



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

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD. THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

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://portal.opengeospatial.org/public_ogc/change_request.php

Copyright notice

Copyright © 2024 Open Geospatial Consortium
To obtain additional rights of use, visit <http://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	xiv
II.	KEYWORDS	xiv
III.	PREFACE	xv
IV.	SECURITY CONSIDERATIONS	xvi
V.	SUBMITTING ORGANIZATIONS	xvii
VI.	SUBMITTERS	xvii
1.	SCOPE	2
2.	CONFORMANCE	4
2.1.	Requirements classes defining resources	4
2.2.	Requirements classes defining origins	5
2.3.	Requirements classes defining parameters	5
2.4.	Requirements classes defining resource representations	5
2.5.	Summary of conformance URLs	6
3.	NORMATIVE REFERENCES	10
4.	TERMS AND DEFINITIONS	13
5.	CONVENTIONS	19
5.1.	Identifiers	19
5.2.	Link relations	19
5.3.	Use of HTTPS	21
6.	REQUIREMENT CLASS “CORE”	23
6.1.	Overview	23
6.2.	Listing available DGGRS (.../dggs)	23
6.3.	Discrete global grid reference system information (.../dggs/{dggrsId})	24
6.4.	Retrieving zone information (.../dggs/{dggrsId}/zones/{zonId})	25
7.	REQUIREMENT CLASS “DATA RETRIEVAL”	30
7.1.	Overview	30
7.2.	Retrieving data from a zone (.../dggs/{dggrsId}/zones/{zonId}/data)	30
8.	REQUIREMENT CLASS “DATA SUBSETTING”	33

8.1. Overview	33
8.2. Parameter subset	33
8.3. Parameter datetime	34
9. REQUIREMENT CLASS “DATA CUSTOM DEPTHS”	38
9.1. Overview	38
9.2. Parameter zone-depth	38
10. REQUIREMENT CLASS “ZONE QUERY”	41
10.1. Overview	41
10.2. Listing zones (.../dggs/{dggrsId}/zones)	41
10.3. Parameter zone-level	43
10.4. Parameter compact-zones	43
10.5. Parameter parent-zone for hierarchical exploration	44
10.6. Parameter limit for paging	44
10.7. Parameter bbox	45
10.8. Parameter bbox-crs	46
10.9. Parameter subset	46
10.10. Parameter subset-crs	48
10.11. Parameter datetime	48
11. REQUIREMENT CLASS “FILTERING ZONE QUERIES WITH CQL2”	51
11.1. Overview	51
11.2. Parameter filter	51
12. REQUIREMENT CLASS “ROOT DGGS”	54
12.1. Overview	54
12.2. Root DGGS (/dggs)	54
13. REQUIREMENT CLASS “COLLECTION DGGS”	56
13.1. Overview	56
13.2. Collection DGGS (/collections/{collectionId}/dggs)	56
14. REQUIREMENTS CLASSES FOR ENCODINGS OF ZONE DATA	58
14.1. Media Types (for zone data)	58
14.2. Requirements Class “DGGS-JSON Zone data encoding”	59
14.3. Requirements Class “DGGS-UB-JSON Zone data encoding”	64
14.4. Requirements Class “GeoTIFF Zone data encoding”	65
14.5. Requirements Class “GeoJSON Zone data encoding”	66
14.6. Requirements Class “FG-JSON Zone data encoding”	68
14.7. Requirements Class “netCDF zone data encoding”	70
14.8. Requirements Class “Zarr zone data encoding”	71
14.9. Requirements Class “CoverageJSON zone data encoding”	72
14.10. Requirements Class “JPEG-XL Zone data encoding”	74
14.11. Requirements Class “PNG Zone data encoding”	75
15. REQUIREMENTS CLASSES FOR ENCODINGS OF ZONE LIST	78

15.1. Media Types (for zone list)	78
15.2. Requirements Class “JSON zone list encoding”	79
15.3. Requirements Class “HTML zone list encoding”	81
15.4. Requirements Class “Binary 64-bit integer zone list encoding”	82
15.5. Requirements Class “GeoJSON zone list encoding”	83
15.6. Requirements Class “FG-JSON zone list encoding”	85
15.7. Requirements Class “GeoTIFF zone list encoding”	86
16. REQUIREMENTS CLASS FOR API DEFINITION OPERATION IDS	89
16.1. Overview	89
16.2. Operation IDs	89
ANNEX A (NORMATIVE) CONFORMANCE CLASS ABSTRACT TEST SUITE	93
A.1. Conformance Class “Core”	93
A.2. Conformance Class “Data Retrieval”	95
A.3. Conformance Class “Data Subsetting”	95
A.4. Conformance Class “Data Custom Depths”	96
A.5. Conformance Class “Zone Query”	96
A.6. Conformance Class “Filtering Zone Queries with CQL2”	97
A.7. Conformance Class “Root DGGS”	97
A.8. Conformance Class “Collection DGGS”	98
A.9. Conformance Class “DGGS-JSON encoding for Zone Data”	98
A.10. Conformance Class “DGGS-UBJSON encoding for Zone Data”	98
A.11. Conformance Class “GeoTIFF encoding for Zone Data”	99
A.12. Conformance Class “GeoJSON encoding for Zone Data”	99
A.13. Conformance Class “FG-JSON encoding for Zone Data”	99
A.14. Conformance Class “netCDF encoding for Zone Data”	100
A.15. Conformance Class “Zarr encoding for Zone Data”	100
A.16. Conformance Class “CoverageJSON encoding for Zone Data”	101
A.17. Conformance Class “JPEG XL Encoding for Zone Data”	101
A.18. Conformance Class “PNG encoding for Zone Data”	101
A.19. Conformance Class “JSON encoding for Zone Listing”	102
A.20. Conformance Class “HTML encoding for Zone Listing”	102
A.21. Conformance Class “Binary 64-bit integer encoding for Zone Listing”	102
A.22. Conformance Class “GeoJSON encoding for Zone Listing”	103
A.23. Conformance Class “FG-JSON encoding for Zone Listing”	103
A.24. Conformance Class “GeoTIFF encoding for Zone Listing”	104
A.25. Conformance Class “Operation IDs”	104
ANNEX B (INFORMATIVE) DISCRETE GLOBAL GRID REFERENCE SYSTEM DEFINITIONS	107
B.1. JSON Schema for defining a DGGRS	107
B.2. ISEA9R DGGRS definition	113
B.3. ISEA3H DGGRS definition	116
B.4. GNOSIS Global Grid DGGRS definition	119
ANNEX C (INFORMATIVE) EXAMPLES	124

C.1. Collection description	124
C.2. Listing available DGGRSs	125
C.3. Retrieving the description of a specific DGGRS	126
C.4. Retrieving information for a specific DGGRS zone	127
C.5. Simple zone queries	132
C.6. Compact zone queries	149
C.7. Filtering zone queries using CQL2 expressions	165
C.8. Retrieving data for a DGGRS Zone	175
 ANNEX D (INFORMATIVE) REVISION HISTORY	178
 BIBLIOGRAPHY	180

LIST OF TABLES

Table 1 – Conformance class URLs	7
Table 2	35
Table 3 – Media Types used for zone data encoding requirements classes	58
Table 4 – Media Types used for zone list encoding requirements classes	78
Table 5 – API operation identifier suffixes	90
Table D.1	178

LIST OF FIGURES

Figure 6 – Example encoding for DGGS-JSON	59
Figure 7 – A more complex example encoding for DGGS-JSON with more variables and additional dimensions beyond those of the DGGRS	60
Figure 8 – JSON Schema for DGGS-JSON	63
Figure 9 – JSON Schema for a DGGS Zone Query response	79
Figure B.1 – JSON Schema for the DGGRS Definition	107
Figure B.2 – DGGRS Definition for ISEA9R	113
Figure B.3 – The ISEA9R squared root rhombuses (level 0) in the rotated and sheared ISEA planar CRS mapped to a 5×6 space whose axes are aligned with the zone edges, compatible with a 2DTMS definition (imagery from NASA Earth Observatory’s Blue Marble: Next Generation)	115
Figure B.4 – The ISEA9R level 0 zone identifiers, shown in the original ISEA planar CRS (GEBCO 2014 bathymetry)	116
Figure B.5 – The ISEA9R level 1 zone identifiers, shown in the original ISEA planar CRS (GEBCO 2014 bathymetry)	116
Figure B.6 – DGGRS Definition for ISEA3H using indexing based on ISEA9R	117

Figure B.7 – The ISEA3H level 0 zones and associated identifiers (derived from ISEA9R level 0), shown on an ISEA planar projection (GEBCO 2014 bathymetry)	118
Figure B.8 – The ISEA3H level 1 zones and associated identifiers (derived from ISEA9R level 0), shown on an ISEA planar projection (GEBCO 2014 bathymetry)	119
Figure B.9 – The ISEA3H level 2 zones and associated identifiers (derived from ISEA9R level 1), shown on an ISEA planar projection (GEBCO 2014 bathymetry)	119
Figure B.10 – DGGRS Definition for the GNOSIS Global Grid	120
Figure B.11 – The GNOSIS Global Grid level 0 zones and associated identifiers (GEBCO 2014 bathymetry)	121
Figure B.12 – The GNOSIS Global Grid level 1 zones and associated identifiers (GEBCO 2014 bathymetry)	121
Figure B.13 – The GNOSIS Global Grid level 2 zones and associated identifiers (GEBCO 2014 bathymetry)	122
Figure C.1 – Sample HTML response of an ISEA3H DGGRS zone information resource for GEBCO data	131
Figure C.2 – GeoJSON response for zone query for ISEA3H DGGRS at level 1 visualized in QGIS	134
Figure C.3 – GeoJSON response for zone query for ISEA3H DGGRS at level 1 for a (40°N, 30°E) to (60°N,50°E) bounding box visualized in QGIS	135
Figure C.4 – Sample HTML response for a zone query for ISEA3H DGGRS at level 1 for a (40°N, 30°E) to (60°N,50°E) bounding box	135
Figure C.5 – Sample HTML response for a zone query for ISEA3H DGGRS at level 8 for a (40°N, 30°E) to (60°N,50°E) bounding box	136
Figure C.6 – GeoJSON response for zone query for ISEA3H DGGRS at level 6 for a Viewfinder Panoramas land elevation collection, visualized in QGIS	137
Figure C.7 – GeoJSON response for zone query for ISEA3H DGGRS at level 10, using a (40°N, 30°E) to (40°N,50°E) bounding box, for a Viewfinder Panoramas land elevation collection, visualized in QGIS	138
Figure C.8 – GeoJSON response for zone query for ISEA3H DGGRS at level 6, for a Natural Earth states and provinces collection, visualized in QGIS	139
Figure C.9 – GeoJSON response for zone query for ISEA3H DGGRS at level 8, for parent zone A6-0-E, for a Viewfinder Panoramas land elevation collection, visualized in QGIS	140
Figure C.10 – GeoJSON response for zone query for ISEA3H DGGRS at level 16, for parent zone E6-317-A, for a Viewfinder Panoramas land elevation collection, visualized in QGIS	141
Figure C.11 – GeoJSON response for zone query for ISEA3H DGGRS at level 2, for parent zone A6-0-E (outline shown), for a Viewfinder Panoramas land elevation collection, visualized in QGIS	142
Figure C.12 – GeoJSON response for zone query for ISEA9R DGGRS at level 3 for a Viewfinder Panoramas land elevation collection visualized in QGIS	143
Figure C.13 – GeoJSON response for zone query for ISEA9R DGGRS at level 4 for a (40°N, 30°E) to (60°N,50°E) bounding box visualized in QGIS	144
Figure C.14 – GeoJSON response for zone query for ISEA9R DGGRS at level 4 for parent zone B6-2, for a Viewfinder Panoramas, visualized in QGIS	145

Figure C.15 – GeoJSON response for zone query for ISEA9R DGGRS at level 8 for parent zone E6-317, for a Viewfinder Panoramas, visualized in QGIS	145
Figure C.16 – GeoJSON response for zone query for GNOSIS Global Grid DGGRS at level 5 for a Viewfinder Panoramas land elevation collection visualized in QGIS	146
Figure C.17 – GeoJSON response for zone query for GNOSIS Global Grid DGGRS at level 7 for a (40°N, 30°E) to (60°N,50°E) bounding box visualized in QGIS	147
Figure C.18 – GeoJSON response for zone query for GNOSIS Global Grid DGGRS at level 8 for parent zone 2-2-9, for a Viewfinder Panoramas, visualized in QGIS	148
Figure C.19 – GeoJSON response for zone query for GNOSIS Global Grid DGGRS at level 13 for parent zone 6-1F-98, for a Viewfinder Panoramas, visualized in QGIS	149
Figure C.20 – GeoJSON compact zones response querying ISEA3H DGGRS at level 8 for a (40°N, 30°E) to (60°N,50°E) bounding box	150
Figure C.21 – GeoJSON compact zones response for querying ISEA3H DGGRS at level 6 for a Viewfinder Panoramas land elevation collection, visualized in QGIS	151
Figure C.22 – GeoJSON compact zones response for querying ISEA3H DGGRS at level 10, using a (40°N, 30°E) to (40°N,50°E) bounding box, for a Viewfinder Panoramas land elevation collection, visualized in QGIS	151
Figure C.23 – GeoJSON compact zones response for querying ISEA3H DGGRS at level 16, for parent zone E6-317-A, for a Viewfinder Panoramas land elevation collection, visualized in QGIS	152
Figure C.24 – GeoJSON compact zones response for querying ISEA9R DGGRS at level 3 for a Viewfinder Panoramas land elevation collection visualized in QGIS	153
Figure C.25 – GeoJSON compact zones response for querying ISEA9R DGGRS at level 4 for a (40°N, 30°E) to (60°N,50°E) bounding box visualized in QGIS	154
Figure C.26 – GeoJSON compact zones response for querying ISEA9R DGGRS at level 4 for parent zone B6-2, for a Viewfinder Panoramas, visualized in QGIS	155
Figure C.27 – GeoJSON compact zones response for querying ISEA9R DGGRS at level 8 for parent zone E6-317, for a Viewfinder Panoramas, visualized in QGIS	156
Figure C.28 – GeoJSON compact zones response for querying GNOSIS Global Grid DGGRS at level 5 for a Viewfinder Panoramas land elevation collection visualized in QGIS	158
Figure C.29 – GeoJSON compact zones response for querying GNOSIS Global Grid DGGRS at level 7 for a (40°N, 30°E) to (60°N,50°E) bounding box visualized in QGIS	161
Figure C.30 – GeoJSON compact zones response for querying GNOSIS Global Grid DGGRS at level 8 for parent zone 2-2-9, for a Viewfinder Panoramas, visualized in QGIS	163
Figure C.31 – GeoJSON compact zones response for querying GNOSIS Global Grid DGGRS at level 13 for parent zone 6-1F-98, for a Viewfinder Panoramas, visualized in QGIS	165
Figure C.32 – GeoJSON compact zones response for querying ISEA3H DGGRS for GEBCO 2014 bathymetry with CQL2 expression Elevation < 0 visualized in QGIS	167
Figure C.33 – GeoJSON compact zones response for querying ISEA3H DGGRS for GEBCO 2014 bathymetry with CQL2 expression Elevation < -4000 visualized in QGIS	167
Figure C.34 – GeoJSON compact zones response for querying ISEA3H DGGRS for GEBCO 2014 bathymetry with CQL2 expression Elevation > 3000 visualized in QGIS	168

Figure C.35 – GeoJSON compact zones response for querying ISEA3H DGGRS for GEBCO 2014 bathymetry with CQL2 expression Elevation < -4000 or Elevation > 3000 visualized in QGIS	168
Figure C.36 – GeoJSON compact zones response for querying ISEA3H DGGRS for GEBCO 2014 bathymetry with a CQL2 expression using S_INTERSECTS() for a WKT triangular polygon visualized in QGIS	169
Figure C.37 – GeoJSON compact zones response for querying ISEA3H DGGRS for GEBCO 2014 bathymetry with a CQL2 expression using both Elevation < -5500 as well as S_INTERSECTS() for a WKT polygon visualized in QGIS	171
Figure C.38 – GeoJSON compact zones response for querying ISEA3H DGGRS for CMIP5 daily surface temperature from Copernicus Climate Data Store comparing delta with CQL2 expression (tasmax - tasmin) > 10 visualized in QGIS	172
Figure C.39 – GeoJSON compact zones response for querying ISEA3H DGGRS for Ottawa buildings (© OpenStreetMap contributors) with CQL2 expression name='National Arts Centre' at level 30, visualized in QGIS	173
Figure C.40 – GeoJSON compact zones response for querying ISEA9R DGGRS for Ottawa buildings (© OpenStreetMap contributors) with CQL2 expression name='East Block' at level 15, visualized in QGIS	174
Figure C.41 – GeoJSON compact zones response for querying GNOSIS Global Grid DGGRS for Ottawa buildings (© OpenStreetMap contributors) with CQL2 expression name='Centre Block' at level 24, visualized in QGIS	175
Figure C.42 – Visualization of a DGGS-JSON response for querying ISEA3H DGGRS for zone A2-0-A of GEBCO 2014 bathymetry at a relative depth of 8	176

LIST OF RECOMMENDATIONS

REQUIREMENTS CLASS 1	23
REQUIREMENTS CLASS 2	30
REQUIREMENTS CLASS 3	33
REQUIREMENTS CLASS 4	38
REQUIREMENTS CLASS 5	41
REQUIREMENTS CLASS 6	51
REQUIREMENTS CLASS 7	54
REQUIREMENTS CLASS 8	56
REQUIREMENTS CLASS 9: REQUIREMENTS CLASS JSON ZONE DATA	62
REQUIREMENTS CLASS 10: REQUIREMENTS CLASS UB-JSON	64
REQUIREMENTS CLASS 11: REQUIREMENTS CLASS GEOTIFF DATA	65
REQUIREMENTS CLASS 12: REQUIREMENTS CLASS GEOJSON	67

REQUIREMENTS CLASS 13: REQUIREMENTS CLASS FG-JSON DATA	68
REQUIREMENTS CLASS 14: REQUIREMENTS CLASS NETCDF	70
REQUIREMENTS CLASS 15: REQUIREMENTS CLASS ZARR	71
REQUIREMENTS CLASS 16: REQUIREMENTS CLASS COVERAGEJSON DATA	72
REQUIREMENTS CLASS 17: REQUIREMENTS CLASS JPEG XL DATA	74
REQUIREMENTS CLASS 18: REQUIREMENTS CLASS PNG DATA	75
REQUIREMENTS CLASS 19: REQUIREMENTS CLASS JSON ZONE LIST	79
REQUIREMENTS CLASS 20: REQUIREMENTS CLASS HTML ZONE LIST	81
REQUIREMENTS CLASS 21: REQUIREMENTS CLASS 64-BIT BINARY ZONE LIST	82
REQUIREMENTS CLASS 22: REQUIREMENTS CLASS GEOJSON ZONE LIST	84
REQUIREMENTS CLASS 23: REQUIREMENTS CLASS FG-JSON ZONE LIST	85
REQUIREMENTS CLASS 24: REQUIREMENTS CLASS GEOTIFF ZONE LIST	86
REQUIREMENTS CLASS 25	89
REQUIREMENT 1	23
REQUIREMENT 2	24
REQUIREMENT 3	26
REQUIREMENT 4	31
REQUIREMENT 5	33
REQUIREMENT 6	34
REQUIREMENT 7	38
REQUIREMENT 8	42
REQUIREMENT 9	43
REQUIREMENT 10	43
REQUIREMENT 11	44
REQUIREMENT 12	45
REQUIREMENT 13	46
REQUIREMENT 14	47
REQUIREMENT 15	48
REQUIREMENT 16	49
REQUIREMENT 17	51
REQUIREMENT 18	54
REQUIREMENT 19	56
REQUIREMENT 20	62

REQUIREMENT 21	64
REQUIREMENT 22	66
REQUIREMENT 23	67
REQUIREMENT 24	68
REQUIREMENT 25	70
REQUIREMENT 26	71
REQUIREMENT 27	73
REQUIREMENT 28	74
REQUIREMENT 29	75
REQUIREMENT 30	79
REQUIREMENT 31	82
REQUIREMENT 32	83
REQUIREMENT 33	84
REQUIREMENT 34	85
REQUIREMENT 35	87
REQUIREMENT 36	89
RECOMMENDATION 1	25
RECOMMENDATION 2	26
RECOMMENDATION 3	27
RECOMMENDATION 4	42
RECOMMENDATION 5	44
RECOMMENDATION 6	66
RECOMMENDATION 7	66
RECOMMENDATION 8	67
RECOMMENDATION 9	67
RECOMMENDATION 10	68
RECOMMENDATION 11	69
RECOMMENDATION 12	69
RECOMMENDATION 13	69
RECOMMENDATION 14	69
RECOMMENDATION 15	70
RECOMMENDATION 16	71
RECOMMENDATION 17	72

RECOMMENDATION 18	73
RECOMMENDATION 19	73
RECOMMENDATION 20	74
RECOMMENDATION 21	75
RECOMMENDATION 22	76
RECOMMENDATION 23	76
RECOMMENDATION 24	76
RECOMMENDATION 25	80
RECOMMENDATION 26	80
RECOMMENDATION 27	80
RECOMMENDATION 28	82
RECOMMENDATION 29	82
RECOMMENDATION 30	84
RECOMMENDATION 31	84
RECOMMENDATION 32	86
RECOMMENDATION 33	86
RECOMMENDATION 34	87
RECOMMENDATION 35	87
RECOMMENDATION 36	87
PERMISSION 1	25
CONFORMANCE CLASS A.1	93
CONFORMANCE CLASS A.2	95
CONFORMANCE CLASS A.3	95
CONFORMANCE CLASS A.4	96
CONFORMANCE CLASS A.5	96
CONFORMANCE CLASS A.6	97
CONFORMANCE CLASS A.7	97
CONFORMANCE CLASS A.8	98
CONFORMANCE CLASS A.9	98
CONFORMANCE CLASS A.10	98
CONFORMANCE CLASS A.11	99
CONFORMANCE CLASS A.12	99
CONFORMANCE CLASS A.13	100

CONFORMANCE CLASS A.14	100
CONFORMANCE CLASS A.15	100
CONFORMANCE CLASS A.16	101
CONFORMANCE CLASS A.17	101
CONFORMANCE CLASS A.18	101
CONFORMANCE CLASS A.19	102
CONFORMANCE CLASS A.20	102
CONFORMANCE CLASS A.21	103
CONFORMANCE CLASS A.22	103
CONFORMANCE CLASS A.23	103
CONFORMANCE CLASS A.24	104
CONFORMANCE CLASS A.25	104

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 specified 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, DGGS, discrete global grid, ISEA, hexagonal grids

PREFACE

This document defines the OGC API – *Discrete Global Grid Systems – Part 1: Core Standard*. Suggested additions, changes and comments on this standard are welcome and encouraged. Such suggestions may be submitted as an issue on the [OGC API – DGGS GitHub repository](#).

OGC API Standards define modular API building blocks to spatially enable Web APIs in a consistent way. OGC API Standards use the OpenAPI specification for describing 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.

SECURITY CONSIDERATIONS

OGC API – DGGS – Part 1 only defines HTTP GET operations. As such the security considerations are limited to those applicable to a “read only” service. However, implementations of OGC API – DGGS have to process resource paths and query parameters in a way that cannot be used by a client to inject malicious queries that makes available unforeseen data or even force the server to perform unwanted or dangerous actions. The following paragraphs enumerate some security considerations.

Due to the flexibility in querying DGGS zones, implementations of the OGC API – DGGS that are not optimized can easily encounter requests that take some time to resolve. If several of these requests are processed simultaneously, the server can become slow or unresponsive. Servers should take advantage of the space partitioning structures of the implemented DGGRSSs that guarantee a deterministic maximum amount of data to process, ensuring that maximum refinement level is used for regions covering a larger area, and refusing requests which would otherwise be problematic for the server. Servers should also set and apply reasonable limits (e.g., maximum number of zones to return, etc.) to prevent Denial of Service attacks.

Some OGC API – DGGS deployments may assign different roles to different users that may result in accessing different collections or geographical areas for which data can be retrieved or zones can be queried. The access control can be described in the OpenAPI definition as discussed in OGC API – Common (https://docs.ogc.org/is/19-072/19-072.html#rc_oas30-security). Servers should take care that all resources in all representations and ways they can be requested (e.g., adding query parameters) are managed consistently.

Another security consideration is that DGGS data or DGGS zone lists returned by a DGGS API endpoint may be positionally incorrect and potentially guide users to wrong locations. One possible way this situation could happen is the manipulation by a malicious actor of a projection library installed on a server deployment.

Using HTTPS queries is preferred to using HTTP. The obvious reason is the DGGS response returned by the DGGS API endpoint should be only accessible to the requesting user. Another reason is related to the public accessibility of the request itself. Request to a DGGS API endpoint reveal geographical extents that might be associated with private or sensitive information. For instance, users commonly visit “home” or “work” or “target places” (such as holiday destinations or churches). These requests can be used to predict personal or private activities.

SUBMITTING ORGANIZATIONS

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

- Pangaea Innovations Pty. Ltd.
- Ecere Corporation
- Landcare Research New Zealand
- Peking University Collaborative Innovation Center for Geospatial Big Data
- European Commission Joint Research Centre
- Natural Resources Canada
- University of Tartu
- European Space Agency

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
Robert Gibb	Landcare Research New Zealand
Fuhu Ren / 任伏虎	Peking University Collaborative Innovation Center for Geospatial Big Data
Peter Strobl	European Commission Joint Research Centre
Ryan Ahola	Natural Resources Canada
Alexander Kmoch	University of Tartu
Ferran Gascon	European Space Agency

1

SCOPE

SCOPE

This OGC Standard specifies a Web API that enables the direct retrieval of geospatial data for a specified area, time and resolution of interest, based on a specific DGGS and indexing scheme and/or the query of DGGS resources to return a list of DGGS Zones from which data is available. Zone queries can include building blocks designed for geospatial queries such as a filter expressed using the [OGC Common Query Language \(CQL2\)](#).

This standard also specifies a number of encodings for retrieving both DGGS data and lists of zones, including JSON and compact binary representations.

An informative annex presents a JSON schema for describing a Discrete Global Grid Reference System (DGGRS), as well as examples of such DGGRS definitions which can be implemented in connection with this standard.

2

CONFORMANCE

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 twenty one 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. However, a Web API with practical use needs to implement either the Data Retrieval and/or the Zone Query requirement class.

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, as well as either the “Data Retrieval” and/or the “Zone Query” requirements class, defined in Annex A (normative).

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

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

2.1. Requirements classes defining resources

In addition to the Core requirement class defining the `../dggs`, `../dggs/{dggrsId}` and `./dggs/{dggrsId}/zones/{zoneId}` resources, either the Zone Data retrieval (`../dggs/{dggrsId}/zones/{zoneId}/data`) or Zone Query requirement classes (`../dggs/{dggrsId}/zones` with support for parameters such as `zone-level=`, `parent-zone=`, `compact-zones=`, `bbox=` and `datetime=`) need to be implemented for a DGGS API implementation to have practical use.

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

2.2. Requirements classes defining origins

These requirement classes define DGGS resources for the API (/dggs) and collections (/collections/{collectionId}/dggs).

- Root DGGS
- Collection DGGS

2.3. Requirements classes defining parameters

Two requirement classes adds support for the subset, properties and zone-depth parameters to zone data resources:

- Zone data subsetting
- Zone data custom depths

A requirement classes adds support for the filter parameter for zone query resources:

- Filtering Zone Queries with CQL2

2.4. Requirements classes defining resource representations

2.4.1. Zone data encodings

The Data Encoding Requirements Classes address support for formats commonly used for encoding geospatial data, as well as encodings based on JSON and UB-JSON using a schema specifically designed for the purpose of encoding one or more depths of zone data referenced to a DGGS, including optional support for multiple fields (properties) and additional dimensions beyond those of the DGGS.

- DGGS-JSON Data
- DGGS-UB-JSON Data
- GeoJSON Data

- FG-JSON Data
- GeoTIFF Data
- netCDF Data
- Zarr Data
- CoverageJSON Data
- JPEG XL Data
- PNG Data

2.4.2. Zone list encodings

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

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

2.4.3. Operation IDs in API definitions

The *Operation IDs* Requirements Class defines requirements for using specific operation IDs in API definitions, which can be combined with requirement classes for specific API definition, such as the OpenAPI 3.0 requirement class of *OGC API – Common – Part 1: Core*, facilitating the identification of DGGS resources.

- **Operation IDs**

2.5. Summary of conformance URIs

Table 1 – Conformance class URIs

CORRESPONDING REQUIREMENTS CLASS	CONFORMANCE CLASS URI
Core	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/core
Zone Data Retrieval	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-retrieval
Zone Query	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-query
Filtering Zone Queries with CQL2	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/cql2-query
Root DGGS	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/root-dggs
Collection DGGS	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/collection-dggs
Data subsetting	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-subsetting
Data custom depths	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-custom-depths
DGGS-JSON Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-json
DGGS-UB-JSON Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-ubjson
GeoJSON Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-geojson
FG-JSON Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-fgjson
GeoTIFF Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-tiff
netCDF Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-netcdf
CoverageJSON Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-coveragejson
Zarr Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-zarr
JPEG XL Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-jpegxl
PNG Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-png
JSON Zone List	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-json
HTML Zone List	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-html
64-bit Binary Zone List	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-uint64

CORRESPONDING REQUIREMENTS CLASS	CONFORMANCE CLASS URI
GeoJSON Zone List	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-geojson
FG-JSON Zone List	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-fjson
GeoTIFF Zone List	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-tiff
Operation IDs	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/operation-ids

3

NORMATIVE REFERENCES

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.

Robert Gibb: OGC 20-040r3, *Topic 21 – Discrete Global Grid Systems – Part 1 Core Reference system and Operations and Equal Area Earth Reference System*. Open Geospatial Consortium (2021). <http://www.opengis.net/doc/AS/dggs/2.0>.

T. Bray (ed.): IETF RFC 8259, *The JavaScript Object Notation (JSON) Data Interchange Format*. RFC Publisher (2017). <https://www.rfc-editor.org/info/rfc8259>.

Charles Heazel: OGC 19-072, *OGC API – Common – Part 1: Core*. Open Geospatial Consortium (2023). <http://www.opengis.net/doc/is/ogcapi-common-1/1.0.0>.

Charles Heazel: *OGC API – Common – Part 2: Geospatial Data (Draft)*. OGC 20-024, Open Geospatial Consortium, <https://docs.ogc.org/DRAFTS/20-024.html>

Open API Initiative: OpenAPI Specification 3.0.3, <https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.3.md>

Ben Domenico: OGC 10-090r3, *OGC Network Common Data Form (NetCDF) Core Encoding Standard version 1.0*. Open Geospatial Consortium (2011) https://portal.ogc.org/files/?artifact_id=43732

Panagiotis (Peter) A. Vretanos: *OGC Common Query Language (CQL2) (Draft)*. OGC 21-065, Open Geospatial Consortium, <https://docs.ogc.org/DRAFTS/21-065.html>

H. Butler, M. Daly, A. Doyle, S. Gillies, S. Hagen, T. Schaub: IETF RFC 7946, *The GeoJSON Format*. RFC Publisher (2016). <https://www.rfc-editor.org/info/rfc7946>.

OGC Features & Geometry JSON (Draft), <https://docs.ogc.org/DRAFTS/21-045r1.html>

Adobe Developers Association: *TIFF Specification Revision 6.0*. (1992) <https://www.itu.int/tudoc/itu-t/com16/tiff-fx/docs/tiff6.pdf>

Emmanuel Devys, Ted Habermann, Chuck Heazel, Roger Lott, Even Rouault: OGC 19-008r4, *OGC GeoTIFF Standard*. Open Geospatial Consortium (2019). <http://www.opengis.net/doc/IS/GeoTIFF/1.1.0>.

ISO/IEC: ISO/IEC 18181-1, ISO, IEC

ISO/IEC: ISO/IEC 18181-2, ISO, IEC

ISO/IEC: ISO/IEC 15948, *ISO/IEC 15948:2004, Information technology – Computer graphics and image processing – Portable Network Graphics (PNG): Functional specification*. ISO, IEC

Chris Little, Jon Blower, Maik Riechert: OGC 21-069r2, OGC CoverageJSON Community Standard. Open Geospatial Consortium (2023). <http://www.opengis.net/doc/CS/covjson/1.0>.

ISO/IEC: ISO/IEC 21320-1, ISO/IEC 21320-1:2015. ISO, IEC

4

TERMS AND DEFINITIONS

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.

4.1. child zone

immediate descendant of a parent zone

Note 1 to entry: adapted from *child cell* in OGC Abstract Specification Topic 21

[**SOURCE:** OGC 20-040r3]

4.2. discrete global grid (DGG)

set of zones at the same refinement level, that uniquely and completely cover a globe

Note 1 to entry: Adapted to use zones rather than cells to avoid potential confusion with the similar concept of grid cells in a *coverage*. As noted in OGC Abstract Specification Topic 21, "While the terms *cell* and *zone* are often used interchangeably, strictly *zone* is the preferred term.".

[**SOURCE:** OGC 20-040r3]

4.3. discrete global grid reference system (DGGRS)

integrated system comprised of a specific discrete global grid system (a parameterized hierarchy of discrete global grids fully establishing zone geometry), spatio-temporal referencing by zonal identifiers and deterministic sub-zone ordering

Note 1 to entry: partially adapted from *discrete global grid system* in OGC Abstract Specification Topic 21

Note 2 to entry: Annex B provides an informative schema for defining a DGGRS with accompanying examples

[SOURCE: OGC 20-040r3]

4.4. discrete global grid system (DGGS)

system comprised of a hierarchy of discrete global grids organized in successive levels of zone refinement, using a specific set of parameters such as its underlying Earth model spheroid and orientation of a base polyhedron, fully establishing the geometry of all zones.

Note 1 to entry: This definition differs from the OGC Abstract Specification Topic 21 definition, following earlier DGGS literature where the term DGGS often refers strictly to a “system of discrete global grids” in the sense of a “set” of grids (e.g., “A Discrete Global Grid System (DGGS) is a series of discrete global grids” [Sahr2003]), excluding the concepts of zonal identifiers (which is re-introduced as part of a discrete global grid reference system) and excluding notions of software functionality or interoperability so as to remain purely a data structure. This definition more closely corresponds to the term DGGS “hierarchy” (4.26) in OGC Abstract Specification Topic 21.

[SOURCE: OGC 20-040r3]

4.5. globe

region of space-time enclosing a celestial body

[SOURCE: OGC 20-040r3]

4.6. parent zone

zone in a lower refinement level of a discrete global grid with immediate descendants

Note 1 to entry: adapted from *parent cell* in OGC Abstract Specification Topic 21

Note 2 to entry: Adapted to say *lower* rather than *higher*, where *lower* refers to a zone covering a larger area of space-time which would normally correspond to lower resolution data.

[SOURCE: OGC 20-040r3]

4.7. refinement level

numerical order of a discrete global grid in the tessellation sequence

Note 1 to entry: The discrete global grid with the least number of zones has a refinement level of 0.

[SOURCE: OGC 20-040r3]

4.8. relative depth

the number of refinement levels separating a finer discrete global grid from a coarser discrete global grid in a discrete global grid system

4.9. sub-zone

zone at a greater refinement level than a parent zone whose geometry is at least partially contained within the geometry of the parent zone

Note 1 to entry: At a relative depth of 1, the sub-zones of the parent zone are its immediate children.

Note 2 to entry: In some DGGSs, the children of the children of a parent zone (relative depth of 2) are not all sub-zones of that parent zone.

Note 3 to entry: A discrete global grid reference system defines a deterministic order for sub-zones

4.10. Web API

An Application Programming Interface (API) using an architectural style that is founded on the technologies of the Web

Note 1 to entry: See Best Practice 24: Use Web Standards as the foundation of APIs (W3C Data on the Web Best Practices) for more detail.

[SOURCE: OGC 19-072]

4.11. zonal identifier

spatio-temporal reference in the form of a label or code that uniquely identifies a zone

Note 1 to entry: synonym of “zone identifier” and “zone ID”.

Note 2 to entry: This standard and the DGGRS definitions described in Annex B require textual identifiers (which may or may not be comprised of only digit characters), while optionally supporting 64-bit integer identifiers for the purpose of compact transmission and internal representations.

[SOURCE: OGC 20-040r3]

4.12. zonal identifier reference system (ZIRS)

reference system establishing a specific association of zone identifiers to zones for one or more discrete global grid systems

Note 1 to entry: synonym of “zone identifier reference system” and “zone indexing scheme”

4.13. zone

particular region of space-time

[SOURCE: OGC 20-040r3]

5

CONVENTIONS

CONVENTIONS

This section provides details of conventions used in this document.

5.1. Identifiers

The normative provisions in this standard are denoted by the URI <https://www.opengis.net/spec/ogcapi-dggs-1/1.0>.

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

5.2. Link relations

To express relationships between resources, [RFC 8288 \(Web Linking\)](#) is used.

Note that the CURIE based on the pattern [ogc-rel:<relation>] is also considered equivalent. For example, [ogc-rel:dggrs-definition] corresponds to <https://www.opengis.net/def/rel/ogc/1.0/dggrs-definition>.

The following [IANA link relation types](#) are used in this document:

- **alternate**: Refers to a substitute for this context.
- **self**: Conveys an identifier for the link's context.
- **service-desc**: Identifies service description for the context that is primarily intended for consumption by machines (Web API definitions are considered service descriptions).
- **service-doc**: Identifies service documentation for the context that is primarily intended for human consumption.

The following link relation types which were previously defined for other OGC API standards are applicable:

- <https://www.opengis.net/def/rel/ogc/1.0/dataset>: The target IRI points to a resource representing the dataset (e.g., the root of an OGC Web API).
- <https://www.opengis.net/def/rel/ogc/1.0/geodata>: The target IRI points to a resource representing a collection of geospatial data.

