



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

Notice for Drafts: This document is not an OGC Standard. This document is distributed for review and comment. This document is subject to change without notice and may not be referred to as an OGC Standard.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

License Agreement

Use of this document is subject to the license agreement at <https://www.ogc.org/license>

Suggested additions, changes and comments on this document are welcome and encouraged. Such suggestions may be submitted using the online change request form on OGC web site: <http://ogc.standardstracker.org/>

Copyright notice

Copyright © 2023 Open Geospatial Consortium

To obtain additional rights of use, visit <https://www.ogc.org/legal>

Note

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.

CONTENTS

I. ABSTRACT	vi
II. KEYWORDS	vi
III. PREFACE	vii
IV. SECURITY CONSIDERATIONS	viii
V. SUBMITTING ORGANIZATIONS	ix
VI. SUBMITTERS	ix
1. SCOPE	2
2. CONFORMANCE	4
2.1. Requirements classes defining resources	4
2.2. Requirements classes defining origins	4
2.3. Requirements classes defining parameters	4
2.4. Requirements classes defining resource representations	5
2.5. Summary of conformance URIs	6
3. NORMATIVE REFERENCES	9
4. TERMS AND DEFINITIONS	12
5. CONVENTIONS	14
5.1. Identifiers	14
6. REQUIREMENT CLASS “CORE”	16
6.1. Overview	16
6.2. Requirements	16
7. REQUIREMENT CLASS “DATA RETRIEVAL”	21
7.1. Overview	21
7.2. Requirements	21
8. REQUIREMENT CLASS “DATA SUBSETTING”	24
8.1. Overview	24
8.2. Requirements	24
9. REQUIREMENT CLASS “DATA CUSTOM DEPTHS”	28

9.1. Overview	28
9.2. Requirements	28
10. REQUIREMENT CLASS “ZONE QUERY”	31
10.1. Overview	31
10.2. Requirements	31
11. REQUIREMENT CLASS “ROOT DGGS”	40
11.1. Overview	40
11.2. Requirements	40
12. REQUIREMENT CLASS “COLLECTION DGGS”	42
12.1. Overview	42
12.2. Requirements	42
13. REQUIREMENTS CLASSES FOR ENCODINGS OF ZONE DATA AND ZONE LIST	44
13.1. Media Types	44
13.2. Zone data encodings	45
13.3. Zone list encodings	47
ANNEX A (INFORMATIVE) CONFORMANCE CLASS ABSTRACT TEST SUITE (NORMATIVE)	50
A.1. Conformance Class A	50
ANNEX B (INFORMATIVE) TITLE	52
ANNEX C (INFORMATIVE) REVISION HISTORY	54
BIBLIOGRAPHY	56

LIST OF TABLES

Table 1 – Conformance class URIs	6
Table 2 – Media Types used for encoding requirements classes	44
Table C.1	54

LIST OF RECOMMENDATIONS

REQUIREMENTS CLASS 1	16
REQUIREMENTS CLASS 2	21

REQUIREMENTS CLASS 3	24
REQUIREMENTS CLASS 4	28
REQUIREMENTS CLASS 5	31
REQUIREMENTS CLASS 6	40
REQUIREMENTS CLASS 7	42
REQUIREMENT 1	16
REQUIREMENT 2	17
REQUIREMENT 3	18
REQUIREMENT 4	21
REQUIREMENT 5	24
REQUIREMENT 6	25
REQUIREMENT 7	28
REQUIREMENT 8	31
REQUIREMENT 9	32
REQUIREMENT 10	33
REQUIREMENT 11	33
REQUIREMENT 12	34
REQUIREMENT 13	35
REQUIREMENT 14	36
REQUIREMENT 15	37
REQUIREMENT 16	38
REQUIREMENT 17	40
REQUIREMENT 18	42
RECOMMENDATION 1	18
RECOMMENDATION 2	34
REQUIREMENT A.1	50



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 \(CQL2\)](#)).



KEYWORDS

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

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



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.



SECURITY CONSIDERATIONS

No security considerations have been made for this document.



SUBMITTING ORGANIZATIONS

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

- Pangaea Innovations Pty. Ltd.
- Ecere Corporation



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

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



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 – CommonPart 1* and/or *Part 2*, as well as within a Web API conforming to additional OGC API standards.

