OGC® DOCUMENT: 21-038

External identifier of this OGC® document: http://www.opengis.net/doc/is/ogcapi-dggs/{m.n}

Open Geospatial Consortium

OGC API - DISCRETE GLOBAL GRID SYSTEMS - PART 1: CORE

STANDARD
Implementation

CANDIDATE SWG DRAFT

Version: 1.0
Submission Date: 2029-03-30
Approval Date: 2029-03-30
Publication Date: 2029-03-30
Editor: Dr. Matthew Brian John Purss, Jérôme Jacovella-St-Louis

CONTENTS

LIST OF TABLES

LIST OF RECOMMENDATIONS

I ABSTRACT

This specification defines building blocks which can be used as part of a Web API to retrieve geospatial data for a specific area, time and resolution of interest, based on a specific Discrete Global Grid System (DGGS) and indexing scheme, as defined in OGC Abstract Topic 21, as well to query the list of DGGS zones from which data is available and/or matching a specify query (in combination with other building blocks defining queries, such as a filter defined using the OGC Common Query Language (CQL2).

II KEYWORDS

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, API, openapi, html, ogcapi

III PREFACE

NOTE: OGC API standards define modular API building blocks to spatially enable Web APIs in a consistent way. The OpenAPI specification is used to define the API building blocks.

OGC API – Discrete Global Grid Systems provides API building blocks to retrieve data and query zones based on Discrete Global Grid Systems (DGGS) concepts defined in OGC Abstract Topic 21. Additional parts for *OGC API – Discrete Global Grid Systems* may be defined in the future to provide additional capabilities.

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.

IV SECURITY CONSIDERATIONS

No security considerations have been made for this document.

V SUBMITTING ORGANIZATIONS

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

- Pangaea Innovations Pty. Ltd.
- Ecere Corporation

VI SUBMITTERS

All questions regarding this submission should be directed to the editors or the submitters:

Name	Affiliation
Matthew Brian John Purss (editor)	Pangaea Innovations Pty. Ltd.
Jérôme Jacovella-St-Louis (editor)	Ecere Corporation

1

SCOPE

1 SCOPE

NOTE: Insert Scope text here. Give the subject of the document and the aspects of that scope covered by the document.

2

CONFORMANCE

2 CONFORMANCE

The one Standardization Target for this standard is Web APIs.

OGC API — Common provides a common foundation for OGC API standards. Some conformance classes of this standard have a dependency on, or are designed to be easily integrated with, conformance classes defined in *OGC API — Common Part 1* and/or *Part 2*, as well as within a Web API conforming to additional OGC API standards.

This standard identifies five Conformance Classes. Each Conformance Class has an associated Requirements Class. The Requirements Classes define the functional requirements which will be tested through the associated Conformance Class. Only the Core requirements class is mandatory, all others are optional.

The Requirements Classes for *OGC API — Discrete Global Grid Systems* are:

2.1. Requirements classes defining resources

- **Core**
- **Zone Data Retrieval**
- **Zone Query**

2.2. Requirements classes defining origins

- **Dataset**
- **Collections**

2.3. Requirements classes defining resource representations

- HTML
- GeoJSON

- GeoTIFF
- netCDF
- CoverageJSON
- JPG
- PNG
- JPEG XL
- (Geo)Zarr
- **OpenAPI 3.0**

The Encoding Requirements Classes address support for formats commonly used for encoding geospatial data.

The *OpenAPI 3.0* Requirements Class defines additonal requirements in addition to those defined in *OGC API — Common — Part 1: Core* to facilitate identifying DGGS resources from an OpenAPI 3.0 API Definition.

Conformance with this standard shall be checked using all the relevant tests specified in Annex A (normative) of this document. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site.

In order to conform to this OGC® interface standard, a software implementation shall implement at minimum the “Core” requirements class defined in Annex A (normative).

All requirements-classes and conformance-classes described in this document are owned by the standard(s) identified.

2.4. Summary of conformance URIs

Table 1 — Conformance class URIs

CORRESPONDING REQUIREMENTS CLASS	CONFORMANCE CLASS URI
Core	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/core
Zone Data Retrieval	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-retrieval
Zone Query	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-query

CORRESPONDING REQUIREMENTS CLASS	CONFORMANCE CLASS URI
Dataset	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/dataset
Collections	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/collections
OpenAPI Specification 3.0	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/oas30

3

NORMATIVE REFERENCES

3 NORMATIVE REFERENCES

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

Identification of Common Molecular Subsequences. Smith, T.F., Waterman, M.S., J. Mol. Biol. 147, 195–197 (1981)

ZIB Structure Prediction Pipeline: Composing a Complex Biological Workflow through Web Services. May, P., Ehrlich, H.C., Steinke, T. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) Euro-Par 2006. LNCS, vol. 4128, pp. 1148–1158. Springer, Heidelberg (2006)

The Grid: Blueprint for a New Computing Infrastructure., Foster, I., Kesselman, C.. Morgan Kaufmann, San Francisco (1999).