In addition, the following link relation types are used for which no applicable registered link relation type could be identified:

- <https://www.opengis.net/def/rel/ogc/1.0/dggrs>: The target IRI points to the description, which includes a link to the definition as well as deployment-specific elements, of the Discrete Global Grid Reference System related to the link's context.
- <https://www.opengis.net/def/rel/ogc/1.0/dggrs-list>: The target IRI points to the descriptive list of the Discrete Global Grid Reference Systems available for the resource of the link's context.
- <https://www.opengis.net/def/rel/ogc/1.0/dggrs-definition>: The target IRI points to the definition of the Discrete Global Grid Reference System using a well-defined schema (see Annex B), for example as available from an authoritative DGGRS registry.
- <https://www.opengis.net/def/rel/ogc/1.0/dggrs-zone-info>: The target IRI points to information about a particular zone of a Discrete Global Grid Reference System, such as its geometry, area/volume, and any relevant links (e.g., to retrieve data from the zone).
- <https://www.opengis.net/def/rel/ogc/1.0/dggrs-zone-data>: The target IRI points to data pertaining to a particular zone of a Discrete Global Grid Reference System. This link relation is typically used with a link template where a zone identifier can be substituted.
- <https://www.opengis.net/def/rel/ogc/1.0/dggrs-zone-parent>: The target IRI points to the information resource of a zone which is a parent of the link's context zone.
- <https://www.opengis.net/def/rel/ogc/1.0/dggrs-zone-child>: The target IRI points to the information resource of a zone which is an immediate child of the link's context zone.
- <https://www.opengis.net/def/rel/ogc/1.0/dggrs-zone-neighbor>: The target IRI points to the information resource of a zone which is a neighbor of the link's context zone.

Each resource representation includes an array of links. Implementations are free to add additional links for all resources provided by the Web API implementing OGC API – DGGS requirement(s).

Used in combination OGC API – Common – Part 1: Core, other link relation types will be used, including:

- <https://www.opengis.net/def/rel/ogc/1.0/conformance>: Refers to a resource that identifies the specifications that the link's context conforms to.

Used in combination with OGC API – Common – Part 2: Geospatial Data, other link relation types will be used, including:

- <https://www.opengis.net/def/rel/ogc/1.0/data>: Refers to the list of collections available for a dataset.

5.3. Use of HTTPS

For simplicity, OGC API – DGGS only refers to the HTTP protocol. This is not meant to exclude the use of HTTPS and simply is a shorthand notation for “HTTP or HTTPS.” Following the recent push towards enhancing web security, public facing web servers are expected to use [HTTPS](#) rather than [HTTP](#). In the context of the OGC API – DGGS Standard, use of HTTPS provides a layer of security to prevent leaking of information related to users requesting dggs data or zones list for a particular geographical area, which may be private or sensitive in nature (for example, their own location, or the location of endangered species).

6

REQUIREMENT CLASS “CORE”

REQUIREMENT CLASS “CORE”

6.1. Overview

The “Core” Requirement Class allows a client to list available DGGRSs for a given resource, retrieve additional information about a particular DGGRS, and retrieve information about a particular DGGRS 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, so that a client can ask a question of the type “**What is here?**” and/or the “Zone Query” Requirement Class, so that a client can ask a question of the type “**Where is it?**”

An [example OpenAPI 3.0 definition for OGC API – DGGS](#) is provided, defined as individual building blocks which can be assembled with the [swagger-cli tool](#) after customizing the API definition for a particular implementation or deployment. The example API definition can be visualized with [SwaggerUI](#).

REQUIREMENTS CLASS 1

TARGET TYPE Web API

LABEL <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/core>

6.2. Listing available DGGRS (.../dggs)

REQUIREMENT 1

LABEL /req/core/dggrs-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.

REQUIREMENT 1

- C The .../dggs resource SHALL include a dggrs array property listing all discrete global grid reference systems supported by the Implementation.
- D Each element of the dggrs array SHALL include a subset of the information available in the individual DGGRS 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:dggrs-definition] relation type.
- F The link relation type to use for linking from the corresponding resource to the list of available DGGRSs for that resource SHALL be [ogc-rel:dggrs-list].

6.3. Discrete global grid reference system information (.../dggs/{dggrsId})

REQUIREMENT 2

LABEL /req/core/dggrs-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/{dggrsId}.
- B The Implementation SHALL support a JSON representation of this .../dggs/{dggrsId} resource.
- C The .../dggs/{dggrsId} resource SHALL include an id property consistent with the {dggrsId} resource path parameter.
- D The .../dggs/{dggrsId} resource SHALL include a link to the resource itself using the self link relation type.
- E The .../dggs/{dggrsId} 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:dggrs-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 in Annex B – Discrete Global Grid Reference System Definitions.
- F The .../dggs/{dggrsId} resource SHALL include a templated link in the linkTemplates array to request information for a particular zone using the link relation type [ogc-rel:dggrs-zone-info] and the template variable {zoneId}.

REQUIREMENT 2

- 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.

RECOMMENDATION 1

LABEL /rec/core/max-refinement

STATEMENT For specifying a maximum refinement level for a discrete global grid reference system associated with a particular geospatial data resource:

- A For DGGS resources associated with a data source, the Implementation SHOULD include a `maxRefinementLevel` integer property in the `.../dggs/{dggrsId}` resource specifying the maximum refinement level at which the full resolution of the data can be retrieved (using a zone-depth relative depth of 0) and/or used for performing the most accurate zone queries (using that value for zone-level).

PERMISSION 1

LABEL /per/core/beyond-max-refinement

STATEMENT For handling requests beyond `maxRefinementLevel`:

- A The Implementation MAY return a 4xx error for data retrieval and/or zone query requests beyond a `maxRefinementLevel` specified in the DGGRS description associated with the request.
- B Alternatively, the Implementation MAY process the request by properly oversampling the data values for handling refinement levels beyond the `maxRefinementLevel`.

6.4. Retrieving zone information (`.../dggs/{dggrsId}/zones/{zonId}`)

REQUIREMENT 3

LABEL /req/core/zone-info

STATEMENT For retrieving information for a particular DGGRS zone:

- A The Implementation SHALL support an HTTP GET operation at a resource path ending with .../dggs/{dggrsId}/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 DGGRS resource (.../dggs/{dggrsId}) using the [ogc-rel:dggrs] link relation type.

RECOMMENDATION 2

LABEL /rec/core/zone-info

STATEMENT For recommending additional things that should be included in zone information resources:

- A The zone information resource SHOULD include a shapeType property indicating the shape type of the zone's geometry (e.g., hexagon or pentagon).
- B The zone information resource SHOULD include a level property indicating the refinement level of the zone.
- C The zone information resource SHOULD include a crs property indicating the Coordinate Reference System (CRS) in which the geometry, centroid and bbox property are specified.
- D The zone information resource SHOULD include a centroid property indicating the centroid of the zone. In the JSON representation, this should be an array of two numbers in the CRS specified in crs.
- E The zone information resource SHOULD include a bbox property indicating the extent (envelope / bounding box) of the zone. In the JSON representation, this should be an array of 4 coordinates for a DGGRS with two spatial dimensions or 6 coordinates for a DGGRS with three spatial dimensions, in the CRS specified in crs.
- F The zone information resource SHOULD include links to its parent(s) zone(s) using link relation [ogc-rel:dggrs-zone-parent].
- G The zone information resource SHOULD include links to its immediate children zone using link relation [ogc-rel:dggrs-zone-child].
- H The zone information resource SHOULD include links to its neighboring zones using link relation [ogc-rel:dggrs-zone-neighbor].
- I The zone information resource SHOULD include an areaMetersSquare property indicating the surface area of the zone in square meters.

RECOMMENDATION 2

- J 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.
- K For a temporal DGGS, the zone information resource SHOULD include a `temporalDurationSeconds` property indicating the amount of time covered by the zone in seconds.
- L 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, and including intermediate points between the vertices of the geometry so as to accurately represent the shape of the zones for DGGRSs not in CRS84. The coordinates should be in the CRS specified in the `crs` property.
- M For a temporal DGGS, the zone information resource SHOULD include a `temporalInterval` property indicating the start and end time of the zone.
- N 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, a `zoneID` property of that feature corresponds to the textual identifier (the `{zoneId}`), the `id` of the feature corresponds to either the textual identifier or the 64-bit unsigned integer identifier of the zone, and the other properties described in this recommendation are properties of that feature.
- O For a zone associated with a particular collection, the implementation SHOULD provide summary statistics in a `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 encoding, this SHOULD be implemented as a `statistics` JSON dictionary property mapping field names to an object containing with each statistic.
- P 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).

RECOMMENDATION 3

LABEL /rec/core/robots-txt

STATEMENT For discouraging automated crawling of zone information resources:

- A Implementations SHOULD include a `robots.txt` file at the root of their Web API discouraging robots from crawling the DGGS zone resources.
- B The content of that Robots.txt file SHOULD include `Disallow: */dggs/*/zones/*` to prevent crawling all DGGS resources under the `/zones/` resource path.

NOTE: The presence of a `robots.txt` file is not a security measure and relies on the voluntary compliance of well-intended crawlers to minimize unnecessary requests. This measure does not

prevent malicious clients from overwhelming the server with numerous requests which may result in Denial of Service attacks.

7

REQUIREMENT CLASS “DATA RETRIEVAL”

REQUIREMENT CLASS “DATA RETRIEVAL”

7.1. Overview

The Data Retrieval requirement class allows to retrieve data from a specific Discrete Global Grid Reference System (DGGRS), that is a specific hierarchy of Discrete Global Grids combined with a particular zone indexing reference system (ZIRS), for an individual zone. It describes an HTTP GET operation, as well as its response. The selected DGGRS is listed as available and described in the Core requirement 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.

This requirement class enables a client to ask a question of the type “**What is here?**”, where the *what* refers to data values associated with a particular dataset, collection, or processing workflow, and the *here* refers to a particular spatio-temporal region identified by the Zone ID.

REQUIREMENTS CLASS 2

TARGET TYPE Web API

PREREQUISITE <https://www.opengis.net/spec/ogcapi-common-1/1.0/req/core>

LABEL <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval>

7.2. Retrieving data from a zone (../dggs/{dggrsId}/zones/{zoneId}/data)

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

REQUIREMENT 4

LABEL /req/data-retrieval/zone-data

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/{dggrsId}/zones/{zoneId}/data.
- B The operation SHALL include a templated link to this resource path in the “Core” .../dggs/{dggrsId} resource link templates, and regular link in the .../dggs/{dggrsId}/zones/{zoneId} resource links using the link relation type [ogc-rel:dggrs-zone-data] for all zones for which data is available.
- 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 .../dggs/{dggrsId} resource SHALL include a defaultDepth property indicating the implementation’s default depth for when the zone-depth parameter is omitted. This default value could be any valid value and/or form as defined in the /req/data-custom-depths/zone-depth-parameter requirement (single depth, range of depths, or list of depths, relative to the {zoneId} hierarchy level).
- G Unless a zone-depth parameter is specified, the response SHALL return a data packet consistent with this defaultDepth property, in accordance with the capabilities of the negotiated data packet encoding.

8

REQUIREMENT CLASS “DATA SUBSETTING”

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). Additionally, this requirement class defines the `properties` parameter to select specific fields to include in the response instead of all fields which are included by defaults.

REQUIREMENTS CLASS 3

TARGET TYPE Web API

PREREQUISITE <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval>

LABEL <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-subsetting>

8.2. 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):

The Implementation SHALL support a subset query parameter for the zone data retrieval operation (resource path ending with .../dggs/{dggrsId}/zones/{zoneId}/data) conforming to the following Augmented Backus Naur Form (ABNF) fragment:

A SubsetSpec: "subset"=axisName(intervalOrPoint)
 axisName: {text}
 intervalOrPoint: interval \| point
 interval: low : high
 low: point \| *
 high: point \| *

REQUIREMENT 5

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.

8.3. 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):

REQUIREMENT 6

```
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 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

8.3.1. Parameter properties

Table 2

Requirement 1	/req/data-subsetting/properties
A	The zone data retrieval operation SHALL support a parameter properties where the value is a comma-separated list of fields to be returned.

B	The implementation SHALL support selecting a field using the identifier corresponding to the top-level property keys of the logical schema of the resource associated with the DGGRS zone data request, and return 400 status code for an unrecognized selected field.
C	Only the selected fields SHALL be returned from the zone data request.
D	If the zone data encodings response can self-describe its list of fields (as with the shema property of the DGGS-JSON encoding), the field description SHALL correspond to the requested list of fields.
E	If the negotiated format of the response has a concept of field order, then the fields SHALL be in the same order as the requested list of selected fields.

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 <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval>

LABEL <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-custom-depths>

9.2. 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 .../dggs/{dggrsId}/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);

REQUIREMENT 7

- 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).

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:

C

- 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,
- ..
- 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.

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”

REQUIREMENT CLASS “ZONE QUERY”

10.1. Overview

The Zone Query requirement class allows to request the list of DGGS zones from a specific Discrete Global Grid Reference System (DGGRS), that is a specific hierarchy of Discrete Global Grids combined with a particular zone indexing reference system (ZIRS), 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 DGGRS is listed as available and described in the Core requirement class, and conforms to [OGC Topic 21](#). The list of zones from a Web API using this requirement 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.

This requirement class enables a client to ask a question of the type “**Where is it?**”, where the *where* refers to the spatio-temporal response encoded as a list of Zone IDs, and the *it* refers to either the availability of data associated with a particular collection, dataset, or processing workflow (where the result itself could simply be this spatio-temporal region), as well as the evaluation of any additional filtering query expressed as per the CQL2 filtering requirement class.

REQUIREMENTS CLASS 5

TARGET TYPE Web API

PREREQUISITE <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/core>

LABEL <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query>

10.2. Listing zones (.../dggs/{dggrsId}/zones)

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

REQUIREMENT 8

LABEL /req/zone-query/zones-list

STATEMENT For retrieving a list of DGGS zones:

- A The Implementation SHALL support an HTTP GET operation at a resource path ending with .../dggs/{dggrsId}/zones.
- B The Implementation SHALL include a link to this resource path in the “Core” .../dggs/{dggrsId} resource links using the link relation type <https://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”, “1-E-15”, “1-F-14”, “1-F-15”, “1-F-16”] }).

The following recommendation describes how the response should include the total surface area of the zones returned from the query (or volume / hypervolume in the case of spatiotemporal and/or 3D DGGS), excluding overlapping regions when returning compact zones.

RECOMMENDATION 4

LABEL /rec/zone-query/zone-total-area

STATEMENT For returning the total area (or volumes) occupied by zones returned by the query

- A The Implementation SHOULD indicate the total surface area (in the case of a 2D DGGS) or total (hyper)volume occupied (in the case of a 3D or 4D DGGS) occupied by the zones being returned, adding only once any overlaps which may occur for compact zone responses where zones are non-congruent.

10.3. 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/{dggrsId}/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.

10.4. 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/{dggrsId}/zones), where a value of true corresponds to the default behavior when the parameter 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.

REQUIREMENT 10

- C When the compact-zones parameter is set 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.

10.5. Parameter parent-zone for hierarchical exploration

The following requirement describes 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 parent-zone zone identifier query parameter.
- B When specified, the response SHALL NOT contain zones which are not this parent zone itself or a sub-zone of that zone.

10.6. Parameter limit for paging

The following recommendation describes 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 5

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).

RECOMMENDATION 5

- 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.

10.7. 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:

The Implementation SHALL support a `bbox` query parameter for the zone query operation (resource path ending with .../dggs/{dggrsId}/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

A
 `* 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 <https://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`

REQUIREMENT 12

style: form
explode: false

- 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 (<https://www.opengis.net/def/crs/OGC/1.3/CRS84> axis order for a 2D bounding box, <https://www.opengis.net/def/crs/OGC/1.3/CRS84h> for a 3D bounding box). For example in <https://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.

10.8. 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 <https://www.opengis.net/def/crs/OGC/1.3/CRS84> as a value.
- C If the bbox-crs is not indicated <https://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.

10.9. 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:

The Implementation SHALL support a subset query parameter for the zone query operation (resource path ending with .../dggs/{dggrsId}/zones) conforming to the following Augmented Backus Naur Form (ABNF) fragment:

```
A      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 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.

G 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.

H The implementation SHALL interpret the coordinates as values for the named axis of the CRS specified in the subset-crs parameter value or in <https://www.opengis.net/def/crs/OGC/1.3/CRS84> (<https://www.opengis.net/def/crs/OGC/1.3/CRS84h> for vertical dimension) if the subset-crs parameter is missing.

I 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.

J 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 <https://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.

10.10. 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, <https://www.opengis.net/def/crs/OGC/1.3/CRS84> as a value SHALL be supported.
- C If the subset-crs is not indicated, <https://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.

10.11. 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:

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):

A 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 “FILTERING ZONE QUERIES WITH CQL2”

REQUIREMENT CLASS “FILTERING ZONE QUERIES WITH CQL2”

11.1. Overview

The Filtering Zone Queries with CQL2 conformance class defines a filter query parameter where an OGC Common Query Language (CQL2) expression can be used to filter the list of returned zones, for example based on the properties / fields of the data, or for performing spatial and/or temporal intersections.

REQUIREMENTS CLASS 6

TARGET TYPE	Web API
PREREQUISITES	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query https://www.opengis.net/spec/cql2/1.0/req/cql2-text
LABEL	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/cql2-query

11.2. Parameter filter

The following requirements describe how a client can specify a filtering CQL2 expression querying a list of zones.

REQUIREMENT 17

LABEL	/req/cql2-query/filter
-------	------------------------

STATEMENT For specifying a CQL2 expression filtering the zones returned from a zone query:

- A The Implementation SHALL support a filter query parameter specified using the CQL2-Text encoding of the OGC Common Query Language for the zone query operation (resource path ending with .../dggs/{dggrsId}/zones).
- B The list of returned zones SHALL only be those for which the CQL2 expression evaluates to true when considering the geometry and the data of the DGGS zones resource being queried.

REQUIREMENT 17

- C The CQL2 expression evaluator SHALL support the queryables declared in the JSON Schema resource linked to from the origin of the DGGRS resources using the [ogc-rel:queryables] link relation type.

NOTE: The queryables would typically include the fields of the data source, which may also be retrieved using the Data Retrieval requirement class.

12

REQUIREMENT CLASS “ROOT DGGS”

REQUIREMENT CLASS “ROOT DGGS”

12.1. 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 7

TARGET TYPE Web API

PREREQUISITE <https://www.opengis.net/spec/ogcapi-common-1/1.0/req/core>

LABEL <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/root-dggs>

12.2. Root DGGS (/dggs)

REQUIREMENT 18

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.

B The Implementation SHALL include a link to the list of available DGGRSs for the API/dataset at /dggs in the links of the landing page using the link relation type [ogc-rel:dggrs-list].

C If the DGGS API allows querying or retrieving data from a dataset, the /dggs and /dggs/{dggrsId} resources SHALL include a link to the landing page using the link relation type [ogc-rel:dataset].

13

REQUIREMENT CLASS “COLLECTION DGGS”

REQUIREMENT CLASS “COLLECTION DGGS”

13.1. 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 8

TARGET TYPE Web API

PREREQUISITE <https://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections>

LABEL <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/collection-dggs>

13.2. Collection DGGS (/collections/{collectionId}/dggs)

REQUIREMENT 19

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 Implementation SHALL include a link to the list of available DGGRSs for the collection at /collections/{collectionId}/dggs in the links of the collection using the link relation type [ogc-rel:dggrs-list].
- C The /collections/{collectionId}/dggs and /collections/{collectionId}/dggs/{dggrsId} resources SHALL include a link to the collection using the link relation type [ogc-rel:geodata].

14

REQUIREMENTS CLASSES FOR ENCODINGS OF ZONE DATA

REQUIREMENTS CLASSES FOR ENCODINGS OF ZONE DATA

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

- DGGS-JSON Data
- DGGS-UB-JSON Data
- GeoJSON Data
- FG-JSON Data
- GeoTIFF Data
- netCDF Data
- Zarr Data
- CoverageJSON Data
- JPEG XL Data
- PNG Data

14.1. Media Types (for zone data)

A table of the media types used in the encoding of the Zone Data requirements class defined in this standard follows.

Table 3 – Media Types used for zone data encoding requirements classes

Encoding	media type
DGGS-JSON zone data	application/json
DGGS-UB-JSON zone data	application/ubjson
GeoJSON data	application/geo+json

JSON-FG data	application/fg+json
GeoTIFF data	image/tiff; application=geotiff
netCDF zone data	application/x-netcdf
Zarr zone data	application/zarr+zip
CoverageJSON zone data	application/prs.coverage+json
PNG zone data	image/png
JPEG XL zone data	image/jxl

14.2. Requirements Class “DGGS-JSON Zone data encoding”

14.2.1. Overview

The JSON Zone Data encoding requirements class defines support for encoding a DGGS Zone data response in JSON, using a schema defined below named “DGGS-JSON”.

This encoding is intended to support one-to-one mapping of sub-zones values, for one or more zone depths, regardless of zone geometry type, in a human-readable format.

Values are encoded as a one-dimensional array following a sub-zone order defined by the DGGRS.

Example encodings follow:

```
{
  "$schema": "https://schemas.opengis.net/ogcapi/dggs/part1/1.0/openapi/schemas/dggs-json/schema",
  "dggrs": "https://www.opengis.net/def/dggrs/OGC/1.0/ISEA3H",
  "zoneId": "C0-2B-A",
  "depths": [ 0, 1 ],
  "representedValue": "area",
  "schema":
  {
    "$schema": "https://json-schema.org/draft/2020-12/schema",
    "$id": "https://example.com/ogcapi/collections/climate/schema",
    "title": "Temperature",
    "type": "object",
    "properties":
```

```

{
  "t": {
    "type": "number",
    "title": "air temperature in celsius",
    "x-ogc-propertySeq": 1,
    "x-ogc-definition": "https://qudt.org/vocab/quantitykind/Temperatur
e",
    "x-ogc-unit": "C"
  }
},
"values":
{
  "t": [
    {
      "depth": 0,
      "shape": { "count": 1, "subZones": 1 },
      "data": [ 22.8 ]
    },
    {
      "depth": 1,
      "shape": { "count": 7, "subZones": 7 },
      "data": [ 25.1, 25.8, 22.0, 23.1, 22.8, 20.0, 17.1 ]
    }
  ]
}
}

```

Figure 6 – Example encoding for DGGS-JSON

```

{
  "$schema": "https://schemas.opengis.net/ogcapi/dggs/part1/1.0/openapi/
schemas/dggs-json/schema",
  "dggrs": "https://www.opengis.net/def/dggrs/OGC/1.0/ISEA3H",
  "zoneId": "C0-2B-A",
  "depths": [ 0, 1 ],
  "representedValue": "area",
  "schema":
  {
    "$schema": "https://json-schema.org/draft/2020-12/schema",
    "$id": "https://example.com/ogcapi/collections/climate/schema",
    "title": "Climate Variables",
    "type": "object",
    "properties":
    {
      "rh": {
        "type": "number",
        "title": "relative humidity",
        "x-ogc-propertySeq": 1,
        "x-ogc-definition": "https://qudt.org/vocab/quantitykind/RelativeHu
midity"
      },
      "t": {
        "type": "number",
        "title": "air temperature in celsius",
        "x-ogc-propertySeq": 2,
        "x-ogc-definition": "https://qudt.org/vocab/quantitykind/Temperatur
e",
        "x-ogc-unit": "C"
      },
      "ua": {
        "type": "number"
      }
    }
  }
}

```

```

        "type": "number",
        "title": "Eastward wind",
        "x-ogc-propertySeq": 3,
        "x-ogc-definition": "https://qudt.org/vocab/quantitykind/Speed",
        "x-ogc-unit": "m/s"
    },
    "va": {
        "type": "number",
        "title": "Northward wind",
        "x-ogc-propertySeq": 4,
        "x-ogc-definition": "https://qudt.org/vocab/quantitykind/Speed",
        "x-ogc-unit": "m/s"
    }
}
},
"dimensions": [
{
    "name": "time",
    "interval": [ "2020-01-01", "2020-03-31" ],
    "grid": { "cellsCount": 3, "resolution": "P1M" }
},
{
    "name": "pressure",
    "definition": "https://qudt.org/vocab/quantitykind/AtmosphericPressure",
    "unit": "hPa",
    "interval": [ 850.0, 50.0 ],
    "grid": { "cellsCount": 3, "coordinates": [ 850.0, 250.0, 50.0 ] }
},
],
"values": {
    "rh": [
        {
            "depth": 0,
            "shape": { "count": 9, "subZones": 1, "dimensions": { "pressure": 3, "time": 3 } },
            "data": [
                25.1, 25.8, 22.0,
                23.1, 22.8, 20.0,
                17.1, 12.8, 14.0,
            ]
        },
        {
            "depth": 1,
            "shape": { "count": 63, "subZones": 7, "dimensions": { "pressure": 3, "time": 3 } },
            "data": [
                25.1, 25.8, 22.0, 23.1, 22.8, 20.0, 17.1, 12.8, 14.0,
                25.1, 25.8, 22.0, 23.1, 22.8, 20.0, 17.1, 12.8, 14.0,
                25.1, 25.8, 22.0, 23.1, 22.8, 20.0, 17.1, 12.8, 14.0,
                25.1, 25.8, 22.0, 23.1, 22.8, 20.0, 17.1, 12.8, 14.0,
                25.1, 25.8, 22.0, 23.1, 22.8, 20.0, 17.1, 12.8, 14.0,
                25.1, 25.8, 22.0, 23.1, 22.8, 20.0, 17.1, 12.8, 14.0,
                25.1, 25.8, 22.0, 23.1, 22.8, 20.0, 17.1, 12.8, 14.0,
                25.1, 25.8, 22.0, 23.1, 22.8, 20.0, 17.1, 12.8, 14.0
            ]
        }
    ],
    "t": [ ... ],
    "ua": [ ... ],
    "va": [ ... ]
}

```

}

Figure 7 – A more complex example encoding for DGGS-JSON with more variables and additional dimensions beyond those of the DGGRS

REQUIREMENTS CLASS 9: REQUIREMENTS CLASS JSON ZONE DATA

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-json>

TARGET TYPE Web API

CONFORMANCE CLASS Conformance class A.9: <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-json>

PREREQUISITES JSON
<https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval>

REQUIREMENT 20

IDENTIFI /req/data-json/content

- A** Every 200 response of the server for zone data with the media type application/json SHALL be a JSON document representing the data values for all selected fields of each included sub-zone.
- B** The schema for the JSON document SHALL follow the JSON Schema for DGGS-JSON described below.
- C** Every zone depth requested using the zone-depth parameter SHALL be included in the response.
- D** At every depth, each individual value SHALL correspond exactly to the data sampled representative of that sub-zone for either the overall area of the sub-zone, the sub-zone centroid or the primary vertex of the sub-zone.
- E** The list of data values SHALL follow the default zone order as specified by the Discrete Global Grid Reference System (for example based on a scanline or space-filling curved defined therein) for which the request is made.
- F** For the purpose of identifying the primary vertex, assuming the following approach results in uniquely mapping a single vertex for every zone of each discrete global grid, for the first sub-zone, the primary vertex SHALL correspond to the vertex furthest away from the next sub-zone in the DGGRS sub-zone order which is at different coordinates for both spatial axes. For example, for square zones using a quad-tree refinement with a top to bottom row by row sub-zone order starting with the top-left zone (whether jumping back to the start of the next row, or alternating between left to right and right to left), the primary vertex will be the top-left corner.
- G** For a DGGRS where this primary vertex identification approach does not result in uniquely mapping a single vertex for every zone of a DGG, the primary vertex SHALL be identified as specified in the DGGRS definition.
- H** Null values SHALL use the null JSON value.

```

{
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$id" : "https://schemas.opengis.net/ogcapi/dggs/part1/1.0/openapi/schemas/
dggs-json/schema",
  "type": "object",
  "properties":
  {
    "dggrs" : { "type": "string", "format": "uri" },
    "zoneId" : { "type": "string" },
    "depths" : { "type": "array", "items": { "type": "integer", "minimum": 0
} },
    "representedValue" : { "type": "string", "enum": [ "area", "centroid",
"vertex" ] },
    "schema" : { "$ref": "https://json-schema.org/draft/2020-12/schema" },
    "dimensions" :
    {
      "type": "array",
      "items":
      {
        "type": "object",
        "properties":
        {
          "name": { "type": "string" },
          "definition": { "type": "string", "format": "uri" },
          "unit": { "type": "string" },
          "unitLang": { "type": "string", "format": "uri" },
          "grid":
          {
            "type": "object",
            "properties":
            {
              "cellsCount" : { "type": "integer" },
              "resolution" : { },
              "coordinates": { "type": "array" }
            }
          },
          "interval": { "type": "array" }
        },
        "required": [ "name", "interval" ]
      }
    },
    "values" :
    {
      "additionalProperties":
      {
        "type": "array",
        "items":
        {
          "type": "object",
          "properties":
          {
            "depth": { "type": "integer" },
            "shape":
            {
              "type": "object",
              "properties":
              {
                "count": { "type": "integer" },
                "subZones": { "type": "integer" },
                "dimensions":
                {
                  "type": "array"
                }
              }
            }
          }
        }
      }
    }
  }
}

```

```

        "type": "object",
        "additionalProperties": { "type": "integer" }
    }
},
"required": [ "count", "subZones" ]
},
"data":
{
    "type": "array",
    "item": "number"
}
}
}
},
"required": [ "dggrs", "zoneId", "depths", "values", "representedValue" ]
}

```

Figure 8 – JSON Schema for DGGS-JSON

14.3. Requirements Class “DGGS-UB-JSON Zone data encoding”

14.3.1. Overview

The UB-JSON Zone Data encoding requirements class defines support for encoding a DGGS Zone data response using the same JSON structure as the above JSON data requirement class, but using the binary UB-JSON encoding.

REQUIREMENTS CLASS 10: REQUIREMENTS CLASS UB-JSON

IDENTIFIER	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-ubjson
------------	---

TARGET TYPE	Web API
-------------	---------

CONFORMANCE CLASS	Conformance class A.10: https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-ubjson
-------------------	---

PREREQUISITES	https://ubjson.org/ https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval
---------------	--

REQUIREMENT 21

IDENTIFI	/req/data-ubjson/content
----------	--------------------------

REQUIREMENT 21

- A Every 200 response of the server for zone data with the media type application/ubjson SHALL be a Universal Binary JSON document representing the data values for all selected fields for a single zone.
- B The schema for the UB-JSON document SHALL follow the same JSON Schema as for DGGS-JSON described above.
- C Every zone depth requested using the zone-depth parameter SHALL be included in the response.
- D At every depth, each individual value SHALL correspond exactly to the data sampled representative of that zone geometry.
- E The list of data values SHALL follow the default zone order as specified by the Discrete Global Grid Reference System (for example based on a scanline or space-filling curved defined therein) for which the request is made.
- F Null values SHALL use the null JSON value.

14.4. Requirements Class “GeoTIFF Zone data encoding”

14.4.1. Overview

The GeoTIFF Zone Data encoding 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.

REQUIREMENTS CLASS 11: REQUIREMENTS CLASS GEOTIFF DATA

IDENTIFIER	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-geotiff
TARGET TYPE	Web API
PREREQUISITES	TIFF V6.0 GeoTIFF https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval

REQUIREMENT 22

IDENTIFIER /req/data-geotiff/content

- A Every 200 response of the server for zone data with the media type `image/tiff` SHALL be a TIFF image representing the data values for all selected fields for a single zone.
- B If the TIFF encoding incorporates a GeoTIFF georeference, this information SHALL be consistent with the DGGRS Zone ID.
- C If the zone geometry is not rectangular, the closest bounding rectangle SHALL be used for referencing the image.
- D A distinct value for each sub-zone implied from the requested zone-depth SHALL correspond to at least one distinct cell value in the response.
- E For implementations supporting Data Custom Depths, each depth of the requested zone depth pyramid SHALL be a separate image (overview) in the response.

RECOMMENDATION 6

IDENTIFIER /rec/data-geotiff/null-values

- A Null values SHOULD be used for cells lying outside the zone geometry.

RECOMMENDATION 7

IDENTIFIER /rec/data-geotiff/crs

- A The CRS of the response SHOULD be consistent with either the DGGRS or the underlying geographic CRS (e.g., CRS84).

14.5. Requirements Class “GeoJSON Zone data encoding”

14.5.1. Overview

The GeoJSON Zone Data encoding 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.

REQUIREMENTS CLASS 12: REQUIREMENTS CLASS GEOJSON

IDENTIFIER	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-geojson
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.12: https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-geojson
PREREQUISITES	GeoJSON https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval

REQUIREMENT 23

IDENTIFI /req/data-geojson/content

- A Every 200 response of the server for zone data with the media type application/geo+json SHALL be a GeoJSON document representing the features, including their geometry and associated properties, for a single zone.
- B Unless otherwise specified by a prior arrangement (for example, a crs output parameter), the coordinate reference system SHALL be CRS84(h) in longitude and latitude (and optional height above the WGS84 ellipsoid).
- C Features whose geometry lie wholly outside of the zone geometry SHALL not be included in the response.

RECOMMENDATION 8

IDENTIFI /rec/data-geojson/clipping

- A For features partially inside of the zone geometry, the geometry included in the response SHOULD be clipped to the boundaries of the zone geometry.

RECOMMENDATION 9

IDENTIFI /rec/data-geojson/generalization

- A The geometry of the features in the response SHOULD be generalized to the scale associated with the hierarchy level implied from the zone-depth relative to the requested zone.

RECOMMENDATION 10

IDENTIFIER /rec/data-geojson/omission

- A For datasets whose features have either a size (such as areal or linear feature) or other properties making some features irrelevant at lower hierarchy levels, such irrelevant features SHOULD be omitted from the response based on the hierarchy level implied from the zone-depth relative to the requested zone.

14.6. Requirements Class “FG-JSON Zone data encoding”

14.6.1. Overview

The FG-JSON Zone Data encoding requirements class defines support for encoding a DGGS Zone data response according to the [FG JSON candidate Standard](#).

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

REQUIREMENTS CLASS 13: REQUIREMENTS CLASS FG-JSON DATA

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-fgjson>

TARGET TYPE Web API

CONFORMANCE CLASS Conformance class A.13: <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-fgjson>

PREREQUISITES OGC 21-045r1
<https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval>

REQUIREMENT 24

IDENTIFIER /req/data-fgjson/content

- A Every 200 response of the server for zone data with the media type application/fg+json SHALL be a [Feature & Geometries JSON](#) document representing the features, including their geometry and associated properties, for a single zone.

REQUIREMENT 24

- B Features whose geometry lie wholly outside of the zone geometry SHALL not be included in the response.

RECOMMENDATION 11

IDENTIFIE /rec/data-fgjson/clipping

- A For features partially inside of the zone geometry, the geometry included in the response SHOULD be clipped to the boundaries of the zone geometry.

RECOMMENDATION 12

IDENTIFIE /rec/data-fgjson/generalization

- A The geometry of the features in the response SHOULD be generalized to the scale associated with the hierarchy level implied from the zone-depth relative to the requested zone.

RECOMMENDATION 13

IDENTIFIE /rec/data-fgjson/omission

- A For datasets whose features have either a size (such as areal or linear feature) or other properties making some features irrelevant at lower hierarchy levels, such irrelevant features SHOULD be omitted from the response based on the hierarchy level implied from the zone-depth relative to the requested zone.

RECOMMENDATION 14

IDENTIFIE /rec/data-fgjson/crs

- A The CRS of the response SHOULD be consistent with either the DGGRS or the underlying geographic CRS (e.g., CRS84).

14.7. Requirements Class “netCDF zone data encoding”

14.7.1. Overview

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

REQUIREMENTS CLASS 14: REQUIREMENTS CLASS NETCDF

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-netcdf>

TARGET TYPE Web API

CONFORMANCE CLASS Conformance class A.14: <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-netcdf>

PREREQUISITES OGC Network Common Data Form (NetCDF) Core Encoding Standard version 1.0
<https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval>

REQUIREMENT 25

IDENTIFIER /req/data-netcdf/content

- A** Every 200 response of the server for zone data with the media type application/netcdf SHALL be a netCDF file representing the data values for all selected fields for a single zone.
- B** If the netCDF encoding incorporates georeferencing information, this information SHALL be consistent with the DGGRS Zone ID.
- C** If the zone geometry is not rectilinear, the closest bounding rectangle (or volume) SHALL be used for referencing the data.
- D** A distinct value for each sub-zone implied from the requested zone-depth SHALL correspond to at least one distinct cell value in the response.

RECOMMENDATION 15

IDENTIFIER /rec/data-netcdf/null-values

- A** Null values SHOULD be used for cells lying outside the zone geometry.

RECOMMENDATION 16

IDENTIFIE /rec/data-netcdf/crs

- A The CRS of the response SHOULD be consistent with either the DGGRS or the underlying geographic CRS (e.g., CRS84).

14.8. Requirements Class “Zarr zone data encoding”

14.8.1. Overview

The Zarr Zone Data encoding requirements class defines support for encoding a DGGS Zone data response according to the [Zarr Storage Specification Version 2](#), packed into a Zip archive.

REQUIREMENTS CLASS 15: REQUIREMENTS CLASS ZARR

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-zarr>

TARGET TYPE Web API

CONFORMANCE CLASS Conformance class A.15: <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-zarr>

PREREQUISITES Zarr Storage Specification Version 2

ISO/IEC 21320-1:2015 Information technology – Document Container File (Zip)
<https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval>

REQUIREMENT 26

IDENTIFI /req/data-zarr/content

- A Every 200 response of the server for zone data with the media type application/zarr+zip SHALL be a Zip file containing the data values for all selected fields for a single zone data encoded following the Zarr Storage Specification Version 2 file.
- B A dimension of the Zarr data SHALL correspond to the sub-zones for which values are being returned following the DGGRS deterministic sub-zone ordering.
- C If multiple relative zone depths are being returned, each depth SHALL be encoded as a separate Zarr dataset within the data packet.
- D Every zone depth requested using the zone-depth parameter SHALL be included in the response.

REQUIREMENT 26

- E At every depth, each individual value SHALL correspond exactly to the data sampled representative of that sub-zone for either the overall area of the sub-zone, the sub-zone centroid or the primary vertex of the sub-zone.
- F The list of data values SHALL follow the default zone order as specified by the Discrete Global Grid Reference System (for example based on a scanline or space-filling curved defined therein) for which the request is made.
- G For the purpose of identifying the primary vertex, assuming the following approach results in uniquely mapping a single vertex for every zone of each discrete global grid, for the first sub-zone, the primary vertex SHALL correspond to the vertex furthest away from the next sub-zone in the DGGRS sub-zone order which is at different coordinates for both spatial axes. For example, for square zones using a quad-tree refinement with a top to bottom row by row sub-zone order starting with the top-left zone (whether jumping back to the start of the next row, or alternating between left to right and right to left), the primary vertex will be the top-left corner.
- H For a DGGRS where this primary vertex identification approach does not result in uniquely mapping a single vertex for every zone of a DGG, the primary vertex SHALL be identified as specified in the DGGRS definition.

