WERL,2019-08-11,244.0,138.0,0.0
7.888899803161621,51.57780075073242,GMM00010424,WERL,2019-08-10,257.0,181.0,0.0
7.888899803161621,51.57780075073242,GMM00010424,WERL,2019-08-09,267.0,136.0,15.0
7.888899803161621,51.57780075073242,GMM00010424,WERL,2019-08-08,257.0,143.0,0.0
7.888899803161621,51.57780075073242,GMM00010424,WERL,2019-08-07,252.0,147.0,0.0
6.535799980163574,51.8307991027832,GME00130294,BOCHOLT,2019-08-11,,,0.0
6.535799980163574,51.8307991027832,GME00130294,BOCHOLT,2019-08-10,,,10.0
6.535799980163574,51.8307991027832,GME00130294,BOCHOLT,2019-08-09,,,64.0
6.535799980163574,51.8307991027832,GME00130294,BOCHOLT,2019-08-08,,,0.0
6.535799980163574,51.8307991027832,GME00130294,BOCHOLT,2019-08-07,,,0.0
...
7.193900108337402,50.73640060424805,GME00130990,BONN-ROLEBER,2019-08-
11,231.0,143.0,0.0
7.193900108337402,50.73640060424805,GME00130990,BONN-ROLEBER,2019-08-
10,256.0,180.0,0.0
7.193900108337402,50.73640060424805,GME00130990,BONN-ROLEBER,2019-08-
09,275.0,157.0,127.0
7.193900108337402,50.73640060424805,GME00130990,BONN-ROLEBER,2019-08-
08,232.0,146.0,0.0
7.193900108337402,50.73640060424805,GME00130990,BONN-ROLEBER,2019-08-
07,243.0,154.0,0.0
7.699999809265137,52.13529968261719,GME00111430,MUENSTER/OSNABRUECK (AIRPORT),2019-08-
11,240.0,166.0,0.0
7.699999809265137,52.13529968261719,GME00111430,MUENSTER/OSNABRUECK (AIRPORT),2019-08-
10,259.0,184.0,0.0
7.699999809265137,52.13529968261719,GME00111430,MUENSTER/OSNABRUECK (AIRPORT),2019-08-
09,265.0,121.0,20.0
7.699999809265137,52.13529968261719,GME00111430,MUENSTER/OSNABRUECK (AIRPORT),2019-08-
08,256.0,125.0,0.0
7.699999809265137,52.13529968261719,GME00111430,MUENSTER/OSNABRUECK (AIRPORT),2019-08-
07,257.0,165.0,0.0
```

Content-Type: application/geo+json

```
{"type":"FeatureCollection","features":[{"type":"Feature","geometry":{"type":"Point","
coordinates":[11.7856,50.7294]},"properties":{"locationCode":"GME00126070","locationNa
me":"SCHMIERITZ-WELTWITZ","phenomenonTime":"2019-08-
09","TMAX":297,"TMIN":127,"PRCP":0}},{"type":"Feature","geometry":{"type":"Point","coo
rdinates":[6.2589,52.4342]},"properties":{"locationCode":"NLE00152479","locationName":
"HEINO","phenomenonTime":"2019-08-09","TMAX":221,"TMIN":108,"PRCP":53}}]}
```

5.6.7. GET /collections/observation/processes/grid:aggregate-time

Retrieve a time series for selected variables for each station in an area, resample the observations to a time series in a 2D grid and apply functions on the values of each time series.

5.6.7.1. Description

This DAPA endpoint retrieves a time series for each station in an area (parameter `box`, `coords` or `coordsRef`) in the selected time interval or at the selected time instant (parameter `datetime`). Each time series contains daily observation values for each selected variable (parameter `variables`) for which a value has been observed at the station during the time interval. The data is then resampled to a time series for each cell in a 2D spatial grid and all values in each time series are aggregated and each of the requested statistical functions (parameter `functions`) is applied to the values.

5.6.7.2. Parameters

5.6.7.2.1. Query Parameters

Name	Description	Required	Default
bbox	Only features that have a geometry that intersects the bounding box are selected. The bounding box is provided as four numbers: • Lower left corner, coordinate axis 1 • Lower left corner, coordinate axis 2 • Upper right corner, coordinate axis 1 • Upper right corner, coordinate axis 2 The CRS of the values is WGS 84 longitude/latitude (http://www.opengis.net/def/crs/OGC/1.3/CRS84). [Double]	-	[6.0,48.5,8.0,50.5]
coordsRef	A URI that returns a GeoJSON feature. For a polygon or multi-polygon the envelope of the geometry is used, for other geometries a buffer is added before determining the envelope.	-	null
datetime	Either a local date, a date-time with offsets or a closed interval. Date and time expressions adhere to RFC 3339. Examples: • A date-time: '2018-02-12T23:20:50Z' • A closed interval: '2018-02-12T00:00:00Z/2018-03-18T12:31:12Z' Selects features that have a temporal property that intersects the value of the parameter.	-	2019-08-07/2019-08-11

Name	Description	Required	Default
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply, i.e., the accept header will be used to determine the format.	-	null
functions	The statistical function(s) to apply when aggregating multiple values to a single value. String	-	["max","mean","min"]
height	The number of grid cells in the vertical direction.	-	null
variables	A comma-separated list of values with names of observable properties that should be returned in the response. More information about the available properties can be retrieved from the '../variables' path. The default is to return all observed properties. String	-	["PRCP","SNOW","SNWD", "TMAX","TMIN","TAVG"]
width	The number of grid cells in the horizontal direction.	-	10

5.6.7.3. Return Type

String

5.6.7.4. Content Type

- text/csv
- image/tiff
- application/geo+json
- application/json
- text/html

5.6.7.5. Responses

Table 19. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	String
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.6.7.6. Example Data

Content-Type: csv

```
"phenomenonTime,longitude,latitude,TMAX,TMIN,PRCP,SNWD\n2019-08-
11,6.9299617,50.000008,206.35806,103.4677,8.454727,0.0\n2019-08-
10,6.9299617,50.000008,229.26971,136.94968,0.19743806,0.0"
```

Content-Type: text/csv

```
longitude,latitude,TMAX_max
5.8,54.17058823529412,302.4419
5.8,53.70588235294118,346.7035
5.8,53.241176470588236,358.9652
5.8,52.7764705882353,386.4441
5.8,52.311764705882354,396.17108
5.8,51.847058823529416,403.03824
5.8,51.38235294117647,395.76236
5.8,50.91764705882353,386.07953
5.8,50.45294117647059,390.37183
5.8,49.98823529411765,392.23697
5.8,49.523529411764706,393.08337
5.8,49.05882352941177,394.32608
5.8,48.59411764705882,396.6705
5.8,48.129411764705885,391.1083
5.8,47.66470588235295,384.11874
...
```

Content-Type: application/geo+json