Grid Information Services for Distributed Resource Sharing. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C. In: 10th IEEE International Symposium on High Performance Distributed Computing, pp. 181–184. IEEE Press, New York (2001)

The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration. Foster, I., Kesselman, C., Nick, J., Tuecke, S. Technical report, Global Grid Forum (2002)

National Center for Biotechnology Information, http://www.ncbi.nlm.nih.gov

ISO: ISO 19101-1:2014, *Geographic information — Reference model — Part 1: Fundamentals*. International Organization for Standardization, Geneva (2014). https://www.iso.org/standard/59164.html.

ISO: ISO 19115-1:2014, *Geographic information — Metadata — Part 1: Fundamentals*. International Organization for Standardization, Geneva (2014). https://www.iso.org/standard/53798.html.

ISO: ISO 19157:2013, *Geographic information — Data quality*. International Organization for Standardization, Geneva (2013). https://www.iso.org/standard/32575.html.

ISO: ISO 19139:2007, *Geographic information — Metadata — XML schema implementation*. ISO (2007).

ISO: ISO 19115-3, *Geographic information — Metadata — Part 3: XML schema implementation for fundamental concepts*. International Organization for Standardization, Geneva https://www.iso.org/standard/80874.html.

Joan Masó and Lucy Bastin: OGC 15-097r1, *OGC® Geospatial User Feedback Standard: Conceptual Model*. Open Geospatial Consortium (2016). https://docs.ogc.org/is/15-097r1/15-097r1.html.

Gerhard Gröger, Thomas H. Kolbe, Claus Nagel, Karl-Heinz Häfele: OGC 12-019, *OGC City Geography Markup Language (CityGML) Encoding Standard*. Open Geospatial Consortium (2012). https://portal.ogc.org/files/?artifact_id=47842.

Jiyeong Lee, Ki-Joune Li, Sisi Zlatanova, Thomas H. Kolbe, Claus Nagel, Thomas Becker: OGC 14-005r3, *OGC® IndoorGML*. Open Geospatial Consortium (2014). https://docs.ogc.org/is/14-005r3/14-005r3.html.

Arliss Whiteside Jim Greenwood: OGC 06-121r9, *OGC Web Service Common Implementation Specification*. Open Geospatial Consortium (2010). https://portal.ogc.org/files/?artifact_id=38867.

4

TERMS AND DEFINITIONS

4 TERMS AND DEFINITIONS

This document uses the terms defined in OGC Policy Directive 49, which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this document and OGC documents do not use the equivalent phrases in the ISO/IEC Directives, Part 2.

This document also uses terms defined in the OGC Standard for Modular specifications (OGC 08-131r3), also known as the ‘ModSpec’. The definitions of terms such as standard, specification, requirement, and conformance test are provided in the ModSpec.

For the purposes of this document, the following additional terms and definitions apply.

This document uses the terms defined in Sub-clause 5.3 of OGC06-121r9, which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

For the purposes of this document, the following additional terms and definitions apply.

4.1. example term

term used for exemplary purposes

Note 1 to entry: An example note.

Example Here’s an example of an example term.

[**SOURCE:** ISO 19101-1:2014]

5

CONVENTIONS

5 CONVENTIONS

This sections provides details and examples for any conventions used in the document. Examples of conventions are symbols, abbreviations, use of XML schema, or special notes regarding how to read the document.

5.1. Identifiers

The normative provisions in this standard are denoted by the URI

`http://www.opengis.net/spec/{standard}/{m.n}`

All requirements and conformance tests that appear in this document are denoted by partial URIs which are relative to this base.

6

REQUIREMENT CLASS "CORE"

6 REQUIREMENT CLASS "CORE"

6.1. Overview

The "Core" Requirement Class allows a client to list available DGGS for a given resource, retrieve additional information about a particular DGGS, and retrieve information about a particular DGGS zone.

The "Core" Requirement Class is the only mandatory Requirements Class, but an implementation with practical use is expected to additionally implement either the "Zone Data Retrieval" Requirement Class, the "Zone Query" Requirement Class, or both.

REQUIREMENTS CLASS 1	
TARGET TYPE	Web API
LABEL	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/core

6.2. Requirements

6.2.1. Listing available DGGS (…/dggs)

REQUIREMENT 1	
LABEL	/req/core/dggs-list
STATEMENT	For retrieving the list of available DGGS:
A	The Implementation SHALL support an HTTP GET operation at a resource path ending with …/dggs.

6.2.2. Retrieving DGGS information (…/dggs/{dggsId})

REQUIREMENT 2	
LABEL	/req/core/dggs-info
STATEMENT	For retrieving information for a particular available DGGS:
A	The Implementation SHALL support an HTTP GET operation at a resource path ending with …/dggs/{dggsId}.

6.2.3. Retrieving zone information (…/dggs/{dggsId}/zones/{zoneId})

REQUIREMENT 3	
LABEL	/req/core/zone-info
STATEMENT	For retrieving information for a particular DGGS zone:
A	The Implementation SHALL support an HTTP GET operation at a resource path ending with …/dggs/{dggsId}/zones/{zoneId}.