RECOMMENDATION 17

IDENTIFIE /rec/data-zarr/no-extra-chunking

- A Since the zone data packet are for a limited set of sub-zones within a single parent zone, the zone data response SHOULD be encoded as a single chunk.

14.9. Requirements Class “CoverageJSON zone data encoding”

14.9.1. Overview

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

REQUIREMENTS CLASS 16: REQUIREMENTS CLASS COVERAGEJSON DATA

IDENTIFIER	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-coveragejson
------------	---

REQUIREMENTS CLASS 16: REQUIREMENTS CLASS COVERAGEJSON DATA

TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.16: https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-coveragejson
PREREQUISITES	OGC 21-069r2 https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval

REQUIREMENT 27

IDENTIFIER /req/data-coveragejson/content

- A Every 200 response of the server for zone data with the media type application/prs.coverage+json SHALL be a CoverageJSON file representing the data values for all selected fields for a single zone.
- B If the CoverageJSON encoding incorporates georeferencing information, this information SHALL be consistent with the DGGRS Zone ID.
- C If the zone geometry is not rectilinear, the closest bounding rectangle (or volume) SHALL be used for referencing the data.
- D A distinct value for each sub-zone implied from the requested zone-depth SHALL correspond to at least one distinct cell value in the response.

RECOMMENDATION 18

IDENTIFIER /rec/data-coveragejson/null-values

- A Null values SHOULD be used for cells lying outside the zone geometry.

RECOMMENDATION 19

IDENTIFIER /rec/data-coveragejson/crs

- A The CRS of the response SHOULD be consistent with either the DGGRS or the underlying geographic CRS (e.g., CRS84).

14.10. Requirements Class “JPEG-XL Zone data encoding”

14.10.1. Overview

The JPEG XL Zone Data encoding 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.

REQUIREMENTS CLASS 17: REQUIREMENTS CLASS JPEG XL DATA

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-jpegxl>

TARGET TYPE Web API

CONFORMANCE CLASS Conformance class A.17: <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-jpegxl>

PREREQUISITES
ISO/IEC 18181-1
ISO/IEC 18181-2

<https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval>

REQUIREMENT 28

IDENTIFIER /req/data-jpegxl/content

- A** Every 200 response of the server for zone data with the media type `image/jxl` SHALL be a [JPEG XL](#) image representing the data values for all selected fields for a single zone.
- B** If the zone geometry is not rectangular, the closest bounding rectangle SHALL be used for referencing the image.
- C** A distinct value for each sub-zone implied from the requested zone-depth SHALL correspond to at least one distinct cell value in the response.
- D** For implementations supporting Data Custom Depths, each depth of the requested zone depth pyramid SHALL be a separate image (overview) in the response.

RECOMMENDATION 20

IDENTIFIER /rec/data-jpegxl/null-values

RECOMMENDATION 20

A Null values SHOULD be used for cells lying outside the zone geometry.

RECOMMENDATION 21

IDENTIFI /rec/data-jpegxl/crs

A The CRS of the response SHOULD be consistent with either the DGGRS or the underlying geographic CRS (e.g., CRS84).

14.11. Requirements Class “PNG Zone data encoding”

14.11.1. Overview

The PNG Zone Data encoding 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.

REQUIREMENTS CLASS 18: REQUIREMENTS CLASS PNG DATA

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-png>

TARGET TYPE Web API

CONFORMANCE CLASS Conformance class A.18: <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-png>

PREREQUISITES ISO/IEC 15948
<https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval>

REQUIREMENT 29

IDENTIFI /req/data-png/content

REQUIREMENT 29

- A Every 200 response of the server for zone data with the media type `image/png` SHALL be a PNG image representing the data values for all selected fields for a single zone.
- B If the zone geometry is not rectangular, the closest bounding rectangle SHALL be used for referencing the image.
- C A distinct value for each sub-zone implied from the requested zone-depth SHALL correspond to at least one distinct cell value in the response.
- D For implementations supporting Data Custom Depths, each depth of the requested zone depth pyramid SHALL be a separate image (overview) in the response.
- E A **Values-Scale**: response header with a numeric real value SHALL be returned indicating the factor by which values were multiplied before an offset was added to result in the encoded 8-bit or 16-bit PNG unsigned integer values.
- F A **Values-Offset**: response header with a numeric real value SHALL be returned indicating the offset which was added after multiplying values by the scale factor to result in the encoded 8-bit or 16-bit PNG unsigned integer values.

RECOMMENDATION 22

IDENTIFIER /rec/data-png/null-values

- A Null values SHOULD be used for cells lying outside the zone geometry.

RECOMMENDATION 23

IDENTIFIER /rec/data-png/crs

- A The CRS of the response SHOULD be consistent with either the DGGRS or the underlying geographic CRS (e.g., CRS84).

RECOMMENDATION 24

IDENTIFIER /rec/data-png/scale-offset

- A The Implementation SHOULD support a values-scale query parameter on Zone data resources when requesting a PNG output to indicate the value scale factor, as described in the Values-Scale response header part of the content requirement.
- B The Implementation SHOULD support a values-offset query parameter on Zone data resources when requesting a PNG output to indicate the value offset, as described in the Values-Offset response header part of the content requirement.

15

REQUIREMENTS CLASSES FOR ENCODINGS OF ZONE LIST

REQUIREMENTS CLASSES FOR ENCODINGS OF ZONE LIST

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

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

15.1. Media Types (for zone list)

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

Table 4 – Media Types used for zone list encoding requirements classes

Encoding	media type
JSON zone list	application/json
HTML zone list	text/html
Binary 64-bit zone list	application/x-binary
GeoJSON zone list	application/geo+json
JSON-FG zone list	application/fg+json
GeoTIFF zone list	image/tiff; application=geotiff

15.2. Requirements Class “JSON zone list encoding”

15.2.1. 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.

REQUIREMENTS CLASS 19: REQUIREMENTS CLASS JSON ZONE LIST

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-json>

TARGET TYPE Web API

CONFORMANCE CLASS Conformance class A.19: <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/zone-json>

PREREQUISITES JSON
<https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query>

REQUIREMENT 30

IDENTIFI /req/zone-json/content

- A** Every 200 response of the server for zone query with the media type application/json SHALL be a JSON document listing the textual identifiers for all zones matching the query.
- B** The schema for the JSON document SHALL follow the JSON Schema for DGGS Zone Query described below, where the zone identifiers are strings within a zones array property.
- C** The links property SHALL include an [ogc-rel:dggrs] link to the Discrete Global Grid Reference System description resource.
- D** The links property SHALL include an [ogc-rel:dggrs-definition] link to the DGGRS definition, using the schema defined in Annex B – DGGRS Definitions or a later version.

```
{  
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
```

```

"$id" : "https://schemas.opengis.net/ogcapi/dggs/part1/1.0/openapi/schemas/
dggs-core/dggs-zones.json",
  "type": "object",
  "required": [ "zones" ],
  "properties":
  {
    "zones" : { "type": "array", "items": { "type": "string" } },
    "returnedAreaMetersSquare" : { "type": "number" },
    "returnedVolumeMetersCube" : { "type": "number" },
    "returnedVolumeMetersSquareSeconds" : { "type": "number" },
    "returnedHyperVolumeMetersCubeSeconds" : { "type": "number" },
    "links" :
    {
      "description": "Links to related resources."
      "items": { "$ref" : "https://schemas.opengis.net/ogcapi/dggs/part1/1.
0/openapi/schemas/common-core/link.yaml"
    }
  }
}

```

Figure 9 – JSON Schema for a DGGS Zone Query response

RECOMMENDATION 25

IDENTIFIE/rec/zone-json/additional-links

- A If DGGS Zone Data Retrieval is supported, the linkTemplates property SHOULD include an [ogc-rel:dggrs-zone-data] link with a {zoneId} variable to retrieve data from each of these DGGRS zones.
- B The links property SHOULD include an [ogc-rel:geodata] link for zone listing pertaining to a particular collection (for Collection DGGS requirement class).

RECOMMENDATION 26

IDENTIFIE/rec/zone-json/zone-order

- A For responses where zones of multiple hierarchy levels are returned when compact-zones is true, the zones SHOULD be listed with the coarser resolution level first (larger zones).

RECOMMENDATION 27

LABEL /rec/zone-json/zone-total-area

STATEMENT For returning the total area (or volumes) occupied by zones returned by the query

- A For a 2D DGGS, the Implementation SHOULD include in the response a returnedAreaMetersSquare property indicating the total surface area occupied by the zones being returned, adding the overlaps which may occur for compact zone responses where zones are non-congruent only once.

RECOMMENDATION 27

- B For a 3D DGGS, the Implementation SHOULD include in the response a `returnedVolumeMetersCube` property indicating the total volume occupied by the zones being returned, still counting overlapping regions only once.
- C For a 2D + Time DGGS, the Implementation SHOULD include in the response a `returnedVolumeMetersSquareSeconds` property indicating the total volume occupied by the zones being returned, still counting overlapping regions only once.
- D For a 3D + Time DGGS, the Implementation SHOULD include in the response a `returnedHyperVolumeMetersCubeSeconds` property indicating the total hypervolume occupied by the zones being returned, still counting overlapping regions only once.

15.3. Requirements Class “HTML zone list encoding”

15.3.1. 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.

REQUIREMENTS CLASS 20: REQUIREMENTS CLASS HTML ZONE LIST

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-html>

TARGET TYPE Web API

CONFORMANCE CLASS Conformance class A.20: <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/zone-html>

PREREQUISITES <https://html.spec.whatwg.org/>
<https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query>

REQUIREMENT 31

IDENTIFI /req/zone-html/content

- A Every 200 response of the server for zone query with the media type `text/html` SHALL be an HTML document listing the textual identifiers for all zones matching the query.

RECOMMENDATION 28

IDENTIFIE /rec/zone-html/zone-information

- A The implementation SHOULD include additional information for each zone ID, such as the area, bounding box and/or a preview of the data available for this zone.

RECOMMENDATION 29

IDENTIFIE /rec/zone-html/zone-data

- A If the implementation also supports Zone Data Retrieval, the implementation SHOULD include a link for each zone to download the associated data .

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

15.4.1. 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.

REQUIREMENTS CLASS 21: REQUIREMENTS CLASS 64-BIT BINARY ZONE LIST

IDENTIFIER

<https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-uint64>

REQUIREMENTS CLASS 21: REQUIREMENTS CLASS 64-BIT BINARY ZONE LIST

TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.21: https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/zone-uint64
PREREQUISITE	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query

REQUIREMENT 32

IDENTIFI /req/zone-uint64/content

- A Every 200 response of the server for zone query with the media type application/x-binary SHALL be a binary response consisting of a first 64-bit integer count defining the number of zones returned, followed by one 64-bit integer for each zone matching the query.
- B The 64-bit integer identifiers SHALL be the one defined by the DGGRS.
- C The endianness of the returned count and zones integers SHALL be little endian.
- D If the DGGRS does not define a 64-bit integer identifier, a 406 “Not Acceptable” response SHALL be returned.

15.5. Requirements Class “GeoJSON zone list encoding”

15.5.1. 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.

REQUIREMENTS CLASS 22: REQUIREMENTS CLASS GEOJSON ZONE LIST

IDENTIFIER	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-geojson
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.22: https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/zone-geojson
PREREQUISITES	GeoJSON https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query

REQUIREMENT 33

IDENTIFI /req/zone-geojson/content

- A Every 200 response of the server for zone query with the media type application/geo+json SHALL be a GeoJSON document consisting of a FeatureCollection, where every Feature represents a single zone.
- B Every feature SHALL have a zoneID property corresponding to the textual identifier of the zone.
- C The geometry of each feature SHALL be the geometry of the zone.

RECOMMENDATION 30

IDENTIFI /rec/zone-geojson/id

- A The Feature id SHOULD have the same value as the zoneID property or use the 64-bit integer identifier.

RECOMMENDATION 31

IDENTIFI /rec/zone-geojson/mid-points

- A For DGGRs not in CRS84, the geometry SHOULD include intermediate points between the vertices of the zone geometry so as to accurately represent the shape of the zones.

15.6. Requirements Class “FG-JSON zone list encoding”

15.6.1. Overview

The FG-JSON zone list encoding requirements class defines support for encoding a DGGS Zone list response according to the [FG-JSON 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.

REQUIREMENTS CLASS 23: REQUIREMENTS CLASS FG-JSON ZONE LIST

IDENTIFIER	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-fgjson
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.23: https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/zone-fgjson
PREREQUISITES	OGC 21-045r1 https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query

REQUIREMENT 34

IDENTIFI /req/zone-fgjson/content

- A Every 200 response of the server for zone query with the media type application/fg+json SHALL be a Features & Geometry JSON document consisting of a FeatureCollection, where every Feature represents a single zone.
- B Every feature SHALL have a zoneID property corresponding to the textual identifier of the zone.
- C The geometry of each feature SHALL be the geometry of the zone.

RECOMMENDATION 32

IDENTIFIE /rec/zone-fgjson/id

- A The Feature id SHOULD have the same value as the zoneID property or use the 64-bit integer identifier.

RECOMMENDATION 33

IDENTIFIE /rec/zone-fgjson/mid-points

- A For DGGRSs not in CRS84, the geometry SHOULD include intermediate points between the vertices of the zone geometry so as to accurately represent the shape of the zones.

15.7. Requirements Class “GeoTIFF zone list encoding”

15.7.1. 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.

REQUIREMENTS CLASS 24: REQUIREMENTS CLASS GEOTIFF ZONE LIST

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-geotiff>

TARGET TYPE Web API

REQUIREMENTS CLASS 24: REQUIREMENTS CLASS GEOTIFF ZONE LIST

PREREQUISITES	TIFF V6.0 GeoTIFF https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query
---------------	---

REQUIREMENT 35

IDENTIFI /req/zone-geotiff/content

- A Every 200 response of the server for zone query with the media type image/tiff SHALL be a Geo TIFF document representing the zones matching the query in a geo-referenced image where each zone corresponds to at least one pixel.
- B The GeoTIFF SHALL be encoded as Pixel-Is-Area.

RECOMMENDATION 34

IDENTIFI /rec/zone-geotiff/crs

- A The CRS of the response SHOULD be consistent with either the DGGRS or the underlying geographic CRS (e.g., CRS84).

RECOMMENDATION 35

IDENTIFI /rec/zone-geotiff/values

- A The value of the pixel SHOULD be a 64-bit integer identifier representing the zone, if the DGGRS provides for such an identifier.

RECOMMENDATION 36

IDENTIFI /rec/zone-geotiff/non-rectilinear

- A For DGGRS where the zones do not correspond to a rectilinear structure, the resolution of the response image SHOULD be high enough so that the shape of the zone geometry is recognizable.

16

REQUIREMENTS CLASS FOR API DEFINITION OPERATION IDS

REQUIREMENTS CLASS FOR API DEFINITION OPERATION IDS

This standard does not mandate any particular API definition language, but if the API is described using a definition language supporting operation identifiers, such as OpenAPI 3.0, allowing to associate the functionality described in these requirements with an operation defined in the language, this requirement class defines how this is achieved.

16.1. Overview

REQUIREMENTS CLASS 25

TARGET TYPE Web API

PREREQUISITE <https://www.opengis.net/spec/ogcapi-common-1/1.0/req/landing-page>

LABEL <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/operation-ids>

16.2. Operation IDs

REQUIREMENT 36

LABEL /req/operation-ids/operation-ids

STATEMENT For specifying the operation identifiers associated with the capabilities defined in OGC API – DGGS

A The API definition SHALL identify the supported operations defined in this Standard using the identifier suffixes defined in table Table 5.

Table 5 – API operation identifier suffixes

ORIGIN	RESOURCE	OPERATION ID SUFFIXES
<i>With the origins described in this document</i>		
DataSet ⁴	DGGRS List ¹	.dataset.getDGGRSList
DataSet ⁴	DGGRS Description ¹	.dataset.getDGGRS
DataSet ⁴	DGGRS Zones ²	.dataset.getDGGRSZones
DataSet ⁴	DGGRS Zone Information ¹	.dataset.getDGGRSZoneInfo
DataSet ⁴	DGGRS Zone Data ³	.dataset.getDGGRSZoneData
Collection ⁵	DGGRS List ¹	.collection.getDGGRSList
Collection ⁵	DGGRS Description ¹	.collection.getDGGRS
Collection ⁵	DGGRS Zones ²	.collection.getDGGRSZones
Collection ⁵	DGGRS Zone Information ¹	.collection.getDGGRSZoneInfo
Collection ⁵	DGGRS Zone Data ³	.collection.getDGGRSZoneData
<i>With other potential origins⁶</i>		
other	DGGRS List ¹	#.getDGGRSList
other	DGGRS Description ¹	#.getDGGRS
other	DGGRS Zones ²	#.getDGGRSZones
other	DGGRS Zone Information ¹	#.getDGGRSZoneInfo
other	DGGRS Zone Data ³	#.getDGGRSZoneData
¹ The DGGRS List (.../dggs), DGGRS Description (.../dggs/{dggrsId}) and DGGRS Zone Information (.../dggs/{dggrsId}/zones/{zoneId}) resources are defined in requirements class “Core”.		
² The DGGRS Zones resource (.../dggs/{dggrsId}/zones) is defined in the “Zone Query” requirements class.		
³ The DGGRS Zone Data resource (.../dggs/{dggrsId}/zones/{zoneId}/data) is defined in the “Data Retrieval” requirements class.		

ORIGIN	RESOURCE	OPERATION ID SUFFIXES
⁴	The DataSet origin is defined in requirements class “Root DGGS” and depends on OGC API – Common – Part 1: Core.	
⁵	The Collection origin is defined in requirements class “Collection DGGS” and depends on the Collections requirements class defined in OGC API – Common – Part 2: Geospatial data.	
⁶	#	represents an optional origin that could be defined in another relevant standard.



A

ANNEX A (NORMATIVE) CONFORMANCE CLASS ABSTRACT TEST SUITE

ANNEX A (NORMATIVE)

CONFORMANCE CLASS ABSTRACT TEST SUITE

A.1. Conformance Class “Core”

CONFORMANCE CLASS A.1

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/core>

**REQUIREMENTS
CLASS** <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/core>

TARGET TYPE Web API

**CONFORMANCE
TESTS**
Abstract test A.1-1: /req/core/dggrs-list
Abstract test A.1-2: /req/core/dggrs-info
Abstract test A.1-3: /req/core/zone-info

A.1.1. Abstract Test for Requirement Listing available DGGRS

ABSTRACT TEST A.1

IDENTIFIER /conf/core/dggrs-list

REQUIREMENT Abstract test A.1-1: /req/core/dggrs-list

**TEST
PURPOSE** Verify that the implementation supports listing the available DGGRSs

**TEST
METHOD** **Given:** a geospatial data resource conforming to the DGGS API Standard, with an API path including ... /dggs or discovered following a link with relation type [ogc-rel:dggrs-list]

ABSTRACT TEST A.1

When: performing a GET operation on the .../dggs resource with an application/json media type specified in the Accept: header (e.g., Accept: application/json)
Then: (assert that all parts of the requirement are fully satisfied)

A.1.2. Abstract Test for Discrete global grid reference system information

ABSTRACT TEST A.2

IDENTIFIER /conf/core/dggrs-info

REQUIREMENT Abstract test A.1-2: /req/core/dggrs-info

TEST PURPOSE Verify that the implementation supports retrieving information for a particular DGGRSs

TEST METHOD
Given: a DGGRS resource identified from listing available DGGRS or discovered by following a link with relation type [ogc-rel:dggrs]
When: performing a GET operation on the .../dggs/{dggrsId} resource with an application/json media type specified in the Accept: header (e.g., Accept: application/json)
Then: (assert that all parts of the requirement are fully satisfied)

A.1.3. Abstract Test for Retrieving zone information

ABSTRACT TEST A.3

IDENTIFIER /conf/core/zone-info

REQUIREMENT Abstract test A.1-3: /req/core/zone-info

TEST PURPOSE Verify that the implementation supports retrieving information for a particular DGGRS zone

TEST METHOD
Given: a DGGRS zone information resource identified from a [ogc-rel:dggrs-zone-info] link template within a DGGRS description resource, replacing the {zoneId} variable with a known valid identifier for the DGGRS, not beyond the maxRefinementLevel if one is specified
When: performing a GET operation on the .../dggs/{dggrsId}/zones/{zoneID} resource with an application/json media type specified in the Accept: header (e.g., Accept: application/json)
Then: (assert that all parts of the requirement are fully satisfied)

A.2. Conformance Class “Data Retrieval”

CONFORMANCE CLASS A.2

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-retrieval>

REQUIREMENTS CLASS <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval>

TARGET TYPE Web API

A.2.1. Abstract Test for Retrieving data from a zone

ABSTRACT TEST A.4

IDENTIFIER /conf/data-retrieval/zone-data

REQUIREMENT /req/data-retrieval/zone-data

TEST PURPOSE Verify that the implementation supports retrieving data for a particular DGGRS zone

TEST METHOD

Given: a DGGRS zone data resource identified from a [ogc-rel:dggrs-zone-data] link template within a DGGRS description resource, replacing the {zoneId} variable with a known valid identifier for the DGGRS, not beyond the maxRefinementLevel if one is specified

When: performing a GET operation on the .../dggs/{dggrsId}/zones/{zoneID}/data resource with supported media type specified in the Accept: header (e.g., Accept: application/json for DGGS-JSON)

Then: (assert that all parts of the requirement are fully satisfied)

A.3. Conformance Class “Data Subsetting”

CONFORMANCE CLASS A.3

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-subsetting>

CONFORMANCE CLASS A.3

REQUIREMENTS CLASS <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-subsetting>

TARGET TYPE Web API

A.4. Conformance Class “Data Custom Depths”

CONFORMANCE CLASS A.4

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-custom-depths>

REQUIREMENTS CLASS <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-custom-depths>

TARGET TYPE Web API

A.5. Conformance Class “Zone Query”

CONFORMANCE CLASS A.5

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/zone-query>

REQUIREMENTS CLASS <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query>

TARGET TYPE Web API

A.5.1. Abstract Test for Listing zones

ABSTRACT TEST A.5

IDENTIFIER /conf/zone-query/zones-list

ABSTRACT TEST A.5

REQUIREMENT /req/zone-query/zones-list

TEST PURPOSE Verify that the implementation supports querying DGGRS zones

TEST METHOD Given: a DGGRS zones query resource identified from a [ogc-rel:dggrs-zone-query] link
When: performing a GET operation on the .../dggs/{dggrsId}/zones resource with supported media type specified in the Accept : header (e.g., Accept: application/json for a JSON zone list)
Then: (assert that all parts of the requirement are fully satisfied)

A.6. Conformance Class “Filtering Zone Queries with CQL2”

CONFORMANCE CLASS A.6

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/cql2-query>

REQUIREMENTS CLASS <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/cql2-query>

TARGET TYPE Web API

A.7. Conformance Class “Root DGGS”

CONFORMANCE CLASS A.7

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/root-dggs>

REQUIREMENTS CLASS <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/root-dggs>

TARGET TYPE Web API

A.8. Conformance Class “Collection DGGS”

CONFORMANCE CLASS A.8

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/collection-dggs>

REQUIREMENTS CLASS <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/collection-dggs>

TARGET TYPE Web API

A.9. Conformance Class “DGGS-JSON encoding for Zone Data”

CONFORMANCE CLASS A.9

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-json>

REQUIREMENTS CLASS Requirements class 9: <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-json>

TARGET TYPE Web API

A.10. Conformance Class “DGGS-UBJSON encoding for Zone Data”

CONFORMANCE CLASS A.10

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-ubjson>

REQUIREMENTS CLASS Requirements class 10: <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-ubjson>

CONFORMANCE CLASS A.10

TARGET TYPE Web API

A.11. Conformance Class “GeoTIFF encoding for Zone Data”

CONFORMANCE CLASS A.11

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-tiff>

REQUIREMENTS CLASS <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-tiff>

TARGET TYPE Web API

A.12. Conformance Class “GeoJSON encoding for Zone Data”

CONFORMANCE CLASS A.12

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-geojson>

REQUIREMENTS CLASS Requirements class 12: <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-geojson>

TARGET TYPE Web API

A.13. Conformance Class “FG-JSON encoding for Zone Data”

CONFORMANCE CLASS A.13

IDENTIFIER	https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-fgjson
REQUIREMENTS CLASS	Requirements class 13: https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-fgjson
TARGET TYPE	Web API

A.14. Conformance Class “netCDF encoding for Zone Data”

CONFORMANCE CLASS A.14

IDENTIFIER	https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-netcdf
REQUIREMENTS CLASS	Requirements class 14: https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-netcdf
TARGET TYPE	Web API

A.15. Conformance Class “Zarr encoding for Zone Data”

CONFORMANCE CLASS A.15

IDENTIFIER	https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-zarr
REQUIREMENTS CLASS	Requirements class 15: https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-zarr
TARGET TYPE	Web API

A.16. Conformance Class “CoverageJSON encoding for Zone Data”

CONFORMANCE CLASS A.16

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-coveragejson>

REQUIREMENTS CLASS Requirements class 16: <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-coveragejson>

TARGET TYPE Web API

A.17. Conformance Class “JPEG XL Encoding for Zone Data”

CONFORMANCE CLASS A.17

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-jpegxl>

REQUIREMENTS CLASS Requirements class 17: <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-jpegxl>

TARGET TYPE Web API

A.18. Conformance Class “PNG encoding for Zone Data”

CONFORMANCE CLASS A.18

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/data-png>

REQUIREMENTS CLASS Requirements class 18: <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-png>

CONFORMANCE CLASS A.18

TARGET TYPE Web API

A.19. Conformance Class “JSON encoding for Zone Listing”

CONFORMANCE CLASS A.19

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/zone-json>

REQUIREMENTS Requirements class 19: <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-json>
CLASS

TARGET TYPE Web API

A.20. Conformance Class “HTML encoding for Zone Listing”

CONFORMANCE CLASS A.20

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/zone-html>

REQUIREMENTS Requirements class 20: <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-html>
CLASS

TARGET TYPE Web API

A.21. Conformance Class “Binary 64-bit integer encoding for Zone Listing”

CONFORMANCE CLASS A.21

IDENTIFIER	https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/zone-uint64
REQUIREMENTS CLASS	Requirements class 21: https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-uint64
TARGET TYPE	Web API

A.22. Conformance Class “GeoJSON encoding for Zone Listing”

CONFORMANCE CLASS A.22

IDENTIFIER	https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/zone-geojson
REQUIREMENTS CLASS	Requirements class 22: https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-geojson
TARGET TYPE	Web API

A.23. Conformance Class “FG-JSON encoding for Zone Listing”

CONFORMANCE CLASS A.23

IDENTIFIER	https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/zone-fgjson
REQUIREMENTS CLASS	Requirements class 23: https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-fgjson
TARGET TYPE	Web API

A.24. Conformance Class “GeoTIFF encoding for Zone Listing”

CONFORMANCE CLASS A.24

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/zone-tiff>

REQUIREMENTS CLASS <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-tiff>

TARGET TYPE Web API

A.25. Conformance Class “Operation IDs”

CONFORMANCE CLASS A.25

IDENTIFIER <https://www.opengis.net/spec/ogcapi-dggs/1.0/conf/operation-ids>

REQUIREMENTS CLASS <https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/operation-ids>

TARGET TYPE Web API

A.25.1. Abstract Test for Operation IDs

ABSTRACT TEST A.6

IDENTIFIER /conf/operation-ids/operation-ids

REQUIREMENT /req/operation-ids/operation-ids

TEST PURPOSE Verify that the correct operation IDs suffixed are used in the API definition to identify DGGS resources

TEST METHOD **Given:** an DGGS API implementation providing an API definition with a concept of operation IDs (such as OpenAPI 3.0)
When: inspecting the operations in the API definition for DGGS resources

ABSTRACT TEST A.6

Then:

- assert that the operation IDs for DGGS resources correspond to those specified in Table 5

B

ANNEX B (INFORMATIVE) DISCRETE GLOBAL GRID REFERENCE SYSTEM DEFINITIONS

ANNEX B (INFORMATIVE)

DISCRETE GLOBAL GRID REFERENCE SYSTEM DEFINITIONS

The following informative schema describes the expected response of the resource linked to by using the [ogc-rel:dgtrs-definition] link relation from the DGGRS description, as specified by the Core requirement class. An important objective for this schema is to support establishing authoritative registries of DGGRSs providing a URI for a DGGRS corresponding to a specific definition.

Note that this schema is preliminary, and a normative version may be defined by a later version of this standard or by a future part of OGC Abstract Specification Topic 21. The schema may change to allow for better description of existing and new classes of DGGS.

The DGGRS definition consists of three main components:

- the definition of the hierarchy of discrete global grids (dggs), including any parameterization such as the spheroid or orientation, establishing both the topology as well as the geometry of every zone,
- the Zonal Identifier Reference System (zirs), with textual identifiers being mandatory while 64-bit integer identifiers (which support the 64-bit integer zone list encoding requirement class) are optional,
- the sub-zone ordering (subZoneOrder), which allows for retrieving zone data using e.g., DGGS-JSON at a zone-depth other than 0 (the values of sub-zones can be provided as a 1D array of values with a pre-established association to sub-zones).

B.1. JSON Schema for defining a DGGRS

```
{
  "$schema": "https://json-schema.org/draft/2019-09/schema",
  "description": "Discrete Global Grid System Reference System (DGGRS) Definition.",
  "type": "object",
  "required": [
    "dggs",
    "zirs",
    "subZoneOrder"
  ],
  "properties": {
    "dggs": {
      "type": "object",
      "required": [
        "id"
      ],
      "properties": {
        "id": {
          "type": "string"
        }
      }
    },
    "zirs": {
      "type": "array",
      "items": {
        "type": "object",
        "required": [
          "id"
        ],
        "properties": {
          "id": {
            "type": "string"
          }
        }
      }
    },
    "subZoneOrder": {
      "type": "array",
      "items": {
        "type": "integer"
      }
    }
  }
}
```

```

        "subZoneOrder"
    ],
    "properties": {
        "title": {
            "type": "string"
        },
        "description": {
            "type": "string"
        },
        "uri": {
            "type": "string",
            "format": "uri",
            "description": "The authoritative URI associated with this DGGRS definition"
        },
        "dggs": {
            "type": "object",
            "description": "The hierarchical series of Discrete Global Grid upon which this DGGRS is based, including any parameters.",
            "properties": {
                "definition": {
                    "description": "The base definition of the hierarchical series of Discrete Global Grid, which may be parameterized.",
                    "type": "object",
                    "required": [
                        "spatialDimensions",
                        "temporalDimensions"
                    ],
                    "properties": {
                        "crs": {
                            "allOf": [
                                {
                                    "description": "The native Coordinate Reference System (CRS) in which the geometry of the zones for this DGGS is defined."
                                },
                                {
                                    "anyOf": [
                                        {
                                            "type": "string",
                                            "format": "uri",
                                            "example": "EPSG:7789"
                                        }
                                    ]
                                }
                            ]
                        },
                        "basePolyhedron": {
                            "type": "string",
                            "description": "The Type/Class of Polyhedron used to construct the Discrete Global Grid System - if it is constructed using a Base Polyhedron.",
                            "example": "icosahedron"
                        },
                        "refinementRatio": {
                            "description": "The ratio of the area of zones between two consecutive hierarchy level (the ratio of child zones to parent zones, also called the aperture).",
                            "example": 9,
                            "type": "integer"
                        },
                        "constraints": {
                            "type": "object",
                            "properties": {

```

```

        "cellAxisAligned": {
            "description": "Set to true if all edges of the geometry of
all zones are aligned with one of the axis of the `crs`.",
            "type": "boolean",
            "default": false
        },
        "cellConformal": {
            "type": "boolean",
            "default": false
        },
        "cellEquiAngular": {
            "type": "boolean",
            "default": false
        },
        "cellEquiDistant": {
            "type": "boolean",
            "default": false
        },
        "cellEqualSized": {
            "description": "Set to true if the area of all zones is the
same for a particular zone geometry type of any specific discrete global grid
of the DGGS hierarchy.",
            "type": "boolean",
            "default": false
        }
    }
},
"spatialDimensions": {
    "description": "Number of Spatial Dimensions defined by the
Discrete Global Grid System.",
    "example": 2,
    "type": "integer"
},
"temporalDimensions": {
    "description": "Number of Temporal Dimensions defined by the
Discrete Global Grid System.",
    "example": 0,
    "type": "integer"
},
"zoneTypes": {
    "type": "array",
    "items": {
        "type": "string",
        "enum": [
            "triangle",
            "square",
            "hexagon",
            "pentagon",
            "rhombus"
        ]
    },
    "example": [
        "hexagon",
        "pentagon"
    ]
},
"refinementStrategy": {
    "description": "The refinement strategy used by the Discrete
Global Grid System",
    "type": "array",
    "items": {
        "type": "string",
        "enum": [

```

```

        "centredChildCell",
        "nestedChildCell",
        "nodeCentredChildCell",
        "edgeCentredChildCell",
        "faceCentredChildCell",
        "solidCentredChildCell"
    ],
},
"example": [ "nestedChildCell" ]
}
}
},
"parameters": {
  "description": "The optional parameters establishing a very specific Discrete Global Grid System, where each zone has a well-defined geometry.",
  "type": "object",
  "properties": {
    "ellipsoid": {
      "type": "string",
      "format": "uri",
      "description": "Globe Reference System Identifier/Specification",
      "example": [
        "EPSG:7019"
      ]
    },
    "orientation": {
      "type": "object",
      "properties": {
        "latitude": {
          "description": "Reference Latitude in decimal degrees to fix the orientation of the polyhedron.",
          "example": 0,
          "type": "number"
        },
        "longitude": {
          "description": "Reference Longitude in decimal degrees to fix the orientation of the polyhedron.",
          "example": 0,
          "type": "number"
        },
        "description": {
          "type": "string"
        }
      },
      "required": [
        "longitude",
        "latitude"
      ]
    }
  }
},
"zirs": {
  "description": "The Zone Identifier Reference System used for this Discrete Global Grid System Reference System.",
  "type": "object",
  "required": [
    "textZIRS"
  ],
  "properties": {
    "textZIRS": {
      "description": "textual zone identifier indexing scheme",

```

```

    "type": "object",
    "required": [
        "description"
    ],
    "properties": {
        "description": {
            "type": "string",
            "examples": [
                "Topological Morton Code (Space Filling Curve) Identifier.  

With a Level 0 alphabetical identifier from 'A' to 'S' followed by a numeric  

reference from 1 to 9 for all subsequent levels representing the arrangement  

of child zones following a Morton (Z-code) subdivision of the parent zone. The  

length of the identifier represents the level of refinement.",
                "Universal Morton Code (Space Filling Curve) Identifier.  

An ellipsoidal bitwise concatenation of longitude and latitude direct  

position to form a direct Morton code. This can be used across multiple DGGS  

infrastructures to provide a common index structure that is fixed in nature  

regardless of the topological zone index used by each DGGS.",
                "An identifier comprised of three separate level, row and  

column components corresponding to an OGC 2D Tile Matrix Set tile identifier,  

where the individual component are represented by uppercase hexadecimal  

numbers separated by a hyphen.",
                "An identifier comprised of an uppercase letter representing  

a level (A for level 0, B for level 1...), followed by a number from 0 to 9  

representing the root rhombus corresponding to the faces of an icosahedron,  

in a staircase order from the top-left to the bottom right of an ISEA planar  

projection after a 60 degrees clockwise rotation, followed by a hyphen, then  

by an uppercase hexadecimal number identifying a sub-zone starting to count  

from 0 in row-major order.",
                "An identifier comprised of an uppercase letter representing  

the ISEA9R level corresponding to the current even or immediate lower odd  

ISEA3H level (A for level 0, B for level 1...), followed by a number from 0 to  

9 representing the root rhombus corresponding to the faces of an icosahedron,  

in a staircase order from the top-left to the bottom right of an ISEA planar  

projection after a 60 degrees clockwise rotation, followed by a hyphen, then  

by an uppercase hexadecimal number identifying a sub-zone starting to count  

from 0 in row-major order, followed by a second hyphen, then by a letter  

identifying a hexagonal or pentagonal zone intersecting the sub-rhombus: A:  

even level regular zone B: even level North pole zone C: even level South  

pole zone D: odd level zone centered on rhombus vertex E: odd level zone top-  

right of rhombus vertex F: odd level zone bottom-right of rhombus vertex G:  

odd level North pole zone H: odd level South pole zone\n"
            ]
        }
    },
    "type": {
        "anyOf": [
            {
                "type": "string"
            },
            {
                "type": "string",
                "enum": [
                    "hierarchicalConcatenation",
                    "ogc2DTMSHexLevelRowCol",
                    "levelRootFaceHexRowMajorSubZone"
                ]
            }
        ]
    }
},
"uint64ZIRS": {
    "description": "64-bit unsigned integer zone indexing scheme",

```

```

    "type": "object",
    "required": [
        "description"
    ],
    "properties": {
        "description": {
            "type": "string",
            "examples": [
                "A 64-bit integer with the 30 least significant bits corresponds to a column, the next 29 bits corresponding to a row, and the 5 most significant bits correspond to a level of an OGC 2D Tile Matrix Set identifier, with individual components using little endian.",
                "A 64-bit integer with the 3 least significant bits corresponding to a hexagonal or pentagonal zone intersecting a sub-rhombus: 0: even level regular zone 1: even level North pole zone 2: even level South pole zone 3: odd level zone centered on rhombus vertex 4: odd level zone top-right of rhombus vertex 5: odd level zone bottom-right of rhombus vertex 6: odd level North pole zone 7: odd level South pole zone, the next 51 bits corresponding to a sub-rhombus index, the next 4 bits corresponding to a root rhombus, the next 5 bits corresponding to an associated ISEA9R level, and the last most significant bit always 0 for a valid zone identifier.\n"
            ]
        },
        "type": {
            "anyOf": [
                {
                    "type": "string"
                },
                {
                    "type": "string",
                    "enum": [
                        "ogc2DTMSHexLevelRowCol"
                    ]
                }
            ]
        }
    }
},
"subZoneOrder": {
    "description": "The ordering used for this Discrete Global Grid System Reference System when encoding the values associated with sub-zones at any given depth relative to a parent zone.",
    "type": "object",
    "required": [
        "description"
    ],
    "properties": {
        "description": {
            "type": "string",
            "examples": [
                "The zones are ordered in tightly packed scanlines. For an odd relative depth, scanlines start along a hexagon edge, whereas for even relative depths, scanlines start on a hexagon vertex. When the parity (even/odd) of the parent zone level corresponds to the parity of the relative depth, zones are ordered left to right along a scanline and scanlines are ordered from top to bottom (considering the ISEA planar projection). When the parity of the parent zone level and relative depth do not match, zones are ordered top to bottom along a scanline and scanlines are ordered from left to right."
            ]
        },
        "type": {

```

```

    "anyOf": [
      {
        "type": "string"
      },
      {
        "type": "string",
        "enum": [
          "scanline",
          "spiralFromCenter",
          "mortonCurve",
          "hilbertCurve"
        ]
      }
    ]
  }
}

```

Figure B.1 – JSON Schema for the DGGRS Definition

The following example JSON DGGRS definitions conform to this schema.