```
{"type":"FeatureCollection","features":[{"type":"Feature","geometry":{"type":"Point","
coordinates":[5.8,54.17058823529412]},"properties":{"phenomenonTime":"2019-01-01/2019-
12-
31","TMAX_max":302.4419}},{"type":"Feature","geometry":{"type":"Point","coordinates":[
14.635000000000002,47.66470588235295]},"properties":{"phenomenonTime":"2019-01-
01/2019-12-31","TMAX_max":346.3743}}]}
```

5.6.8. GET /collections/observation/processes/grid:retrieve

Retrieve a time series for selected variables for each station in an area and resample the observations to a time series for each cell in a 2D grid.

5.6.8.1. Description

This DAPA endpoint retrieves a time series for each station in an area (parameter `box`, `coords` or `coordsRef`) in the selected time interval or at the selected time instant (parameter `datetime`). Each time series contains daily observation values for each selected variable (parameter `variables`) for which a value has been observed at the station during the time interval. The data is then resampled to a time series for each cell in a 2D spatial grid.

5.6.8.2. Query Parameters

Name	Description	Required	Default
bbox	Only features that have a geometry that intersects the bounding box are selected. The bounding box is provided as four numbers: * Lower left corner, coordinate axis 1 * Lower left corner, coordinate axis 2 * Upper right corner, coordinate axis 1 * Upper right corner, coordinate axis 2 The CRS of the values is WGS 84 longitude/latitude (http://www.opengis.net/def/crs/OGC/1.3/CRS84). Double	-	[6.0,48.5,8.0,50.5]
coordsRef	A URI that returns a GeoJSON feature. For a polygon or multi-polygon the envelope of the geometry is used, for other geometries a buffer is added before determining the envelope.	-	null
datetime	Either a local date, a date-time with offsets or a closed interval. Date and time expressions adhere to RFC 3339. Examples: * A date-time: '2018-02-12T23:20:50Z' * A closed interval: '2018-02-12T00:00:00Z/2018-03-18T12:31:12Z' Selects features that have a temporal property that intersects the value of the parameter.	-	2019-08-07/2019-08-11
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply, i.e., the accept header will be used to determine the format.	-	null
height	The number of grid cells in the vertical direction.	-	null
variables	A comma-separated list of values with names of observable properties that should be returned in the response. More information about the available properties can be retrieved from the '../variables' path. The default is to return all observed properties. String	-	["PRCP","SNOW","SNWD", "TMAX","TMIN","TAVG"]
width	The number of grid cells in the horizontal direction.	-	10

5.6.8.3. Return Type

String

5.6.8.4. Content Type

- text/csv

- image/tiff
- application/geo+json
- application/json
- text/html

5.6.8.5. Responses

Table 20. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	String
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.6.8.6. Example Data

Content-Type: application/csv

```
"phenomenonTime,longitude,latitude,TMAX,TMIN,PRCP,SNWD\n2019-08-
11,6.9299617,50.000008,206.35806,103.4677,8.454727,0.0\n2019-08-
10,6.9299617,50.000008,229.26971,136.94968,0.19743806,0.0"
```

5.6.9. GET /collections/observation/processes/position:aggregate-time

Retrieve a time series for selected variables at a position.

5.6.9.1. Description

This DAPA endpoint returns observation values at the selected location (parameter `coords` or `coordsRef`) in the selected time interval or at the selected time instant (parameter `datetime`). All values in the time interval for each requested variable (parameter `variables`) are aggregated and each of the requested statistical functions (parameter `functions`) is applied to the aggregated values.

5.6.9.2. Query Parameters

Name	Description	Required	Default
coords	A Well Known Text representation of a POINT geometry as defined in Simple Feature Access - Part 1: Common Architecture.	-	POINT(7.0218 49.9174)

Name	Description	Required	Default
coordsRef	A URI that returns a GeoJSON feature. The centroid of the feature geometry is used as the sampling geometry.	-	null
datetime	Either a local date, a date-time with offsets or a closed interval. Date and time expressions adhere to RFC 3339. Examples: • A date-time: '2018-02-12T23:20:50Z' • A closed interval: '2018-02-12T00:00:00Z/2018-03-18T12:31:12Z' Selects features that have a temporal property that intersects the value of the parameter.	-	2019-08-07/2019-08-11
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply, i.e., the accept header will be used to determine the format.	-	null
functions	The statistical function(s) to apply when aggregating multiple values to a single value. String	-	["max","mean","min"]
variables	A comma-separated list of values with names of observable properties that should be returned in the response. More information about the available properties can be retrieved from the '../variables' path. The default is to return all observed properties. String	-	["PRCP","SNOW","SNWD", "TMAX","TMIN","TAVG"]

5.6.9.3. Return Type

String

5.6.9.4. Content Type

- text/csv
- application/geo+json
- application/json
- text/html

5.6.9.5. Responses

Table 21. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	String

Code	Message	Datatype
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.6.9.6. Example Data

Content-Type: text/csv

```
"phenomenonTime,longitude,latitude,TMAX,TMIN,PRCP,SNWD\n2019-08-
11,6.9299617,50.000008,206.35806,103.4677,8.454727,0.0\n2019-08-
10,6.9299617,50.000008,229.26971,136.94968,0.19743806,0.0"
```

Content-Type: text/csv

```
PRCP_count,PRCP_max,PRCP_mean,PRCP_min,PRCP_std-
dev,TMAX_count,TMAX_max,TMAX_mean,TMAX_min,TMAX_std-
dev,TMIN_count,TMIN_max,TMIN_mean,TMIN_min,TMIN_std-dev
5,175.7282,49.449417,0.1784924,74.594795,5,261.97256,230.02454,206.25876,23.499502,5,1
48.28212,132.24287,107.17521,17.294535
```

Content-Type: application/geo+json

