

OGC API - Coverages - Part 1

Core

Open Geospatial Consortium

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: 2023-xx-xx

External identifier of this OGC® document: <http://www.opengis.net/doc/is/ogcapi-coverages-1/1.0>

Internal reference number of this OGC® document: 19-087

Version: 0.0.7

Category: OGC® Implementation Specification

Editors: Charles Heazel, Jérôme Jacovella-St-Louis

OGC API - Coverages - Part 1: Core

Copyright notice

Copyright © 2019-2023 Open Geospatial Consortium

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

Warning

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.

Document type: OGC® Implementation
Specification

Document stage: Draft

Document language: English

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 NON-INFRINGEMENT 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.

Table of Contents

1. Scope	7
1.1. API Scope	7
1.2. Content Scope	7
2. Conformance	8
2.1. Requirements classes defining resources	8
2.2. Requirements classes defining query parameters	8
2.3. Requirements classes defining resource representations	9
2.4. Summary of conformance URIs	9
3. References	11
4. Terms and Definitions	13
4.1. Coverage	13
4.2. Regular grid	13
4.3. Irregular grid	13
4.4. Displaced grid	13
4.5. Mesh	13
4.6. Partition [of a coverage]	13
4.7. Scene	13
4.8. Sensor model	14
4.9. Transformation grid	14
5. Conventions	15
5.1. Identifiers	15
5.2. Examples	15
5.3. Schema	15
5.4. UML Notation	15
5.5. Namespace Prefix Conventions	15
5.6. Link relations	16
5.7. Use of HTTPS	17
6. Overview	18
6.1. General	18
7. Requirements Class "Core"	21
7.1. Overview	21
7.2. Requirements	22
7.2.1. Extended collection list response (<code>/collections</code>)	22
7.2.2. Extended collection description response (<code>/collections/{collectionId}</code>)	22
7.2.3. Collection schema response (<code>/collections/{collectionId}/schema</code>)	23
7.2.4. Coverage data retrieval (<code>/collections/{collectionId}/coverage</code>)	24
8. Requirements Class "Subsetting"	25
8.1. Overview	25

8.2. Subsetting Examples	25
8.3. Requirements	25
8.3.1. Parameter <code>bbox</code>	25
8.3.2. Parameter <code>bbox-crs</code>	26
8.3.3. Parameter <code>datetime</code>	27
8.3.4. Parameter <code>subset</code>	28
8.3.5. Parameter <code>subset-crs</code>	30
8.3.6. Permission for Slicing Sparse Dimensions	30
9. Requirements Class "Scaling"	32
9.1. Overview	32
9.2. Scaling Examples	32
9.3. Requirements	32
9.3.1. Parameter <code>scale-factor</code>	32
9.3.2. Parameter <code>scale-axes</code> parameters	32
9.3.3. Parameter <code>scale-size</code>	33
9.3.4. Downsampling permission	34
10. Requirements Class "Field Selection"	35
10.1. Overview	35
10.2. Field Selection Examples	35
10.3. Requirements	35
10.3.1. Parameter <code>properties</code>	35
11. Requirement Class "Coordinate Reference System"	37
11.1. Overview	37
11.2. Requirements	37
11.2.1. Parameter <code>crs</code>	37
12. Requirements Class "Coverage Tiles"	39
12.1. Overview	39
12.2. Coverage Tiles Examples	39
12.3. Requirements	39
12.3.1. Coverage tilesets (<code>/collections/{collectionId}/coverage/tiles</code>)	39
13. Requirements Class "Scenes"	41
13.1. Overview	41
13.2. Requirements	41
13.2.1. List of scenes (<code>/collections/{collectionId}/scenes</code>)	41
13.2.2. Individual scene (<code>/collections/{collectionId}/scenes/{sceneId}</code>)	42
13.2.3. Scene coverage (<code>/collections/{collectionId}/scenes/{sceneId}/coverage</code>)	43
13.2.4. Scene coverage tiles (<code>/collections/{collectionId}/scenes/{sceneId}/coverage/tiles</code>)	44
13.2.5. Scene queryables	44
14. Requirements Classes for Encodings	45
14.1. Media Types	45
14.2. Requirements Class "HTML"	46