B.2. ISEA9R DGGRS definition

The following DGGRS is for an axis-aligned and equal-area DGGS based on the [Icosahedral Snyder Equal-Area \(ISEA\)](#) planar projection (see [PROJ](#) and [geogrid](#) for open-source implementations) using rhombuses with a refinement ratio of 9. When the ISEA projection is rotated 60 degrees, sheared horizontally by 30 degrees and units are mapped to a 5×6 space, the zones become square and their edges are aligned with the CRS axes, while remaining equal-area. A [2D Tile Matrix Set](#) can also be defined in this transformed CRS, allowing for compatibility with [OGC API – Tiles](#), [WMTS](#) and traditional client and visualization software with no built-in DGGS functionality. This transformed CRS no longer preserves angles and distances, although the topology, indexing and surface geometry on the Earth model remains the same as in the original ISEA CRS where angles and distances are preserved.

```

{
  "title": "ISEA9R",
  "description": "An ISEA9R Discrete Global Grid Reference System using the standard ISEA orientation and WGS84 authalic sphere, a {level}{rootRhombus}-{hexSubZone} indexing system, and scanline-based sub-zone ordering. The ten root rhombuses are formed by combining two icosahedron triangles at their base. Depending on whether the ISEA planar CRS is rotated, sheared and scaled or not, the geometry of the zones are either squares or rhombuses, and the zone edges are either axis-aligned, or the angles and distances are preserved. In either case, the topology, indexing and surface geometry on the Earth model remains the same. The ISEA9R DGGS is a dual of the ISEA3H DGGS even levels (a vertex/node in ISEA9R is the centroid of an ISEA3H zone).",
  "uri": "https://www.opengis.net/def/dggrs/OGC/1.0/ISEA9R",
  "links": [ { "rel": "describedby", "href": "https://portal.ogc.org/files/?artifact_id=107432" } ],
  "dggs": {

```

```

"definition": {
    "spatialDimensions": 2,
    "temporalDimensions": 0,
    "crs": "https://www.opengis.net/def/crs/OGC/0/ISEAr60hs30",
    "links": [
        { "rel": "related", "href": "https://doi.org/10.3138/27H7-8K88-4882-1752" },
        { "rel": "related", "href": "https://proj.org/en/9.3/operations/projections/isea.html" },
    ],
    "basePolyhedron": "icosahedron",
    "refinementRatio": 9,
    "refinementStrategy": [ "nestedChildCell" ],
    "constraints": {
        "cellAxisAligned": true,
        "cellEqualSized": true
    },
    "zoneTypes": [ "square" ],
},
"parameters": {
    "ellipsoid": "Authalic sphere based on [EPSG:7030] (WGS84) -- NOTE: No identifier defined yet? [EPSG:7048] exists for Authalic sphere based on GRS80 [EPSG:7019]",
    "orientation": {
        "description": "Standard ISEA orientation with a single icosahedron vertex falling on land",
        "latitude": 58.28252559,
        "longitude": 11.25
    }
},
"zirs": {
    "textZIRS": {
        "description": "An identifier comprised of an uppercase letter representing a level (A for level 0, B for level 1...), followed by a number from 0 to 9 representing the root rhombus corresponding to the faces of an icosahedron, in a staircase order from the top-left to the bottom right of an ISEA planar projection after a 60 degrees clockwise rotation, followed by a hyphen, then by an uppercase hexadecimal number identifying a sub-zone starting to count from 0 in row-major order.",
        "type": "levelRootFaceHexRowMajorSubZone"
    },
    "uint64ZIRS": {
        "description": "A 64-bit integer with the 30 least significant bits corresponds to a column, the next 29 bits corresponding to a row, and the 5 most significant bits correspond to a level of an OGC 2D Tile Matrix Set identifier, with individual components using little endian.",
        "type": "ogc2DTMSHexLevelRowCol"
    }
},
"subZoneOrder": {
    "description": "The zones are ordered as scanlines in row-major order, left to right, then top to bottom, based on a CRS derived from the planar ISEA projection after rotating it 60 degrees clockwise and applying a 30 degrees horizontal shear.",
    "type": "scanline"
}
}

```

Figure B.2 – DGGRS Definition for ISEA9R

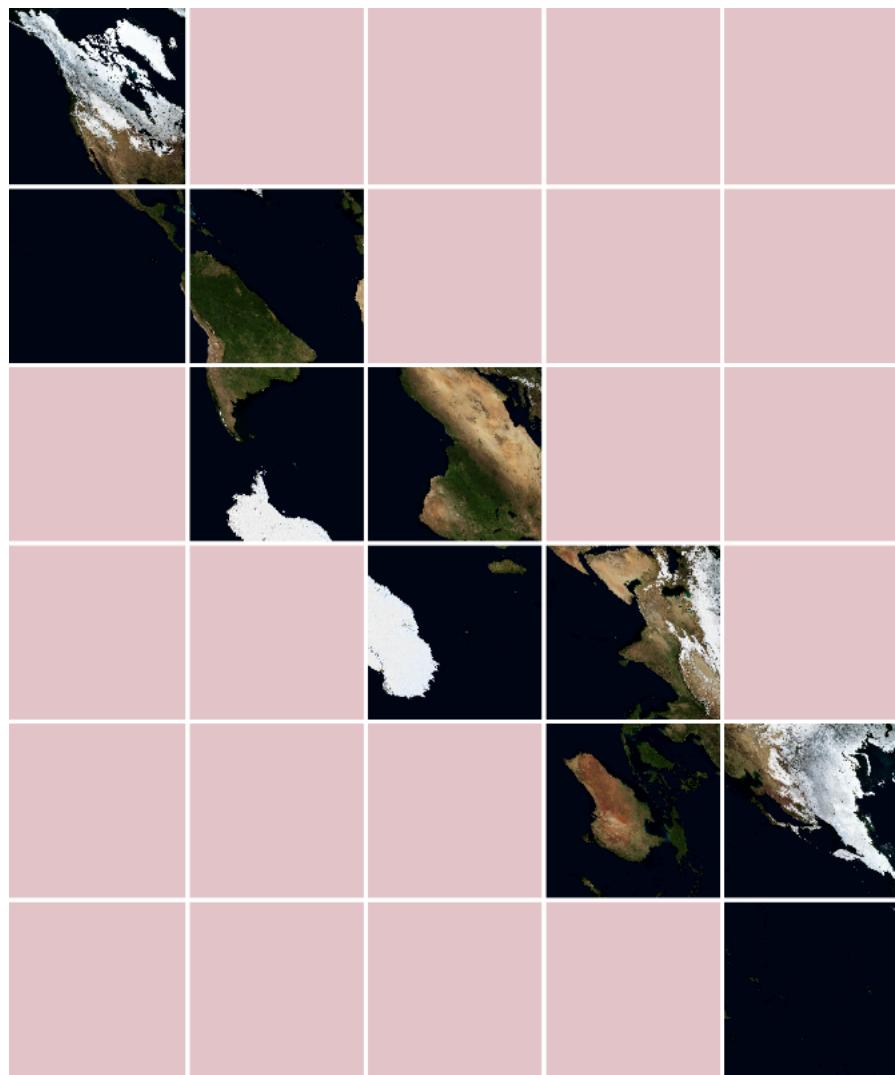


Figure B.3 – The ISEA9R squared root rhombuses (level 0) in the rotated and sheared ISEA planar CRS mapped to a 5×6 space whose axes are aligned with the zone edges, compatible with a 2DTMS definition (imagery from NASA Earth Observatory's [Blue Marble: Next Generation](#))

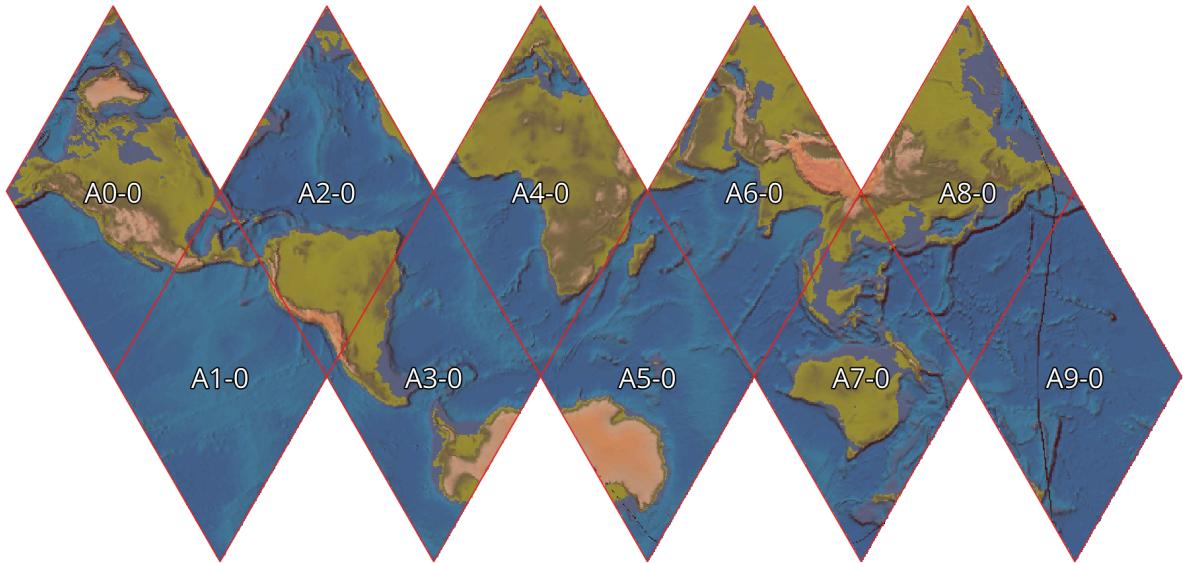


Figure B.4 – The ISEA9R level 0 zone identifiers, shown in the original ISEA planar CRS (GEBCO 2014 bathymetry)

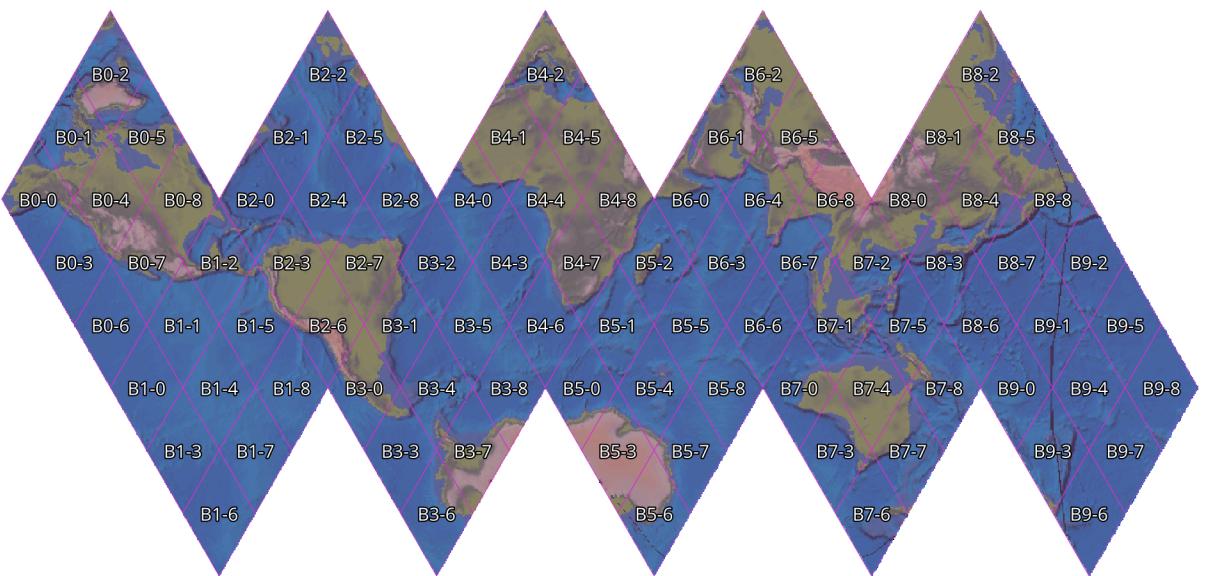


Figure B.5 – The ISEA9R level 1 zone identifiers, shown in the original ISEA planar CRS (GEBCO 2014 bathymetry)

B.3. ISEA3H DGGRS definition

The following DGGRS is for an ISEA3H DGGS, also based on the ISEA planar projection, with hexagonal and pentagonal zones, and a refinement ratio of 3, using a zone reference system

based in part on its dual relationship with the ISEA9R DGGS. The DGGS is considered equal-area, although the 12 pentagonal zones at any level occupy 5/6th of the area of hexagonal zones.

```
{
  "title": "ISEA3H",
  "description": "An ISEA3H Discrete Global Grid Reference System using the standard ISEA orientation and WGS84 authalic sphere, a zone indexing system based on an ISEA9R, and scanline-based ordering",
  "uri": "https://www.opengis.net/def/dggrs/OGC/1.0/ISEA3H",
  "links": [ { "rel": "describedby", "href": "https://portal.ogc.org/files/?artifact_id=107432" } ],
  "dggs": {
    "definition": {
      "spatialDimensions": 2,
      "temporalDimensions": 0,
      "crs": "https://www.opengis.net/def/crs/OGC/0/ISEA",
      "links": [
        { "rel": "related", "href": "https://doi.org/10.3138/27H7-8K88-4882-1752" },
        { "rel": "related", "href": "https://proj.org/en/9.3/operations/projections/isea.html" },
      ],
      "basePolyhedron": "icosahedron",
      "refinementRatio": 3,
      "refinementStrategy": [ "nestedChildCell", "nodeCentredChildCell" ],
      "constraints": {
        "cellEqualSized": true,
        "cellEquiAngular": true,
        "cellEquiDistant": true
      },
      "zoneTypes": [ "hexagon", "pentagon" ],
    },
    "parameters": {
      "ellipsoid": "Authalic sphere based on [EPSG:7030] (WGS84) -- NOTE: No identifier defined yet? [EPSG:7048] exists for Authalic sphere based on GRS80 [EPSG:7019]",
      "orientation": {
        {
          "description": "Standard ISEA orientation with a single icosahedron vertex falling on land",
          "latitude": 58.28252559,
          "longitude": 11.25
        }
      }
    },
    "zirs": {
      "textZIRS": {
        "description": "An identifier comprised of an uppercase letter representing the ISEA9R level corresponding to the current even or immediate lower odd ISEA3H level (A for level 0, B for level 1...), followed by a number from 0 to 9 representing the root rhombus corresponding to the faces of an icosahedron, in a staircase order from the top-left to the bottom right of an ISEA planar projection after a 60 degrees clockwise rotation, followed by a hyphen, then by an uppercase hexadecimal number identifying a sub-zone starting to count from 0 in row-major order, followed by a second hyphen, then by a letter identifying a hexagonal or pentagonal zone intersecting the sub-rhombus: A: even level regular zone B: even level North pole zone C: even level South pole zone D: odd level zone centered on rhombus vertex E:"
      }
    }
  }
}
```

```

        odd level zone top-right of rhombus vertex F: odd level zone bottom-right of
        rhombus vertex G: odd level North pole zone H: odd level South pole zone\n",
        "type": "levelRootFaceHexRowMajorSubZone"
    },
    "uint64ZIRS": {
        "description": "A 64-bit integer with the 3 least significant bits
corresponding to a hexagonal or pentagonal zone intersecting a sub-rhombus:
0: even level regular zone 1: even level North pole zone 2: even level South
pole zone 3: odd level zone centered on rhombus vertex 4: odd level zone top-
right of rhombus vertex 5: odd level zone bottom-right of rhombus vertex 6:
odd level North pole zone 7: odd level South pole zone, the next 51 bits
corresponding to a sub-rhombus index, the next 4 bits corresponding to a root
rhombus, the next 5 bits corresponding to an associated ISEA9R level, and the
last most significant bit always 0 for a valid zone identifier."
    }
},
"subZoneOrder": {
    "description": "The zones are ordered in tightly packed scanlines. For
an odd relative depth, scanlines start along a hexagon edge, whereas for even
relative depths, scanlines start on a hexagon vertex. When the parity (even/
odd) of the parent zone level corresponds to the parity of the relative depth,
zones are ordered left to right along a scanline and scanlines are ordered
from top to bottom (considering the ISEA planar projection). When the parity
of the parent zone level and relative depth do not match, zones are ordered
top to bottom along a scanline and scanlines are ordered from left to right.",
    "type": "scanline"
}
}

```

Figure B.6 – DGGRS Definition for ISEA3H using indexing based on ISEA9R

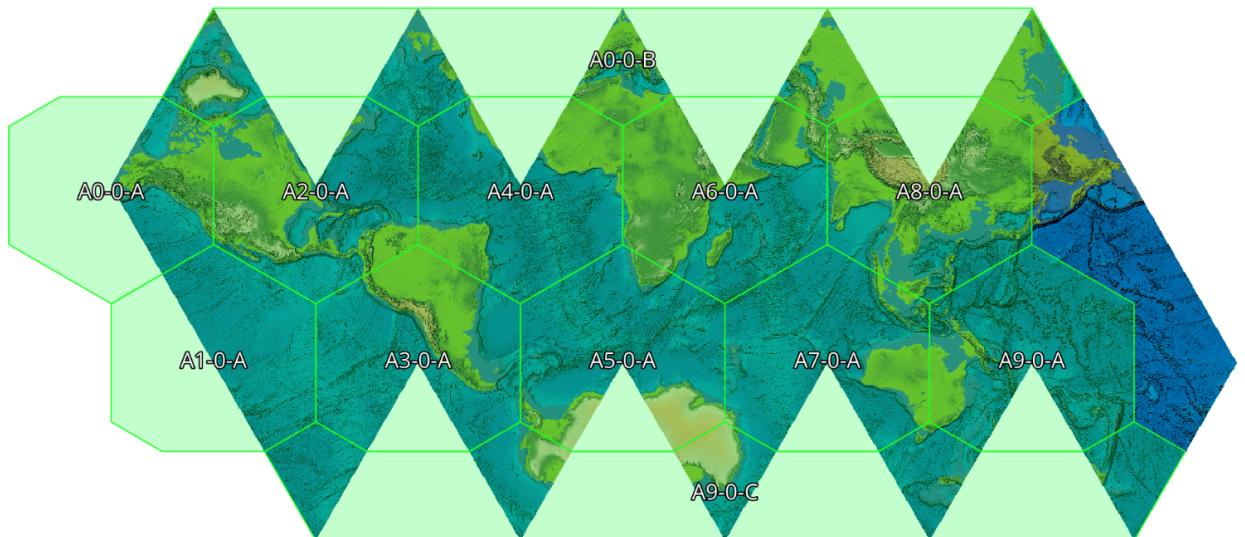


Figure B.7 – The ISEA3H level 0 zones and associated identifiers (derived from ISEA9R level 0), shown on an ISEA planar projection ([GEBCO 2014 bathymetry](#))

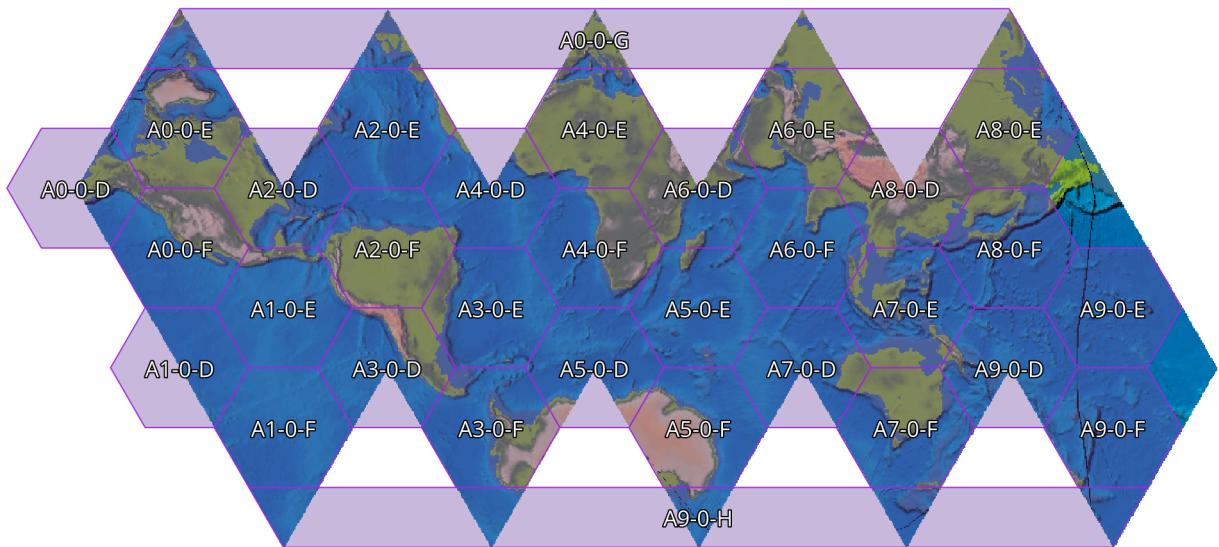


Figure B.8 – The ISEA3H level 1 zones and associated identifiers (derived from ISEA9R level 0), shown on an ISEA planar projection (GEBCO 2014 bathymetry)

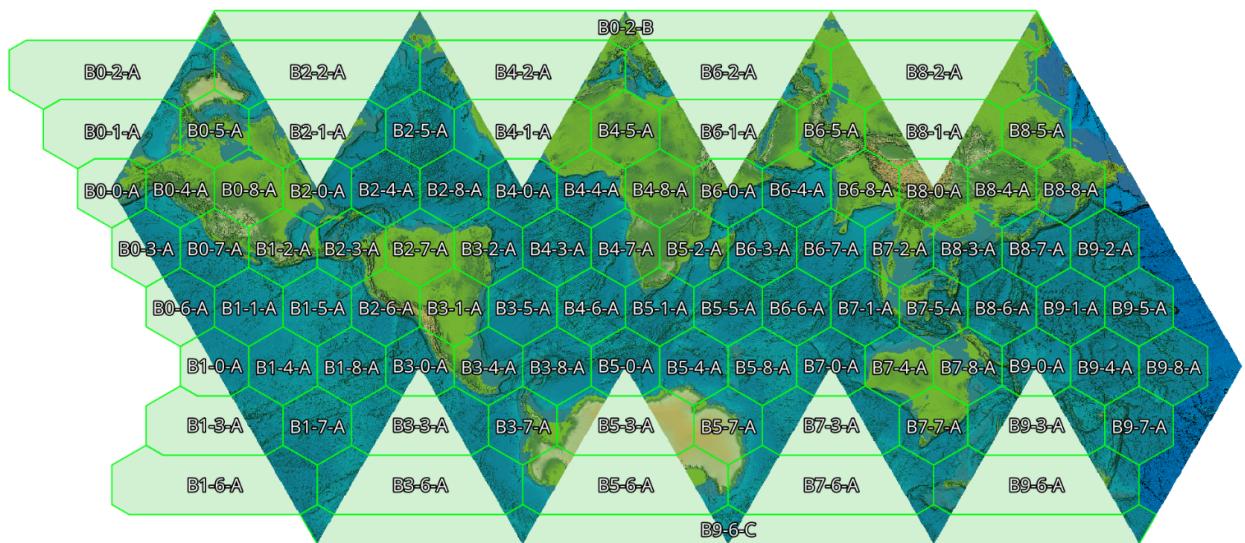


Figure B.9 – The ISEA3H level 2 zones and associated identifiers (derived from ISEA9R level 1), shown on an ISEA planar projection (GEBCO 2014 bathymetry)

B.4. GNOSIS Global Grid DGGRS definition

The following DGGRS is based on the [variable width 2D Tile Matrix Set of the same name](#). The GNOSIS Global Grid is also axis-aligned, with the axes being the geographic latitude and longitude, and already corresponds to a registered 2DTMS. Through the variable width capability of the tile matrices, this DGGS approximates equal area. For example, at any refinement level, only 4 zones touch each pole. Since the error budget still exceeds the 1%,

it cannot be considered equal area. However, the error is the same along a particular row of zones following latitude lines, and can easily be calculated. Since its native axes are latitude and longitude, no additional reprojection is required to import data to or export data from this DGGS.

```
{
  "title": "GNOSISGlobalGrid",
  "description": "A Discrete Global Grid Reference System based on the GNOSIS Global Grid 2D Tile Matrix Set, a {hexLevel}-{hexRow}-{hexCol} indexing system, and scanline-based sub-zone ordering. Making use of variable matrix widths, zones are divided by 4, unless they touch the pole, in which case the half touching the pole is not split longitude-wise. The error budget is ~25.8/+66.0% from mean zone size (+/- 45.9% from median) up to level 25 (~849 cm² mean zones).",
  "uri": "https://www.opengis.net/def/dggrs/OGC/1.0/GNOSISGlobalGrid",
  "links": [ { "rel": "describedby", "href": "https://docs.ogc.org/is/17-083r4/17-083r4.html#toc58" } ],
  "dggs": {
    "definition": {
      "spatialDimensions": 2,
      "temporalDimensions": 0,
      "crs": "https://www.opengis.net/def/crs/EPSG/0/4326",
      "links": [
        { "rel": "related", "href": "https://www.opengis.net/def/tilematrixset/OGC/1.0/GNOSISGlobalGrid" }
      ],
      "basePolyhedron": "octahedron",
      "refinementRatio": 4,
      "refinementStrategy": [ "nestedChildCell" ],
      "constraints": {
        "cellAxisAligned": true
      },
      "zoneTypes": [ "rectangle" ],
    },
    "parameters": {
      "ellipsoid": "[EPSG:7030]"
      "orientation": {
        {
          "latitude": 0,
          "longitude": 0
        }
      }
    },
    "zirs": {
      "textZIRS": {
        "description": "An identifier comprised of three uppercase hexadecimal numbers separated by hyphens representing the level, row and column of the GNOSIS Global Grid 2D Tile Matrix Set.",
        "type": "ogc2DTMSHexLevelRowCol"
      },
      "uint64ZIRS": {
        "description": "A 64-bit integer with the 30 least significant bits corresponds to a column, the next 29 bits corresponding to a row, and the 5 most significant bits correspond to a level of an OGC 2D Tile Matrix Set identifier, with individual components using little endian.",
        "type": "ogc2DTMSHexLevelRowCol"
      }
    },
    "subZoneOrder": {
  
```

```

    "description": "The zones are ordered as scanlines in row-major order,
left to right, then top to bottom, based on the EPSG:4326 CRS.",
    "type": "scanline"
}
}

```

Figure B.10 – DGGRS Definition for the GNOSIS Global Grid

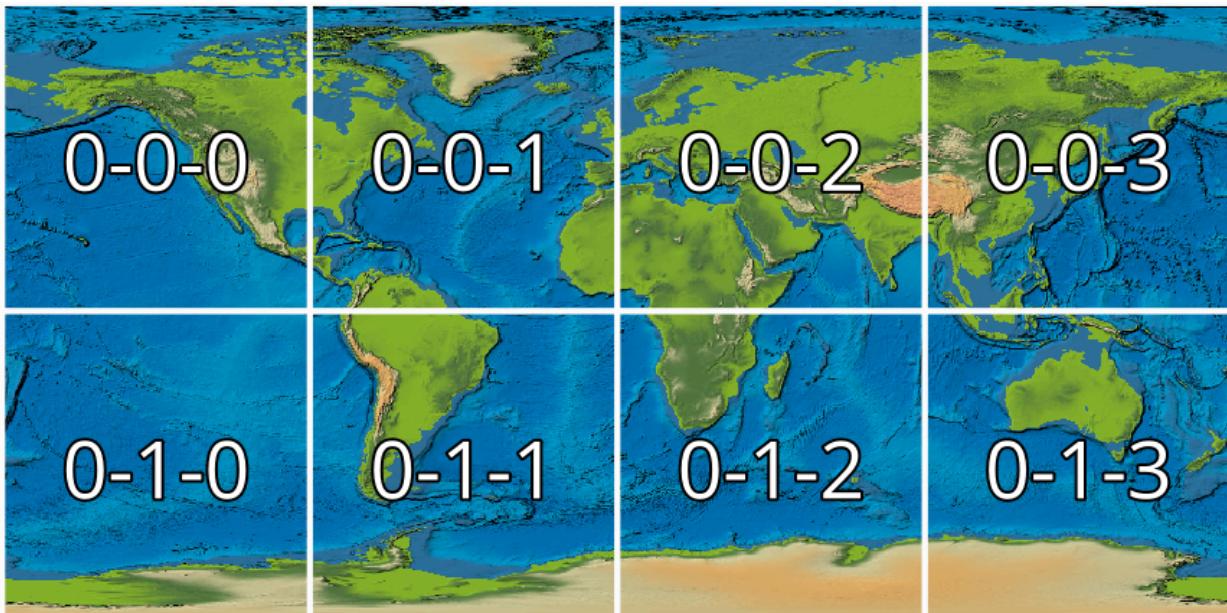


Figure B.11 – The GNOSIS Global Grid level 0 zones and associated identifiers ([GEBCO 2014 bathymetry](#))

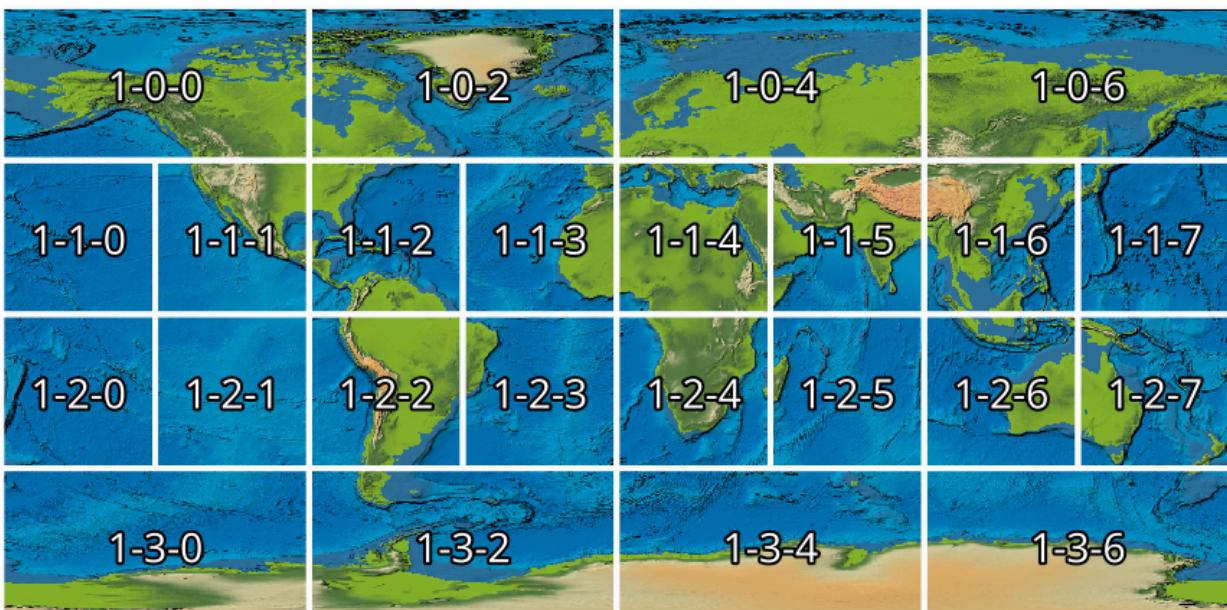


Figure B.12 – The GNOSIS Global Grid level 1 zones and associated identifiers ([GEBCO 2014 bathymetry](#))



Figure B.13 – The GNOSIS Global Grid level 2 zones and associated identifiers ([GEBCO 2014 bathymetry](#))

C

ANNEX C (INFORMATIVE) EXAMPLES

ANNEX C (INFORMATIVE) EXAMPLES

This annex provides a set of examples illustrating requests and responses for retrieving DGGS zone data and performing zone queries using capabilities defined by this DGGS API.

C.1. Collection description

DGGS resources can be attached to an *OGC API – Common – Part 2* collection origin. The following global GEBCO 2014 bathymetry collection will be used as an example:

<https://maps.gnosis.earth/ogcapi/collections/gebco>

An example JSON response for this collection description resource, as defined by Common – Part 2 and the Collection DGGS Requirement Class, is presented below, including a link to the list of available DGGRSSs:

```
{
  "id" : "gebco",
  "title" : "GEBCO 2014",
  "description" : "General Bathymetric Chart of the Oceans (2014)",
  "attribution" : "General Bathymetric Chart of the Oceans (<a href='https://www.gebco.net/news_and_media/gebco_2014_grid.html'>GEBCO_2014</a>)",
  "extent" : {
    "spatial" : {
      "bbox" : [ [ -180, -90, 180, 90 ] ],
      "grid" : [
        { "cellsCount" : 65537, "resolution" : 0.0054931640625 },
        { "cellsCount" : 32769, "resolution" : 0.0054931640625 }
      ]
    } },
  "minScaleDenominator" : 2183915.0938621787354,
  "minCellSize" : 0.0054931640625,
  "links" : [
    { "rel" : "self", "title" : "Information about this GEBCO data collection",
      "href" : "/ogcapi/collections/gebco"
    },
    { "rel" : "[ogc-rel:schema]", "title" : "Schema",
      "href" : "/ogcapi/collections/gebco/schema"
    },
    { "rel" : "[ogc-rel:queryables]", "title" : "Queryables",
      "href" : "/ogcapi/collections/gebco/queryables"
    },
  ],
}
```

```

        { "rel" : "[ogc-rel:dggrs-list]", "title" : "Available DGGRSs for this
GEBCO data collection",
          "href" : "/ogcapi/collections/gebco/dggs"
        }
      ]
    }
  ]
}

```

C.2. Listing available DGGRSs

Whether accessing DGGS resources for the root of an API/dataset or for a particular collection, the first DGGS-specific request made by a client would be to list the available DGGRSs. Following the link obtained from the previous example response, a client can determine which DGGRSs are supported by the API at that end-point:

\https://maps.gnosis.earth/ogcapi/collections/gebco/dggs

An example JSON response for this DGGRS list, as defined by the Core requirement class, is presented below:

```

{
  "links" : [
    { "rel" : "[ogc-rel:geodata]", "href" : "/ogcapi/collections/gebco" }
  ],
  "dggrs" : [
    {
      "id" : "GNOSISGlobalGrid",
      "title" : "GNOSIS Global Grid",
      "uri" : "[ogc-dggrs:GNOSISGlobalGrid]",
      "links" : [
        { "rel" : "self", "title" : "GNOSISGlobalGrid DGGRS for GEBCO",
          "href" : "/ogcapi/collections/gebco/dggs/GNOSISGlobalGrid" },
        { "rel" : "[ogc-rel:dggrs-definition]", "title" :
"GNOSISGlobalGrid DGGRS definition",
          "href" : "/ogcapi/dggrs/GNOSISGlobalGrid"
        }
      ]
    },
    {
      "id" : "ISEA9R",
      "title" : "ISEA Aperture 9 Rhombic",
      "uri" : "[ogc-dggrs:ISEA9R]",
      "links" : [
        { "rel" : "self", "title" : "ISEA9R DGGRS for GEBCO",
          "href" : "/ogcapi/collections/gebco/dggs/ISEA9R"
        },
        { "rel" : "[ogc-rel:dggrs-definition]", "title" : "ISEA9R DGGRS
definition",
          "href" : "/ogcapi/dggrs/ISEA9R"
        }
      ]
    },
    {
      "id" : "ISEA3H",
      "title" : "ISEA Aperture 3 Hexagonal",
      "uri" : "[ogc-dggrs:ISEA3H]"
      "links" : [

```

```

        {
          "rel" : "self", "title" : "ISEA3H DGGRS for GEBCO",
          "href" : "/ogcapi/collections/gebco/dggs/ISEA3H"
        },
        {
          "rel" : "[ogc-rel:dggrs-definition]", "title" : "ISEA3H DGGRS
definition",
          "href" : "/ogcapi/dggrs/ISEA3H"
        }
      ]
    }
}

```

C.3. Retrieving the description of a specific DGGRS

A client would then select a particular DGGRS of which it already has built-in knowledge, possibly using a software library implementing support for one or more DGGRS available from the server.

IMPORTANT

This DGGS API is intended primarily for client / server exchanges where both parties are familiar and can efficiently deal with the same DGGRS. Use cases where the client itself also acts as a server as part of data integration workflows are particularly relevant, where the combination of advanced spatial queries with the ability to request data for specific zones of interest can potentially greatly accelerate analytic processing through parallel distributed processing and lazy evaluation.

As a fallback, a client without native support for a particular DGGRS can still retrieve and interpret data from a server implementing support for both zone queries and data retrieval by performing a two-steps operation. First, the client can perform a zone query for an area of interest using the bbox query parameter. Then, the client can request data for the zones returned by this query in a georeferenced encoding which it understands and which is not tightly coupled with the DGGS (excluding e.g.,the DGGS-JSON encoding), such as GeoJSON or GeoTIFF. However, for such use cases, OGC API – DGGS does not present significant advantages over other OGC API data access mechanisms such as [OGC API – Coverages](#), [OGC API – Features](#) or [OGC API – Tiles](#).