This standard identifies nineteen 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

- Root DGGS
- Collection DGGS

2.3. Requirements classes defining parameters

- Zone data subsetting
- Zone data custom depths

2.4. Requirements classes defining resource representations

- GeoJSON Data
- GeoTIFF Data
- netCDF Data
- CoverageJSON Data
- JPEG XL Data
- PNG Data
- JSON Zone List
- HTML Zone List
- 64-bit Binary Zone List
- GeoJSON Zone List
- GeoTIFF Zone List
- **OpenAPI 3.0**

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

The Zone List Encoding Requirements Classes address support for formats efficiently and/or intuitively encoding a list of DGGs zone identifiers.

The *OpenAPI 3.0* Requirements Class defines additional 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.

2.5. 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
Root DGGS	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/root-dggs
Collection DGGS	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/collection-dggs
Data subsetting	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-subsetting
Data custom depths	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-custom-depths
GeoJSON Data	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-geojson
GeoTIFF Data	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-tiff
netCDF Data	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-netcdf
CoverageJSON Data	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-coveragejson
JPEG XL Data	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-jpegxl
PNG Data	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-png
JSON Zone List	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-json
HTML Zone List	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-html
64-bit Binary Zone List	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-uint64
GeoJSON Zone List	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-geojson
GeoTIFF Zone List	http://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-tiff

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



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

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

This section 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”

6.1. 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 discrete global grid reference systems:

A The Implementation SHALL support an HTTP GET operation at a resource path ending with .../dggs.

B The Implementation SHALL support a JSON representation of this .../dggs resource.

C The .../dggs resource SHALL include a dggs array property listing all discrete global grid reference systems supported by the Implementation.

REQUIREMENT 1

D	Each element of the dggs array SHALL include a subset of the information available in the individual DGGs information described below, including at minimum the id, title, uri (if applicable), and links.
E	The links property SHALL include at minimum a link to the discrete global grid reference system using the self link relation type, as well as a link to the discrete global grid reference system using the [ogc-rel:dggs-definition] relation type.

6.2.2. Discrete global grid reference system information (.../dggs/{dggsId})

REQUIREMENT 2

LABEL /req/core/dggs-info

STATEMENT For retrieving information for a particular available discrete global grid reference system:

A	The Implementation SHALL support an HTTP GET operation at a resource path ending with .../dggs/{dggsId}.
B	The Implementation SHALL support a JSON representation of this .../dggs/{dggsId} resource.
C	The .../dggs/{dggsId} resource SHALL include an id property consistent with the {dggsId} resource path parameter.
D	The .../dggs/{dggsId} resource SHALL include a link to the resource itself using the self link relation type.
E	The .../dggs/{dggsId} resource SHALL include a link to a definition of the discrete global grid reference system (describing both the discrete global grid as well as the particular indexing scheme to identify zones by identifiers), using the link relation type [ogc-rel:dggs-definition]. The schema for that definition is likely to evolve in order to support describing a growing number of classes of DGGs. A first draft of this schema is available here .
F	The .../dggs/{dggsId} resource SHALL include a templated link to request information for a particular zone using the link relation type [ogc-rel:dggs-zone-info].
G	If the discrete global grid reference system (the combination of the discrete global grid and indexing system) is registered with an authority, the resource SHALL include uri property corresponding to that registered discrete global grid reference system.
H	If the discrete global grid reference system is based on a particular coordinate reference system, the resource shall specify that CRS in a crs property, preferably as a URI (if one is available).
I	The Implementation SHALL include a short title property identifying the discrete global grid reference system intended for display to a human.
J	The Implementation SHALL include a description property providing a summary description of the discrete global grid reference system used by this particular DGGs instance.