14.2.1. Overview	46
14.2.2. Requirements	46
14.3. Requirements Class "GeoTIFF"	47
14.3.1. Overview	47
14.4. Requirements Class "netCDF"	47
14.4.1. Overview	47
14.5. Requirements Class "CIS JSON"	47
14.5.1. Overview	47
14.5.2. Requirements	49
14.6. Requirements Class "CoverageJSON"	50
14.6.1. Overview	50
14.7. Requirements Class "LAS"	50
14.7.1. Overview	50
14.8. Requirements Class "LASZip"	51
14.8.1. Overview	51
14.9. Requirements Class "PNG"	51
14.9.1. Overview	51
14.10. Requirements Class "JPEG XL"	51
14.10.1. Overview	51
14.11. Requirements Class "JPEG 2000"	51
14.11.1. Overview	51
14.12. Requirements Class "(Geo)Zarr"	51
14.12.1. Overview	51
15. Requirements class "OpenAPI 3.0"	52
15.1. Overview	52
15.2. Requirements	52
Annex A: Conformance Class Abstract Test Suite (Normative)	53
A.1. Conformance Class A	53
A.1.1. Requirement 1	53
A.1.2. Requirement 2	53
Annex B: Examples (Informative)	54
B.1. Basic Coverage Retrieval Example	54
B.2. Subsetting Example	54
B.3. Scaling Example	54
B.4. Field Selection (Range Subsetting) Example	54
B.5. Coverage Tiles Example	54
Annex C: Revision History	55
Annex D: Bibliography	56

i. Abstract

<Insert Abstract Text here>

ii. Keywords

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

ogcdoc, OGC document, geographic information, spatial data, spatial things, dataset, distribution, API, json, html, OpenAPI, REST, Coverages

iii. Preface

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 Inc. shall not be held responsible for identifying any or all such patent rights.

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

iv. Submitting organizations

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

- Heazeltech LLC
- Ecere Corporation

v. Submitters

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

Name	Affiliation
Charles Heazel (Editor)	HeazelTech
Jérôme Jacovella-St-Louis < jerome@ecere.com > (Editor)	Ecere Corporation
Stephan Meissl < stephan.meissl@eox.at >	EOX IT Services GmbH
Tom Kralidis < tom.kralidis@canada.ca >	Meteorological Service of Canada

Chapter 1. Scope

This [OGC API - Coverages](#) standard establishes an access mechanism for coverages as defined by the [OGC Abstract Specification Topic 6 / ISO 19123 Schema for coverage geometry and functions](#) through a Web API which can be described by an API description language such as the [OpenAPI specification](#).

1.1. API Scope

The functionality provided by *OGC API - Coverages* addresses a similar scope as that of the [OGC Web Coverage Service \(WCS\) 2.1 Interface Standard](#). It is expected that *OGC API - Coverages* and WCS implementations will be able to interoperate, allowing developers to pick the solution best suited for their requirements, and also to implement one as a façade on top of the other.

The OGC is using an incremental approach to their API development. The initial goal is to develop a relatively simple API standard which will meet the needs of a large percentage of implementors. Additional capabilities will be added based on community demand. The functionality covered in *Part 1* is:

- Description of a coverage, including its domain (dimensions/axes) and range (schema of the fields for the values of each cell),
- Retrieval of a coverage with support for content type negotiation,
- Subsetting (trimming or slicing along one or more dimensions/axes of its domain),
- Field selection (retrieving only certain fields of the range, also known as range subsetting),
- Scaling (retrieving a downsampled or upsampled coverage),
- Selection of an output Coordinate Reference System (CRS) which may differ from the native CRS,
- Retrieving part of a coverage as tiles (in combination with [OGC API - Tiles](#), each tile corresponding to a specific subset trimmed along two spatial dimensions, and the response downsampled for lower resolution tile matrices),
- Listing, retrieving metadata for and retrieving data from individual scenes of which an overall coverage may consist.

1.2. Content Scope

The *OGC API - Coverages* standard provides access to content which complies with [OGC Abstract Specification Topic 6: Schema for coverage geometry and functions](#). This includes, but is not limited to, support for gridded (regular and irregular) and multi-point coverages (e.g., point clouds).