In this example, the client selects the ISEA3H DGGRS (because it has built-in support for it) and follows the link to obtain a complete description of the DGGRS:

<https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H>

An example JSON response for this DGGRS description, as defined by the Core requirement class, is presented below. Note how the description is more comprehensive than the summary description included for each item in the previous DGGRS list response, including a description, a default (relative) depth as well as additional links and link templates.

```
{
  "id" : "ISEA3H",

```

```

    "title" : "ISEA Aperture 3 Hexagonal",
    "description" : "A Discrete Global Grid Reference System based on the
Icosahedral Snyder Equal Area projection, with aperture 3 hexagonal zones,
using an indexing scheme based on ISEA9R.",
    "uri" : "[ogc-dggrs:ISEA3H]",
    "defaultDepth": 10,
    "maxRefinementLevel": 18,
    "links" : [
        { "rel" : "self", "title" : "ISEA3H DGGRS for GEBCO",
          "href" : "/ogcapi/collections/gebco/dggs/ISEA3H"
        },
        { "rel" : "[ogc-rel:dggrs-definition]", "title" : "ISEA3H DGGRS
definition",
          "href" : "/ogcapi/dggrs/ISEA3H"
        },
        { "rel" : "[ogc-rel:dggrs-zone-query]", "title" : "ISEA3H DGGRS Zone
Query",
          "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones"
        },
        { "rel" : "[ogc-rel:geodata]", "href" : "/ogcapi/collections/gebco" }
    ],
    "linkTemplates" : [
        {
            "rel" : "[ogc-rel:dggrs-zone-info]",
            "title" : "DGGRS zone information for a particular ISEA3H zone",
            "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/{zoneId}"
        },
        {
            "rel" : "[ogc-rel:dggrs-zone-data]",
            "title" : "Data retrieval for a particular ISEA3H zone",
            "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/{zoneId}/data"
        }
    ]
}

```

See the informative annex B for examples of DGGRS definition responses which the client could obtain by following the [ogc-rel:dggrs-definition] links. The annex includes provisional definitions for the ISEA3H, ISEA9R and GNOSIS Global Grid used in these examples which will likely be used to initially populate an authoritative registry, Additional DGGRS entries would be submitted, reviewed and registered based on the needs of the community.

C.4. Retrieving information for a specific DGGRS zone

Later examples will demonstrate how to list available zones by performing zone queries, but for now these examples will skip directly to retrieving information about a particular zone. This capability, also defined in the Core requirement class, exists primarily for completeness of the API resource tree, for allowing to explore datasets using a browser interface, as well as for educational, demonstration and debugging purposes. An actual DGGS API client for visualization or performing analytics would normally not access this resource at all, since URL templates allow to directly access the data for a particular zone. While the Core requirement class makes a number of recommendations about statistical information and topological relationships that should be included in the zone information resource, clients may not rely on any of this information being present. Clients can instead rely on their own local software libraries to establish topological relationships, can use zone queries to pose questions that can be answered

by a list of zone IDs, or can retrieve zone data packets containing values from multiple sub-zones to perform their own analytics. A future extension could also introduce mandatory requirements making some of the recommended content such as summary statistics or additional useful information mandatory, which clients could then rely on for conforming implementations.

This example assumes a client does for some reason request information in a JSON representation about a particular ISEA3H DGGRS zone, say E6-317-A, based on its built-in support for that DGGRS (perhaps this client is an Executable Test Suite, since implementing this resource is still required to conform to the Core conformance class, including at minimum an `id` and links to the DGGRS and data retrieval, if applicable). The GET request URL and response for this zone information resource associated with the GEBCO collection would then look like the following:

\https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-317-A

```
{  
  "id" : "E6-317-A",  
  "links" : [  
    { "rel" : "[ogc-rel:geodata]", "href" : "/ogcapi/collections/gebco" },  
    { "rel" : "[ogc-rel:dggrs]", "title" : "ISEA3H DGGRS for GEBCO",  
      "href" : "/ogcapi/collections/gebco/dggs/ISEA3H"  
    },  
    { "rel" : "[ogc-rel:dggrs-definition]", "title" : "ISEA3H DGGRS  
definition",  
      "href" : "/ogcapi/dggrs/ISEA3H"  
    },  
    { "rel" : "[ogc-rel:dggrs-zone-data]", "title" : "ISEA3H Zone E6-317-A  
data for GEBCO",  
      "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-317-A/data"  
    },  
  
    { "rel" : "[ogc-rel:dggrs-zone-parent]", "title": "Parent zone D6-65-E",  
      "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/D6-65-E" },  
    { "rel" : "[ogc-rel:dggrs-zone-parent]", "title": "Parent zone D6-4A-F",  
      "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/D6-4A-F" },  
    { "rel" : "[ogc-rel:dggrs-zone-parent]", "title": "Parent zone D6-66-D  
(centroid child)",  
      "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/D6-66-D" },  
  
    { "rel" : "[ogc-rel:dggrs-zone-child]", "title": "Child zone E6-317-D  
(centroid child)",  
      "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-317-D" },  
    { "rel" : "[ogc-rel:dggrs-zone-child]", "title": "Child zone E6-317-E  
(vertex child)",  
      "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-317-E" },  
    { "rel" : "[ogc-rel:dggrs-zone-child]", "title": "Child zone E6-317-F  
(vertex child)",  
      "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-317-F" },  
    { "rel" : "[ogc-rel:dggrs-zone-child]", "title": "Child zone E6-316-E  
(vertex child)",  
      "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-316-E" },  
    { "rel" : "[ogc-rel:dggrs-zone-child]", "title": "Child zone E6-2C5-F  
(vertex child)",  
      "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-2C5-F" },  
    { "rel" : "[ogc-rel:dggrs-zone-child]", "title": "Child zone E6-2C5-E  
(vertex child)",  
      "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-2C5-E" },  
    { "rel" : "[ogc-rel:dggrs-zone-child]", "title": "Child zone E6-2C6-F  
(vertex child)",  
      "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-2C6-F" }  
  ]  
}
```

```

        {
          "rel" : "[ogc-rel:dggrs-zone-neighbor]", "title": "Neighboring zone E6-2C5-A (left)",
          "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-2C5-A" },
        {
          "rel" : "[ogc-rel:dggrs-zone-neighbor]", "title": "Neighboring zone E6-369-A (right)",
          "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-369-A" },
        {
          "rel" : "[ogc-rel:dggrs-zone-neighbor]", "title": "Neighboring zone E6-2C6-A (top-left)",
          "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-2C6-A" },
        {
          "rel" : "[ogc-rel:dggrs-zone-neighbor]", "title": "Neighboring zone E6-318-A (top-right)",
          "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-318-A" },
        {
          "rel" : "[ogc-rel:dggrs-zone-neighbor]", "title": "Neighboring zone E6-316-A (bottom-left)",
          "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-316-A" },
        {
          "rel" : "[ogc-rel:dggrs-zone-neighbor]", "title": "Neighboring zone E6-368-A (bottom-right)",
          "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-368-A" }
        ],
      "shapeType" : "hexagon",
      "areaMetersSquare" : 7765218521.3124237060547,
      "level": 8,
      "crs": "[OGC:CRS84]",
      "centroid": [ 34.8301691522937, 45.493496839407 ],
      "bbox": [ 34.1122890215028, 45.0307065899164, 35.5548602543606, 45.
9545736562164 ],
      "statistics" : {
        "Elevation" : {
          "minimum": -56.415161132812,
          "maximum": 499.67742919922,
          "average": 34.243211554643,
          "stdDev": 58.802922870029
        }
      },
      "geometry" : {
        "type" : "Polygon",
        "coordinates" : [
          [ [35.0325792437483, 45.9545736562164], [35.0675730384768, 45.
9355692576877],
          [35.1025417108968, 45.9165538942826], [35.1374852871664, 45.
8975275789752],
          [35.1724037934654, 45.8784903247104], [35.2072972559966, 45.
859442144404],
          [35.2421657009844, 45.8403830509428], [35.2770091546753, 45.
8213130571846],
          [35.3118276433383, 45.8022321759581], [35.3466211932639, 45.
7831404200626],
          [35.3813898307646, 45.7640378022689], [35.4161335821748, 45.
7449243353179],
          [35.4508524738509, 45.725800031922], [35.4855465321706, 45.
7066649047642],
          [35.5202157835335, 45.6875189664983], [35.5548602543606, 45.
6683622297491],
          [35.5409621444686, 45.6377043376462], [35.527097993237, 45.
6070377035762],
          [35.5132676379054, 45.576362404245], [35.4994709165275, 45.
5456785157748],
          [35.4857076679667, 45.5149861137079], [35.4719777318922, 45.
4842852730117],
          [35.4582809487756, 45.4535760680823], [35.4446171598862, 45.
4228585727495],

```

[35.4309862072879, 45.3921328602809], [35.4173879338344, 45.3613990033857],
[35.4038221831664, 45.3306570742198], [35.390288799707, 45.2999071443891],
[35.376787628658, 45.2691492849539], [35.3633185159963, 45.2383835664335],
[35.3498813084697, 45.2076100588098], [35.3020092883842, 45.1959626800988],
[35.2541667920171, 45.1842941045825], [35.2063537245807, 45.1726044068217],
[35.1585699921521, 45.1608936605836], [35.1108155016624, 45.1491619388553],
[35.0630901608873, 45.1374093138529], [35.0153938784362, 45.1256358570339],
[34.9677265637424, 45.1138416391074], [34.9200881270535, 45.1020267300452],
[34.8724784794212, 45.0901911990919], [34.8248975326921, 45.0783351147756],
[34.7773451994982, 45.0664585449175], [34.7298213932474, 45.0545615566424],
[34.6823260281141, 45.0426442163884], [34.6348590190306, 45.0307065899164],
[34.6001893100315, 45.0496582358553], [34.565495668238, 45.0685988693406],
[34.5307780684752, 45.0875284779916], [34.4960364855943, 45.1064470494011],
[34.4612708944723, 45.1253545711343], [34.4264812700118, 45.1442510307293],
[34.3916675871419, 45.163136415697], [34.3568298208172, 45.1820107135215],
[34.3219679460187, 45.2008739116594], [34.2870819377534, 45.2197259975403],
[34.2521717710543, 45.2385669585668], [34.2172374209808, 45.2573967821143],
[34.1822788626187, 45.2762154555309], [34.1472960710798, 45.2950229661379],
[34.1122890215028, 45.3138193012292], [34.1249230545728, 45.34474018811],
[34.1375863824436, 45.3756544566371], [34.1502791448219, 45.4065620478253],
[34.1630014821059, 45.4374629022505], [34.1757535353903, 45.4683569600458],
[34.1885354464684, 45.4992441608991], [34.2013473578361, 45.5301244440496],
[34.2141894126955, 45.5609977482844], [34.2270617549581, 45.5918640119347],
[34.2399645292484, 45.6227231728739], [34.2528978809076, 45.6535751685123],
[34.2658619559968, 45.6844199357953], [34.2788569013007, 45.7152574111995],
[34.2918828643314, 45.7460875307289], [34.3049399933312, 45.7769102299116],
[34.3532293127776, 45.7889076311011], [34.4015496229923, 45.8008834941054],
[34.4499010211884, 45.8128377421031], [34.4982836054328, 45.8247702974718],
[34.5466974746559, 45.8366810817765], [34.5951427286622, 45.8485700157587],
[34.64361946814, 45.8604370193242], [34.6921277946726, 45.8722820115316],
[34.7406678107482, 45.8841049105797], [34.7892396197712, 45.8959056337956],

```

[34.8378433260726, 45.9076840976223], [34.8864790349218, 45.
919440217605],
[34.9351468525365, 45.9311739083794], [34.9838468860951, 45.
9428850836581],
[35.0325792437483, 45.9545736562164] ]
}
}
}

```

An HTML representation of that zone information resource, allowing a user to browse through parent, children and neighboring zones, could look like the following:

Zone *E6-317-A*

for ISEA3H DGGS (General Bathymetric Chart of the Oceans)

Hierarchy Level: 8

Associated ISEA9R Zone: [E6-317](#)

Zone Surface Area: 7765.22 kilometers square (**0.11% less** than reference mean zone for level 8)

Centroid: { lat: 45.493496839407, lon: 34.8301691522937 }

Extent: { { lat: 45.0307065899164, lon: 34.1122890215028 }, { lat: 45.9545736562164, lon: 35.5548602543606 } }

[Download data](#)

Parent: [D6-65-E](#)



Parent: [D6-4A-F](#)



Parent: [D6-66-D](#)

(centroid child)



Child	Zone ID	Map
Centroid	E6-317-D	
Vertex	E6-317-E	
Vertex	E6-317-F	
Vertex	E6-316-E	
Vertex	E6-2C5-F	
Vertex	E6-2C5-E	
Vertex	E6-2C6-F	

Neighbor	Zone ID	Map
Left	E6-2C5-A	
Right	E6-369-A	
Top-Left	E6-2C6-A	
Top-Right	E6-318-A	
Bottom-Left	E6-316-A	
Bottom-Right	E6-368-A	

Figure C.1 – Sample HTML response of an ISEA3H DGGRS zone information resource for GEBCO data

C.5. Simple zone queries

Having selected a particular DGGRS, one use case for the API is for a client to perform spatial queries by requesting a list of zones. The simplest case is to simply request a list of all available zones for a discrete global grid of a given refinement level (zone-level), which in the case of a particular collection implies zones for which the collection has data.

C.5.1. Querying a particular refinement level

If no zone-level parameter is provided, the server is free to pick a default zone level for such queries, which would result in a reasonable number of zones returned, based on the area and resolution of the data collection. In this example, the client will explicitly request a list of zones at refinement level 1.

By default, if compact-zones=false is not specified by the client, zone queries are compacted, meaning that if all sub-zones of a parent zone would be included in the response, they get replaced by that parent zone so that fewer zones need to be enumerated. This can result in zones of a lower refinement level being returned than the requested zone-level, and therefore a mix of zones of different levels. A full globe response would therefore always result in the list of all zones of refinement level 0. In these examples, the client will explicitly set compact-zones=false.

Following the [ogc-rel:dggrs-zone-query] link, specifying these parameters, and negotiating a JSON zone list response using an Accept: application/json header, the client requests from the end-point:

```
\https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H/zones?zone-level=1&compact-zones=false
```

An example JSON response for this Zone Query JSON, as defined by the Zone Query requirement class, is presented below:

```
{
  "zones" : [
    "A0-0-D", "A0-0-E", "A0-0-F", "A0-0-G", "A1-0-D", "A1-0-E",
    "A1-0-F", "A2-0-D", "A2-0-E", "A2-0-F", "A3-0-D", "A3-0-E",
    "A3-0-F", "A4-0-D", "A4-0-E", "A4-0-F", "A5-0-D", "A5-0-E",
    "A5-0-F", "A6-0-D", "A6-0-E", "A6-0-F", "A7-0-D", "A7-0-E",
    "A7-0-F", "A8-0-D", "A8-0-E", "A8-0-F", "A9-0-D", "A9-0-E",
    "A9-0-F", "A9-0-H"
  ],
  "returnedAreaMetersSquare" : 4080524973792710,
  "links" : [
    { "rel" : "[ogc-rel:dggrs]", "title" : "ISEA3H DGGS for GEBCO",
      "href" : "/ogcapi/collections/gebco/dggs/ISEA3H"
    },
    { "rel" : "[ogc-rel:dggrs-definition]", "title" : "ISEA3H DGGRS
definition",
      "href" : "/ogcapi/dggrs/ISEA3H"
    },
    { "rel" : "[ogc-rel:geodata]", "href" : "/ogcapi/collections/gebco" }
  ],
}
```

```
"linkTemplates" : [
  { "rel" : "[ogc-rel:dggrs-zone-data]", "title" : "ISEA3H data for GEBCO",
    "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/{zoneId}/data",
  }
]
```

The server returned all 32 zones of ISEA3H level 1 (which corresponds to a truncated icosahedron – the traditional soccer ball pattern), because the collection has global coverage.

Servers may support additional representations of zone lists beyond JSON, such as a more compact binary 64-bit integers encoding, a GeoJSON representation including zone geometry or an HTML representation for exploring datasets in a browser through the DGGS API. Note that regardless of encoding, zone query responses can also be compressed for additional bandwidth efficiency if the client and server negotiate a content encoding e.g., using Content-Encoding: gzip.

CAUTION

The transfer of zone geometry in zone queries is not an efficient use of the DGGS API for practical analytics purposes, since clients should already have built-in knowledge of the selected DGGRS in their local software libraries and can efficiently infer the geometry of a zone from its zone ID alone as needed. In many use cases, the geometry may not be needed at all.

The ability to retrieve zone lists in a geospatial data format such as GeoJSON or GeoTIFF which can readily be visualized in GIS tools is useful for **educational, demonstration and debugging** purposes. For example, this functionality can be used to generate grids, such as those seen in annex B (a crs parameter to request an ISEA planar projection was used in that case to override the default GeoJSON CRS84), as well as the zone visualizations seen in the examples in this section.

A visualization of the response from negotiating application/geo+json for the above request is seen below.

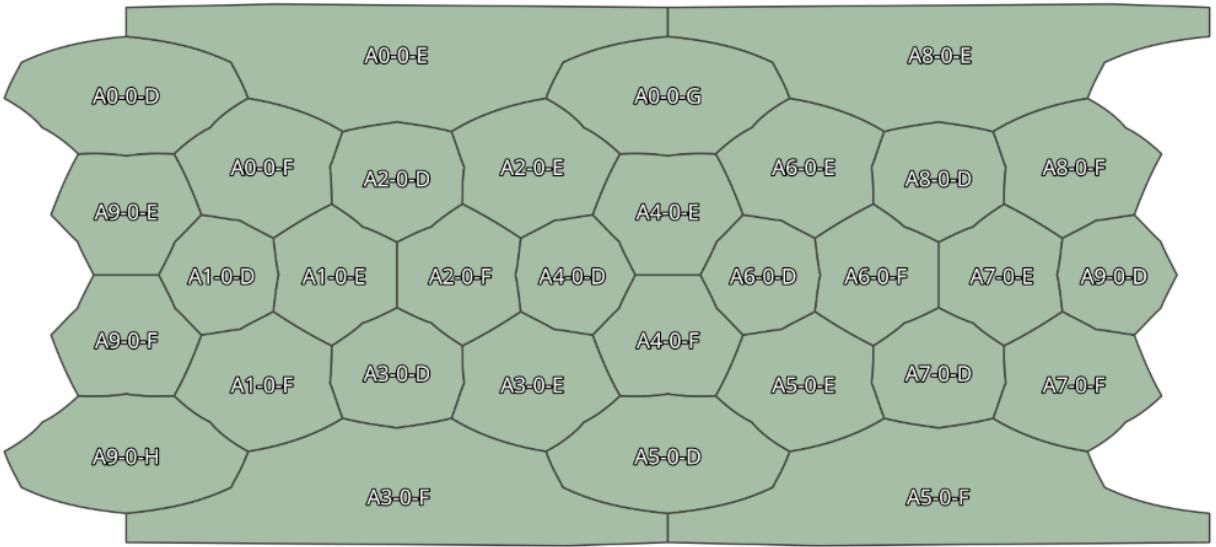


Figure C.2 – GeoJSON response for zone query for ISEA3H DGGRS at level 1 visualized in QGIS

C.5.2. Querying for a spatio-temporal subset

To request a list of zones for an area of interest, the `bbox` or `subset` query parameter can be used. For temporal datasets, the `datetime` can be used as well as `subset=time(...)` to constrain the query to a time of interest. For restricting the same zone query to a bounding box area from (40°N, 30°E) to (60°N, 50°E), the client could make either of the following requests:

```
\u2022 https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H/zones?bbox=30,40,50,60&zone-level=1&compact-zones=false
```

```
\u2022 https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H/zones?subset=Lat\(40:60\),Lon\(30:50\)&zone-level=1&compact-zones=false
```

and obtain the following response (negotiating a JSON zone list again):

```
{
  "zones" : [ "A0-0-G", "A6-0-E" ],
  "returnedAreaMetersSquare" : 31137344609299.56640625,
  "links" : [
    { "rel" : "[ogc-rel:dggrs]", "title" : "ISEA3H DGGS for GEBCO",
      "href" : "/ogcapi/collections/gebco/dggs/ISEA3H"
    },
    { "rel" : "[ogc-rel:dggrs-definition]", "title" : "ISEA3H DGGRS
definition",
      "href" : "/ogcapi/dggrs/ISEA3H"
    },
    { "rel" : "[ogc-rel:geodata]", "href" : "/ogcapi/collections/gebco" }
  ],
  "linkTemplates" : [
    { "rel" : "[ogc-rel:dggrs-zone-data]", "title" : "ISEA3H data for GEBCO",
      "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/{zoneId}/data",
    }
  ]
}
```

This time, the server returned only two zones of ISEA3H level 1.

A visualization of the response from negotiating application/geo+json for the above request, together with the bounding box used, is seen below.

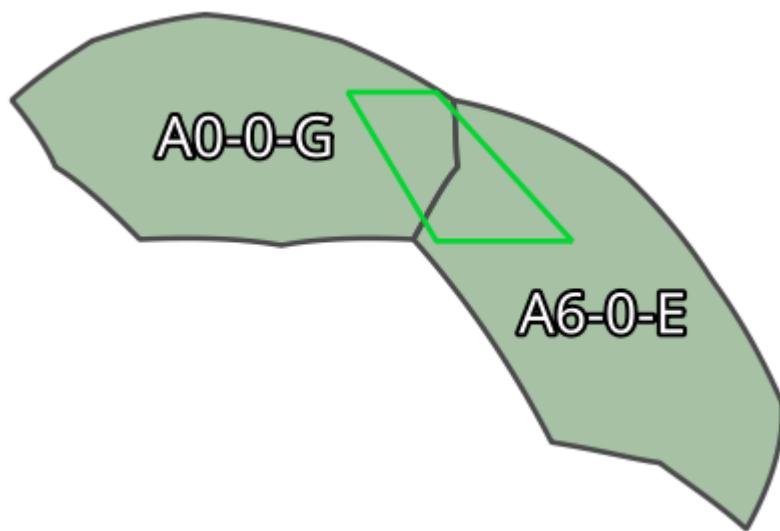


Figure C.3 – GeoJSON response for zone query for ISEA3H DGGRS at level 1 for a (40°N, 30°E) to (60°N, 50°E) bounding box visualized in QGIS

As seen below, an HTML representation could include rendered previews of the data for the zone area, summary information, as well as links to retrieve the data and access the zone information resource for each zone. Note that the significant variance in area (-11.31% and +6.66% from the reference mean zone (whose area is calculated by dividing the Earth surface area by the total number of zones) is due to the fact that A0-0-G is a pentagon (in the ISEA projection – on the Earth surface it looks more like a decagon) which occupies 5/6th the area of hexagons such as A6-0-E (which on the Earth surface is one of the hexagons appearing as a nonagon), as well as the fact that at level 1, the number of pentagons (12) is significant compared to the total number of zones (32).

Zone ID	Map	Data	Zone Geometry	Area	Variance	Min Lat	Min Lon	Max Lat	Max Lon
Level 1 (variance relative to 15,939,550.68 km ² reference mean zone surface area)									
A0-0-G		data	geojson.io download	14,136,336.1 km ²	-11.31 %	39.532480763194	-29.2549078771889164	79.260655977152	51.7549078778955
A6-0-E		data	geojson.io download	17,001,008.51 km ²	+6.66 %	12.1872690128254	27.142896755173	58.6687287802954	83.1792006259808

Total surface area: **31,137,344.61 km²**

(~15,939,550.68 km² level 1 zone precision)

Figure C.4 – Sample HTML response for a zone query for ISEA3H DGGRS at level 1 for a (40°N, 30°E) to (60°N, 50°E) bounding box

Of course with a higher zone level, many more zones can be returned:

[\\https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H/zones?
bbox=30,40,50,60&zone-level=8&compact-zones=false](https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H/zones?bbox=30,40,50,60&zone-level=8&compact-zones=false)

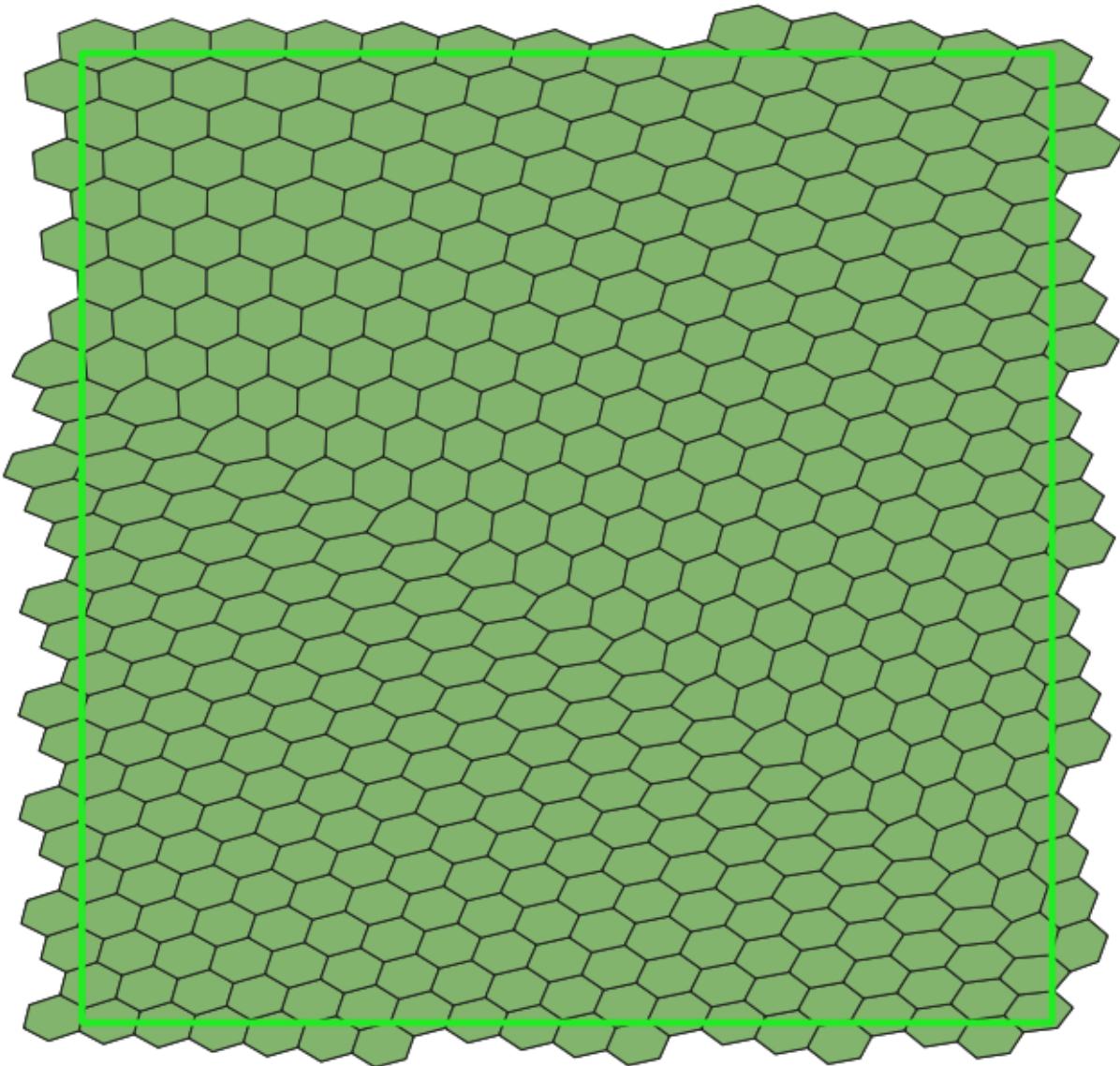


Figure C.5 – Sample HTML response for a zone query for ISEA3H DGGRS at level 8 for a (40°N, 30°E) to (60°N,50°E) bounding box

C.5.3. Zones representing data availability

The primary purpose of the Zone Query requirement class is to ask questions of the type “Where is it?”. The answer is a region of space-time represented by a list of zones. This region might correspond to where data is available for a particular collection, may be the result of a processing workflow, or the result of evaluating a CQL2 query expression as demonstrated in later examples. This result could otherwise have been communicated as a multi-polygon vector mask or as a rasterized 1-bit image or opacity channel.

The following example illustrates the case of a collection (elevation data from [Viewfinder Panoramas](#)) which does not have data for everywhere on the globe (only over land), using a higher zone level of 6 for the query.

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=6&compact-zones=false



Figure C.6 – GeoJSON response for zone query for ISEA3H DGGRS at level 6 for a [Viewfinder Panoramas](#) land elevation collection, visualized in QGIS

Using a bounding box to restrict the query to an area of interest, a higher zone level of 10 could be used:

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=10&bbox=30,40,50,50&compact-zones=false

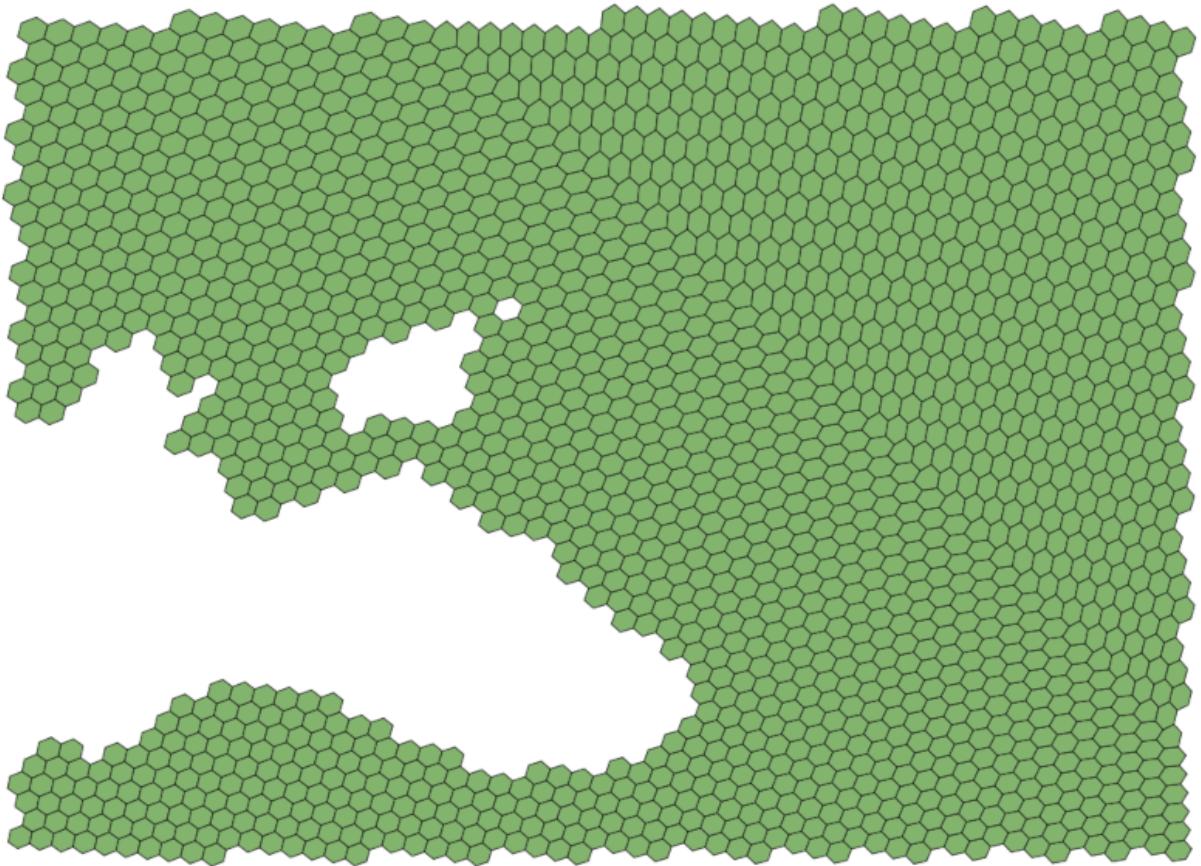


Figure C.7 – GeoJSON response for zone query for ISEA3H DGGRS at level 10, using a (40°N, 30°E) to (40°N, 50°E) bounding box, for a [Viewfinder Panoramas](#) land elevation collection, visualized in QGIS

The DGGS API works independently of the nature of the data, whether dealing with gridded coverage or vector features, allowing to easily perform complex analytics involving heterogeneous data sources. The previous examples all used gridded coverage data sources. The following example illustrates the same capabilities using the Natural Earth states and provinces collection:

```
\https://maps.gnosis.earth/ogcapi/collections/  
NaturalEarth:cultural:ne_10m_admin_1_states_provinces/dggs/ISEA3H/zones?zone-  
level=6&compact-zones=false
```

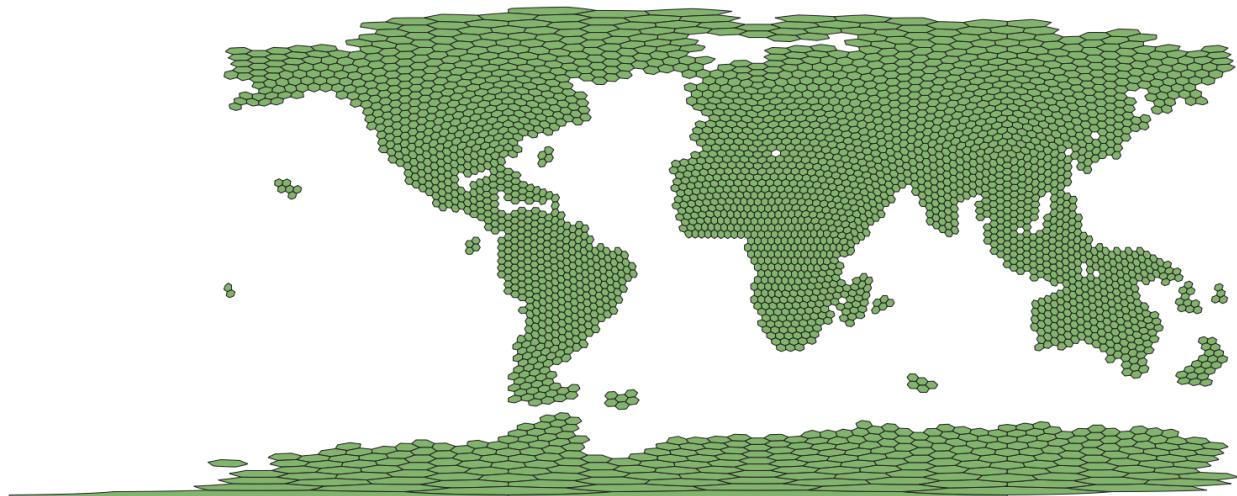


Figure C.8 – GeoJSON response for zone query for ISEA3H DGGRS at level 6, for a [Natural Earth states and provinces collection](#), visualized in QGIS

The similarity between the response for this collection of states and provinces features compared to the same request earlier for a gridded elevation coverages showcases the potential for DGGS zone queries to facilitate data integration.