7

REQUIREMENT CLASS "DATA RETRIEVAL"

7 REQUIREMENT CLASS "DATA RETRIEVAL"

7.1. Overview

The Data Retrieval conformance class allows to retrieve data from a specific Discrete Global Grid System (DGGS) in a particular indexing scheme from an individual zone. It describes an HTTP GET operation, as well as its response. The selected DGGS is listed as available and described in the Core conformance class, and conforms to OGC Topic 21.

The data for a particular zone is retrieved based on a URI template including a variable representing a Zone ID.

The resource from which data is retrieved can either be for a particular collection of geospatial data, for a dataset as a whole, or in connection with *OGC API — Processes — Part 3: Workflows & Chaining*, the output of a processing workflow.

REQUIREMENTS CLASS 2	
TARGET TYPE	Web API
PREREQUISITE	http://www.opengis.net/spec/ogcapi-common-1/1.0/req/core
LABEL	http://www.opengis.net/spec/ogcapi-dggs-1/0.0/req/data-retrieval

7.2. Requirements

7.2.1. Retrieving data from a zone (`../dggs/{dggsId}/zones/{zoneId}/data`)

The following requirements describe how a client can retrieve data from a single DGGS zone at the resource path …`/dggs/{dggsId}/zones/{zoneId}/data`.

REQUIREMENT 4	
LABEL	/req/data-retrieval/zone-data

REQUIREMENT 4	
STATEMENT	For retrieving data for a single DGGS zone:
A	The operation SHALL support an HTTP GET operation at a resource path ending with …/`dggs/{dggsId}/zones/{zoneId}/data`.
B	The operation SHALL satisfy requirement /req/data-retrieval/rc-zone-depth-definition.
C	The operation SHALL provide a templated link to this resource path using the link relation type http://www.opengis.net/def/rel/ogc/1.0/dggs-zone-data.
D	The response of the HTTP GET operation SHALL have a status code of 200.
E	The content of the response SHALL be a data packet corresponding precisely to the area covered by the DGGS zone.
F	The response SHALL satisfy requirement /req/data-retrieval/rc-zone-depth-response.
G	The selection of an encoding for the response SHALL be consistent with HTTP content negotiation.

7.2.2. Parameter `zone-depth`

REQUIREMENT 5	
LABEL	/req/data-retrieval/zone-depth-parameter
STATEMENT	Parameter to specify the DGGS resolution levels beyond the specified DGGS zone's hierarchy level to include in the response, when retrieving data for that zone
A	The implementation SHALL support a `zone-depth` parameter for the HTTP GET operation on a resource path ending with …/`dggs/{dggsId}/zones/{zoneId}/data`.
B	The implementation SHALL accept the following types of values for the `zone-depth` parameter: • A single positive integer value — representing a specific zone depth to return (e.g., `zone-depth=5`); • A range of positive integer values in the form "{low}-{high}" — representing a continuous range of zone depths to return (e.g., `zone-depth=1-8`); or, • A comma separated list of at least two (2) positive integer values — representing a set of specific zone depths to return (e.g., `zone-depth=1,3,7`). Some or all of these forms of the `zone-depth` parameter may not be supported with particular data packet encodings (the data encoding may support a fixed depth, a range of depths, and/or an arbitrary selection of depths).
C	For each zone depth to be included in the response, the interpretation of a selected depth (whether requesting a single depth, a range of depths, or a list of depths) SHALL be: • 0 corresponding to a single set of range (properties / field) value(s) for the requested zone, • 1 corresponding to all zones of the next deeper hierarchy level associated with the requested zone by the indexing scheme,

REQUIREMENT 5	
	• .. • *n* corresponding to all zones for the *n*'th deeper level in the hierarchy level associated with the requested zone by the indexing scheme.
D	The association of zones of deeper hierarchy levels with the requested zone SHALL be based on the DGGS reference system, which takes into consideration both the grid definition as well as the indexing system in use for the DGGS resource.
E	If a `zone-depth` is specified, the operation SHALL return the data at the resolution(s) / scale(s) specified.
F	If the `zone-depth` parameter is omitted, the default value described in the DGGS reference system resource (.../`dggs/{dggsId}`) SHALL be assumed by the server, in accordance with the capabilities of the data packet encoding. This default value which could be any valid value and/or form as defined above (single depth, range of depths, or list of depths).

NOTE 1: A use case for a `zone-depth` of 0 would be to query the single set of values for a specific DGGS zone.

NOTE 2: For use cases such as visualization and performing analysis over a certain area, a non-zero `zone-depth` would normally be used to avoid an overwhelming number of server round-trips. In this case, more than a single value would be returned for each zone request, with values returned for descendent zones at `zone-depth` levels deeper than the requested zone's level. For example, requesting data for a level 10 zone with a `zone-depth` of 8 would return individual values for each level 18 zones contained within that level 10 zone being requested.