The content of the coverage can be encoded using any suitable logical model and physical encodings, such as those defined by [Coverage Implementation Schema \(CIS\) 1.1](#), [netCDF](#), [CoverageJSON](#), [GeoTIFF](#) (including [Cloud Optimized GeoTIFF \(COG\)](#)), or [LAS/LAZ](#) (including [Cloud Optimized Point Cloud \(COPC\)](#)).

Chapter 2. Conformance

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

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

The Requirements Classes for *OGC API - Coverages* are:

2.1. Requirements classes defining resources

- [Core](#)
- [Coverage Tiles](#)
- [Coverage Scenes](#)

The *Core* Requirements Class is the minimal useful service interface for an OGC Coverages API. The requirements specified in this Requirements Class are mandatory for all implementations of *OGC API - Coverages*.

The *Coverage Tiles* Requirements Class defines how to combine the *OGC API - Tiles* building blocks with the Coverages API to request coverage tiles.

The *Scenes* Requirements Class defines how to present separate components of a coverage as individual scenes, in addition to an overall coverage of the whole collection.

2.2. Requirements classes defining query parameters

- [Subsetting](#)
- [Scaling](#)
- [Field Selection](#)
- [Coordinate Reference System](#)

The *Subset* Requirements Class provides capabilities to select a sub-set of a [Coverage](#) using a multi-dimensional "bounding box" which is suitable for any coordinate reference system and any

dimension.

The *Scaling* Requirements Class defines parameters for retrieving data from n-dimensional Range Sets at a resolution different than the original.

The *Field Selection* (Range Subsetting) Requirements Class defines parameters for selecting a subset of the bands (defined in the Range Type) to retrieve from Range Sets.

The *Coordinate Reference System* Requirements Class defines the ability to request coverage data in an alternate output coordinate reference system.

2.3. Requirements classes defining resource representations

- [HTML](#)
- [GeoTIFF](#)
- [netCDF](#)
- [CIS JSON](#)
- [CoverageJSON](#)
- [LAS](#)
- [LASZip](#)
- [PNG](#)
- [JPEG XL](#)
- [JPEG 2000](#)
- [\(Geo\)Zarr](#)
- **[OpenAPI 3.0](#)**

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

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

2.4. Summary of conformance URIs

Table 1. Conformance class URIs

Corresponding requirements class	Conformance class URI
Core	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/core
Scaling	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/scaling
Subsetting	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/subsetting

Corresponding requirements class	Conformance class URI
Field Selection	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/fieldselection
CRS	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/crs
Coverage Tiles	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/tiles
Coverage Scenes	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/scenes
HTML	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/html
GeoTIFF	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/geotiff
netCDF	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/netcdf
CIS JSON	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/cisjson
CoverageJSON	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/coveragejson
LAS	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/las
LASZip	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/laszip
PNG	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/png
JPEG XL	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/jpegxl
JPEG 2000	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/jpeg2000
(Geo)Zarr	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/zarr
OpenAPI 3.0	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/oas30