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} providing information for a valid individual zones of the discrete global grid reference system. |
| B | The zone information resource SHALL support a JSON representation. |
| C | The zone information resource SHALL include an id property corresponding to the {zoneId} resource path parameter. |
| D | The zone information resource SHALL include a link back to the corresponding DGGS resource (.../dggs) using the [ogc-rel : dggs] link relation type. |

RECOMMENDATION 1

LABEL /rec/core/zone-info

- | | |
|----------|---|
| A | The zone information resource SHOULD include an areaMetersSquare property indicating the surface area of the zone in square meters. |
| B | For a DGGS with three spatial dimension, the zone information resource SHOULD include a volumeMetersCube property indicating the volume of the zone in cubic meters. |
| C | For a temporal DGGS, the zone information resource SHOULD include a temporalDurationSeconds property indicating the amount of time covered by the zone in seconds. |
| D | The zone information resource SHOULD include a geometry property indicating the 2D and/or 3D spatial geometry of the zone, using GeoJSON or OGC Features & Geometry JSON for the JSON encoding. |
| E | For a temporal DGGS, the zone information resource SHOULD include a temporalInterval property indicating the start and end time of the zone. |
| F | The implementation SHOULD support a GeoJSON and/or OGC Features & Geometry JSON representation of the zone information resource where the top-level object is a feature representing the zone geometry, the feature ID corresponds to the {zoneId}, and the other properties described in this recommendation are properties of that feature. |
| G | For a zone associated with a particular collection, the implementation SHOULD provide summary statistics (minimum, maximum, average, stdDev) pertaining to this zone for each field (fields of the range of a coverage, or relevant numeric properties of a feature collection) of the data. In the JSON |

RECOMMENDATION 1

encoding, this SHOULD be implemented as a JSON dictionary mapping field names to an object with each statistic.

H

For a zone associated with a particular collection, the implementation SHOULD provide `areaMetersSquareWithData`, `volumeMetersCubeWithData`, `temporalDurationSecondsWithData` properties corresponding to the respective properties defined above for the overall zones, but considering only the portions of the zone where there is data (e.g., regions of the zone excluding NODATA values for a gridded coverage, or within geometry for a feature collection).



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	<code>/req/data-retrieval/zone-data</code>
-------	--

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 <code>.../dggs/{dggsId}/zones/{zoneId}/data</code> . |
| B | The operation SHALL include a templated link to this resource path in the “Core” <code>.../dggs/{dggsId}</code> resource links, and regular link in the <code>.../dggs/{dggsId}/zones/{zoneId}</code> resource links using the link relation type <code>[ogc-rel:dggs-zone-data]</code> . |
| C | The response of the HTTP GET operation SHALL have a status code of 200. |
| D | The content of the response SHALL be a data packet corresponding precisely to the area covered by the DGGS zone. |
| E | The selection of an encoding for the response SHALL be consistent with HTTP content negotiation. |
| F | The <code>.../dggs/{dggsId}</code> resource SHALL include a <code>defaultDepth</code> property indicating the implementation’s default depth for when the <code>zone-depth</code> parameter is omitted. This default value could be any valid value and/or form as defined in the <code>/req/data-custom-depths/zone-depth-parameter</code> requirement (single depth, range of depths, or list of depths, relative to the <code>{zoneId}</code> hierarchy level). |
| G | Unless a <code>zone-depth</code> parameter is specified, the response SHALL return a data packet consistent with this <code>defaultDepth</code> property, in accordance with the capabilities of the negotiated data packet encoding. |



8

REQUIREMENT CLASS “DATA SUBSETTING”

8.1. Overview

The Data Subsetting requirements class extends the zone data retrieval requirements with the ability to subset data on additional dimensions beyond those defined by the discrete global grid reference systems (DGGRS), for example along time or vertical dimension if those axes are not part the DGGRS definition, or along additional dimensions such as atmospheric pressure levels. Two parameters are defined for this purpose: `datetime` (specifically for time) and `subset` (for any dimension, including time).

REQUIREMENTS CLASS 3

TARGET TYPE	Web API
PREREQUISITE	http://www.opengis.net/spec/ogcapi-dggs-1/0.0/req/data-retrieval
LABEL	http://www.opengis.net/spec/ogcapi-dggs-1/0.0/req/data-subsetting

8.2. Requirements

8.2.1. Parameter `subset`

REQUIREMENT 5

LABEL	/req/data-subsetting/subset
STATEMENT	For specifying a multi-dimensional subset for the zone data being retrieved (excluding dimensions of the discrete global grid reference system):
A	<p>The Implementation SHALL support a subset query parameter for the zone data retrieval operation (resource path ending with <code>.../dggs/{dggsId}/zones/{zoneId}/data</code>) conforming to the following Augmented Backus Naur Form (ABNF) fragment:</p> <pre> SubsetSpec: "subset"=axisName(intervalOrPoint) axisName: {text} intervalOrPoint: interval \ point interval: low : high </pre>