7.2.3. Parameter `subset`

REQUIREMENT 6	
LABEL	/req/data-retrieval/subset
STATEMENT	For specifying a multi-dimensional subset for the zone data being retrieved:

A	The Implementation SHALL support a `subset` query parameter for the zone data retrieval operation (resource path ending with .../`dggs/{dggsId}/zones/{zoneId}/data`) conforming to the following Augmented Backus Naur Form (ABNF) fragment:
	`SubsetSpec: "subset"=axisName(intervalOrPoint)`
	`axisName: {text}`
	`intervalOrPoint: interval | point`
	`interval: low : high`
	`low: point | *`
	`high: point | *`
	`point: {number} | "{text}"`
	
	`Where:`
	`\" = double quote = ASCII code 0x42,`
	`{number} is an integer or floating-point number, and`

REQUIREMENT 6	
	`{text}` is some general ASCII text (such as a time and date notation in ISO 8601).
B	The implementation SHALL support as axis names `Lat` and `Lon` for geographic CRS and `x` and `y` for projected CRS, which are to be interpreted as the best matching spatial axis in the CRS definition.
C	If a third spatial dimension is supported (if the resource's spatial extent bounding box is three dimensional), the implementation SHALL also support a `h` dimension (elevation above the ellipsoid in EPSG:4979 or CRS84h) for geographic CRS and `z` for projected CRS, which are to be interpreted as the vertical axis in the CRS definition.
D	The implementation SHALL return a 400 error status code if an axis name does not correspond to one of the axes of the Coordinate Reference System (CRS) of the target resource.
E	For a CRS where an axis can wrap around, such as subsetting across the dateline (anti-meridian) in a geographic CRS, a *low* value greater than *high* SHALL be supported to indicate an extent crossing that wrapping point.
F	The implementation SHALL interpret the coordinates as values for the named axis of the CRS specified in the `subset-crs` parameter value or in http://www.opengis.net/def/crs/OGC/1.3/CRS84 (http://www.opengis.net/def/crs/OGC/1.3/CRS84h for vertical dimension) if the `subset-crs` parameter is missing.
G	If the `subset` parameter including any of the dimensions corresponding to those of the map bounding box is used with a `bbox`, the server SHALL return a 400 client error.
H	The implementation SHALL interpret multiple `subset` parameters, as if all dimension subsetting values were provided in a single `subset` parameter (comma separated). Example: `subset=Lat(-90:90)&subset=Lon(-180:180)` is equivalent to `subset=Lat(-90:90),Lon(-180:180)`

NOTE 1: A subset parameter for http://www.opengis.net/def/crs/OGC/1.3/CRS84 will read as subset=Lon(left_lon:right_lon),Lat(lower_lat:upper_lat).

NOTE 2: When the *inteval* values fall partially outside of the range of valid values defined by the CRS for the identified axis, the service is expected to return the non-empty portion of the resource resulting from the subset.

7.2.4. Parameter `datetime`

REQUIREMENT 7	
LABEL	/req/data-retrieval/datetime
STATEMENT	For specifying a multi-dimensional subset for which to retrieve data from a zone:
A	The Implementation SHALL support a `subset` query parameter for the zone data retrieval operation (resource path ending with `.../dggs/{dggsId}/zones/{zoneId}/data`) conforming to the following Augmented Backus Naur Form (ABNF) fragment: `SubsetSpec: "subset"=axisName(intervalOrPoint)`

REQUIREMENT 7

```
  axisName:         {text}
  intervalOrPoint:  interval \| point
  interval:         low : high
  low:              point \| *
  high:             point \| *
  point:            {number} \| "{text}"

  Where:
     \" = double quote = ASCII code 0x42,
     {number} is an integer or floating-point number, and
     {text} is some general ASCII text (such as a time and date notation in
ISO 8601).
```

B	The implementation SHALL support as axis names `Lat` and `Lon` for geographic CRS and `x` and `y` for projected CRS, which are to be interpreted as the best matching spatial axis in the CRS definition.
C	If a third spatial dimension is supported (if the resource's spatial extent bounding box is three dimensional), the implementation SHALL also support a `h` dimension (elevation above the ellipsoid in EPSG:4979 or CRS84h) for geographic CRS and `z` for projected CRS, which are to be interpreted as the vertical axis in the CRS definition.
D	The implementation SHALL return a 400 error status code if an axis name does not correspond to one of the axes of the Coordinate Reference System (CRS) of the target resource.
E	For a CRS where an axis can wrap around, such as subsetting across the dateline (anti-meridian) in a geographic CRS, a *low* value greater than *high* SHALL be supported to indicate an extent crossing that wrapping point.
F	The implementation SHALL interpret the coordinates as values for the named axis of the CRS specified in the `subset-crs` parameter value or in http://www.opengis.net/def/crs/OGC/1.3/CRS84 (http://www.opengis.net/def/crs/OGC/1.3/CRS84h for vertical dimension) if the `subset-crs` parameter is missing.
G	If the `subset` parameter including any of the dimensions corresponding to those of the map bounding box is used with a `bbox`, the server SHALL return a 400 client error.
H	The implementation SHALL interpret multiple `subset` parameters, as if all dimension subsetting values were provided in a single `subset` parameter (comma separated). Example: `subset=Lat(-90:90)&subset=Lon(-180:180)` is equivalent to `subset=Lat(-90:90),Lon(-180:180)`

NOTE 1: A subset parameter for http://www.opengis.net/def/crs/OGC/1.3/CRS84 will read as subset=Lon(left_lon:right_lon),Lat(lower_lat:upper_lat).