Chapter 3. References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

- Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T.: IETF RFC 2616, **HTTP/1.1**, [RFC 2616](#)
- Rescorla, E.: IETF RFC 2818, **HTTP Over TLS**, [RFC 2818](#)
- Klyne, G., Newman, C.: IETF RFC 3339, **Date and Time on the Internet: Timestamps**, [RFC 3339](#)
- Berners-Lee, T., Fielding, R., Masinter, L.: IETF RFC 3986, **Uniform Resource Identifier (URI): Generic Syntax**, [RFC 3986](#)
- Duerst, M., Suignard, M.: IETF RFC 3987, **Internationalized Resource Identifiers (IRIs)**, [RFC 3987](#)
- Gregorio, J., Fielding, R., Hadley, M., Nottingham, M., Orchard, D.: IETF RFC 6570, **URI Template**, [RFC 6570](#)
- Nottingham, M.: IETF RFC 8288, **Web Linking**, [RFC 8288](#)
- OGC 19-072: **OGC API - Common - Part 1: Core**, (Draft) https://github.com/opengeospatial/oapi_common/blob/Master/19-072.pdf
- OGC 20-024: **OGC API - Common - Part 2: Geospatial Data**, (Draft) https://github.com/opengeospatial/oapi_common/blob/Master/20-024.pdf
- OGC 09-146: **OGC Coverage Implementation Schema (CIS)**, version 1.1, [CIS](#)
- OGC 19-008: **OGC GeoTIFF Standard**, Version 1.1, <http://docs.opengeospatial.org/is/19-008r4/19-008r4.html>
- OGC Schema: **OGC JSON Schema for Coverage Implementation Schema**, version 1.1, 2017, [CIS Schema](#)
- OGC 10-090: **OGC Network Common Data Form (NetCDF) Core Encoding Standard**, Version 1.0, http://portal.opengeospatial.org/files/?artifact_id=43732
- OGC 17-089: **OGC Web Coverage Service (WCS) Interface Standard - Core**, Version 2.1, ([WCS 2.1](#))
- Open API Initiative: **OpenAPI Specification 3.0.2**, [OpenAPI](#)
- **Schema.org**: [Schema.org](#)
- W3C: **HTML5**, W3C Recommendation, [HTML5](#)
- OGC: OGC 07-011, **Abstract Specification Topic 6: The Coverage Type and its Subtypes**, version 7.0 (identical to ISO 19123:2005), 2007
- OGC: OGC 08-094, **OGC® SWE Common Data Model Encoding Standard**, version 2, 2011
- OGC: OGC 12-000, **OGC® SensorML: Model and XML Encoding Standard**, version 2, 2014
- OGC: OGC 09-146r2, **GML 3.2.1 Application Schema – Coverages**, version 1.0.1, 2012

- OGC: OGC 16-083, **Coverage Implementation Schema – ReferenceableGridCoverage Extension**, version 1, 2017
- OGC: OGC 09-110r4, **Web Coverage Service (WCS) Core Interface Standard**, version 2, 2012
- OGC: OGC 13-102r2, **Name type specification – Time and index coordinate reference system definitions** (OGC Policy Document), version 1, 2014
- IETF: RFC 7159, **The JavaScript Object Notation (JSON) Data Interchange Format**, <https://www.ietf.org/rfc/rfc7159.txt>, 2014
- W3C: W3C JSON-LD 1.0, **A JSON-based Serialization for Linked Data**. <http://www.w3.org/TR/json-ld/>, 2014
- W3C: W3C JSON-LD 1.0 **Processing Algorithms and API**. <http://www.w3.org/TR/json-ld-api>, 2014
- W3C: W3C RDF 1.1 **Concepts and Abstract Syntax**. <https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/>, 2014

Chapter 4. Terms and Definitions

This document uses the terms defined in Sub-clause 5 of [OGC API - Common - Part 1: Core](#) (OGC 19-072), which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

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

4.1. Coverage

function which returns values from its range for any direct position within its domain (as defined in OGC Abstract Topic 6)

4.2. Regular grid

grid whose grid lines have a constant distance along each grid axis

4.3. Irregular grid

Grid whose grid lines have individual distances along each grid axis

4.4. Displaced grid

grid whose direct positions are topologically aligned to a grid, but whose geometric positions can vary arbitrarily

4.5. Mesh

coverage consisting of a collection of curves, surfaces, or solids, respectively

4.6. Partition [of a coverage]

separately stored coverage acting, by being referenced in the coverage on hand, as one of its components

4.7. Scene

component of a coverage with its domain being a subset of the overall coverage for a coverage consisting of multiple scenes, similar to a partition

(in the context of Earth Observation imagery data products and the *Scenes* requirements class defined in this standard)

Unlike Coverage Implementation Schema (CIS) *coverages by partitioning*, scenes are an optional capability of the API with corresponding individual resource paths rather than a potential aspect of

a coverage resource. A client can either retrieve a coverage for the overall coverage or for an individual scene, but will never receive a special "partitioned coverage" response necessitating special handling considerations.