```
{"type":"FeatureCollection","features":[{"type":"Feature","geometry":{"type":"Point","
coordinates":[7.0218,49.9174]},"properties":{"phenomenonTime":"2019-08-07/2019-08-
11","TMAX_mean":230.02454,"TMIN_std-
dev":17.294535,"TMIN_max":148.28212,"TMIN_count":5,"TMAX_count":5,"PRCP_mean":49.44941
7,"TMAX_max":261.97256,"PRCP_min":0.1784924,"TMAX_min":206.25876,"PRCP_count":5,"TMAX_
std-dev":23.499502,"TMIN_min":107.17521,"PRCP_std-
dev":74.594795,"TMIN_mean":132.24287}}]}
```

5.6.10. GET /collections/observation/processes/position:retrieve

Retrieve a time series for selected variables at a position.

5.6.10.1. Description

This DAPA endpoint returns a time series with daily observation values at the selected location (parameter `coords` or `coordsRef`) in the selected time interval or at the selected time instant (parameter `datetime`). The time series contains values for each selected variable (parameter `variables`) for which a value can be interpolated at the location.

5.6.10.2. Query Parameters

Name	Description	Required	Default
coords	A Well Known Text representation of a POINT geometry as defined in Simple Feature Access - Part 1: Common Architecture.	-	POINT(7.0218 49.9174)
coordsRef	A URI that returns a GeoJSON feature. The centroid of the feature geometry is used as the sampling geometry.	-	null
datetime	Either a local date, a date-time with offsets or a closed interval. Date and time expressions adhere to RFC 3339. Examples: • A date-time: '2018-02-12T23:20:50Z' • A closed interval: '2018-02-12T00:00:00Z/2018-03-18T12:31:12Z' Selects features that have a temporal property that intersects the value of the parameter.	-	2019-08-07/2019-08-11
f	Select the output format of the response. If no value is provided, the standard HTTP rules apply: The accept header will be used to determine the format.	-	null
variables	A comma-separated list of values with names of observable properties that should be returned in the response. More information about the available properties can be retrieved from the '../variables' path. The default is to return all observed properties. String	-	["PRCP","SNOW","SNWD","TMAX","TMIN","TAVG"]

5.6.10.3. Return Type

String

5.6.10.4. Content Type

- text/csv
- application/geo+json
- application/json
- text/html

5.6.10.5. Responses

Table 22. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	String

Code	Message	Datatype
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the `Accept` header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

5.6.10.6. Example Data

Content-Type: application/json

```
"phenomenonTime,longitude,latitude,TMAX,TMIN,PRCP,SNWD\n2019-08-
11,6.9299617,50.000008,206.35806,103.4677,8.454727,0.0\n2019-08-
10,6.9299617,50.000008,229.26971,136.94968,0.19743806,0.0"
```

Content-Type: text/csv

```
phenomenonTime,TMAX,TMIN,PRCP
2019-08-11,208.7299,107.17521,12.385428
2019-08-10,230.10042,146.47885,0.1784924
2019-08-09,261.97256,148.28212,175.7282
2019-08-08,243.06105,122.76392,0.21927235
2019-08-07,206.25876,136.51428,58.735703
```

Content-Type: application/geo+json