NOTE 2: When the *inteval* values fall partially outside of the range of valid values defined by the CRS for the identified axis, the service is expected to return the non-empty portion of the resource resulting from the subset.

8

REQUIREMENT CLASS "ZONE QUERY"

8 REQUIREMENT CLASS "ZONE QUERY"

8.1. Overview

The Zone Query conformance class allows to request the list of DGGS zones from a specific Discrete Global Grid System (DGGS) in a particular indexing scheme for which there is data available, or matching a particular query (e.g., using a filtering parameter). It describes an HTTP GET operation, as well as its response. The selected DGGS is listed as available and described in the Core conformance class, and conforms to OGC Topic 21. The list of zones from a Web API using this conformance class can either be for a particular collection of geospatial data, for a dataset as a whole, or in connection with *OGC API — Processes — Part 3: Workflows & Chaining*, the output of a processing workflow.

REQUIREMENTS CLASS 3	
TARGET TYPE	Web API
PREREQUISITE	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/core
LABEL	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query

8.2. Requirements

8.2.1. Listing zones (…/dggs/{dggsId}/zones)

The following requirements describe how a client can retrieve the list of zones from which data is available at the resource path …/dggs/{dggsId}/zones.

REQUIREMENT 8	
LABEL	/req/zone-query/zones-list
STATEMENT	For retrieving a list of DGGS zones:

REQUIREMENT 8	
A	The Implementation SHALL support an HTTP GET operation at a resource path ending with .../dggs/{dggsId}/zones.
B	The Implementation SHALL provide a link to this resource path using the link relation type http://www.opengis.net/def/rel/ogc/1.0/dggs-zone-query.
C	The response of the HTTP GET operation SHALL have a status code of 200.
D	The content of the response SHALL be a list of zones fully covering where data is available (in the case where the resource is associated with a particular dataset), and matching any additional query parameters specified by the client (e.g., a filtering query parameter), without any redundancy.
E	Unless the zones is a compact list of zones (see compact-zones parameter), the zones returned SHALL all be of the same DGGS hierarchy level.
F	The selection of an encoding for the returned list of zones SHALL be consistent with HTTP content negotiation.
G	The implementation SHALL support at minimum a JSON encoding (media type application/json) where each zone ID is specified as a string within an array of zones assigned to a zones property within a JSON object (e..g, { "zones": ["1-E-14"s, "1-E-15", "1-F-14", "1-F-15", "1-F-16"] }.

8.2.2. Parameter zone-level

The following requirements describe how a client can specify the DGGS hierarchy level at which to retrieve the list of zones.

REQUIREMENT 9	
LABEL	/req/zone-query/zone-level
STATEMENT	For specifying a level at which to return a list of DGGS zones using a zone-level query parameter:
A	The Implementation SHALL support a zone-level query parameter for the zone query operation (resource path ending with .../dggs/{dggsId}/zones).
B	If a compact zones list is used (the default), the zones returned in the response SHALL be of the DGGS hierarchy level specified by the zone-level query parameter, or of a lower hierarchy level standing in for a compact representation of multiple zones at the requested hierarchy level.
C	If a compact zones list is not used, the zones returned in the response SHALL be of the DGGS hierarchy level specified by the zone-level query parameter.

8.2.3. Parameter `compact-zones`

By default, implementations return a compact list of zones where children zones fully covering a parent are recursively replaced by the parent zones, allowing to express large areas in a much more compact list of zones. The following requirements describe how a client can disable returning a compact list of zones.

REQUIREMENT 10	
LABEL	/req/zone-query/compact-zones
STATEMENT	For specifying whether to retrieve a list of DGGS zones using a `compact-zones` parameter:
A	The Implementation SHALL support a boolean `compact-zones` query parameter for the zone query operation (resource path ending with …/`dggs/{dggsId}/zones`), where a value of `true` corresponds to the default behavior when the paramter is not specified, and a value of `false` disables the use of compact-zones in the response.
B	When the `compact-zones` parameter is to false, the zones list response SHALL NOT be a compact list, and SHALL explicitly list every individual zone at the requested or default DGGS hierarchy level.
C	When the `compact-zones` parameter is to true (or unspecified), the zones list response SHALL be a compact list, where children zones completely covering the area of a parent zone SHALL be replaced by that parent zone, in a recursive manner all the way to the lowest DGGS hierarchy level.

8.2.4. Parameter `limit` for paging

The following requirements describe how a client can specify a limit to the number of zones to be returned and page through large list of zones as multiple requests and responses.

REQUIREMENT 11	
LABEL	/req/zone-query/limit
STATEMENT	For specifying a paging limit for the list of zones using a `limit` query parameter:
A	The Implementation SHALL support a parameter `limit` integer query parameter, with a minimum value of 1.
B	The response SHALL not contain more zones than specified by the optional `limit` parameter (if specified).
C	If the API definition specifies a maximum value for the `limit` parameter, the response SHALL not contain more zones than this maximum value.