4.8. Sensor model

mathematical model for estimating geolocations from recorded sensor data such as digital imagery

4.9. Transformation grid

grid whose direct positions are given by some transformation algorithm not further specified in this standard

Chapter 5. 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 <http://www.opengis.net/spec/ogcapi-coverages-1/1.0>.

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

5.2. Examples

Most of the examples provided in this standard are encoded in JSON. JSON was chosen because it is widely understood by implementers and easy to include in a text document. This convention should NOT be interpreted as a requirement that JSON must be used. Implementors are free to use any format they desire as long as there is a Conformance Class for that format and the API advertises its support for that Conformance Class.

5.3. Schema

JSON Schema is used throughout this standard to define the structure of resources. These schema are typically represented using YAML encoding. This convention is for the ease of the user. It does not prohibit the use of another schema language or encoding. Nor does it indicate that JSON schema is required. Implementations should use a schema language and encoding appropriate for the format of the resource.

5.4. UML Notation

Diagrams using the Unified Modeling Language (UML) adhere to the following conventions:

- UML elements having a package name of “GML” are those defined in the UML model of GML 3.2.1
- UML elements having a package name of “SWE Common” are those defined in the UML model of SWE Common 2.0
- UML elements not qualified with a package name, or with “CIS”, are those defined in this standard.

Further, in any class where an attribute name or association role name is identical to a name in some superclass the local definition overrides the superclass definition.

5.5. Namespace Prefix Conventions

UML diagrams and XML code fragments adhere to the namespace conventions shown in [Table 2](#).

The namespace prefixes used in this document are not normative and are merely chosen for convenience. The namespaces to which the prefixes correspond are normative, however.

Whenever a data item from a CIS-external namespace is referenced this constitutes a normative dependency on the data structure imported together with all requirements defined in the namespace referenced.

Table 2. Namespace mapping conventions

UML prefix	GML prefix	Namespace URL	Description
CIS	cis	http://www.opengis.net/cis/1.1	Coverage Implementation Schema 1.1
CIS10	cis10	http://www.opengis.net/gmlcov/1.0	Coverage Implementation Schema 1.0
GML	gml	http://www.opengis.net/gml/3.2	GML 3.2.1
GML33	gml33	http://www.opengis.net/gml/3.3	GML 3.3
SWE Common	swe	http://www.opengis.net/swe/2.0	SWE Common 2.0
SML	sml	http://www.opengis.net/sensorml/2.0	SensorML 2.0

5.6. Link relations

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

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.
- **item**: The target IRI points to a resource that is a member of the collection represented by the context IRI (used by the templated link pointing to coverage tiles in a coverage tileset).
- **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 specified in the *Two Dimensional Tile Matrix Set and Tileset Metadata* standard are used:

- <http://www.opengis.net/def/rel/ogc/1.0/tiling-scheme>: The target IRI points to a resource that describes the TileMatrixSet according to the 2D-TMS standard.
- <http://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).
- <http://www.opengis.net/def/rel/ogc/1.0/geodata>: The target IRI points to a resource

representing a collection of geospatial data.

The following link relation types specified in the *OGC API - Tiles* standard are used:

- <http://www.opengis.net/def/rel/ogc/1.0/tilesets-coverage>: The target IRI points to a resource that describes how to provide tile sets of the context resource in coverage format.
- <http://www.opengis.net/def/rel/ogc/1.0/tiling-schemes>: The target IRI points to a resource that lists one or more TileMatrixSets according to the 2D-TMS standard.

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

- <http://www.opengis.net/def/rel/ogc/1.0/coverage>: The target IRI points to a resource representing the coverage, including all its components (domain set, range type, range set, metadata).
- <http://www.opengis.net/def/rel/ogc/1.0/coverage-schema>: The target IRI points to a resource representing the schema of the coverage fields or properties, including the title, type and description of each field, as well as semantic annotations. This corresponds to the *range type* of the Coverage Implementation Schema (CIS). However, an 'application/json' representation of such target resource is expected to be described using JSON Schema rather than CIS JSON.