C.5.4. Querying sub-zones of a parent zone

The Zone Query requirement class defines the parent-zone query parameter as one more way how to restrict the query spatially (and temporally in the case of a temporal DGGRS), which is particularly useful for hierarchical exploration. For example, to restrict the query to sub-zones of A6-0-E rather than using a bounding box, the client could perform the following request (at level 8):

```
\https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=8&parent-zone=A6-0-E&compact-zones=false
```

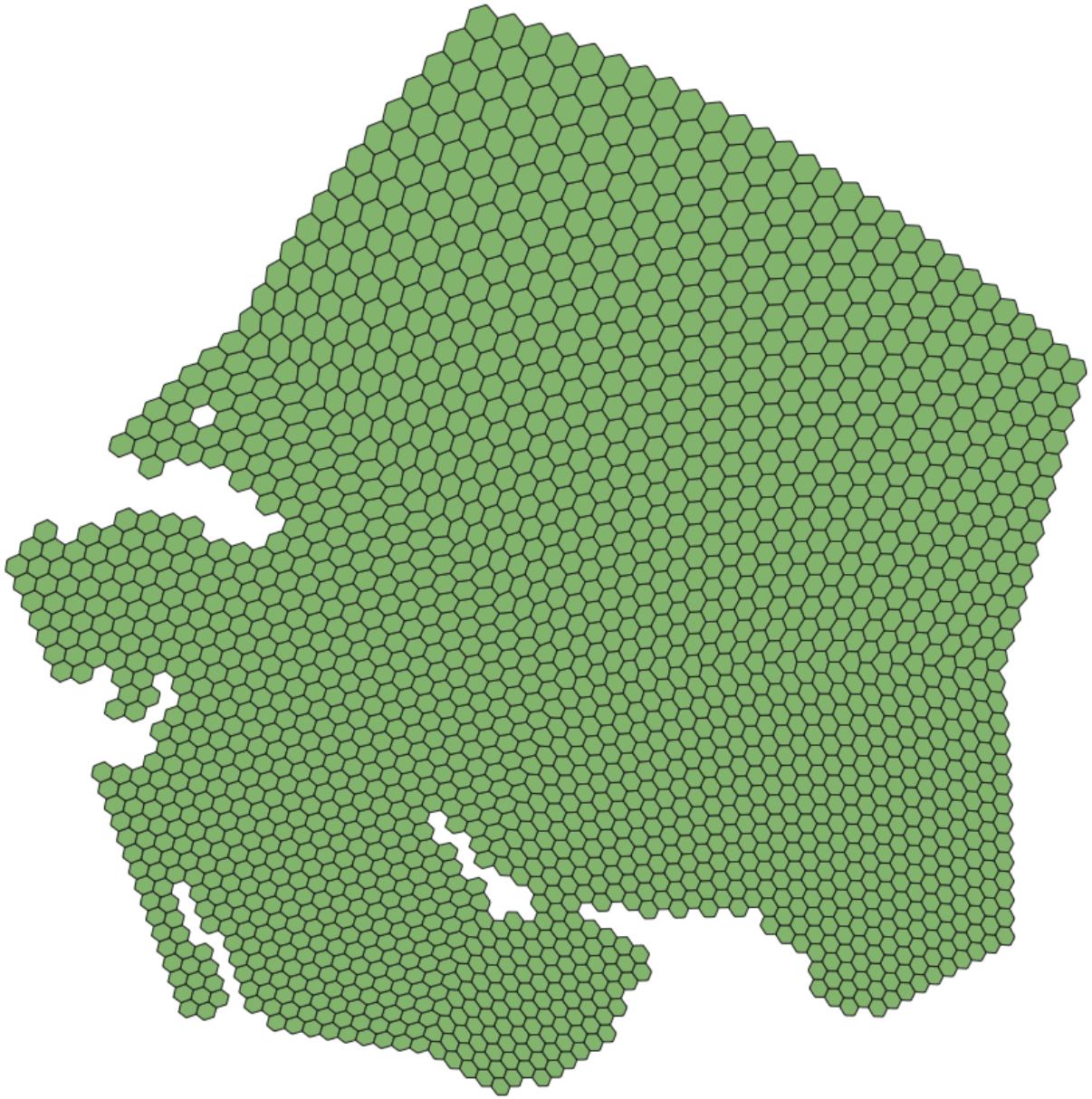


Figure C.9 – GeoJSON response for zone query for ISEA3H DGGRS at level 8, for parent zone [A6-0-E](#), for a [Viewfinder Panoramas](#) land elevation collection, visualized in QGIS

The client could then pick one of the returned zones, say E6-317-A, for deeper exploration at level 16:

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=16&parent-zone=E6-317-A&compact-zones=false

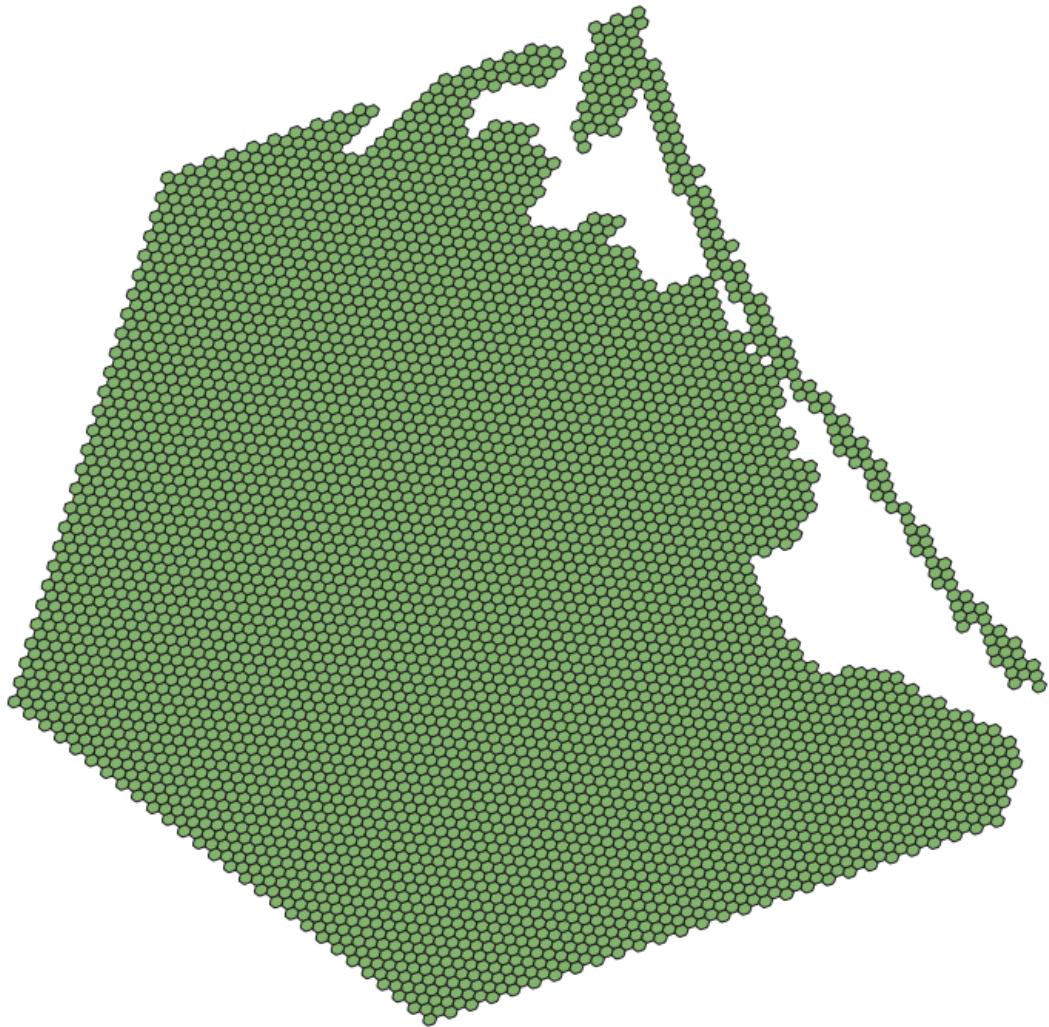


Figure C.10 – GeoJSON response for zone query for ISEA3H DGGRS at level 16, for parent zone [E6-317-A](#), for a [Viewfinder Panoramas](#) land elevation collection, visualized in QGIS

Going back to A6-0-E (a level 1 zone), this could also be used to retrieve the immediate children by requesting the next level (2), to step down the hierarchy a single refinement level at a time:

```
\https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=2&parent-zone=A6-0-E&compact-zones=false
```

The 7 children zones (1 centroid child and 6 vertex children) are returned, since they all contain data.

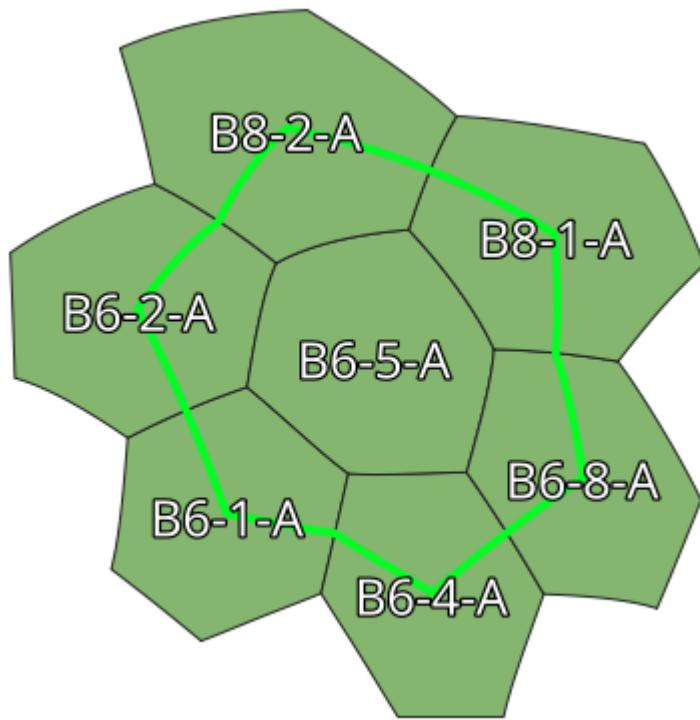


Figure C.11 – GeoJSON response for zone query for ISEA3H DGGRS at level 2, for parent zone A6-0-E (outline shown), for a Viewfinder Panoramas land elevation collection, visualized in QGIS

C.5.5. ISEA9R (rhombus) Zone Queries

Although all examples so far used the ISEA3H DGGRS, this DGGS API is completely agnostic of the DGGRS. Future planned parts for OGC Abstract Topic 21 will standardize new categories of DGGS, including volumetric (Part 2), temporal (Part 3) and axis-aligned (Part 4). The DGGS API functionality, as demonstrated in this examples section, is ready to handle all any DGGRS, including DGGRS based on these new DGGS types.

The following examples demonstrates equivalent zone queries using the ISEA9R DGGS, a dual DGGS of ISEA3H even refinement levels, which can be considered axis-aligned in a CRS derived from rotating and shearing the ISEA planar projection. In the ISEA planar projection, the ISEA9R zones are shaped as rhombuses.

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA9R/zones?zone-level=3&compact-zones=false

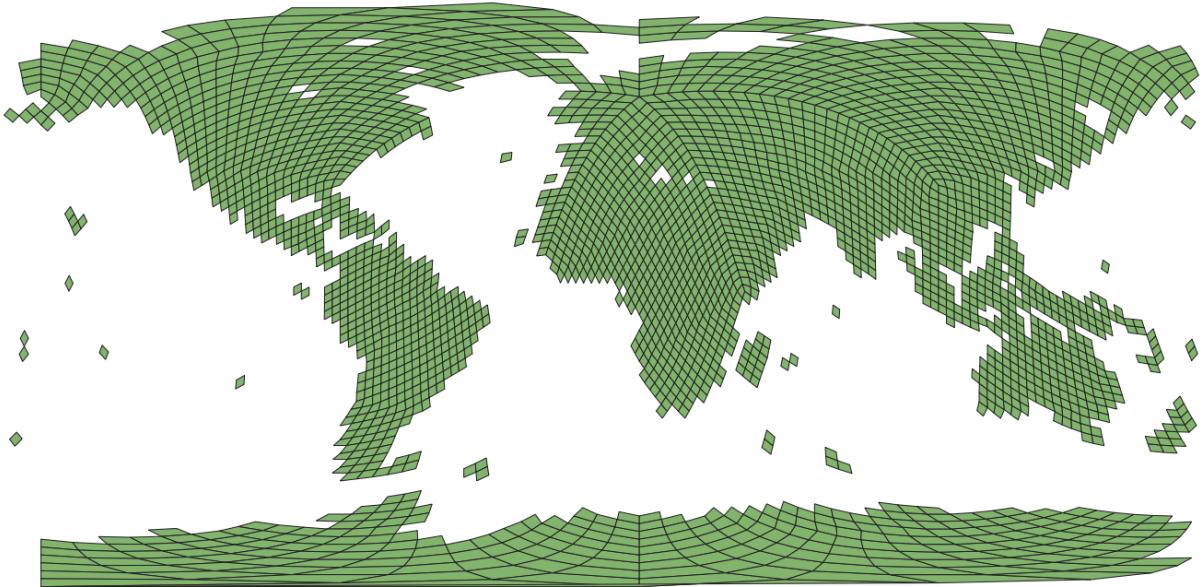


Figure C.12 – GeoJSON response for zone query for ISEA9R DGGRS at level 3 for a [Viewfinder Panoramas](#) land elevation collection visualized in QGIS

[https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA9R/zones?
bbox=30,40,50,60&zone-level=4&compact-zones=false](https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA9R/zones?bbox=30,40,50,60&zone-level=4&compact-zones=false)

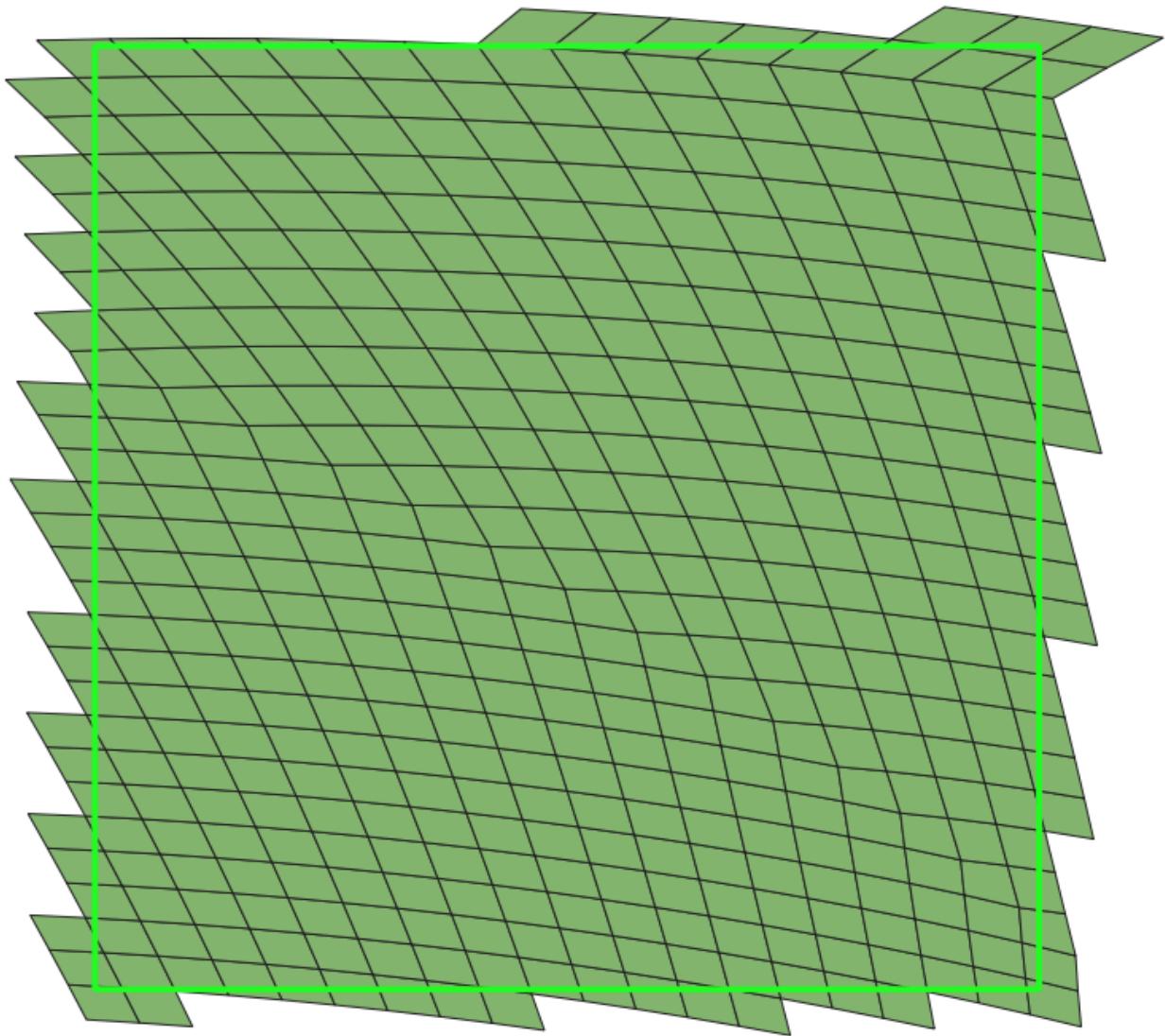


Figure C.13 – GeoJSON response for zone query for ISEA9R DGGRS at level 4 for a (40°N, 30°E) to (60°N, 50°E) bounding box visualized in QGIS

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA9R/zones?zone-level=4&parent-zone=B6-2&compact-zones=false

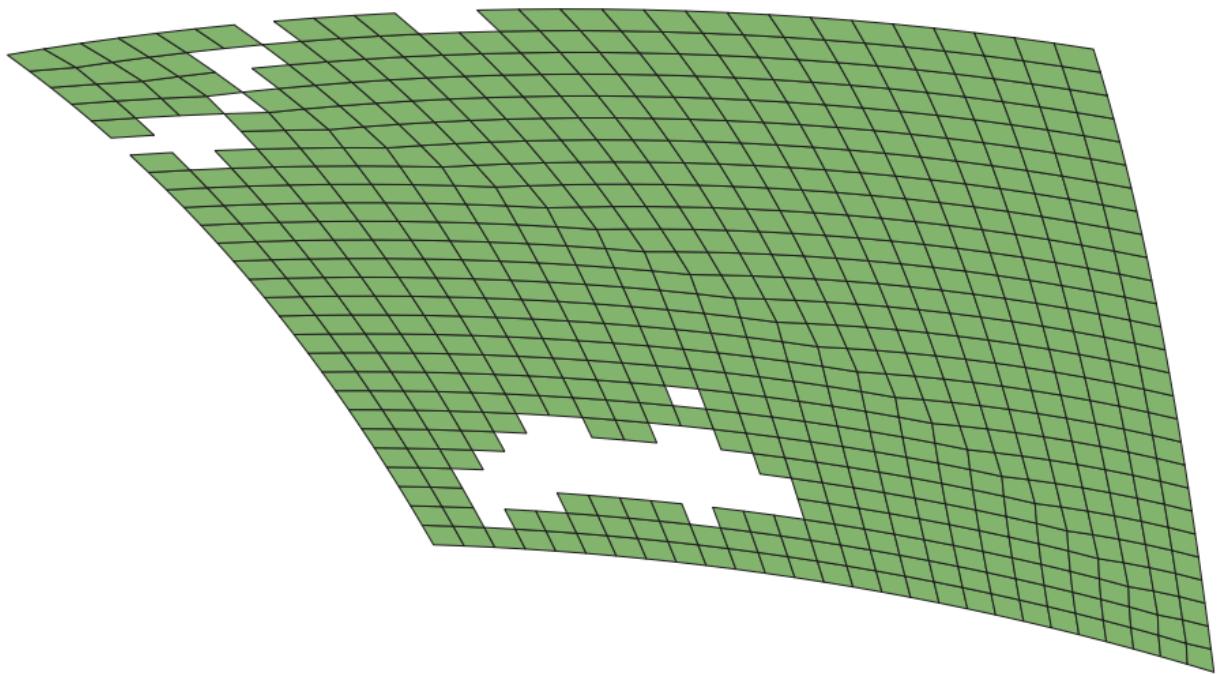


Figure C.14 – GeoJSON response for zone query for ISEA9R DGGRS at level 4 for parent zone B6-2, for a Viewfinder Panoramas, visualized in QGIS

[\\https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA9R/zones?zone-level=8&parent-zone=E6-317&compact-zones=false](https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA9R/zones?zone-level=8&parent-zone=E6-317&compact-zones=false)

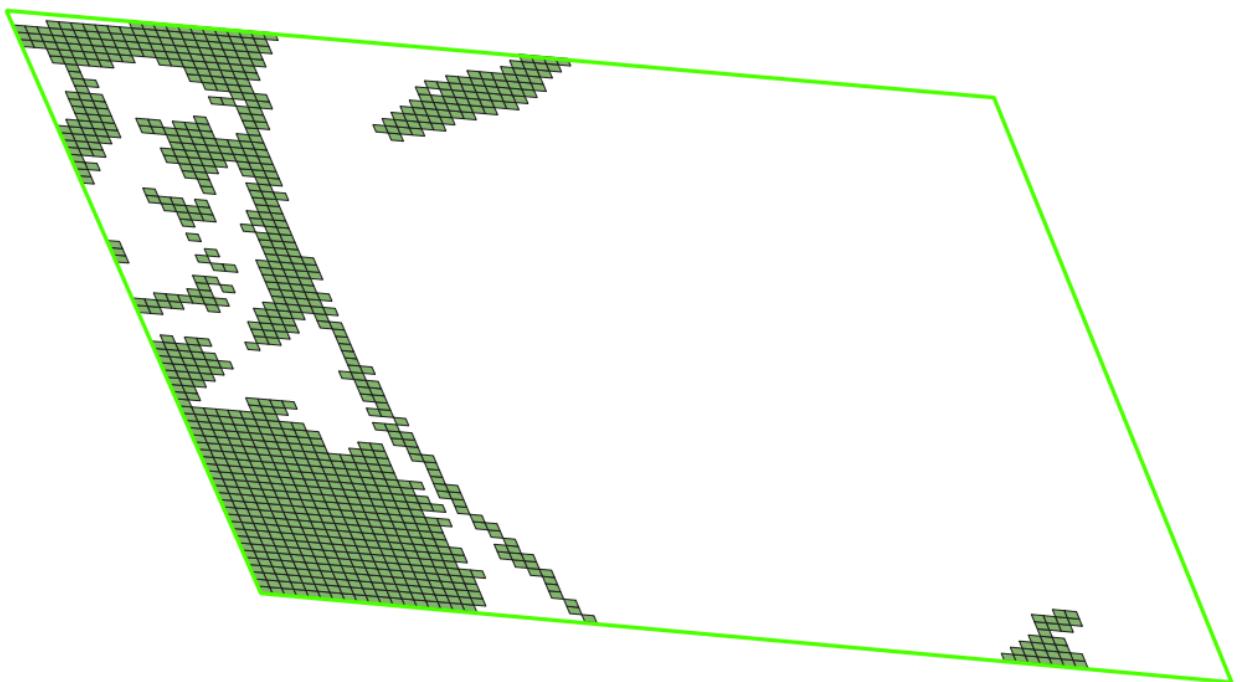


Figure C.15 – GeoJSON response for zone query for ISEA9R DGGRS at level 8 for parent zone E6-317, for a Viewfinder Panoramas, visualized in QGIS

Although ISEA9R is axis-aligned, like ISEA3H, it is equal area, with an error budget below 1%.

C.5.6. GNOSIS Global Grid (WGS84 rectangles) Zone Queries

The next zone query examples use the GNOSIS Global Grid DGGRS. The GNOSIS Global Grid is also axis-aligned, with the axes being the geographic latitude and longitude. On a plate carée projection, the zones are shaped as rectangles, which conveniently correspond to typical EPSG:4326 / CRS84 bounding boxes.

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=5&compact-zones=false

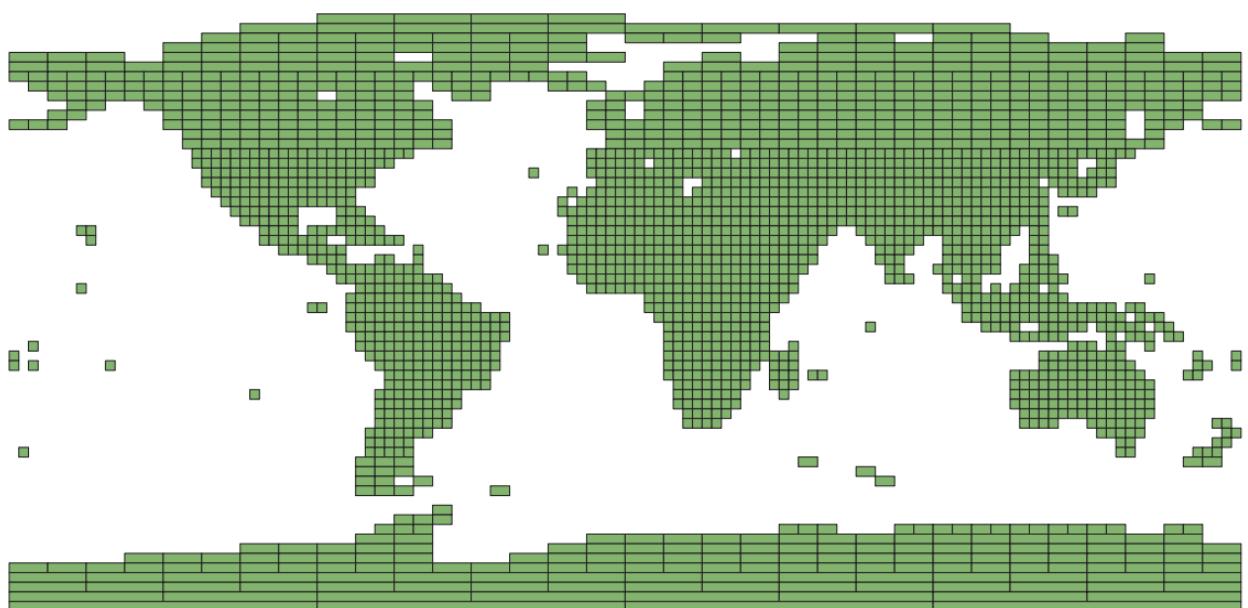


Figure C.16 – GeoJSON response for zone query for GNOSIS Global Grid DGGRS at level 5 for a [Viewfinder Panoramas](#) land elevation collection visualized in QGIS

<https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/GNOSISGlobalGrid/zones?bbox=30,40,50,60&zone-level=7&compact-zones=false>

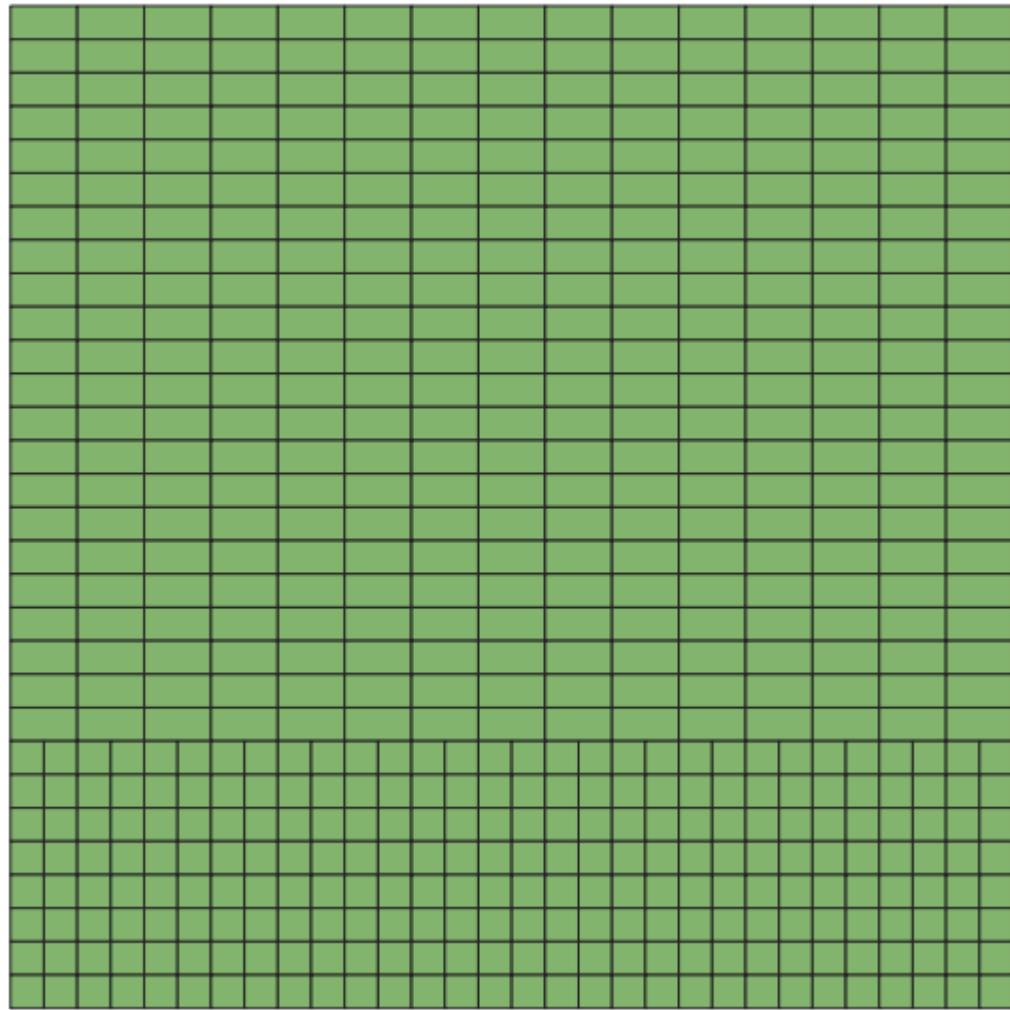


Figure C.17 – GeoJSON response for zone query for GNOSIS Global Grid DGGRS at level 7 for a (40°N, 30°E) to (60°N, 50°E) bounding box visualized in QGIS

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/GNOSISGlobalGrid/zones?zone-level=8&parent-zone=2-2-9&compact-zones=false



Figure C.18 – GeoJSON response for zone query for GNOSIS Global Grid DGGRS at level 8 for parent zone 2-2-9, for a [Viewfinder Panoramas](#), visualized in QGIS

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/GNOSISGlobalGrid/zones?zone-level=13&parent-zone=6-1F-98&compact-zones=false



Figure C.19 – GeoJSON response for zone query for GNOSIS Global Grid DGGRS at level 13 for parent zone [6-1F-98](#), for a [Viewfinder Panoramas](#), visualized in QGIS

C.6. Compact zone queries

The following examples are reprises of some of the earlier ones leaving compact zones enabled, showcasing the fewer zones being returned to communicate an identical spatial area.

C.6.1. ISEA3H compact zone queries

Compacting ISEA3H zones is complicated by the fact that sub-zones are not fully congruent. The approach demonstrated here skips a refinement level, taking advantage of the fact that 7 of the grand-children of a zone are congruent. Overlaps are seen, as 6 holes are left at the vertices of the grand-parent, each occupying 1/3rd the size of a congruent grand-child. Note how $7 + 6 \times 1/3 = 9$ (3×3), which corresponds to the two subsequent refinements at the refinement ratio of 3. If the neighboring grand-parent of the same refinement level cannot be used, then the non-congruent vertex children must be included, resulting in these overlaps.

[\\https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H/zones?
bbox=30,40,50,60&zone-level=8](https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H/zones?bbox=30,40,50,60&zone-level=8)

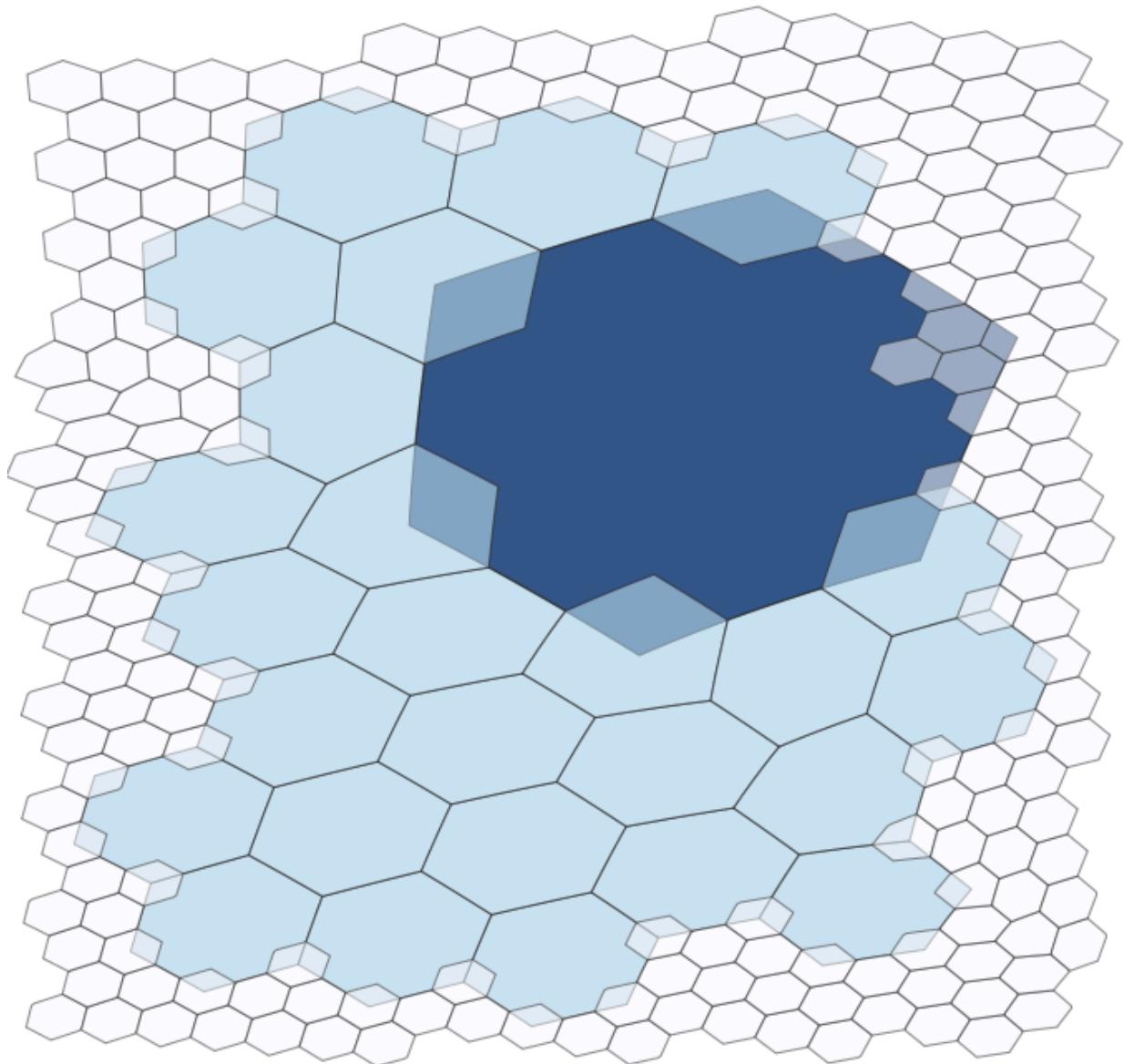


Figure C.20 – GeoJSON compact zones response querying ISEA3H DGGRS at level 8 for a (40°N, 30°E) to (60°N, 50°E) bounding box

[\\https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/
zones?zone-level=6](https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=6)

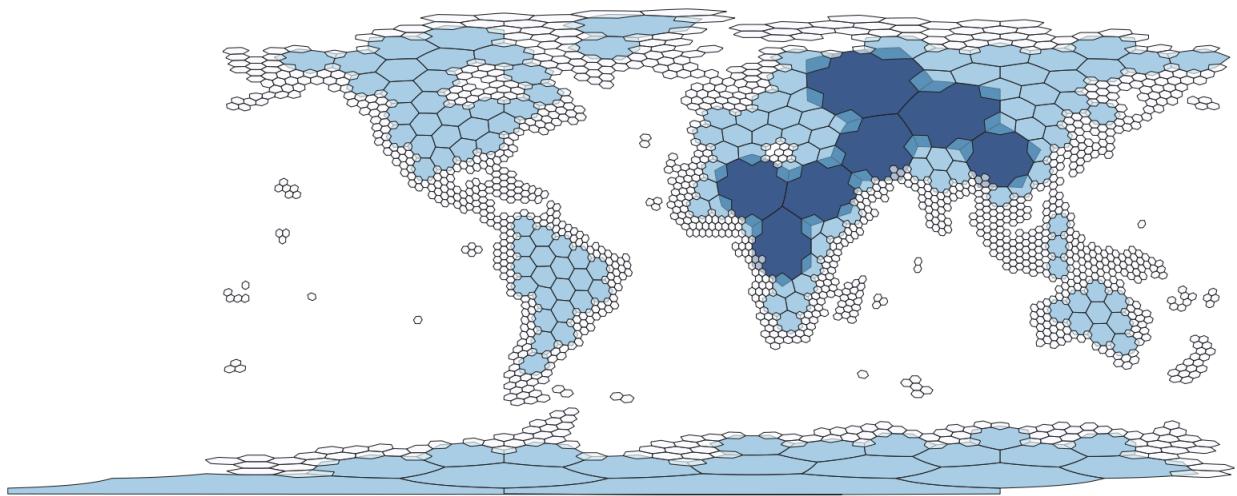


Figure C.21 – GeoJSON compact zones response for querying ISEA3H DGGRS at level 6 for a [Viewfinder Panoramas](#) land elevation collection, visualized in QGIS

\https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=10&bbox=30,40,50,50

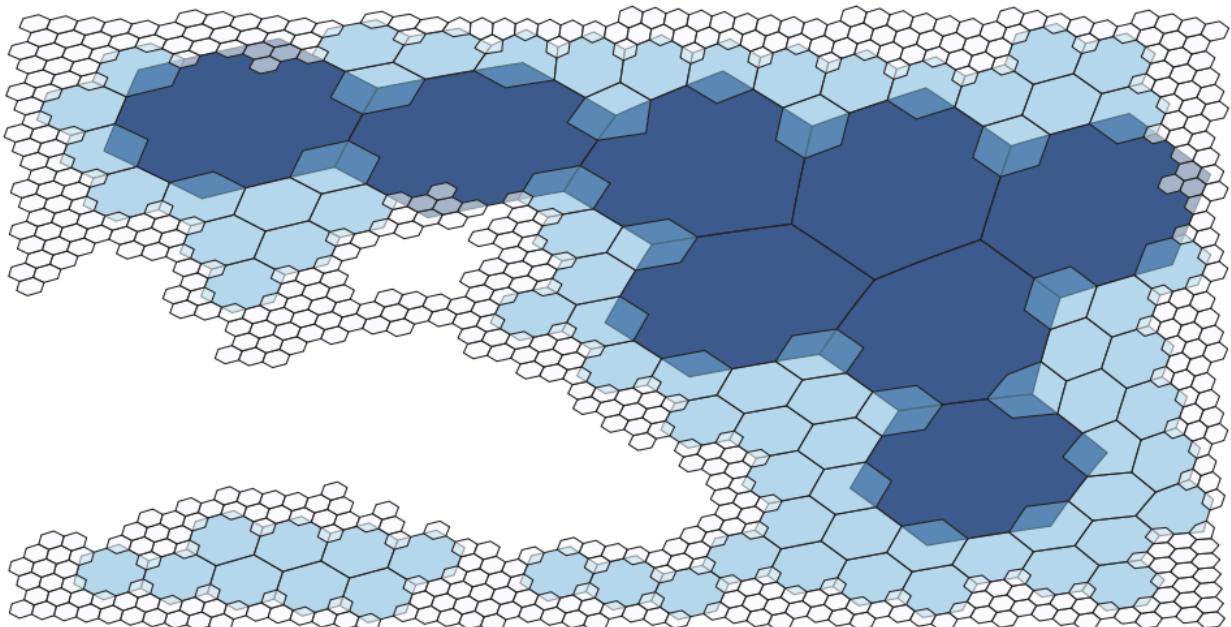


Figure C.22 – GeoJSON compact zones response for querying ISEA3H DGGRS at level 10, using a (40°N, 30°E) to (40°N, 50°E) bounding box, for a [Viewfinder Panoramas](#) land elevation collection, visualized in QGIS

\https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=16&parent-zone=E6-317-A

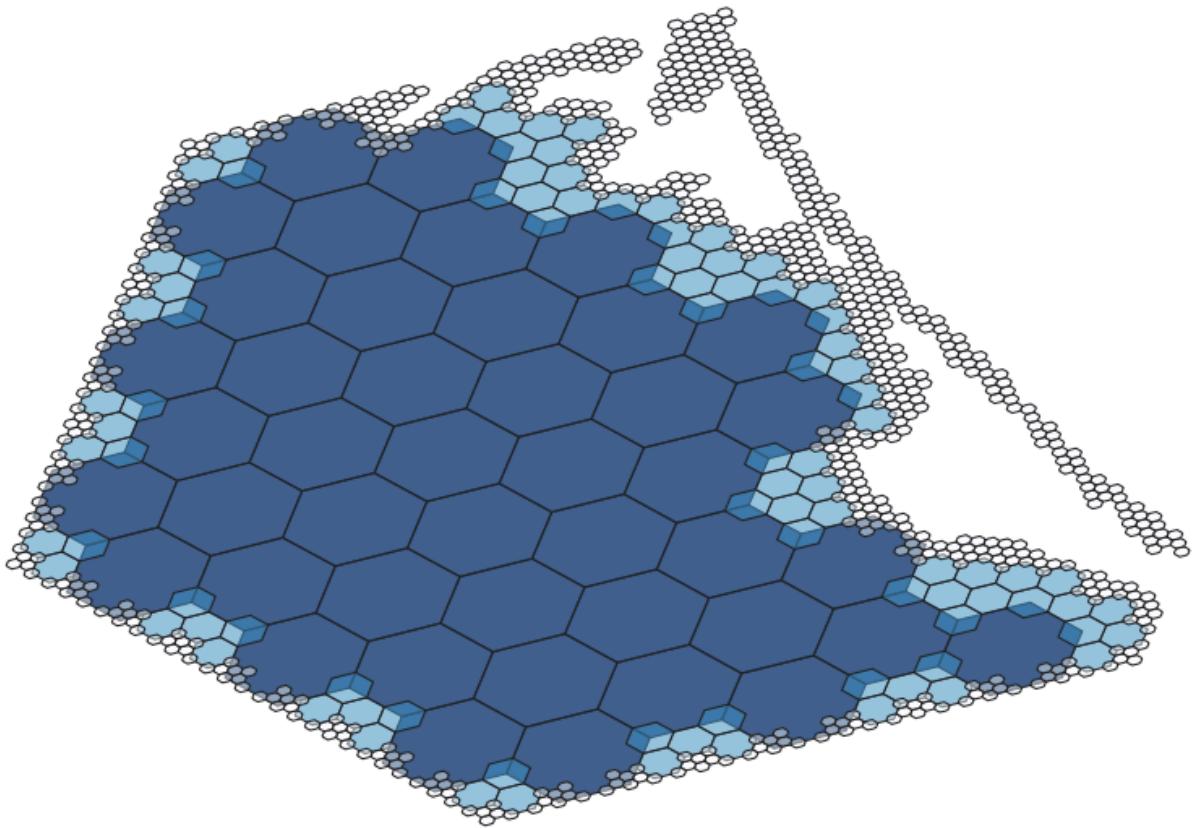


Figure C.23 – GeoJSON compact zones response for querying ISEA3H DGGRS at level 16, for parent zone E6-317-A, for a Viewfinder Panoramas land elevation collection, visualized in QGIS

C.6.2. ISEA9R compact zone queries

Compacting ISEA9R zones can be achieved quite simply given that all sub-zones are fully congruent.