REQUIREMENT 11	
D	If the value of the `limit` parameter is larger than the maximum value, this SHALL NOT result in an error (but instead be replaced by the maximum as the parameter value).
E	If using compact zones, the parent zones SHALL count as a single zone, rather than the number of children zones they stand in for.
F	If an implementation does not return the full list of zones for the request, a link with relation type `next` SHALL be included in a `links` array property of the response, which a client can request to resume listing the zones.

8.2.5. Parameter `bbox`

REQUIREMENT 12	
LABEL	/req/zone-query/bbox
STATEMENT	For specifying a spatial bounding box for which to return a list of DGGS zones:

A	The Implementation SHALL support a `bbox` query parameter for the zone query operation (resource path ending with `.../dggs/{dggsId}/zones`) with the characteristics defined in the OpenAPI Specification 3.0 fragment:
	`bbox:`
	`  name: bbox`
	`  in: query`
	`  description:`
	`  Bounding box of the rendered map. The bounding box is provided as four or six coordinates`
	
	`  * 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 coordinate reference system and axis order of the values are indicated in the `bbox-crs` parameter or if the parameter is missing in http://www.opengis.net/def/crs/OGC/1.3/CRS84`
	`  required: false`
	`  schema:`
	`  type: array`
	`  oneOf:`
	`  - minItems: 4`
	`  maxItems: 4`
	`  - minItems: 6`
	`  maxItems: 6`
	`  items:`
	`  type: number`
	`  format: double`
	`  style: form`
	`  explode: false`

B	`bbox` SHALL be a comma separated list of four or six floating point numbers. If the bounding box consists of six numbers, the first three numbers are the coordinates of the lower bound corner of a

REQUIREMENT 12	
	three-dimensional bounding box and the last three are the coordinates of the upper bound corner. The axis order is determined by the `bbox-crs` parameter value or longitude and latitude if the parameter is missing (http://www.opengis.net/def/crs/OGC/1.3/CRS84 axis order for a 2D bounding box, http://www.opengis.net/def/crs/OGC/1.3/CRS84h for a 3D bounding box). For example in http://www.opengis.net/def/crs/OGC/1.3/CRS84 the order is left_lon, lower_lat, right_lon, upper_lat.
C	The returned list of zone IDs SHALL only contain zones inside or intersecting with the spatial extent of the geographical area of bounding box.

8.2.6. Parameter `bbox-crs`

REQUIREMENT 13	
LABEL	/req/zone-query/bbox-crs
STATEMENT	For specifying the CRS in used for the `bbox` parameter using the `bbox-crs` parameter
A	The list of zones resource SHALL support a `bbox-crs` parameter specifying the CRS used for the `bbox` parameter.
B	For Earth centric data, the implementation SHALL support http://www.opengis.net/def/crs/OGC/1.3/CRS84 as a value.
C	If the `bbox-crs` is not indicated http://www.opengis.net/def/crs/OGC/1.3/CRS84 SHALL be assumed.
D	The native CRS (`storageCRS`) SHALL be supported as a value. Other conformance classes may allow additional values (see `crs` parameter definition).
E	The CRS expressed as URIs or as safe CURIEs SHALL be supported.
F	If the bbox parameter is not used, the `bbox-crs` SHALL be ignored.

8.2.7. Parameter `subset`

REQUIREMENT 14	
LABEL	/req/zone-query/subset
STATEMENT	For specifying a multi-dimensional subset for which to return a list of DGGS zones:

REQUIREMENT 14	

A	The Implementation SHALL support a `subset` query parameter for the zone query operation (resource path ending with .../`dggs/{dggsId}/zones`) conforming to the following Augmented Backus Naur Form (ABNF) fragment:
	`SubsetSpec: "subset"=axisName(intervalOrPoint)`
	`axisName: {text}`
	`intervalOrPoint: interval | point`
	`interval: low : high`
	`low: point | *`
	`high: point | *`
	`point: {number} | "{text}"`
	
	`Where:`
	`\" = double quote = ASCII code 0x42,`
	`{number} is an integer or floating-point number, and`
	`{text} is some general ASCII text (such as a time and date notation in ISO 8601).`

B	The implementation SHALL support as axis names `Lat` and `Lon` for geographic CRS and `x` and `y` for projected CRS, which are to be interpreted as the best matching spatial axis in the CRS definition.
C	If a third spatial dimension is supported (if the resource's spatial extent bounding box is three dimensional), the implementation SHALL also support a `h` dimension (elevation above the ellipsoid in EPSG:4979 or CRS84h) for geographic CRS and `z` for projected CRS, which are to be interpreted as the vertical axis in the CRS definition.
D	The implementation SHALL return a 400 error status code if an axis name does not correspond to one of the axes of the Coordinate Reference System (CRS) of the target resource.
E	For a CRS where an axis can wrap around, such as subsetting across the dateline (anti-meridian) in a geographic CRS, a *low* value greater than *high* SHALL be supported to indicate an extent crossing that wrapping point.
F	The implementation SHALL interpret the coordinates as values for the named axis of the CRS specified in the `subset-crs` parameter value or in http://www.opengis.net/def/crs/OGC/1.3/CRS84 (http://www.opengis.net/def/crs/OGC/1.3/CRS84h for vertical dimension) if the `subset-crs` parameter is missing.
G	If the `subset` parameter including any of the dimensions corresponding to those of the map bounding box is used with a `bbox`, the server SHALL return a 400 client error.
H	The implementation SHALL interpret multiple `subset` parameters, as if all dimension subsetting values were provided in a single `subset` parameter (comma separated). Example: `subset=Lat(-90:90)&subset=Lon(-180:180)` is equivalent to `subset=Lat(-90:90),Lon(-180:180)`