```
{"type":"FeatureCollection","features":[{"type":"Feature","geometry":{"type":"Point","
coordinates":[7.0218,49.9174]},"properties":{"phenomenonTime":"2019-08-
11","TMAX":208.7299,"TMIN":107.17521,"PRCP":12.385428}},{"type":"Feature","geometry":{
"type":"Point","coordinates":[7.0218,49.9174]},"properties":{"phenomenonTime":"2019-
08-
10","TMAX":230.10042,"TMIN":146.47885,"PRCP":0.1784924}},{"type":"Feature","geometry":
{"type":"Point","coordinates":[7.0218,49.9174]},"properties":{"phenomenonTime":"2019-
08-
09","TMIN":148.28212,"PRCP":175.7282,"TMAX":261.97256}},{"type":"Feature","geometry":{
"type":"Point","coordinates":[7.0218,49.9174]},"properties":{"phenomenonTime":"2019-
08-
08","TMAX":243.06105,"TMIN":122.76392,"PRCP":0.21927235}},{"type":"Feature","geometry"
:{"type":"Point","coordinates":[7.0218,49.9174]},"properties":{"phenomenonTime":"2019-
08-07","TMAX":206.25876,"TMIN":136.51428,"PRCP":58.735703}}]}
```

5.7. Models

5.7.1. LandingPage

Field Name	Required	Type	Description	Format
extent		extent		
externalDocs		externalDocs		
title		String		
links	X	List of link		
description		String		

5.7.2. extent

Field Name	Required	Type	Description	Format
spatial		extent_spatial		
temporal		extent_temporal		

5.7.3. extent_spatial

Field Name	Required	Type	Description	Format
crs	X	string		
bbox		List of number		

5.7.4. extent_temporal

Field Name	Required	Type	Description	Format
interval	X	List of string		
trs	X	string		

5.7.5. externalDocs

Field Name	Required	Type	Description	Format
url	X	string		
description		string		

5.7.6. ConformanceDeclaration

Field Name	Required	Type	Description	Format
conformsTo		List of String		
title		String		
links		List of link		
description		String		

5.7.7. Collection

Field Name	Required	Type	Description	Format
nativeCrs		String		
extent		extent		
crs		List of String		
storageCrs		String		
itemType		String		
storageCrsCoordinateEpoch		BigDecimal		
nativeCrsCoordinateEpoch		BigDecimal		
id	X	String		

Field Name	Required	Type	Description	Format
title		String		
links		List of link		
description		String		

5.7.8. Collections

Field Name	Required	Type	Description	Format
collections		List of Collections_collections		
crs		List of String		
title		String		
links		List of link		
description		String		

5.7.9. Collections_collections

Field Name	Required	Type	Description	Format
nativeCrs		String		
extent		extent		
crs		List of String		
storageCrs		String		
itemType		String		
storageCrsCoordinateEpoch		BigDecimal		
nativeCrsCoordinateEpoch		BigDecimal		
id	X	String		
title		String		
links		List of link		
description		String		

5.7.10. featureCollectionGeoJson_observation

Field Name	Required	Type	Description	Format
type	X	String		*Enum:* FeatureCollection,
features	X	List of Observations		
links		List of link		
timeStamp		Date	This property indicates the time and date when the response was generated.	date-time
numberMatched		Integer	The number of features of the feature type that match the selection parameters like 'bbox'.	
numberReturned		Integer	The number of features in the feature collection. A server may omit this information in a response, if the information about the number of features is not known or difficult to compute. If the value is provided, the value shall be identical to the number of items in the \"features\" array.	

5.7.11. Observations

Daily observations from land-based stations worldwide. The archive includes the following meteorological elements: Daily maximum temperature, daily minimum temperature, temperature at the time of observation, precipitation (i.e., rain, melted snow), snowfall, snow depth, other elements where available.

Field Name	Required	Type	Description	Format
type	X	String		*Enum:* Feature,
geometry	X	pointGeoJSON		
properties	X	featureGeoJson_observation_properties		

Field Name	Required	Type	Description	Format
id		String		
links		List of link		

5.7.12. featureGeoJson_observation_properties

Field Name	Required	Type	Description	Format
id		Integer		int32
phenomenonTime		Date		date-time
observedProperty		String		
unitOfMeasurement		String		
result		Integer		int32
locationCode		String		
locationName		String		
locationLink		String		
geometry		pointGeoJSON		

5.7.13. pointGeoJSON

Field Name	Required	Type	Description	Format
type	X	String		*Enum:* Point,
coordinates	X	List of number		

5.7.14. featureCollectionGeoJson_station

Field Name	Required	Type	Description	Format
type	X	String		*Enum:* FeatureCollection,
features	X	List of Stations		
links		List of link		
timeStamp		Date	This property indicates the time and date when the response was generated.	date-time
numberMatched		Integer	The number of features of the feature type that match the selection parameters like 'bbox'.	

Field Name	Required	Type	Description	Format
numberReturned		Integer	The number of features in the feature collection. A server may omit this information in a response, if the information about the number of features is not known or difficult to compute. If the value is provided, the value shall be identical to the number of items in the \"features\" array.	

5.7.15. Stations

The land-based stations contributing to the GHCN-Daily observation dataset.

Field Name	Required	Type	Description	Format
type	X	String		*Enum:* Feature,
geometry	X	pointGeoJSON		
properties	X	featureGeoJson_station_properties		
id		String		
links		List of link		

5.7.16. featureGeoJson_station_properties

Field Name	Required	Type	Description	Format
id		String		
name		String		
elevation		BigDecimal		
wmoId		String		
observations		String		
geometry		pointGeoJSON		

5.7.17. Queryables

Field Name	Required	Type	Description	Format
queryables		List of Queryables_queryables		

Field Name	Required	Type	Description	Format
title		String		
links		List of link		
description		String		

5.7.18. Queryables_queryables

Field Name	Required	Type	Description	Format
title		String		
mediaTypes		List of String		
id	X	String		
type	X	String		
language		String		
min		Object		
max		Object		
pattern		String		
values		List of String		
description		String		
required		Boolean		

5.7.19. Variables

Field Name	Required	Type	Description	Format
variables		List of Variables_variables		
title		String		
links		List of link		
description		String		

5.7.20. Variables_variables

Field Name	Required	Type	Description	Format
title		String		
uom		String		
id	X	String		
description		String		

5.7.21. link

Field Name	Required	Type	Description	Format
href	X	String		
rel		String		
type		String		
hreflang		String		
title		String		
length		Integer		

5.7.22. exception

Information about the exception: an error code plus an optional description.

Field Name	Required	Type	Description	Format
code	X	String		
description		String		

Chapter 6. DAPA Design 1 for DGGS

This section provides the details of the API implementation implemented in the OGC Testbed 16 Discrete Global Grid System (DGGS) thread. Only the data access API related functionality will be discussed.

6.1. DAPA

6.1.1. GET /collections/{collectionId}/processes

List the available data retrieval patterns.

6.1.1.1. Parameters

6.1.1.1.1. Path Parameters

Name	Description	Required	Default
collectionId	local identifier of a collection	X	null

6.1.1.1.2. Query Parameters

Name	Description	Required	Default
f	The format of the response. If no value is provided, the standard HTTP rules apply: The HTTP Accept header shall be used to determine the format.	-	null

6.1.1.2. Return Type

dapa

6.1.1.3. Content Type

- application/json
- text/html

6.1.1.4. Responses

Table 23. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	dapa
400	A query parameter has an invalid value.	exception

Code	Message	Datatype
406	Content negotiation failed. For example, the 'Accept' header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

6.1.2. GET /collections/{collectionId}/variables

Fetch the observable properties included in this observation collection.

6.1.2.1. Parameters

6.1.2.1.1. Path Parameters

Name	Description	Required	Default
collectionId	local identifier of a collection	X	null

6.1.2.1.2. Query Parameters

Name	Description	Required	Default
f	The format of the response. If no value is provided, the standard HTTP rules apply: The HTTP Accept header shall be used to determine the format.	-	null

6.1.2.2. Return Type

Variables

6.1.2.3. Content Type

- application/json
- text/html

6.1.2.4. Responses

Table 24. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	Variables
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the 'Accept' header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

6.1.3. GET /collections/{collectionId}/processes/area:aggregate-space

Retrieve a time series for selected variables for each station in an area and apply functions on the values of each time step.

6.1.3.1. Description

This DAPA endpoint returns a time series for an area (parameter `bbox`, `coords` or `coordsRef`) in the selected time interval or at the selected time instant (parameter `datetime`). All values in the area for each requested variable (parameter `variables`) are aggregated for each time step. Finally, each of the requested statistical functions (parameter `functions`) are applied to the aggregated values.

6.1.3.2. Parameters

6.1.3.2.1. Path Parameters

Name	Description	Required	Default
collectionId	local identifier of a collection	X	null

6.1.3.2.2. Query Parameters

Name	Description	Required	Default
bbox	Only features that have a geometry that intersects the bounding box are selected. The bounding box is provided as four or six numbers, depending on whether the CRS includes a vertical axis (height or depth): • Lower left corner, coordinate axis 1 • Lower left corner, coordinate axis 2 • Minimum value, coordinate axis 3 (optional) • Upper right corner, coordinate axis 1 • Upper right corner, coordinate axis 2 • Maximum value, coordinate axis 3 (optional) The CRS of the values is WGS 84 in longitude/latitude order (CRS84) unless a different coordinate reference system is specified in the parameter `bbox-crs`. For WGS 84, the longitude/latitude values are - for most cases - in the sequence of minimum longitude, minimum latitude, maximum longitude and maximum latitude. However, in cases where the box spans the antimeridian the first value (west-most box edge) is larger than the third value (east-most box edge). If the vertical axis is included, the third and the sixth number are the bottom and the top of the 3-dimensional bounding box. If a feature has multiple spatial geometry properties, it is the decision of the server whether only a single spatial geometry property is used to determine the extent or all relevant geometries. Big Decimal	-	null
geom	A Well Known Text representation of a (MULTI)POLYGON geometry as defined in Simple Feature Access - Part 1: Common Architecture.	-	null
zones	Comma separated list of zone identifiers String	-	null

Name	Description	Required	Default
datetime	Either a date-time or an interval, open or closed. Date and time expressions adhere to RFC 3339. Open intervals are expressed using double-dots. Examples: • A date-time: \"2018-02-12T23:20:50Z\" • A closed interval: \"2018-02-12T00:00:00Z/2018-03-18T12:31:12Z\" • Open intervals: \"2018-02-12T00:00:00Z/..\" or \"../2018-03-18T12:31:12Z\" Only features that have a temporal property that intersects the value of 'datetime' are selected. If a feature has multiple temporal properties, it is the decision of the server whether only a single temporal property is used to determine the extent or all relevant temporal properties.	-	null
f	The format of the response. If no value is provided, the standard HTTP rules apply: The HTTP Accept header shall be used to determine the format.	-	null
functions	The statistical function(s) to apply when aggregating multiple values to a single value. String	-	["max","mean","min"]
variables	A comma-separated list of values with names of observable properties that should be returned in the response. More information about the available properties can be retrieved from the '../variables' path. The default is to return all observed properties. String	-	null

6.1.3.3. Return Type

String

6.1.3.4. Content Type

- text/csv
- application/geo+json
- application/json
- text/html

6.1.3.5. Responses

Table 25. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	String
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the 'Accept' header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

6.1.3.6. Example Request