REQUIREMENT 5

```

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 time for a temporal dataset, unless this temporal axis is an axis of the discrete global grid reference system.
C	If a third vertical spatial dimension is supported (if the resource's spatial extent bounding box is three dimensional) and that dimension is not part of the discrete global grid system definition, 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 support as axis names any additional dimension (beyond spatial and temporal) as described in the extent property of the collection or dataset description.
E	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 data or an axis defined in the relevant extent property.
F	If a subset parameter including any of the dimensions corresponding to the axes of the discrete global grid reference system is used, the server SHALL return a 400 client error.
G	The implementation SHALL interpret multiple subset parameters, as if all dimension subsetting values were provided in a single subset parameter (comma separated). Example: subset=time("2018-02-12T16:00:0Z":"2018-02-12T20:00:00Z")&subset=atm_pressure_hpa(500:750) is equivalent to subset=time("2018-02-12T16:00:0Z":"2018-02-12T20:00:00Z"),atm_pressure_hpa(500:750)

NOTE: When the *interval* 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 data resulting from the subset.

8.2.2. Parameter *datetime*

REQUIREMENT 6

LABEL /req/data-subsetting/datetime

STATEMENT For specifying a time instant or interval for which to retrieve data from a zone for a non-temporal DGGS:

A The implementation SHALL support a *datetime* parameter expressed corresponding to either a date-time instant or a time interval, conforming to the following syntax (using [ABNF](#)):
 interval-closed = date-time "/" date-time

REQUIREMENT 6

```
interval-open-start = [".."] "/" date-time
interval-open-end   = date-time "/" [".."]
interval            = interval-closed / interval-open-start / interval-open-end
datetime            = date-time / interval
```

B The syntax of date-time is specified by [RFC 3339, 5.6](#).

C Only the portions of the data within the specified interval SHALL be part of the zone data response, performing a trim operation for an interval or a slicing operation for an instant (in the case of a gridded coverage), or a filtering operation for feature data.

D The implementation SHALL support a double-dot (..) or an empty string for the start/end as indicating an unbounded or half-bounded interval (only having a start or end).

E If a datetime parameter is specified requesting zone data where no temporal dimension applies, the implementation SHALL either ignore the parameter or return a 4xx client error.

Note: ISO 8601-2 distinguishes unbounded start/end timestamps (double-dot) and unknown start/end timestamps (empty string). For queries, an unspecified start/end has the same effect as an unbounded start/end.

Examples:

Example 1 — A date-time: February 12, 2018, 23:20:52 GMT:

```
datetime=2018-02-12T23:20:52Z
```

Example 2 — Intervals: February 12, 2018, 00:00:00 GMT to March 18, 2018, 12:31:12 GMT:

```
datetime=2018-02-12T00:00:00Z/2018-03-18T12:31:12Z
```

February 12, 2018, 00:00:00 UTC or later:

```
datetime=2018-02-12T00:00:00Z/..
```

March 18, 2018, 12:31:12 UTC or earlier:

```
datetime=../2018-03-18T12:31:12Z
```



9

REQUIREMENT CLASS “DATA CUSTOM DEPTHS”

REQUIREMENT CLASS “DATA CUSTOM DEPTHS”

9.1. Overview

The Data Custom Depths requirements class extends the zone data retrieval requirements class allowing a client to customize which depths to include as part of a zone data response relative to the requested hierarchy level for which data is being requested. This can be either a single depth, a range of depths, or a list of depths. The depths to include in the response are specified using the zone-depth query parameter. This depth customization ability can be restricted by the capability of the particular zone data response encoding negotiated.

REQUIREMENTS CLASS 4

TARGET TYPE	Web API
PREREQUISITE	http://www.opengis.net/spec/ogcapi-dggs-1/0.0/req/data-retrieval
LABEL	http://www.opengis.net/spec/ogcapi-dggs-1/0.0/req/data-custom-depths

9.2. Requirements

9.2.1. Parameter zone-depth

REQUIREMENT 7

LABEL	/req/data-custom-depths/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 <code>.../dggs/{dggsId}/zones/{zoneId}/data</code> .

REQUIREMENT 7

B	<p>The implementation SHALL accept the following types of values for the zone-depth parameter:</p> <ul style="list-style-type: none"> • 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	<p>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:</p> <ul style="list-style-type: none"> • 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, • .. • <i>n</i> corresponding to all zones for the <i>n</i>'th deeper level in the hierarchy level associated with the requested zone by the indexing scheme.
D	<p>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.</p>
E	<p>If a zone-depth is specified, the operation SHALL return the data at the resolution(s) / scale(s) specified.</p>

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.



10

REQUIREMENT CLASS “ZONE QUERY”

10.1. 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 5

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

10.2. Requirements

10.2.1. 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 <code>.../dggs/{dggsId}/zones</code> .
B	The Implementation SHALL include a link to this resource path in the "Core" <code>.../dggs/{dggsId}</code> resource links 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 <code>compact-zones</code> 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 <code>application/json</code>) where each zone ID is specified as a string within an array of zones assigned to a <code>zones</code> property within a JSON object (e.g., <code>{ "zones": ["1-E-14"s, "1-E-15", "1-F-14", "1-F-15", "1-F-16"] }</code>).

10.2.2. 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 <code>zone-level</code> query parameter for the zone query operation (resource path ending with <code>.../dggs/{dggsId}/zones</code>).
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 <code>zone-level</code> 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 <code>zone-level</code> query parameter.

10.2.3. 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 <code>compact-zones</code> query parameter for the zone query operation (resource path ending with <code>.../dggs/{dggsId}/zones</code>), where a value of <code>true</code> corresponds to the default behavior when the parameter is not specified, and a value of <code>false</code> disables the use of compact-zones in the response. |
| B | When the <code>compact-zones</code> parameter is to <code>false</code> , 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 <code>compact-zones</code> parameter is to <code>true</code> (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. |

10.2.4. Parameter `parent-zone` for hierarchical exploration

The following requirement describe how a client can specify a parent zone to only return zones within this parent zone, allowing to explore a large list in a hierarchical manner (in combination with `zone-level`) as multiple requests and responses.

REQUIREMENT 11

LABEL `/rec/zone-query/parent-zone`

STATEMENT For specifying a parent zone within which to restrict zone listing using a `parent-zone` query parameter:

- | | |
|----------|--|
| A | The Implementation SHALL support a parameter <code>parent-zone</code> zone identifier query parameter. |
| B | When specified, the response SHALL NOT contain zones which are not this parent zone itself or a descendant of that zone. |

10.2.5. Parameter `limit` for paging

The following recommendation 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.

RECOMMENDATION 2

LABEL `/req/zone-query/limit`

STATEMENT For specifying a paging limit for the list of zones using a `limit` query parameter:

- A** The Implementation **SHOULD** support a parameter `limit` integer query parameter, with a minimum value of 1.
- B** The response **SHOULD** 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 **SHOULD** not contain more zones than this maximum value.
- D** If the value of the `limit` parameter is larger than the maximum value, this **SHOULD NOT** result in an error (but instead be replaced by the maximum as the parameter value).
- E** If using compact zones, the parent zones **SHOULD** 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` **SHOULD** be included in a `links` array property of the response, which a client can request to resume listing the zones.

10.2.6. 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
```

REQUIREMENT 12

- * 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 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.

10.2.7. 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.

REQUIREMENT 13

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.

10.2.8. 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:

A	<p>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:</p> <pre>SubsetSpec: "subset"=axisName(intervalOrPoint) axisName: {text} intervalOrPoint: interval \ point interval: low : high low: point \ * high: point \ * point: {number} \ "{text}"</pre> <p>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).</p>
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 support as axis names time for a temporal dataset.
E	The implementation SHALL support as axis names any additional dimension (beyond spatial and temporal) as described in the extent property of the collection or dataset description.
F	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 data or an axis defined in the relevant extent property.