NOTE 1: A subset parameter for http://www.opengis.net/def/crs/OGC/1.3/CRS84 will read as subset=Lon(left_lon:right_lon),Lat(lower_lat:upper_lat).

NOTE 2: When the *inteval* values fall partially outside of the range of valid values defined by the CRS for the identified axis, the service is expected to return the non-empty portion of the resource resulting from the subset.

NOTE 3: For the operation of returning a list of zone IDs, there normally is no value in preserving dimensionality, therefore a *slicing* operation (using the *point* notation) is usually

equivalent to a *trimming* operation (using the *interval* notation) when the low and high bounds of an interval are the same. Therefore, use of the point notation is encouraged in these cases.

8.2.8. Parameter `subset-crs`

REQUIREMENT 15	
LABEL	/req/zone-query/subset-crs
STATEMENT	For specifying the CRS in used for the `subset` parameter using the `subset-crs` parameter
A	The zone listing operation SHALL support a parameter `subset-crs` with the characteristics identifying the CRS in which the `subset` parameter is specified with a URI or safe CURIE.
B	For Earth centric data, http://www.opengis.net/def/crs/OGC/1.3/CRS84 as a value SHALL be supported.
C	If the `subset-crs` is not indicated, http://www.opengis.net/def/crs/OGC/1.3/CRS84 SHALL be assumed.
D	The native CRS (`storageCRS`) SHALL be supported as a value. Other requirements classes may allow additional values (see `crs` parameter definition).
E	CRSs expressed as URIs or as safe CURIEs SHALL be supported.
F	If no `subset` parameter referring to an axis of the CRS is used, the `subset-crs` SHALL be ignored.

8.2.9. Parameter `datetime`

REQUIREMENT 16	
LABEL	/req/zone-query/datetime
STATEMENT	For specifying a multi-dimensional subset for which to return a list of DGGS zones:

A	The Implementation SHALL support a `subset` query parameter for the zone query operation (resource path ending with .../`dggs/{dggsId}/zones`) conforming to the following Augmented Backus Naur Form (ABNF) fragment:
	`SubsetSpec: "subset"=axisName(intervalOrPoint)`
	`axisName: {text}`
	`intervalOrPoint: interval | point`
	`interval: low : high`
	`low: point | *`
	`high: point | *`
	`point: {number} | "{text}"`
	
	`Where:`
	`\" = double quote = ASCII code 0x42,`

REQUIREMENT 16	
	`{number}` is an integer or floating-point number, and `{text}` is some general ASCII text (such as a time and date notation in ISO 8601).
B	The implementation SHALL support as axis names `Lat` and `Lon` for geographic CRS and `x` and `y` for projected CRS, which are to be interpreted as the best matching spatial axis in the CRS definition.
C	If a third spatial dimension is supported (if the resource's spatial extent bounding box is three dimensional), the implementation SHALL also support a `h` dimension (elevation above the ellipsoid in EPSG:4979 or CRS84h) for geographic CRS and `z` for projected CRS, which are to be interpreted as the vertical axis in the CRS definition.
D	The implementation SHALL return a 400 error status code if an axis name does not correspond to one of the axes of the Coordinate Reference System (CRS) of the target resource.
E	For a CRS where an axis can wrap around, such as subsetting across the dateline (anti-meridian) in a geographic CRS, a *low* value greater than *high* SHALL be supported to indicate an extent crossing that wrapping point.
F	The implementation SHALL interpret the coordinates as values for the named axis of the CRS specified in the `subset-crs` parameter value or in http://www.opengis.net/def/crs/OGC/1.3/CRS84 (http://www.opengis.net/def/crs/OGC/1.3/CRS84h for vertical dimension) if the `subset-crs` parameter is missing.
G	If the `subset` parameter including any of the dimensions corresponding to those of the map bounding box is used with a `bbox`, the server SHALL return a 400 client error.
H	The implementation SHALL interpret multiple `subset` parameters, as if all dimension subsetting values were provided in a single `subset` parameter (comma separated). Example: `subset=Lat(-90:90)&subset=Lon(-180:180)` is equivalent to `subset=Lat(-90:90),Lon(-180:180)`

NOTE 1: A subset parameter for http://www.opengis.net/def/crs/OGC/1.3/CRS84 will read as subset=Lon(left_lon:right_lon),Lat(lower_lat:upper_lat).