```
https://tb16.geo-solutions.it/geoserver/ogc/dggs/collections/dggs:s2-
h3/dapa/area:aggregate-space?geom=Polygon%20%28%28148.80385517012322794%20-
35.30225535688022376%2C%20149.12542760302881106%20-
35.12379107474414752%2C%20149.2062416175809858%20-
35.21470684111535832%2C%20149.39817490214244344%20-
35.31067348339608003%2C%20149.37123689729170906%20-
35.34771324006582915%2C%20149.22981237182537484%20-
35.34097873885314556%2C%20149.14394748136368207%20-
35.45714888477191096%2C%20149.16078373439538041%20-
35.49923951735117811%2C%20149.14731473197002742%20-
35.5901552837223818%2C%20149.08502059575272369%20-
35.58510440781287087%2C%20149.11364222590660233%20-
35.69454005251895978%2C%20149.096805972874904%20-
35.84774995510747431%2C%20149.04966446438612593%20-
35.92688034435649058%2C%20148.96548319922760584%20-
35.89489146359625238%2C%20148.89813818710078408%20-
35.85280083101698523%2C%20148.87456743285639504%20-
35.81407744904406343%2C%20148.8897200605849207%20-
35.74673243691724167%2C%20148.85268030391517868%20-
35.76020143934260886%2C%20148.7870189170915296%20-
35.70969268024749255%2C%20148.76344816284714057%20-
35.65076579463652706%2C%20148.76176453754396789%20-
35.47735238840996175%2C%20148.76176453754396789%20-
35.47735238840996175%2C%20148.76176453754396789%20-
35.47735238840996175%2C%20148.80385517012322794%20-
35.30225535688022376%29%29&datetime=2019/2020&f=json&resolution=6&functions=min,max,co
unt
```

6.1.4. GET /collections/{collectionId}/processes/area:aggregate-space-time

Retrieve a time series for selected variables for each zone in an area and apply functions on all values.

6.1.4.1. Description

This DAPA endpoint returns observation values for an area (parameter `bbox`, `coord` or `coordRef`) in the selected time interval or at the selected time instant (parameter `datetime`). All values for each requested variable (parameter `variables`) are aggregated and each of the requested statistical functions (parameter `functions`) is applied to the aggregated values.

6.1.4.2. Parameters

6.1.4.3. Path Parameters

Name	Description	Required	Default
collectionId	local identifier of a collection	X	null

6.1.4.3.1. Query Parameters

Name	Description	Required	Default
bbox	Only features that have a geometry that intersects the bounding box are selected. The bounding box is provided as four or six numbers, depending on whether the CRS includes a vertical axis (height or depth): • Lower left corner, coordinate axis 1 • Lower left corner, coordinate axis 2 • Minimum value, coordinate axis 3 (optional) • Upper right corner, coordinate axis 1 • Upper right corner, coordinate axis 2 • Maximum value, coordinate axis 3 (optional) The CRS of the values is WGS 84 in longitude/latitude order (CRS84) unless a different coordinate reference system is specified in the parameter `bbox-crs`. For WGS 84 longitude/latitude the values are in most cases the sequence of minimum longitude, minimum latitude, maximum longitude and maximum latitude. However, in cases where the box spans the antimeridian the first value (west-most box edge) is larger than the third value (east-most box edge). If the vertical axis is included, the third and the sixth number are the bottom and the top of the 3-dimensional bounding box. If a feature has multiple spatial geometry properties, it is the decision of the server whether only a single spatial geometry property is used to determine the extent or all relevant geometries.	-	null
geom	A Well Known Text representation of a (MULTI)POLYGON geometry as defined in Simple Feature Access - Part 1: Common Architecture.	-	null
zones	Comma separated list of zone identifiers String	-	null

Name	Description	Required	Default
resolution		X	null
datetime	Either a date-time or an interval, open or closed. Date and time expressions adhere to RFC 3339. Open intervals are expressed using double-dots. Examples: • A date-time: \"2018-02-12T23:20:50Z\" • A closed interval: \"2018-02-12T00:00:00Z/2018-03-18T12:31:12Z\" • Open intervals: \"2018-02-12T00:00:00Z/..\" or \"../2018-03-18T12:31:12Z\" Only features that have a temporal property that intersects the value of 'datetime' are selected. If a feature has multiple temporal properties, it is the decision of the server whether only a single temporal property is used to determine the extent or all relevant temporal properties.	-	null
f	The format of the response. If no value is provided, the standard HTTP rules apply: The HTTP Accept header shall be used to determine the format.	-	null
functions	The statistical function(s) to apply when aggregating multiple values to a single value. String	-	["max","mean","min"]
variables	A comma-separated list of values with names of observable properties that should be returned in the response. More information about the available properties can be retrieved from the '../variables' path. The default is to return all observed properties. String	-	null

6.1.4.4. Return Type

String

6.1.4.5. Content Type

- text/csv
- application/geo+json
- application/json
- text/html

6.1.4.6. Responses

Table 26. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	String
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the 'Accept' header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

6.1.5. GET /collections/{collectionId}/processes/area:aggregate-time

Retrieve a time series for selected variables for each station in an area and apply functions on the values of each time series.

6.1.5.1. Description

This DAPA endpoint returns a time series for each station in an area (parameter `box`, `coords` or `coordsRef`) in the selected time interval or at the selected time instant (parameter `datetime`). Each time series contains daily observation values for each selected variable (parameter `variables`) for which a value has been observed at the station during the time interval.

6.1.5.2. Parameters

6.1.5.2.1. Path Parameters

Name	Description	Required	Default
collectionId	local identifier of a collection	X	null

6.1.5.2.2. Query Parameters

Name	Description	Required	Default
bbox	Only features that have a geometry that intersects the bounding box are selected. The bounding box is provided as four or six numbers, depending on whether the CRS includes a vertical axis (height or depth): • Lower left corner, coordinate axis 1 • Lower left corner, coordinate axis 2 • Minimum value, coordinate axis 3 (optional) • Upper right corner, coordinate axis 1 • Upper right corner, coordinate axis 2 • Maximum value, coordinate axis 3 (optional) The CRS of the values is WGS 84 in longitude/latitude order (CRS84) unless a different coordinate reference system is specified in the parameter `bbox-crs`. For WGS 84 longitude/latitude the values are in most cases the sequence of minimum longitude, minimum latitude, maximum longitude and maximum latitude. However, in cases where the box spans the antimeridian the first value (west-most box edge) is larger than the third value (east-most box edge). If the vertical axis is included, the third and the sixth number are the bottom and the top of the 3-dimensional bounding box. If a feature has multiple spatial geometry properties, it is the decision of the server whether only a single spatial geometry property is used to determine the extent or all relevant geometries. Big Decimal	-	null
geom	A Well Known Text representation of a (MULTI)POLYGON geometry as defined in Simple Feature Access - Part 1: Common Architecture.	-	null
zones	Comma separated list of zone identifiers String	-	null

Name	Description	Required	Default
datetime	Either a date-time or an interval, open or closed. Date and time expressions adhere to RFC 3339. Open intervals are expressed using double-dots. Examples: • A date-time: \"2018-02-12T23:20:50Z\" • A closed interval: \"2018-02-12T00:00:00Z/2018-03-18T12:31:12Z\" • Open intervals: \"2018-02-12T00:00:00Z/..\" or \"../2018-03-18T12:31:12Z\" Only features that have a temporal property that intersects the value of 'datetime' are selected. If a feature has multiple temporal properties, it is the decision of the server whether only a single temporal property is used to determine the extent or all relevant temporal properties.	-	null
f	The format of the response. If no value is provided, the standard HTTP rules apply, i.e., the HTTP Accept header shall be used to determine the format.	-	null
functions	The statistical function(s) to apply when aggregating multiple values to a single value. String	-	["max","mean","min"]
variables	A comma-separated list of values with names of observable properties that should be returned in the response. More information about the available properties can be retrieved from the '../variables' path. The default is to return all observed properties. String	-	null

6.1.5.3. Return Type

String

6.1.5.4. Content Type

- text/csv
- application/geo+json
- application/json
- text/html

6.1.5.5. Responses

Table 27. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	String
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the 'Accept' header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

6.1.6. GET /collections/{collectionId}/processes/area:retrieve

Retrieve a time series for selected variables for each station in an area.

6.1.6.1. Description

This DAPA endpoint returns observation values at the selected location (parameter `coord` or `coordRef`) in the selected time interval or at the selected time instant (parameter `datetime`). All values in the time interval for each requested variable (parameter `variables`) are aggregated and each of the requested statistical functions (parameter `functions`) is applied to the aggregated values.

6.1.6.2. Parameters

6.1.6.2.1. Path Parameters

Name	Description	Required	Default
collectionId	local identifier of a collection	X	null

6.1.6.2.2. Query Parameters

Name	Description	Required	Default
bbox	Only features that have a geometry that intersects the bounding box are selected. The bounding box is provided as four or six numbers, depending on whether the CRS includes a vertical axis (height or depth): • Lower left corner, coordinate axis 1 • Lower left corner, coordinate axis 2 • Minimum value, coordinate axis 3 (optional) • Upper right corner, coordinate axis 1 • Upper right corner, coordinate axis 2 • Maximum value, coordinate axis 3 (optional) The CRS of the values is WGS 84 in longitude/latitude order (CRS84) unless a different coordinate reference system is specified in the parameter `bbox-crs`. For WGS 84 longitude/latitude the values are in most cases the sequence of minimum longitude, minimum latitude, maximum longitude and maximum latitude. However, in cases where the box spans the antimeridian the first value (west-most box edge) is larger than the third value (east-most box edge). If the vertical axis is included, the third and the sixth number are the bottom and the top of the 3-dimensional bounding box. If a feature has multiple spatial geometry properties, it is the decision of the server whether only a single spatial geometry property is used to determine the extent or all relevant geometries.	-	null
geom	A Well Known Text representation of a (MULTI)POLYGON geometry as defined in Simple Feature Access - Part 1: Common Architecture.	-	null
zones	Comma separated list of zone identifiers String	-	null

Name	Description	Required	Default
datetime	Either a date-time or an interval, open or closed. Date and time expressions adhere to RFC 3339. Open intervals are expressed using double-dots. Examples: • A date-time: \"2018-02-12T23:20:50Z\" • A closed interval: \"2018-02-12T00:00:00Z/2018-03-18T12:31:12Z\" • Open intervals: \"2018-02-12T00:00:00Z/..\" or \"../2018-03-18T12:31:12Z\" Only features that have a temporal property that intersects the value of 'datetime' are selected. If a feature has multiple temporal properties, it is the decision of the server whether only a single temporal property is used to determine the extent or all relevant temporal properties.	-	null
f	The format of the response. If no value is provided, the standard HTTP rules apply: The HTTP Accept header shall be used to determine the format.	-	null
variables	A comma-separated list of values with names of observable properties that should be returned in the response. More information about the available properties can be retrieved from the '../variables' path. The default is to return all observed properties. String	-	null

6.1.6.3. Return Type

String

6.1.6.4. Content Type

- text/csv
- application/geo+json
- application/json
- text/html

6.1.6.5. Responses

Table 28. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	String
400	A query parameter has an invalid value.	exception

Code	Message	Datatype
406	Content negotiation failed. For example, the 'Accept' header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

6.1.7. GET /collections/{collectionId}/processes/position:aggregate-time

Retrieve a time series for selected variables at a position.

6.1.7.1. Description

This DAPA endpoint returns observation values at the selected location (parameter `coords` or `coordsRef`) in the selected time interval or at the selected time instant (parameter `datetime`). All values in the time interval for each requested variable (parameter `variables`) are aggregated and each of the requested statistical functions (parameter `functions`) is applied to the aggregated values.

6.1.7.2. Parameters

6.1.7.2.1. Path Parameters

Name	Description	Required	Default
collectionId	local identifier of a collection	X	null

6.1.7.2.2. Query Parameters

Name	Description	Required	Default
geom	A Well Known Text representation of a (MULTI)POLYGON geometry as defined in Simple Feature Access - Part 1: Common Architecture.	-	null

Name	Description	Required	Default
datetime	Either a date-time or an interval, open or closed. Date and time expressions adhere to RFC 3339. Open intervals are expressed using double-dots. Examples: • A date-time: \"2018-02-12T23:20:50Z\" • A closed interval: \"2018-02-12T00:00:00Z/2018-03-18T12:31:12Z\" • Open intervals: \"2018-02-12T00:00:00Z/..\" or \"../2018-03-18T12:31:12Z\" Only features that have a temporal property that intersects the value of 'datetime' are selected. If a feature has multiple temporal properties, it is the decision of the server whether only a single temporal property is used to determine the extent or all relevant temporal properties.	-	null
f	The format of the response. If no value is provided, the standard HTTP rules apply, i.e., the HTTP Accept header shall be used to determine the format.	-	null
functions	The statistical function(s) to apply when aggregating multiple values to a single value. String	-	["max","mean","min"]
variables	A comma-separated list of values with names of observable properties that should be returned in the response. More information about the available properties can be retrieved from the '../variables' path. The default is to return all observed properties. String	-	null

6.1.7.3. Return Type

String

6.1.7.4. Content Type

- text/csv
- application/geo+json
- application/json
- text/html

6.1.7.5. Responses

Table 29. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	String
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the 'Accept' header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

6.1.8. GET /collections/{collectionId}/processes/position:retrieve

Retrieve a time series for selected variables at a position.

6.1.8.1. Description

This DAPA endpoint returns a time series with daily observation values at the selected location (parameter `coords` or `coordsRef`) in the selected time interval or at the selected time instant (parameter `datetime`). The time series contains values for each selected variable (parameter `variables`) for which a value can be interpolated at the location.

6.1.8.2. Parameters

6.1.8.2.1. Path Parameters

Name	Description	Required	Default	collectionId

6.1.8.2.2. Query Parameters

Name	Description	Required	Default
geom	A Well Known Text representation of a (MULTI)POLYGON geometry as defined in Simple Feature Access - Part 1: Common Architecture.	-	null
zones	Comma separated list of zone identifiers String	-	null

Name	Description	Required	Default
datetime	Either a date-time or an interval, open or closed. Date and time expressions adhere to RFC 3339. Open intervals are expressed using double-dots. Examples: • A date-time: \"2018-02-12T23:20:50Z\" • A closed interval: \"2018-02-12T00:00:00Z/2018-03-18T12:31:12Z\" • Open intervals: \"2018-02-12T00:00:00Z/..\" or \"../2018-03-18T12:31:12Z\" Only features that have a temporal property that intersects the value of 'datetime' are selected. If a feature has multiple temporal properties, it is the decision of the server whether only a single temporal property is used to determine the extent or all relevant temporal properties.	-	null
f	The format of the response. If no value is provided, the standard HTTP rules apply: The HTTP Accept header shall be used to determine the format.	-	null
variables	A comma-separated list of values with names of observable properties that should be returned in the response. More information about the available properties can be retrieved from the '../variables' path. The default is to return all observed properties. String	-	null

6.1.8.3. Return Type

String

6.1.8.4. Content Type

- text/csv
- application/geo+json
- application/json
- text/html

6.1.8.5. Responses

Table 30. HTTP response codes

Code	Message	Datatype
200	The operation was executed successfully.	String

Code	Message	Datatype
400	A query parameter has an invalid value.	exception
406	Content negotiation failed. For example, the 'Accept' header submitted in the request did not support any of the media types supported by the server for the requested resource.	Variables_variables
500	A server error occurred.	exception

Chapter 7. DAPA API Design 2

The DAPA described in this section of the ER was developed by EOX and DLR participants based on inputs from the D110 Expert Use Cases [https://gitlab.ogc.org/ogc/t16-d005-data-access-and-processing-er/-/wikis/D110-Use-cases-(DLR)].

This definition focuses on the provision of raster/coverage data but might also be applied for scattered time-series data.

Overview of URL endpoints:

```
/{collection}/dapa/
    fields/
    cube/
    area/
    timeseries/
        area/
        position/
    value/
        area/
        position/
```

The first hierarchy level after /dapa/ describes the output type of the data requested (except for "fields"):

- cube: 2d raster time-series (each with one or multiple bands).
- area: Single 2d raster (with one or multiple bands).
- timeseries: 1d time-series (each with values from one or multiple fields).
- value: Single value (with values from one or multiple fields).

The second hierarchy level after /dapa/ describes the input geometry (if not implicitly given from the output type):

- area: Polygon/Bounding box
- position: Point

Using this definition, aggregation is automatically conducted based on the `aggregate` parameter to achieve the output type requested (see the Parameter section below).

7.1. /{collection}/dapa/fields

Output fields/variables/properties/bands to be included in the request/processing/aggregation.

The `fields` parameter for the DAPA request consists of values either from the selected collection (e.g., all band names from Sentinel-2) or declared dynamically (e.g., bands algebra NDVI=(B08-B04/B08+B04)) (see the Parameter section below).

7.2. /{collection}/dapa/cube

2d raster time-series (each with one or multiple bands)

- Available only if the collection is a multi-temporal raster dataset.

Parameters:

- bbox/geom/cell
- time
- (fields) defaults to all fields (bands) available
- format

Example UC 1.1: Produce an animated video of ozone concentration from Sentinel-5p for a given time and space.