REQUIREMENT 14

G	For a CRS where an axis can wrap around, such as subsetting across the dateline (anti-meridian) in a geographic CRS, a <i>low</i> value greater than <i>high</i> SHALL be supported to indicate an extent crossing that wrapping point.
H	The implementation SHALL interpret the coordinates as values for the named axis of the CRS specified in the <code>subset-crs</code> 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 <code>subset-crs</code> parameter is missing.
I	If the <code>subset</code> parameter including any of the dimensions corresponding to those of the map bounding box is used with a <code>bbox</code> , the server SHALL return a 400 client error.
J	The implementation SHALL interpret multiple <code>subset</code> parameters, as if all dimension subsetting values were provided in a single <code>subset</code> parameter (comma separated). Example: <code>subset=Lat(-90:90)&subset=Lon(-180:180)</code> is equivalent to <code>subset=Lat(-90:90),Lon(-180:180)</code>

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 *interval* 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.

10.2.9. 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 <code>subset-crs</code> with the characteristics identifying the CRS in which the <code>subset</code> 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 <code>subset-crs</code> is not indicated, http://www.opengis.net/def/crs/OGC/1.3/CRS84 SHALL be assumed.
D	The native CRS (<code>storageCRS</code>) SHALL be supported as a value. Other requirements classes may allow additional values (see <code>crs</code> parameter definition).

REQUIREMENT 15

- 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.

10.2.10. 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 `datetime` parameter expressed corresponding to either a date-time instant or a time interval, conforming to the following syntax (using [ABNF](#)):
- ```
interval-closed = date-time "/" date-time
interval-open-start = [".."] "/" date-time
interval-open-end = date-time "/" [".."]
interval = interval-closed / interval-open-start / interval-open-end
datetime = date-time / interval
```
- B** The syntax of `date-time` is specified by [RFC 3339, 5.6](#).
- C** Only the zones with data whose geometry intersect with the specified temporal interval SHALL be part of the zone list response.
- D** The implementation SHALL support a double-dot (..) or an empty string for the start/end as indicating an unbounded or half-bounded interval (only having a start or end).
- E** If a `datetime` parameter is specified requesting zone data where no temporal dimension applies, the implementation SHALL either ignore the parameter or return a 4xx client error.



11

# REQUIREMENT CLASS “ROOT DGGS”

---



## 11.1. Overview

The “Root DGGS” 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 6 |                                                                                                                                       |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| TARGET TYPE          | Web API                                                                                                                               |
| PREREQUISITE         | <a href="http://www.opengis.net/spec/ogcapi-common-1/1.0/req/core">http://www.opengis.net/spec/ogcapi-common-1/1.0/req/core</a>       |
| LABEL                | <a href="http://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/root-dggs">http://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/root-dggs</a> |

## 11.2. Requirements

### 11.2.1. Root DGGS (/dggs)

| REQUIREMENT 17 |                                                                                                                                                                                              |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LABEL          | /req/root-dggs/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. |



12

# REQUIREMENT CLASS “COLLECTION DGGs”

---

## 12.1. Overview

The “Collection DGGS” 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 7

|              |                                                                                                                                                   |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| TARGET TYPE  | Web API                                                                                                                                           |
| PREREQUISITE | <a href="http://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections">http://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections</a>     |
| LABEL        | <a href="http://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/collection-dggs">http://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/collection-dggs</a> |

## 12.2. Requirements

### 12.2.1. Collection DGGS (/collections/{collectionId}/dggs)

#### REQUIREMENT 18

**LABEL** /req/collection-dggs/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.
- B** The /collections/{collectionId}/dggs and /collections/{collectionId}/dggs/{dggsId} resources SHALL include a link to the collection using the link relation type [ogc-rel:geodata].



13

# REQUIREMENTS CLASSES FOR ENCODINGS OF ZONE DATA AND ZONE LIST

---

## REQUIREMENTS CLASSES FOR ENCODINGS OF ZONE DATA AND ZONE LIST

This standard does not mandate any particular encoding or format in which to return the `../dggs/zones` and `../dggs/zones/{zoneId}/data` resources. DGGS Zone Data and Zone Lists can be encoded in any suitable data format. However, it does define requirements class for encodings which are expected to be commonly supported in implementations of this standard. These requirements classes include:

### For zone data:

- GeoJSON Data
- GeoTIFF Data
- netCDF Data
- CoverageJSON Data
- JPEG XL Data
- PNG Data

### For zone list:

- HTML Zone List
- JSON Zone List
- 64-bit Binary Zone List
- GeoJSON Zone List
- GeoTIFF Zone List

### 13.1. Media Types

A table of the media types used in the encoding requirements classes defined in this standard follows.