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA9R/zones?zone-level=3

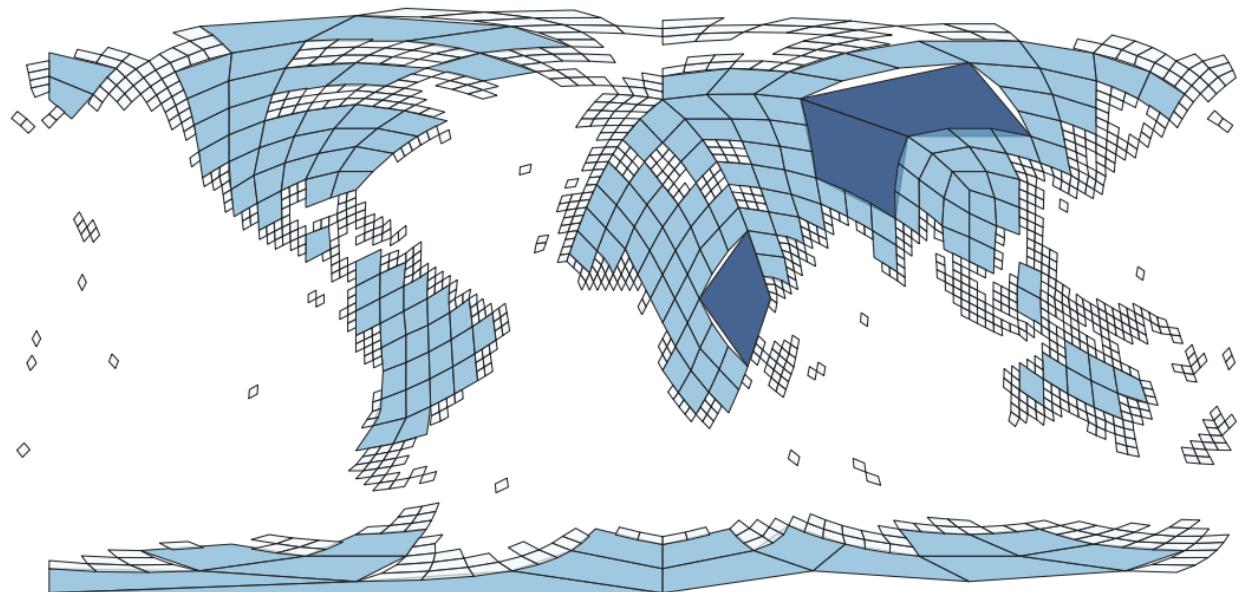


Figure C.24 – GeoJSON compact zones response for querying ISEA9R DGGRS at level 3 for a [Viewfinder Panoramas](#) land elevation collection visualized in QGIS

[https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA9R/zones?
bbox=30,40,50,60&zone-level=4](https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA9R/zones?bbox=30,40,50,60&zone-level=4)

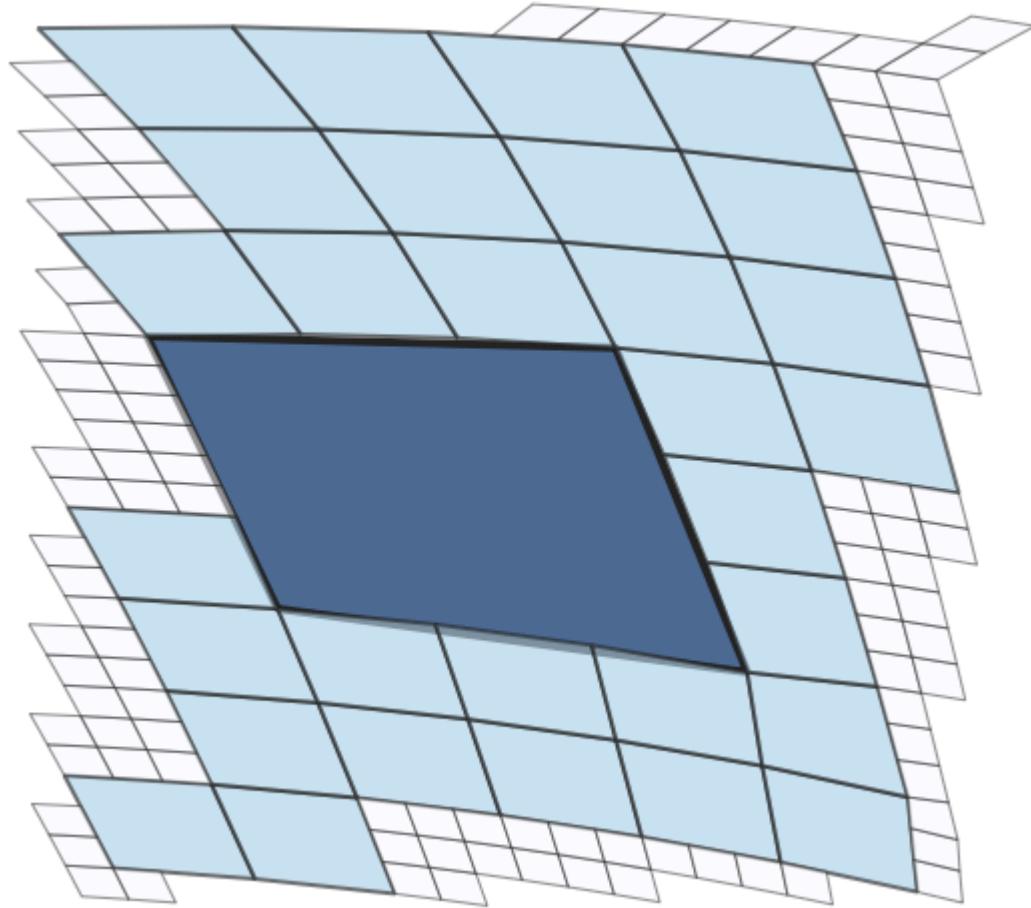


Figure C.25 – GeoJSON compact zones response for querying ISEA9R DGGRS at level 4 for a (40°N, 30°E) to (60°N, 50°E) bounding box visualized in QGIS

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA9R/zones?zone-level=4&parent-zone=B6-2

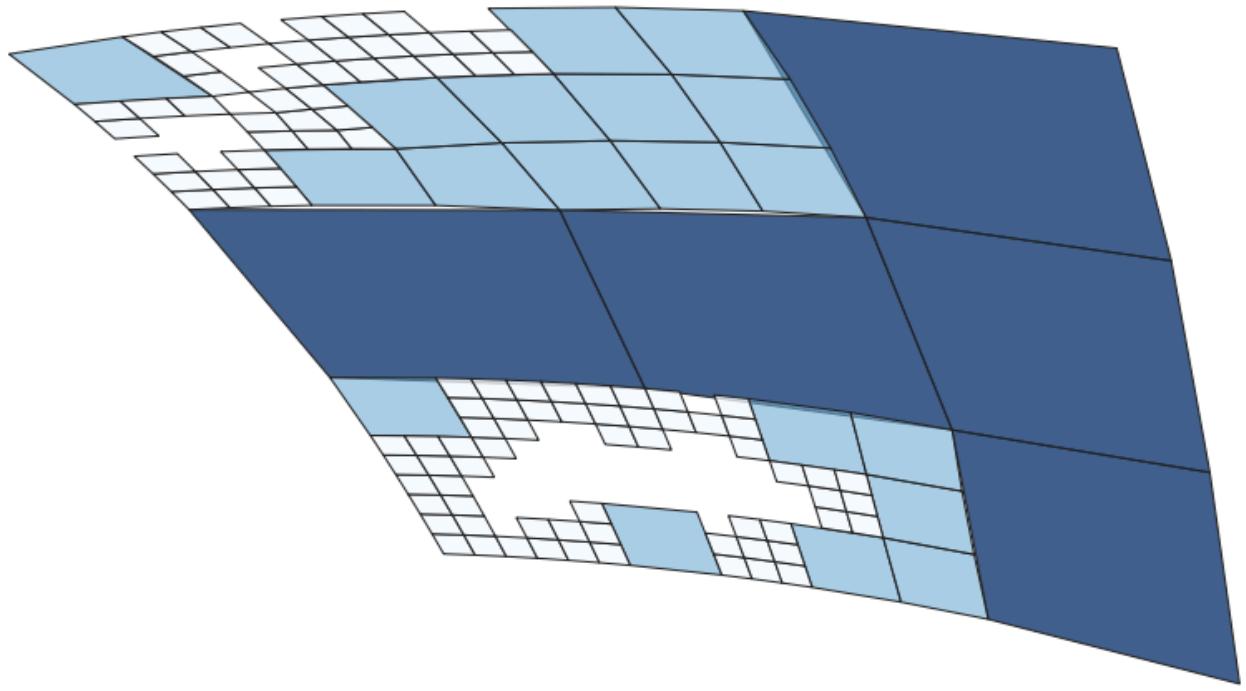


Figure C.26 – GeoJSON compact zones response for querying ISEA9R DGGRS at level 4 for parent zone B6-2, for a Viewfinder Panoramas, visualized in QGIS

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA9R/zones?zone-level=8&parent-zone=E6-317

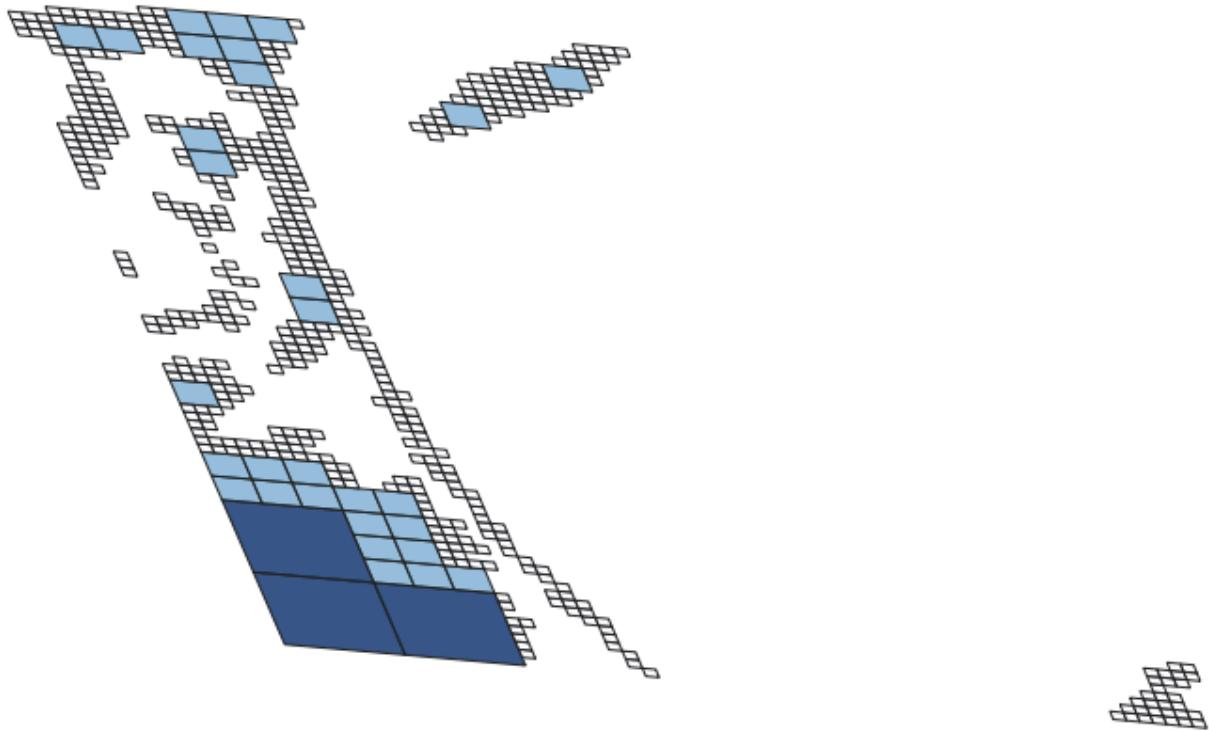


Figure C.27 – GeoJSON compact zones response for querying ISEA9R DGGRS at level 8 for parent zone [E6-317](#), for a [Viewfinder Panoramas](#), visualized in QGIS

C.6.3. GNOSIS Global Grid compact zone queries

Compacting GNOSIS Global Grid zones is also simple to achieve since sub-zones are also fully congruent.

To illustrate how the width of the zones attempts to remain equal to the height, the following examples are visualized using the world Mercator projection. As with the previous examples for ISEA3H and ISEA9R, the zones of a given refinement level are filled with the same color. Note that the zones closer to the poles are wider in longitude than zones closer to the equator, and can be seen in a 2-to-1 configuration at jumps in the variable widths coalescence factor such as at the 45° parallel, but these zones still occupy a similar area within the error budget. However, in these illustrations, the polar zones appear much larger than zones of the same level closer to the equator because the Mercator projection enlarges features away from the equator.

[\\https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=5](https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=5)

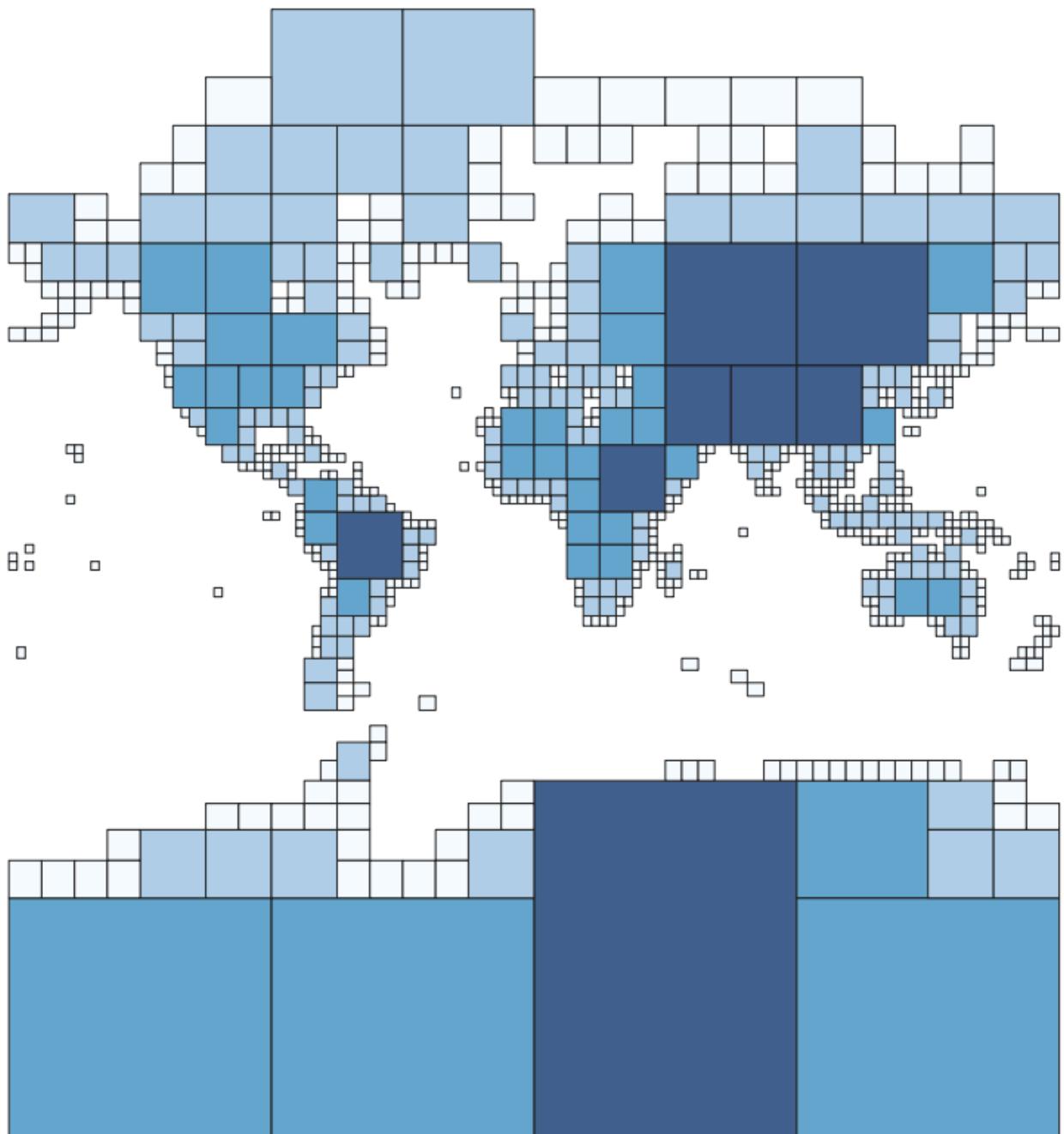


Figure C.28 – GeoJSON compact zones response for querying GNOSIS Global Grid DGGRS at level 5 for a [Viewfinder Panoramas](#) land elevation collection visualized in QGIS

\https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/GNOSISGlobalGrid/zones?
bbox=30,40,50,60&zone-level=7

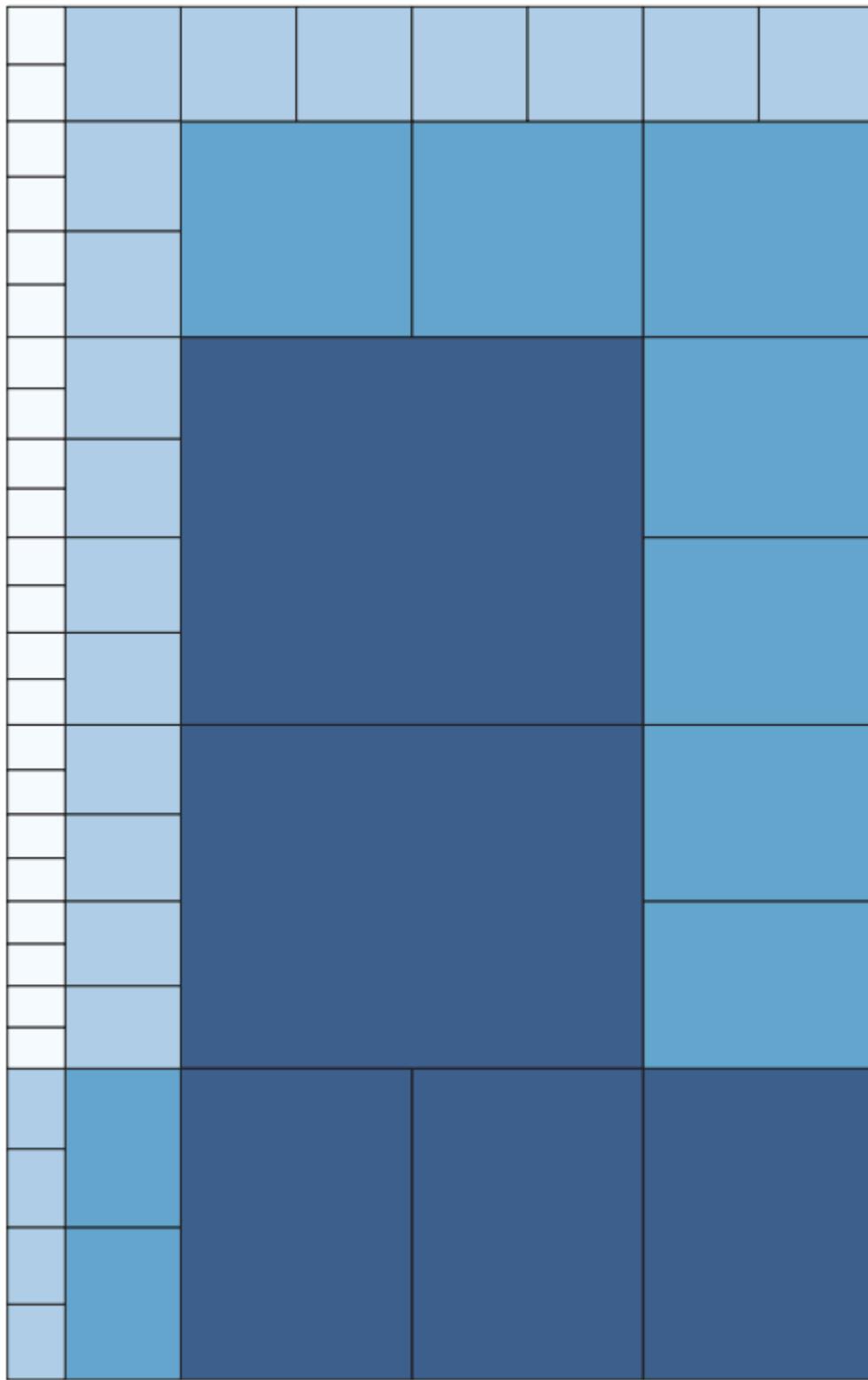


Figure C.29 – GeoJSON compact zones response for querying GNOSIS Global Grid DGGRS at level 7 for a (40°N, 30°E) to (60°N, 50°E) bounding box visualized in QGIS

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/GNOSISGlobalGrid/zones?zone-level=8&parent-zone=2-2-9

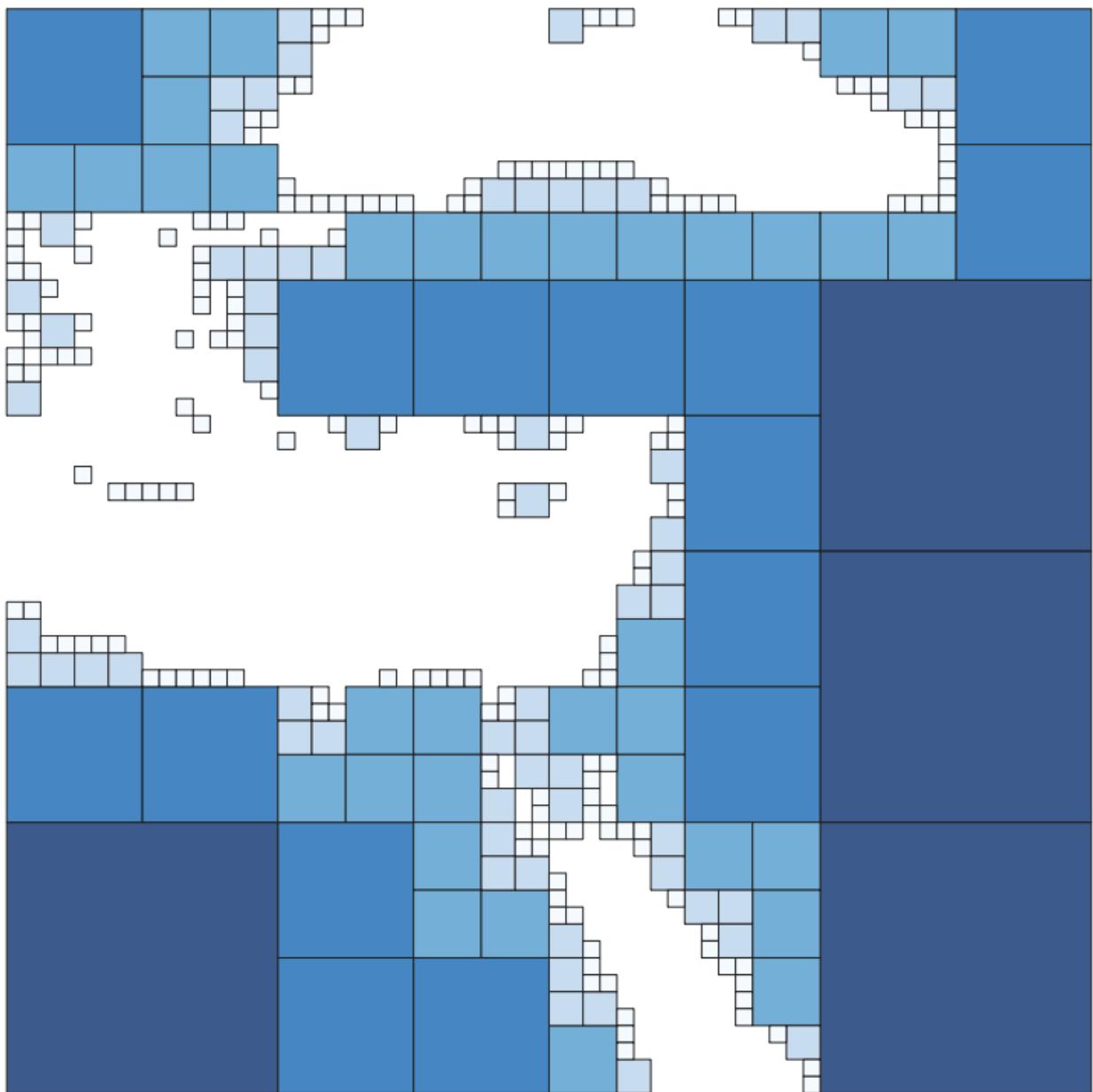


Figure C.30 – GeoJSON compact zones response for querying GNOSIS Global Grid DGGRS at level 8 for parent zone 2-2-9, for a Viewfinder Panoramas, visualized in QGIS

[\\https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/GNOSISGlobalGrid/zones?zone-level=13&parent-zone=6-1F-98](https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/GNOSISGlobalGrid/zones?zone-level=13&parent-zone=6-1F-98)

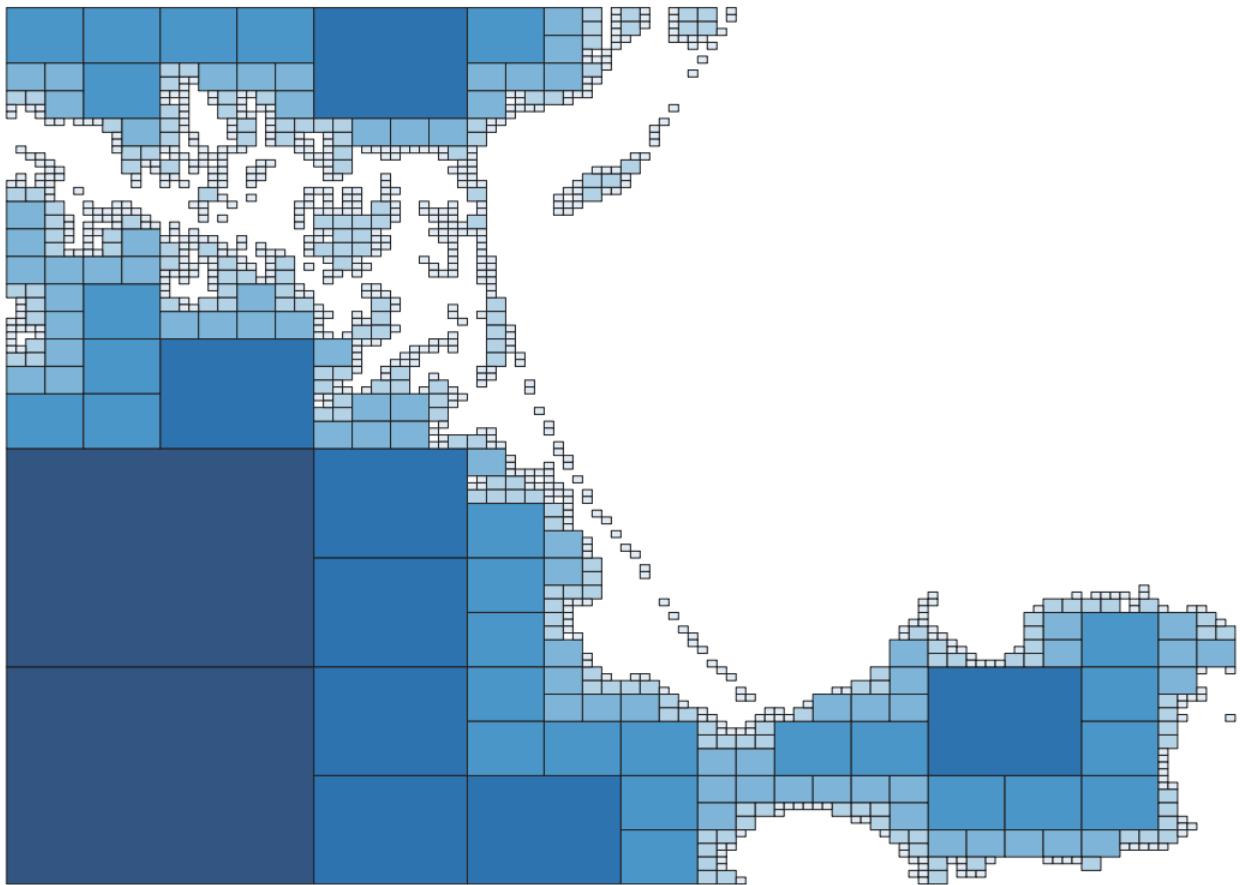


Figure C.31 – GeoJSON compact zones response for querying GNOSIS Global Grid DGGRS at level 13 for parent zone [6-1F-98](#), for a [Viewfinder Panoramas](#), visualized in QGIS

C.7. Filtering zone queries using CQL2 expressions

The Zone Query capabilities is most useful for use cases where the retrieved zone list is the result of some operation, whether a processing workflow or the evaluation of a query expressed by the client. The CQL2 filter capability allows to express such queries using the [OGC Common Query Language \(CQL2\)](#). Note that the processing workflow could also itself include CQL2 expressions using [input/output field modifiers](#).

The CQL2 requirement class in this standard specifically calls for the [CQL2-Text encoding](#), which is an intuitive representation with many similarities to SQL. CQL2 defines minimum capabilities ([Basic CQL2](#)) as well as additional optional capabilities, such as [arithmetic operations](#) and [spatial relation functions](#). The API implementation's conformance resource (/conformance as defined in [OGC API – Common – Part 1: Core “Landing Page” requirement class](#)) would allow the client to know which capabilities are supported.

Basic CQL2 includes support for logical operators as well as to compare the value of queryables against literals using relational operators. The following examples will use the GEBCO collection from earlier ones to illustrate these capabilities. The previous simple zone queries using this collection were not particularly interesting since it covers the entire globe, but the use of CQL2 will enable the use of its data values. First, the client would retrieve the queryables by following the [ogc-rel:queryables] link from the collection description:

<https://maps.gnosis.earth/ogcapi/collections/gebco/queryables>

The application/schema+json response would look like so:

```
{  
    "$schema" : "https://json-schema.org/draft/2020-12/schema",  
    "$id" : "https://maps.gnosis.earth/ogcapi/collections/gebco/queryables",  
    "title" : "General Bathymetric Chart of the Oceans",  
    "type" : "object",  
    "properties" : {  
        "Elevation" : {  
            "type" : "number",  
            "x-ogc-property-seq" : 1,  
            "x-ogc-definition": "https://qudt.org/vocab/quantitykind/Altitude",  
            "x-ogc-unit": "m"  
        },  
        "rec.geom" : {  
            "x-ogc-role" : "primary-geometry",  
            "format" : "geometry-point"  
        }  
    }  
}
```

This informs the client that an Elevation queryable is available for use in CQL2 expressions. Although this collection is a gridded coverage, a rec.geom queryable is still included in the schema which represents the geometry of each individual grid cell. Since the value associated with these cells represent a point in this collection, the format indicates geometry-point. Later examples will make use of this rec.geom property.

The client could then submit the following zone query to return only zones where elevation is below sea level (`Elevation < 0`), at refinement level 6:

[https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?filter=Elevation < 0&zone-level=6](https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?filter=Elevation%3C0&zone-level=6)

The following image is a visualization of the list of zones returned.