Used in combination with *OGC API - Features Part 1: Core* or *OGC API - Common Part 1: Core*, other link relation types will be used, including:

- <http://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 - Features Part 1: Core* or *OGC API - Common Part 2: Geospatial data*, other link relation types will be used, including:

- <http://www.opengis.net/def/rel/ogc/1.0/data>: Refers to the list of collections available for a dataset.
- <http://www.opengis.net/def/rel/ogc/1.0/data-meta>: The target IRI points to a resource representing general metadata for the collection of geospatial data (e.g., ISO-19115 — not the domain/application metadata as defined in CIS).

Each resource representation includes an array of links. Implementations are free to add additional links for all resources provided by the Web API.

5.7. Use of HTTPS

For simplicity, this document in general 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." In fact, most servers are expected to use [HTTPS](#), not [HTTP](#).

Chapter 6. Overview

6.1. General

OGC API - Coverages - Part 1: Core, hereafter sometimes referred to as the *Coverages API*, specifies the fundamental API building blocks for retrieving coverage data. The spatial data community uses the term *coverage* for homogeneous collections of values located in space/time such as: spatio-temporal sensor, image, simulation, and statistical data. The *W3C/OGC Spatial Data on the Web Best Practices* provide an overview of [Coverages: describing properties that vary with location \(and time\)](#).

The OGC API family of standards enable access to resources using the HTTP protocol and its associated operations (GET, PUT, POST, etc.). *OGC API - Common* defines a set of capabilities which are shared by several OGC API standards. Some of these other OGC API standards define capabilities to access data using a particular mechanism tailored to a specific resource type. These standards can be integrated within the framework defined by *OGC API - Common*, including the concept of a landing page, API definition and conformance declaration (*Part 1*), as well as providing access to one or more collections of geospatial data (*Part 2*). OGC API standards also re-use common Web API building blocks such as resource paths (including associated path parameters), response definitions for specific representations, schemas for request payloads and responses, or query parameters. This common framework ensures consistency and allow for the integration of multiple OGC API standards within a single Web API implementation.

This *OGC API - Coverages* standard defines an API with two goals:

- Provide access to **Coverages** conformant to the [OGC Abstract Specification Topic 6: Schema for coverage geometry and functions](#).
- Provide functionality comparable to that of the [OGC Web Coverage Service \(WCS\) standard](#).

This API standard allows efficient access to multidimensional geospatial data cubes, particularly when additional optional requirements classes complementing the functionality defined in the "Core" requirements class allowing to subset the fields, area, time and resolution of interest, are implemented.

This data cube functionality can be further extended by integrating it with advanced processing capabilities as defined by *OGC API - Processes - Part 1: Core* and *OGC API - Processes - Part 3: Workflows and Chaining (draft)* (the *collection input* and *collection output* requirements classes in particular).

A future part extension to this standard is also foreseen for deriving new coverage fields based on a simple expression language such as the [OGC Common Query Language \(CQL2\)](#), together with standardized mathematical and aggregation functions.

Resources exposed through an OGC API may be accessed through a Universal Resource Identifier (URI). URIs are composed of three sections:

- Dataset distribution API: The endpoint corresponding to a dataset distribution, where the landing page resource as defined in *OGC API - Common - Part 1: Core* is available (subsequently

referred to as Base URI or `{root}`)

- Access Paths: Unique paths to Resources
- Query: Parameters to adjust the representation of a Resource or Resources, such as requesting a particular subset or downsampled resolution of the data

Access Paths are used to build resource identifiers. Most of these paths are fixed, with the exception of the *OGC API - Common - Part 1 /api* service description / documentation.

All resources are also accessible by following links starting from the landing page, with a specific link relation type used from one resource to another, as detailed in [Link relations](#).

The following table summarizes the resource paths defined in this standard, as well as those inherited from *OGC API - Common*.