**Table 2** — Media Types used for encoding requirements classes

| Encoding | media type |
|----------|------------|
|----------|------------|

|                           |                                 |
|---------------------------|---------------------------------|
| GeoTIFF data or zone list | image/tiff; application=geotiff |
| GeoJSON data or zone list | application/geo+json            |
| netCDF zone data          | application/x-netcdf            |
| CoverageJSON zone data    | application/prs.coverage+json   |
| PNG zone data             | image/png                       |
| JPEG XL zone data         | image/jxl                       |
| JSON zone list            | application/json                |
| HTML zone list            | text/html                       |
| Binary 64-bit zone list   | application/x-binary            |

## 13.2. Zone data encodings

### 13.2.1. Requirements Class “GeoTIFF Zone data encoding”

#### 13.2.1.1. Overview

The GeoTIFF requirements class defines support for encoding a DGGS Zone data response according to the [OGC GeoTIFF standard](#).

GeoTIFF is a commonly used format for representing 2-dimensional gridded coverages.

For a DGGS with non-rectangular zones (e.g., hexagonal zones), a GeoTIFF representation can be the minimal bounding rectangle of the zone shape, with NODATA values used outside the bounds of the zone.

### 13.2.2. Requirements Class “GeoJSON Zone data encoding”

#### 13.2.2.1. Overview

The GeoJSON requirements class defines support for encoding a DGGS Zone data response according to the [GeoJSON specification](#).

GeoJSON is a commonly used format for representing features with simple geometries and related properties. GeoJSON is simple to understand and well supported by tools and software libraries. Since most Web developers are comfortable with using JSON-based formats, supporting GeoJSON is recommended for a vector representation of DGGS Zone data.

### **13.2.3. Requirements Class “netCDF zone data encoding”**

#### **13.2.3.1. Overview**

The netCDF requirements class defines support for encoding a DGGS Zone data response according to the [OGC netCDF standard](#).

### **13.2.4. Requirements Class “CoverageJSON zone data encoding”**

#### **13.2.4.1. Overview**

The CoverageJSON requirements class defines support for encoding a DGGS Zone data response according to the [OGC CoverageJSON Community Standard](#).

### **13.2.5. Requirements Class “JPEG-XL Zone data encoding”**

#### **13.2.5.1. Overview**

The JPEG XL requirements class defines support for encoding a DGGS Zone data response according to the [JPEG XL ISO/IEC 18181 Standard](#).

JPEG XL supports a large number of bands, image dimensions, and floating point values.

### **13.2.6. Requirements Class “PNG Zone data encoding”**

#### **13.2.6.1. Overview**

The PNG requirements class defines support for encoding a DGGS Zone data response according to the [W3C Portable Network Graphics \(PNG\) Specification \(ISO/IEC 15948:2003\)](#).

Because PNG encoding is limited to an integer data values, this requirements class defines additional parameters and response headers allowing a client to request a specific scale factor and offset to be used to quantize data values, and allowing the implementation to inform the client of the scale factor and offset used for that quantization.

## 13.3. Zone list encodings

---

### 13.3.1. Requirements Class “JSON zone list encoding”

#### 13.3.1.1. Overview

The JSON zone list encoding requirements class defines the ability to retrieve a JSON response for the DGGS zone list resource.

The response consists of a simple JSON object with a "zones" array property listing the zone identifiers as strings. The object can also include a "links" objects linking back to other resources of the DGGS.

This requirement class provides an easy to implement, reasonably efficient and interoperable mechanism to exchange lists of zone identifiers.

### 13.3.2. Requirements Class “HTML zone list encoding”

#### 13.3.2.1. Overview

The HTML zone list encoding requirements class defines at a high level the ability to retrieve an HTML response for the DGGS zone list resource intended primarily for users accessing the API from a Web browser.

The exact content of the HTML response is not prescribed, leaving the flexibility for implementations to choose a preferred approach.

HTML is the core language of the World Wide Web. An API that supports HTML will support browsing the spatial resources with a web browser and will also enable search engines to crawl and index those resources.