NOTE 2: When the *inteval* values fall partially outside of the range of valid values defined by the CRS for the identified axis, the service is expected to return the non-empty portion of the resource resulting from the subset.

NOTE 3: For the operation of returning a list of zone IDs, there normally is no value in preserving dimensionality, therefore a *slicing* operation (using the *point* notation) is usually equivalent to a *trimming* operation (using the *interval* notation) when the low and high bounds of an interval are the same. Therefore, use of the point notation is encouraged in these cases.

9

REQUIREMENT CLASS "DATASET"

9 REQUIREMENT CLASS "DATASET"

9.1. Overview

The "Dataset" Requirement Class defines the availability of DGGS resources applying to a whole dataset or API, as defined by *OGC API — Common — Part 1: Core*.

REQUIREMENTS CLASS 4	
TARGET TYPE	Web API
PREREQUISITE	http://www.opengis.net/spec/ogcapi-common-1/1.0/req/core
LABEL	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/dataset

9.2. Requirements

9.2.1. Collection DGGS (/dggs)

REQUIREMENT 17	
LABEL	/req/dataset/dggs
STATEMENT	For API/dataset-wide DGGS resources:
A	The Implementation SHALL support DGGS resources for the dataset as a whole, for the "Core" requirement class, as well as any resources defined in additional supported requirements classes.

10

REQUIREMENT CLASS "COLLECTIONS"

10 REQUIREMENT CLASS "COLLECTIONS"

10.1. Overview

The "Collections" Requirement Class defines the availability of DGGS resources applying to one or more collections of geospatial data, as defined by *OGC API — Common — Part 2: Geospatial data*.

REQUIREMENTS CLASS 5	
TARGET TYPE	Web API
PREREQUISITE	http://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections
LABEL	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/collections

10.2. Requirements

10.2.1. Collection DGGS (`/collections/{collectionId}/dggs`)

REQUIREMENT 18	
LABEL	/req/collections/dggs
STATEMENT	For collection DGGS resources:
A	The Implementation SHALL support DGGS resources for at least one collection, for the "Core" requirement class, as well as any resources defined in additional supported requirements classes.

11

MEDIA TYPES FOR ANY DATA ENCODING(S)

11 MEDIA TYPES FOR ANY DATA ENCODING(S)

A section describing the MIME-types to be used is mandatory for any standard involving data encodings. If no suitable MIME type exists in http://www.iana.org/assignments/media-types/index.html then this section may be used to define a new MIME type for registration with IANA.

A

ANNEX A (INFORMATIVE) CONFORMANCE CLASS ABSTRACT TEST SUITE (NORMATIVE)

A

ANNEX A (INFORMATIVE) CONFORMANCE CLASS ABSTRACT TEST SUITE (NORMATIVE)

NOTE: Ensure that there is a conformance class for each requirements class and a test for each requirement (identified by requirement name and number)

A.1. Conformance Class A

A.1.1. Requirement 1

REQUIREMENT A.1	
TEST PURPOSE	Verify that…
TEST METHOD	Inspect…

A.1.2. Requirement 2

B

ANNEX B (INFORMATIVE) TITLE

B

ANNEX B (INFORMATIVE) TITLE

NOTE: Place other Annex material in sequential annexes beginning with “B” and leave final two annexes for the Revision History and Bibliography

C

ANNEX C (INFORMATIVE) REVISION HISTORY

C

ANNEX C (INFORMATIVE) REVISION HISTORY

Table C.1

DATE	RELEASE	EDITOR	PRIMARY CLAUSES MODIFIED	DESCRIPTION
2021-05-17	0.1	Matthew Purss	all	initial version
2022-07-22	0.2	Jerome St Louis	all	Renamed Part 1 to core — github commit — 316d37f
2022-07-22	0.3	Jerome St Louis	all	Initial set up of conformance classes clauses — github commit — b76073c
2022-07-26	0.4	Gobe Hobona	all	Fixes problem that was breaking auto-build — github commit — a13c2a8
2022-10-04	0.5	Jerome St Louis	7-Data Retrieval	Initial take at Data Retrieval Conformance Class — github commit — b67f290
2022-11-11	0.6	Jerome St Louis	7-Data Retrieval	Split multi-done retrieval into separate conformance class — github commit — 06a4260
2022-11-11	0.7	Jerome St Louis	8-zone query	Initial progress on Zone Query conformance class — github commit — a208502
2022-11-11	0.8	Jerome St Louis	21-038; 0-front material	Added preface, abstract, Jerome added as editor — github commit — 71e76c6
2022-11-11	0.9	Jerome St Louis	7-Data Retrieval	Clarification regarding resolution — github commit — c43bfad
2022-11-11	0.10	Jerome St Louis	8-zone query	Clarifications — github commit — cee3685
2023-02-17	0.11	Jerome St Louis	7-Data Retrieval	Added zone-depth parameter — github commit — 2b8fa58

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] *OGC: OGC Web Service Common Implementation Specification*, OGC 06-121r9

[2] *OGC: OGC Testbed 12 Annex B: Architecture* (2015).