```
/S5P/dapa/cube?time=2018-05-04T12:12:12Z/2018-06-
04T12:12:12Z&bbox=0,2,3,4&fields=SO2&format=video/avi
```

Results in an animated video.

Example UC 2.1: Retrieve a raster time-series of NDVI calculated from Sentinel-2 scenes for a given time and space.

```
/S2/dapa/cube?time=2018-05-04T12:12:12Z/2018-06-
04T12:12:12Z&bbox=0,2,3,4&fields=NDVI=(B08-B04/B08+B04)&format
=application/metalink
```

Results in a metalink file with download links to multiple raster files (one per time).

7.3. /{collection}/dapa/area

Single 2d raster (e.g., results in TIFF with num(aggregates) x num(fields) bands).

- If the collection is a **multi-temporal raster dataset**, **aggregation over time** is automatically conducted based on the `aggregate` parameter.
- If the collection is a **single raster dataset**, **no aggregation over time** is conducted.

Parameters:

- bbox/geom/cell
- time
- aggregate
- (fields) defaults to all fields (bands) available

- (format) defaults to image/tiff

Example UC 1.2: Retrieve the maximum sulphor dioxide concentration in a given time span as a single coverage (aggregation over time).

```
/S5/dapa/area?time=2018-05-04T12:12:12Z/2018-06-
04T12:12:12Z&bbox=0,2,3,4&aggregate=max&fields=SO2
```

Results in TIFF with a single field/band: SO2_max

Example UC 2.2: Retrieve the minimum and maximum NDVI and NDBI in a given time span (aggregation over time).

```
/S2/dapa/area?time=2018-05-04T12:12:12Z/2018-06-
04T12:12:12Z&bbox=0,2,3,4&aggregate=min,max&fields=NDVI=(B04-B08)/(B04+B08),NDBI=(B01-
B02)/(B01+B02)
```

Results in TIFF with 4 fields/bands: NDVI_min, NDVI_max, NDBI_min, NDBI_max.

7.4. /{collection}/dapa/timeseries

7.4.1. /{collection}/dapa/timeseries/area

1d time-series (each with values from one or multiple fields)

- If the collection is a **multi-temporal raster dataset**, **aggregation over space** is automatically conducted base on the `aggregate` parameter.
- Can not be used for a single raster dataset

Parameters: - bbox/geom/cell - time - aggregate - (fields) defaults to all fields (bands) available - (format) defaults to text/csv

Example UC 1.3: Retrieve the maximum sulphor dioxide concentration in a given area as a time-series (aggregation over space).