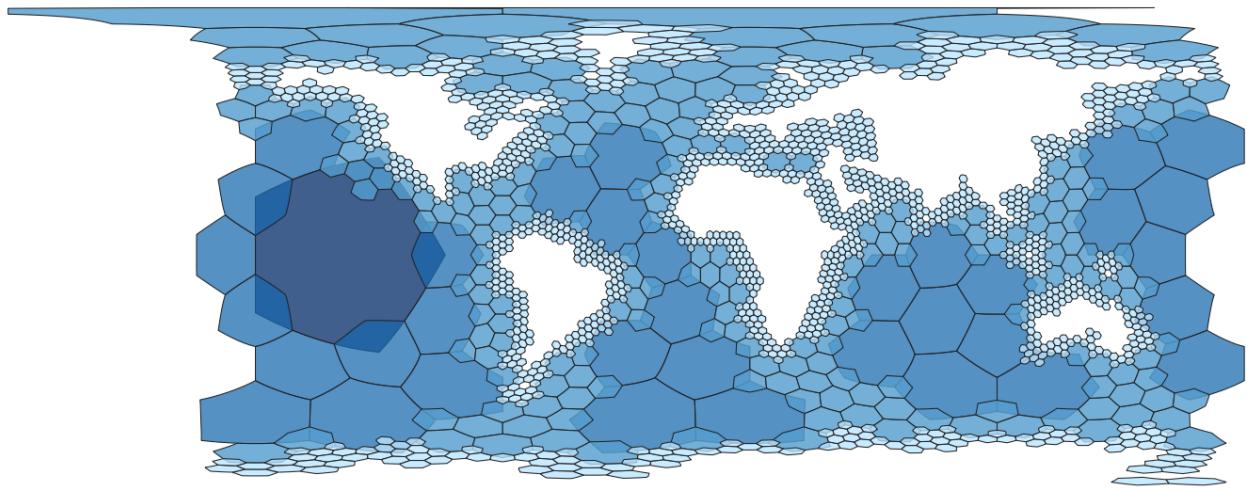


Figure C.32 – GeoJSON compact zones response for querying ISEA3H DGGRS for GEBCO 2014 bathymetry with CQL2 expression $Elevation < 0$ visualized in QGIS

The client could request instead for deeper elevations e.g., $Elevation < -4000$:

[https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?filter=Elevation < -4000&zone-level=6](https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?filter=Elevation%3C-4000&zone-level=6)

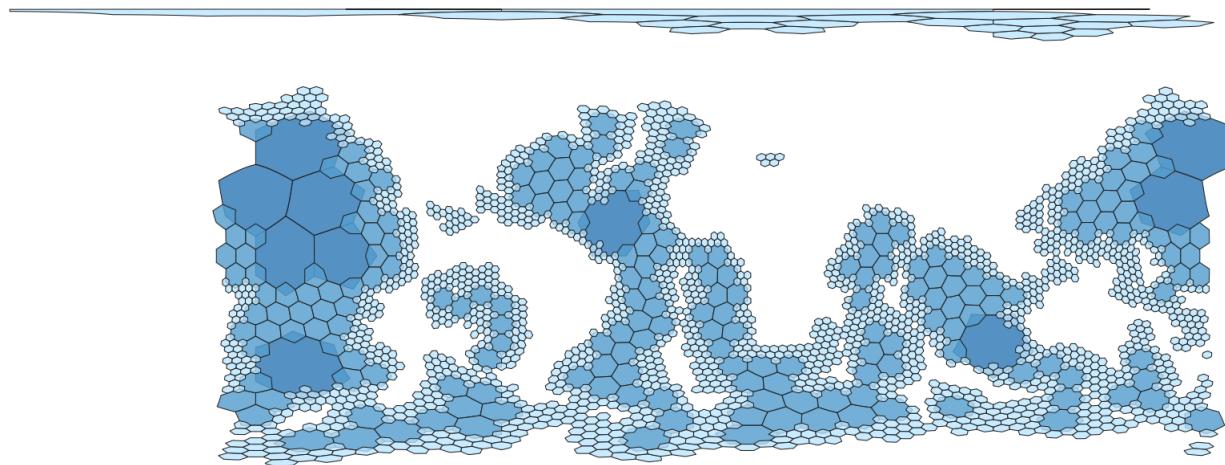


Figure C.33 – GeoJSON compact zones response for querying ISEA3H DGGRS for GEBCO 2014 bathymetry with CQL2 expression $Elevation < -4000$ visualized in QGIS

or for high elevations e.g., $Elevation > 3000$:

[https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?filter=Elevation > 3000&zone-level=6](https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?filter=Elevation%3E3000&zone-level=6)

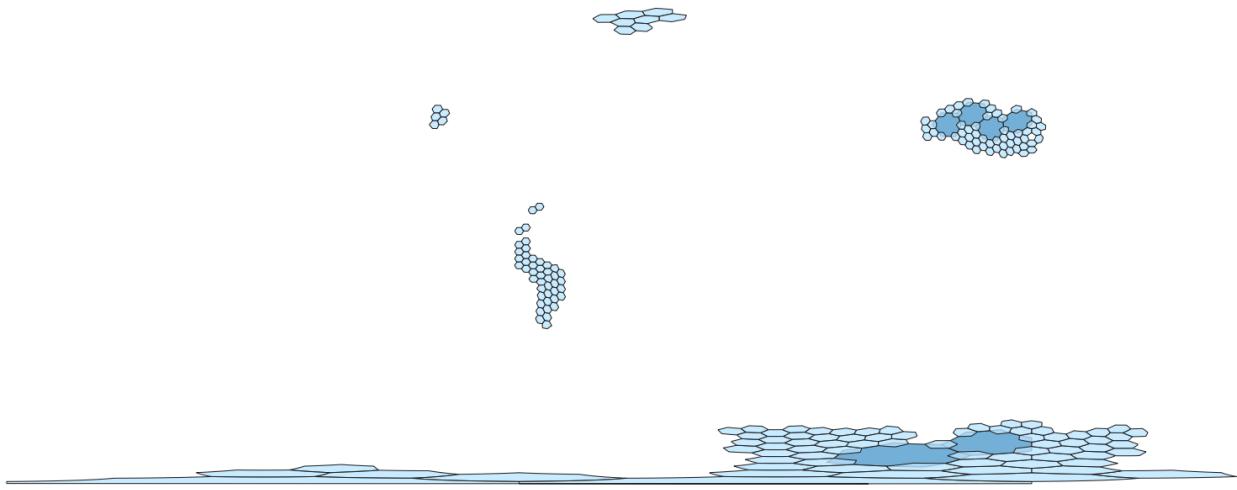


Figure C.34 – GeoJSON compact zones response for querying ISEA3H DGGRS for GEBCO 2014 bathymetry with CQL2 expression Elevation > 3000 visualized in QGIS

or for either deep or high elevations e.g., Elevation < -4000 or Elevation > 3000:

[https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?filter=Elevation < -4000 or Elevation > 3000&zone-level=6](https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?filter=Elevation%3C-4000%20or%20Elevation%3G3000&zone-level=6)

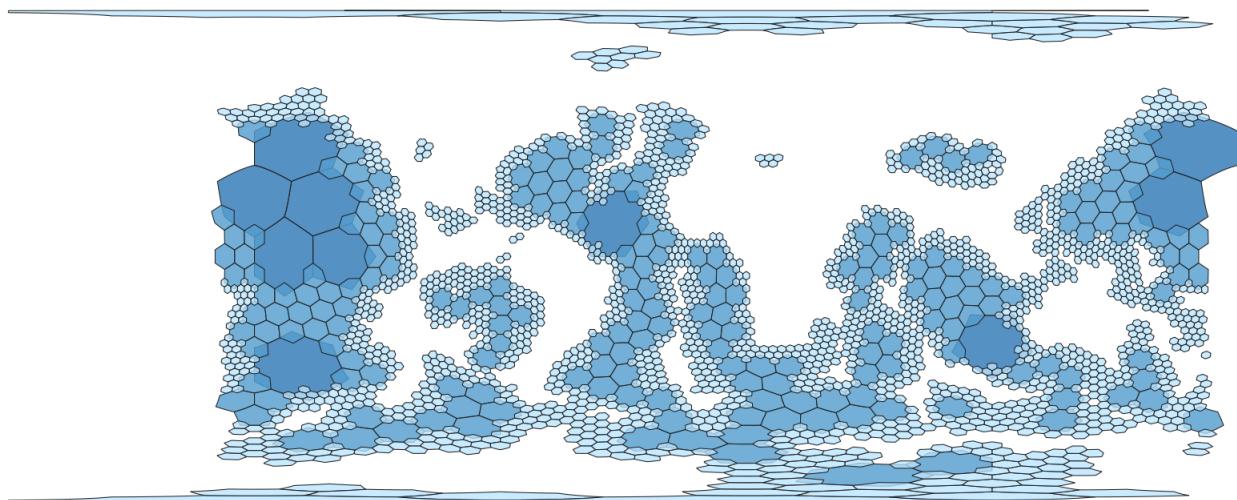


Figure C.35 – GeoJSON compact zones response for querying ISEA3H DGGRS for GEBCO 2014 bathymetry with CQL2 expression Elevation < -4000 or Elevation > 3000 visualized in QGIS

This last visualization highlights the fact that the Andes and Antarctica mountains are actually very close to very deep sea.

If the implementation support spatial relation functions, a query could test intersection with a geometry, which can be expressed using a WKT literal in CQL2-Text. The following example returns zones intersecting the Bermuda Triangle, making use of the `rec.geom` queryable, which represents the geometry of the individual coverage grid cells, mentioned earlier, at refinement level 9.

[https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?zone-level=9&filter=S_INTERSECTS\(rec.geom,POLYGON-64.8%2032.3,-65.5%2018.3,-80.3%2025.2,-64.8%2032.3\)&f=json\[“https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?zone-level=9&filter=S_INTERSECTS\(rec.geom,POLYGON-64.8 32.3,-65.5 18.3,-80.3 25.2,-64.8 32.3\)”\]](https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?zone-level=9&filter=S_INTERSECTS(rec.geom,POLYGON-64.8%2032.3,-65.5%2018.3,-80.3%2025.2,-64.8%2032.3)&f=json[\)

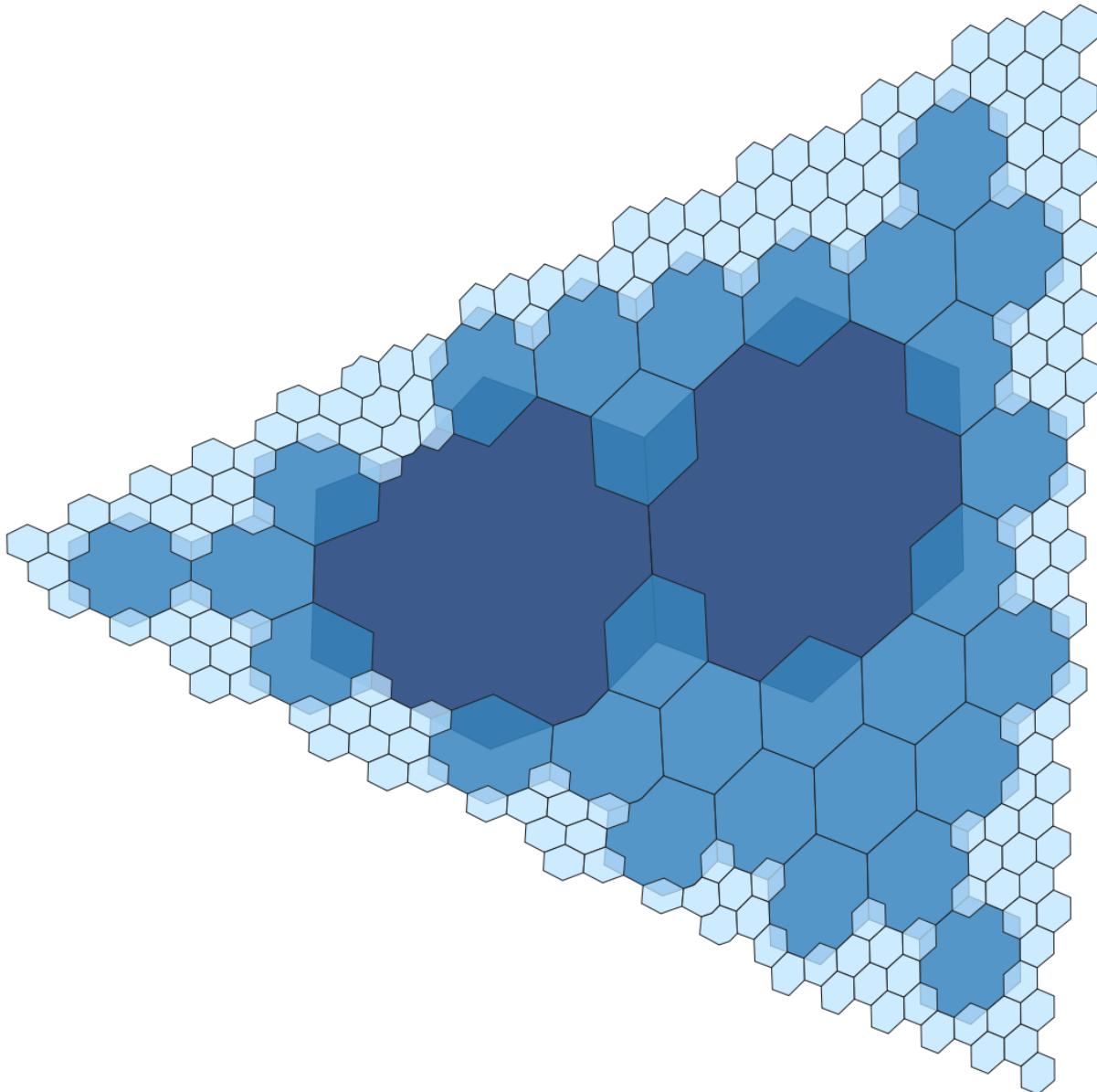


Figure C.36 – GeoJSON compact zones response for querying ISEA3H DGRS for [GEBCO 2014 bathymetry](#) with a CQL2 expression using `S_INTERSECTS()` for a WKT triangular polygon visualized in QGIS

This spatial filter could also be combined together with a relational operator as in the following query for Elevation < -5500 within the Bermuda Triangle:

[https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?zone-level=11&filter=Elevation%20<%20-5500%20and%20S_INTERSECTS\(rec.geom,POLYGON-64.8%2032.3,-65.5%2018.3,-80.3%2025.2,-64.8%2032.3\)&f=json\[“https://maps.gnosis.earth/](https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?zone-level=11&filter=Elevation%20<%20-5500%20and%20S_INTERSECTS(rec.geom,POLYGON-64.8%2032.3,-65.5%2018.3,-80.3%2025.2,-64.8%2032.3)&f=json[\)

ogcapi/collections/gebco/dggs/isea3h/zones?zone-level=11&filter=Elevation < -5500 and S_INTERSECTS(rec.geom,POLYGON-64.8 32.3,-65.5 18.3,-80.3 25.2,-64.8 32.3)"]

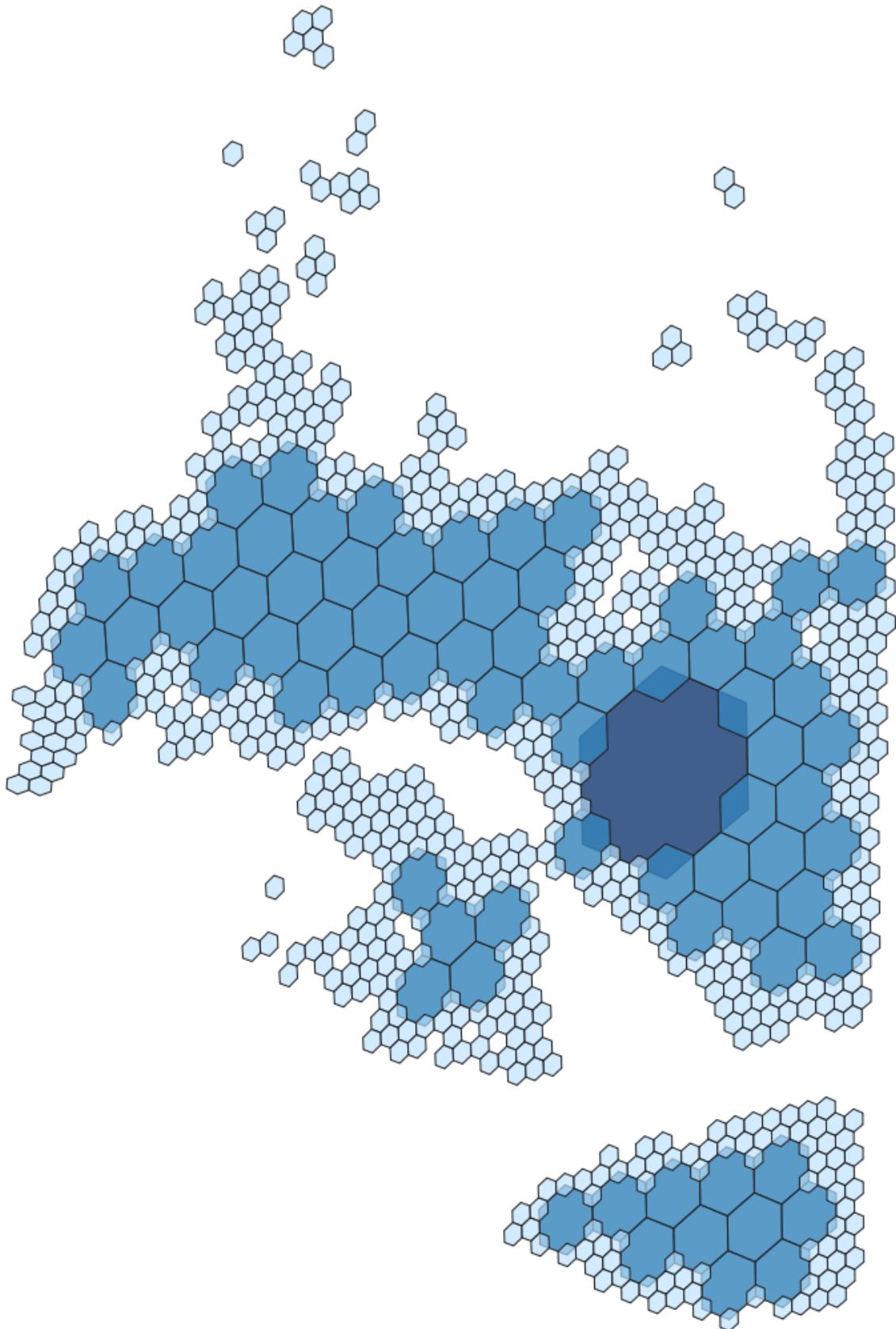


Figure C.37 – GeoJSON compact zones response for querying ISEA3H DGGRS for [GEBCO 2014 bathymetry](#) with a CQL2 expression using both Elevation < -500 as well as S_INTERSECTS() for a WKT polygon visualized in QGIS

To illustrate the use of CQL2 arithmetic operators, the following example will use a collection of daily climate CMIP5 data from the [Copernicus Climate Data Store](#). The request will be for zones where the difference between two different fields of the coverage, the maximum (tasmax) and minimum (tasmin) daily near-surface air temperature, is greater than 10 degrees Kelvin. In addition, this request will use the datetime parameter to subset the temporal dimension for June 1, 2023.

[https://maps.gnosis.earth/ogcapi/collections/climate:cmip5:singlePressure/dggs/isea3h/zones?zone-level=7&filter=\(tasmax-tasmin\)>10&datetime=2023-06-01](https://maps.gnosis.earth/ogcapi/collections/climate:cmip5:singlePressure/dggs/isea3h/zones?zone-level=7&filter=(tasmax-tasmin)>10&datetime=2023-06-01)

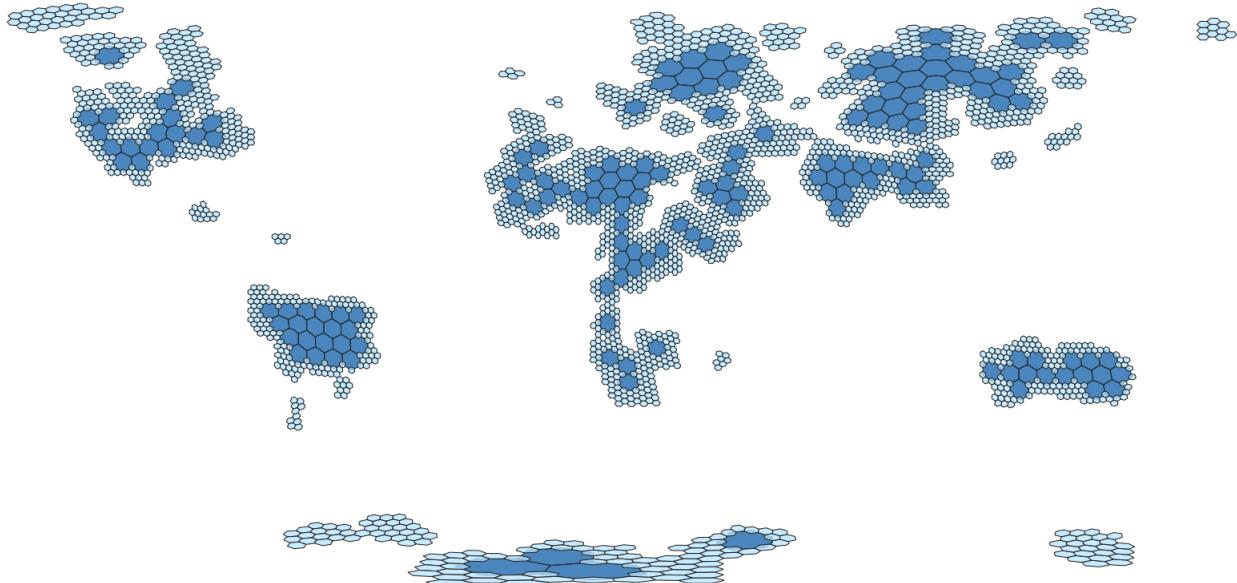


Figure C.38 – GeoJSON compact zones response for querying ISEA3H DGGRS for CMIP5 daily surface temperature from [Copernicus Climate Data Store](#) comparing delta with CQL2 expression ($\text{tasmax} - \text{tasmin}$) > 10 visualized in QGIS

CQL2 expressions can also be used for queries using attributes of vector features. The following examples use a collection of building polygon features in Ottawa extracted from the OpenStreetMap dataset, filtering buildings using both a spatial subset and the name attribute of the features.

This first example illustrates querying for the National Arts Centre at ISEA3H level 30:

[https://maps.gnosis.earth/ogcapi/collections/osm:ottawa:buildings/dggs/ISEA3H/zones?subset=Lat\(45.4198:45.4264\),Lon\(-75.704:-75.6935\)&filter=name='National Arts Centre'&zone-level=30](https://maps.gnosis.earth/ogcapi/collections/osm:ottawa:buildings/dggs/ISEA3H/zones?subset=Lat(45.4198:45.4264),Lon(-75.704:-75.6935)&filter=name='National Arts Centre'&zone-level=30)

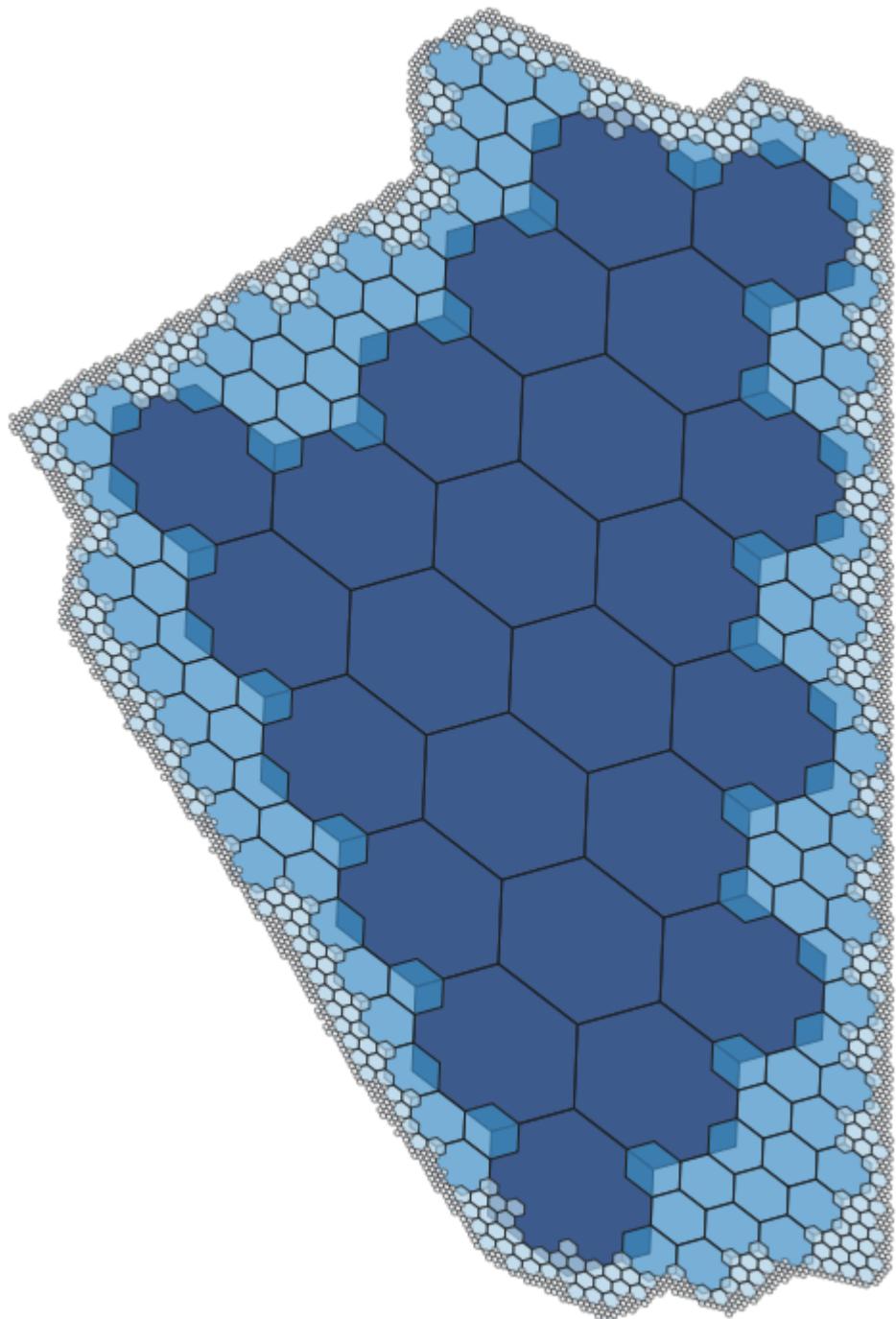


Figure C.39 — GeoJSON compact zones response for querying ISEA3H DGGRS for Ottawa buildings (© [OpenStreetMap](#) contributors) with CQL2 expression name='National Arts Centre' at level 30, visualized in QGIS

This second example illustrates querying for the East Block at ISEA9R level 15 (corresponding to ISEA3H level 30):

[https://maps.gnosis.earth/ogcapi/collections/osm:ottawa:buildings/dggs/ISEA9R/zones?
subset=Lat\(45.4198:45.4264\),Lon\(-75.704:-75.6935\)&filter=name='East Block'&zone-level=15](https://maps.gnosis.earth/ogcapi/collections/osm:ottawa:buildings/dggs/ISEA9R/zones?subset=Lat(45.4198:45.4264),Lon(-75.704:-75.6935)&filter=name='East Block'&zone-level=15)

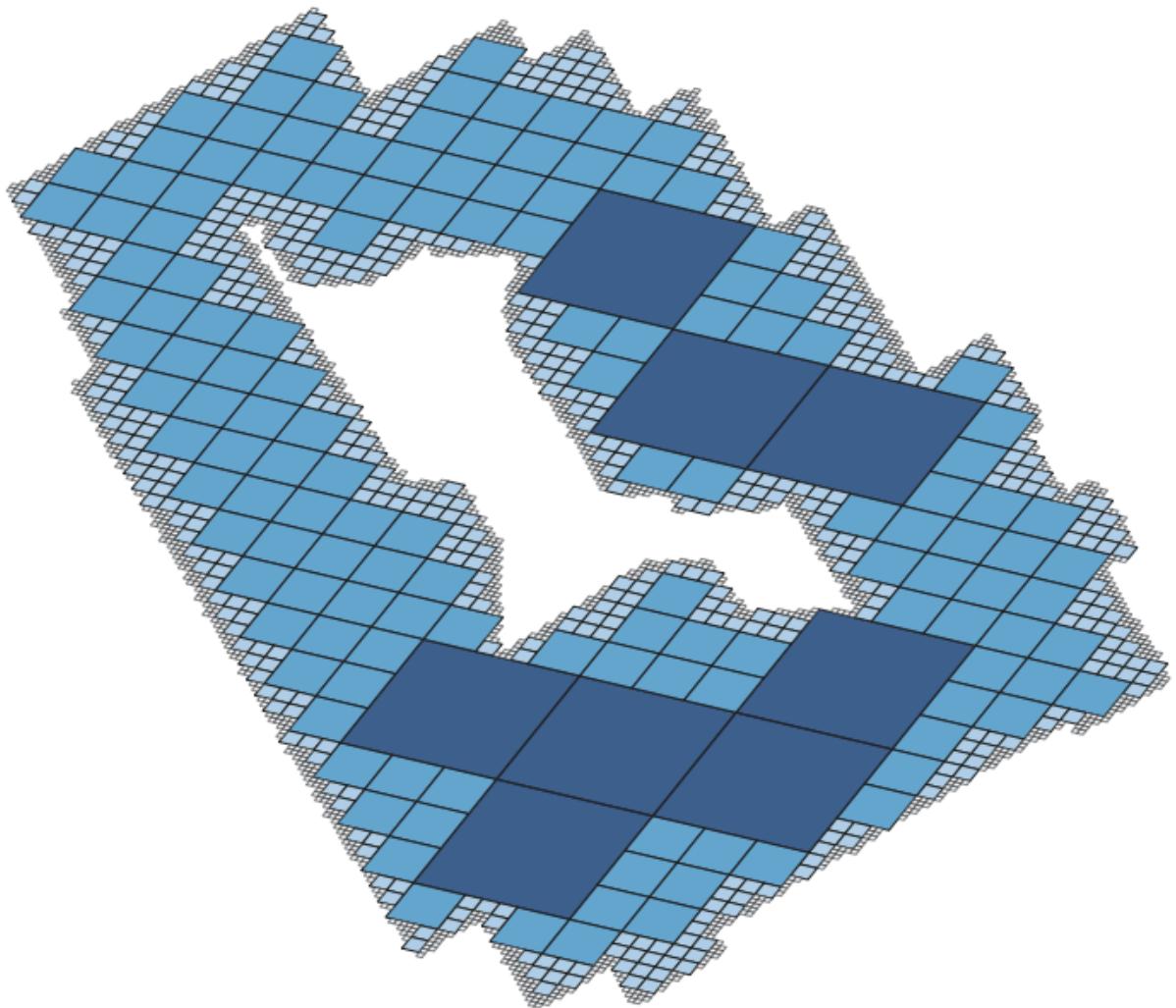


Figure C.40 – GeoJSON compact zones response for querying ISEA9R DGGRS for Ottawa buildings (© [OpenStreetMap](#) contributors) with CQL2 expression name= 'East Block' at level 15, visualized in QGIS

This third example illustrates querying for the Centre Block at GNOSIS Global Grid level 24 (corresponding approximately to same resolution):

[https://maps.gnosis.earth/ogcapi/collections/osm:ottawa:buildings/dggs/GNOSISGlobalGrid/zones?subset=Lat\(45.4198:45.4264\),Lon\(-75.704:-75.6935\)&filter=name='Centre Block'&zone-level=24](https://maps.gnosis.earth/ogcapi/collections/osm:ottawa:buildings/dggs/GNOSISGlobalGrid/zones?subset=Lat(45.4198:45.4264),Lon(-75.704:-75.6935)&filter=name='Centre Block'&zone-level=24)

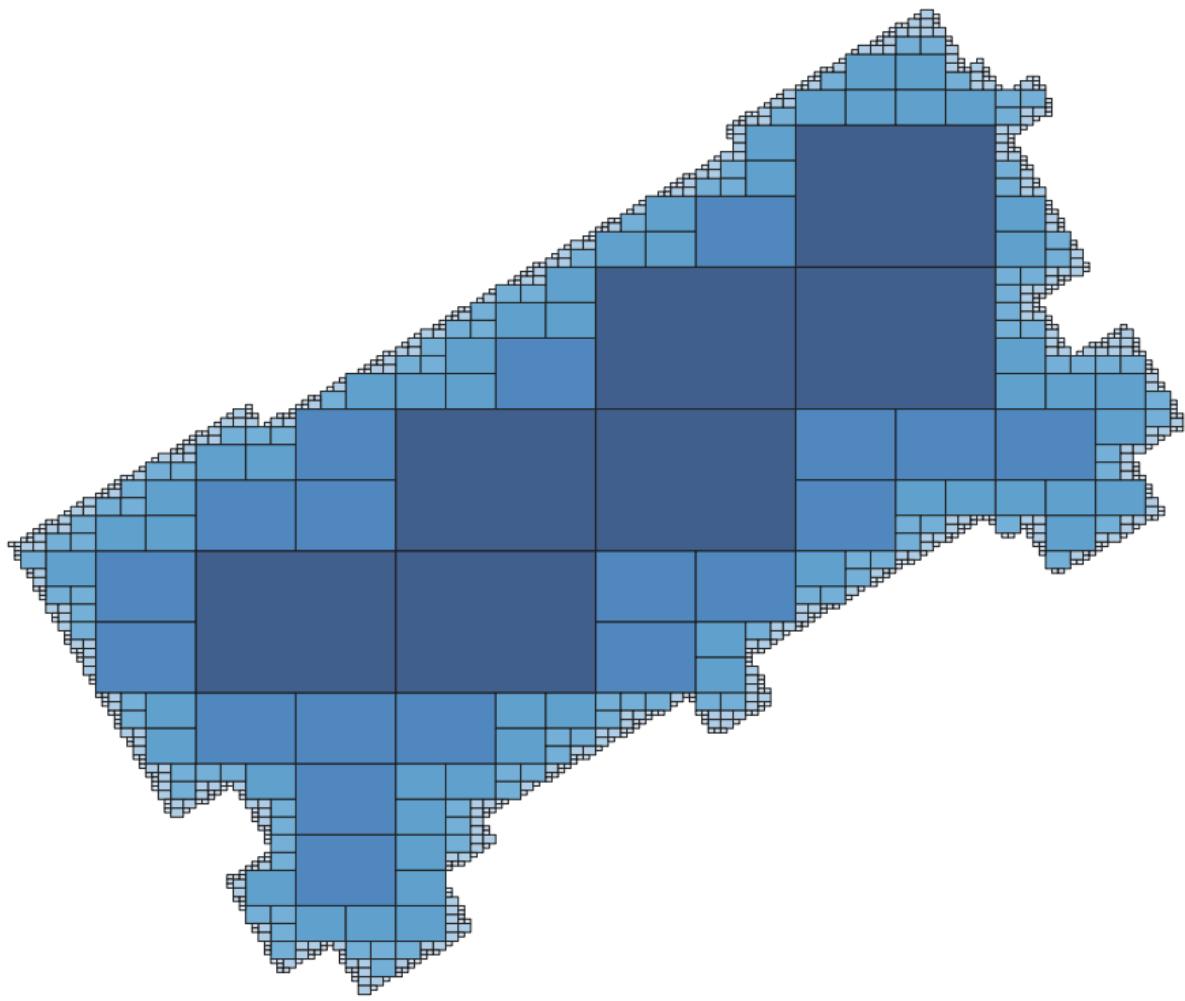


Figure C.41 – GeoJSON compact zones response for querying GNOSIS Global Grid DGGRS for Ottawa buildings (© OpenStreetMap contributors) with CQL2 expression name= 'Centre Block' at level 24, visualized in QGIS

A future extension could also enable cross-collection queries, possibly using a `joinCollections` parameter which would make queryables from different collections available.

C.8. Retrieving data for a DGGRS Zone

The following example retrieves elevation data for the GEBCO collection for ISEA3H zone A2-0-A at a relative depth of 8:

<https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H/zones/A2-0-A/data?zone-depth=8>

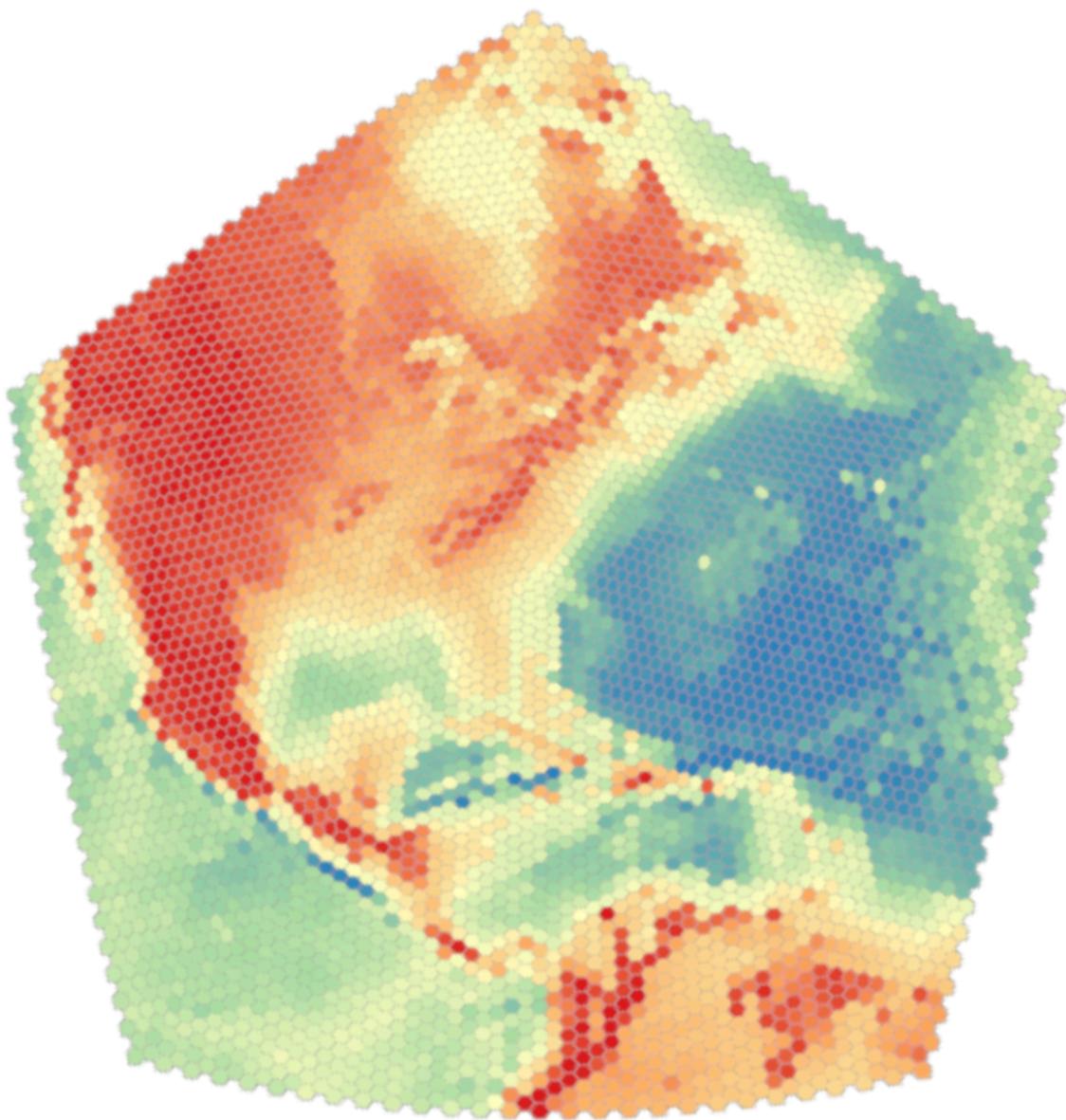


Figure C.42 – Visualization of a DGGS-JSON response for querying ISEA3H DGGRS for zone A2-0-A of GEBCO 2014 bathymetry at a relative depth of 8

D

ANNEX D (INFORMATIVE) REVISION HISTORY

ANNEX D (INFORMATIVE) REVISION HISTORY

Table D.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-zone 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
2024-06-18	1.0. Orc1	Jerome St Louis	all Initial draft ready for OAB review



BIBLIOGRAPHY



BIBLIOGRAPHY

1. Robert Thomas, Sara Saeedi: OGC 23-010, *Towards a Federated Marine SDI: Connecting Land and Sea to Protect the Arctic Environment Engineering Report*. Open Geospatial Consortium (2023). <http://www.opengis.net/doc/PER/FMSDI3>.
2. Robert Gibb, Byron Cochrane, Matthew Purss: OGC 20-039r2, *OGC Testbed-16: DGGS and DGGS API Engineering Report*. Open Geospatial Consortium (2021). <http://www.opengis.net/doc/PER/t16-D017>.
3. Jérôme Jacovella-St-Louis, Flexible real-time data processing and visualization workflows emerging from OGC API modules (Draft), OGC 21-033 (2021). <https://maps.ecere.com/moaw/DiscussionPaper-Draft3.pdf>
4. Universal Binary JSON Specification. <https://ubjson.org/>
5. Zarr Storage Specification Version 2. https://zarr-specs.readthedocs.io/en/latest/v2/v2_0.html
6. Michael F. Goodchild, Yang Shiren, A hierarchical spatial data structure for global geographic information systems, CVGIP: Graphical Models and Image Processing, Volume 54, Issue 1, Pages 31-44, ISSN 1049-9652 (1992). [https://doi.org/10.1016/1049-9652\(92\)90032-S](https://doi.org/10.1016/1049-9652(92)90032-S)
7. John P. Snyder, An Equal-Area Map Projection for Polyhedral Globes, *Cartographica*, 29(1), 10-21. urn:doi:10.3138/27H7-8K88-4882-1752 (1992). <https://doi.org/10.3138/27H7-8K88-4882-1752>
8. Kevin Sahr, Denis White, A. Jon Kimerling, Geodesic Discrete Global Grid Systems, *Cartography and Geographic Information Science*, Vol. 30, No. 2, pp. 121-134 (2003). <https://cs.sou.edu/~sahrk/dgg/pubs/gdgg03.pdf>
9. Robert Gibb, The rHEALPix Discrete Global Grid System. *IOP Conference Series: Earth and Environmental Science*. 34. 012012. 10.1088/1755-1315/34/1/012012 (2016). <https://dx.doi.org/10.1088/1755-1315/34/1/012012>