### 13.3.3. Requirements Class “Binary 64-bit integer zone list encoding”

#### 13.3.3.1. Overview

The binary 64-bit integer zone list encoding requirements class defines the ability to retrieve a binary response for the DGGS zone list resource for DGGRS whose zone identifiers can be expressed as a single 64-bit integer.



The response consists of a 64-bit integer count of zones, followed by that count of zones, also 64-bit integers. With compact zones and additional HTTP content encoding compression, this provides an optimal way to exchange DGGS zone lists.

### 13.3.4. Requirements Class “GeoJSON zone list encoding”

#### 13.3.4.1. Overview

The GeoJSON zone list encoding requirements class defines support for encoding a DGGS Zone list response according to the [GeoJSON specification](#).

The response is a feature collection where each feature represents a zone in the list. The geometry of the zone should be polygons or multipolygons (e.g., in the case of zone geometry which must be split on the anti-meridian or pole). The zone identifier is represented in the `id` field of each feature.

While not intended to efficiently exchange zone list, as the response carries the geometry of the zones which could easily be computed client-side which takes considerable bandwidth and is not necessary for a DGGS client working natively with the same DGGRS, this requirement class provides an easy way to readily visualize the response in a variety of tools. This requirements class is therefore intended for convenience, demonstration and educational purposes.

### 13.3.5. Requirements Class “GeoTIFF zone list encoding”

#### 13.3.5.1. Overview

The GeoTIFF zone list encoding requirements class defines support for encoding a DGGS Zone list response according to the [OGC GeoTIFF standard](#), intended primarily for DGGS with rectangular zones.

The response is a 2D gridded coverage where each cell is a zone in the list, in the coordinate reference system of that DGGS.

For DGGS with non-rectangular zones, the resolution would need to be higher than a single pixel per the most detailed zone to be returned in order to be able to recognize the zone geometry, and would therefore be very sub-optimal. While for rectangular zones it would be easy to identify zones with a one-to-one correspondence, recognizing non-rectangular zones would be significantly more difficult.

While not intended to efficiently exchange zone list, this requirement class provides an easy way to readily visualize the response in a variety of tools. This requirements class is therefore intended for convenience, demonstration and educational purposes.



A

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





# 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

---



## 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



# ANNEX C (INFORMATIVE) REVISION HISTORY

---



## 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 – <a href="#">github commit – 316d37f</a>                                     |
| 2022-07-22 | 0.3     | Jerome St Louis | all                      | Initial set up of conformance classes clauses – <a href="#">github commit – b76073c</a>              |
| 2022-07-26 | 0.4     | Gobe Hobona     | all                      | Fixes problem that was breaking auto-build – <a href="#">github commit – a13c2a8</a>                 |
| 2022-10-04 | 0.5     | Jerome St Louis | 7-Data Retrieval         | Initial take at Data Retrieval Conformance Class – <a href="#">github commit – b67f290</a>           |
| 2022-11-11 | 0.6     | Jerome St Louis | 7-Data Retrieval         | Split multi-done retrieval into separate conformance class – <a href="#">github commit – 06a4260</a> |
| 2022-11-11 | 0.7     | Jerome St Louis | 8-zone query             | Initial progress on Zone Query conformance class – <a href="#">github commit – a208502</a>           |
| 2022-11-11 | 0.8     | Jerome St Louis | 21-038; 0-front material | Added preface, abstract, Jerome added as editor – <a href="#">github commit – 71e76c6</a>            |
| 2022-11-11 | 0.9     | Jerome St Louis | 7-Data Retrieval         | Clarification regarding resolution – <a href="#">github commit – c43bfad</a>                         |
| 2022-11-11 | 0.10    | Jerome St Louis | 8-zone query             | Clarifications – <a href="#">github commit – cee3685</a>                                             |
| 2023-02-17 | 0.11    | Jerome St Louis | 7-Data Retrieval         | Added zone-depth parameter – <a href="#">github commit – 2b8fa58</a>                                 |



# BIBLIOGRAPHY

---





## BIBLIOGRAPHY

---

- [1] OGC: *OGC Web Service Common Implementation Specification*, [OGC 06-121r9](#)
- [2] OGC: *OGC Testbed 12 Annex B: Architecture* (2015).