```
/S5/dapa/timeseries/area?time=2018-05-04T12:12:12Z/2018-06-
04T12:12:12Z&bbox=0,2,3,4&aggregate=max&fields=SO2
```

Results in CSV with two columns: date and SO2_max.

7.4.2. /{collection}/dapa/timeseries/position

1d time-series (each with values from one or multiple fields)

- Extraction of a time series at a point specified in the request.
- No aggregation is conducted because only a single pixel is extracted.

- Can not be used for a single raster dataset.

Parameters: - point - time - (fields) defaults to all fields (bands) available - (format) defaults to text/csv

Example UC 1.3: Retrieve the maximum sulphor dioxide concentration at a given point as a time-series.

```
/S5/dapa/timeseries/area?time=2018-05-04T12:12:12Z/2018-06-
04T12:12:12Z&point=2,3&fields=SO2
```

Results in CSV with two columns: date and SO2.

7.5. /{collection}/dapa/value

7.5.1. /{collection}/dapa/value/area

Single value (with values from one or multiple fields)

- If the collection is a multi-temporal dataset, **aggregation over space and time** is automatically conducted base on the `aggregate` parameter.
- If the collection is a single dataset/coverage, **aggregation over space** is automatically conducted base on the `aggregate` parameter.

Note: If multiple methods are given in the `aggregate` parameter or multiple `fields` are given, text/plain is not sufficient.

Parameters: - bbox/geom/cell - time - aggregate - (fields) defaults to all fields (bands) available - (format) defaults to text/plain

Example: Retrieve the minimum sulphor dioxide concentration in a given area and time span (aggregated over space and time)

```
/S5P/value/area?time=2018-05-04T12:12:12Z/2018-06-
04T12:12:12Z&bbox=0,2,3,4&aggregate=min&fields=SO2&format=text
/plain
```

Results in a single value

7.5.2. /{collection}/dapa/value/position

Single value (with values from one or multiple fields)

- If the collection is a **multi-temporal raster dataset**, **aggregation over time** is automatically conducted based on the `aggregate` parameter.
- If the collection is a **single raster dataset**, **no aggregation** is conducted.

Note: If multiple methods are given in the `aggregate` parameter or multiple `fields` are given, text/plain is not sufficient.

Parameters: - point - time - aggregate - (fields) defaults to all fields (bands) available - (format) defaults to text/plain.

Example: Retrieve the minimum value of all fields (bands) of Sentinel-5p at a given point in a given time span (aggregated over time).

```
/S5P/value/position?time=2018-05-04T12:12:12Z/2018-06-
04T12:12:12Z&point=2,3&aggregate=min&format=application/json
```

Results in JSON file with a single value for each field (band): SO2_min, O3_min, NO2_min, and so forth.

7.6. Parameters

7.6.1. `point`

Specific location in `x,y`, WKT (?), or reference to point (Link to feature?). To be implemented.

Example:

```
&point=14.2,15.3
```

7.6.2. `bbox`

Bounding box in `minx,miny,maxx,maxy` or reference to geometry (Link to feature?). To be implemented.

Example:

```
&bbox=12.3,0.3,14.4,2.3
```

7.6.3. `geom`

WKT geometry or reference to geometry (Link to feature?). To be implemented.

Example:

```
&geom=POLYGON ((...))
```

7.6.4. `time`

Time point or time span

Examples:

Start/End

```
&time=2018-05-04T12:12:12Z/2018-06-04T12:12:12Z
```

Instant

```
&time=2018-05-04T12:12:12Z
```

Start and period after

```
&time=2018-05-04T12:12:12Z/P5D
```

End and period before

```
&time=P5D/2018-05-04T12:12:12Z
```

Whole day

```
&time=2018-05-04
```

Whole month

```
&time=2018-05
```

Whole year

```
&time=2018
```

7.6.5. `fields`

Comma-separated list of fields, derived (calculated) fields are possible

Syntax:

```
field-parameter = field-selection , { ',' , field-selection }

field-selection = identifier | ( identifier , '=' , expression )
```

Examples:

Listing:

```
&fields=B04,B08
```

Simple aliasing:

```
&fields=NIR=B08,RED=B04
```

Derived field:

```
&fields=NDVI=(B04-B08)/(B04+B08)
```

Combined example:

```
&fields=NIR=B08,RED=B04,NDVI=(RED-NIR)/(RED+NIR)
```

7.6.6. `aggregate`

Specify aggregation

Syntax:

```
aggregate-param = method { ',' , method }

method = identifier  [ '(' , method-arg , ')' ]
```

Examples:

Aggregation (min+max)

```
aggregate=max,min
```

Aggregation (min+max+linear average)

```
aggregate=max,min,avg(linear)
```

7.6.7. `cql`

Metadata filter using CQL

Example:

```
cql=cloudCover < 0.5
```

7.6.8. datafilter

Syntax:

```
datafilter-parameter = datafilter , { ',' , datafilter }

datafilter = boolean-expression
```

Example:

```
datafilter=area(NDVI > 1) >= 0.5
```

7.6.9. format

7.6.9.1. cube formats (~3D)

- AVI
- application/x-netcdf
- application/metalink
- WCS-EO DatsetSeries / CIS

7.6.9.2. coverage (~2D)

- image/tiff

7.6.9.3. timeseries (~1D)

- text/csv
- application/json

7.6.9.4. value (~0D)

- text/plain
- application/json

Appendix A: Revision History

Table 31. Revision History

Date	Editor	Release	Primary clauses modified	Descriptions
June 2, 2020	Luis Bermudez	.1	all	TOC and cleaning
Sep 29, 2020	Luis Bermudez	.2	all	Included two APIS, minor edits in the first sections
Nov 13, 2020	Luis Bermudez	.3	all	Added latest version of the DAPA API (1.0.0-beta1), added DGGS API (2.19-SNAPSHOT DGGS 1.0 server), added harmonization notes from the wiki https://gitlab.ogc.org/ogc/t16-d005-data-access-and-processing-er/-/wikis/DAPA-harmonization
Nov 17, 2020	Luis Bermudez	.4	all	Incorporated Carl’s comment 20201116 and change title of the two DAPA API designs

Date	Editor	Release	Primary clauses modified	Descriptions
Nov 19, 2020	Luis Bermudez	.5	all	Incorporated Gobe’s comment 20201119. Rearranged sections to match swagger order, removed patterns column in tables, formatted list inside tables, fix internal links (anchors), rename the API headings to look more like Swagger.
Dec 14, 2020	Luis Bermudez	.6	all	Clarify various sections about the 2 flavors of the API.