Table 3. *Coverages API Resources*

Resource URI	Description
OGC API - Common - Part 1	
<code>{root}/</code>	Landing page for this dataset distribution
<code>{root}/api</code>	API description (e.g., Open API Specification) and documentation (e.g., HTML)
<code>{root}/conformance</code>	Declaration of supported conformance classes
OGC API - Common - Part 2 (extended with additional properties for the Coverages API)	
<code>{root}/collections</code>	The list of all collections available, some or all of which may be accessible using this <i>Coverages API</i> . Each of these collections contains a minimal subset of the object collection resource object described immediately below.
<code>{root}/collections/{collectionId}</code>	Description for the collection with the unique identifier <code>{collectionId}</code> , which may be accessible as a coverage. The resource includes elements such as an <code>id</code> , <code>title</code> , <code>description</code> , available <code>crs</code> and <code>extent</code> . This <code>extent</code> describes the domain of the coverage for each dimension, including the overall envelope, detailed sub-intervals where data is available, and/or a regular or irregular <code>grid</code> . This object also includes links to resources pertaining to this collection. For coverages, a link to the record schema described below will be included. This resource is comparable to a WCS DescribeCoverage response, with the exception that the schema, corresponding to Coverage Implementation Schema (CIS) <i>range type</i> , needs to be retrieved separately.
OGC API - Coverages - Part 1	

Resource URI	Description
<code>{root}/collections/{collectionId}/schema</code>	Returns the schema for the coverage fields or properties of values available at each direct position. At minimum, a <i>JSON Schema</i> representation of this resource is available. This resource is comparable to the CIS <i>range type</i> portion of the WCS DescribeCoverage response, and is retrieved separately from the collection description to accommodate more complex record schemas including several record fields and/or detailed semantic annotations.
<code>{root}/collections/{collectionId}/coverage</code>	Returns the coverage data, including any self-describing information (such as the <i>domain set</i> , <i>range type</i> and <i>metadata</i> components in addition to the <i>range set</i> of CIS). This resource is comparable to a WCS GetCoverage response.
OGC API - Tiles - Part 1	
<code>{root}/collections/{collectionId}/coverage/tiles</code>	Returns the list of tilesets available for this coverage.
<code>{root}/collections/{collectionId}/coverage/tiles/{tileSetId}</code>	Returns an individual coverage tileset for a particular 2D Tile Matrix Set
<code>{root}/collections/{collectionId}/coverage/tiles/{tileSetId}/{tileMatrix}/{tileRow}/{tileCol}</code>	Returns an individual coverage tile for a particular 2D Tile Matrix Set, tile matrix, tile row and tile column
OGC API - Coverages - Part 1 (Scenes requirements class)	
<code>{root}/collections/{collectionId}/scenes</code>	Returns the list of scenes available for this coverage (for multi-scenes coverages, when the <i>Scenes</i> requirement class is supported)
<code>{root}/collections/{collectionId}/scenes/{sceneId}</code>	Returns the scene metadata for an individual scene
<code>{root}/collections/{collectionId}/scenes/{sceneId}/coverage</code>	Returns the coverage data for an individual scene
OGC API - Tiles - Part 1 (for multi-scene coverage)	
<code>{root}/collections/{collectionId}/scenes/{sceneId}/coverage/tiles</code>	Returns the list of tilesets available for this scene's coverage
<code>{root}/collections/{collectionId}/scenes/{sceneId}/coverage/tiles/{tileSetId}</code>	Returns an individual scene coverage tileset for a particular 2D Tile Matrix Set
<code>{root}/collections/{collectionId}/scenes/{sceneId}/coverage/tiles/{tileSetId}/{tileMatrix}/{tileRow}/{tileCol}</code>	Returns an individual scene coverage tile for a particular 2D Tile Matrix Set, tile matrix, tile row and tile column

Where:

- `{root}` = URI of the landing page for the API distributing the dataset
- `{collectionId}` = an identifier for a specific coverage (collection)

Chapter 7. Requirements Class "Core"

7.1. Overview

The "Core" Requirement Class of the *Coverages API* defines the mandatory requirements for any implementation of a Web API claiming conformance with this *OGC API - Coverages* standard. All other requirement classes are optional.

The requirement class depends on *OGC API - Common - Part 2: Geospatial data*, which defines how to list available collections of geospatial data and associate access mechanisms for these collections.

This *Coverages API* is presented as one such access mechanism for collections of geospatial data which can be modeled as a coverage, defined a function returning homogeneous values for any direct position part of its domain.

This requirement class extends the `/collections` and `/collections/{collectionId}` resources defined by *OGC API - Common - Part 2* with additional information for collections accessible using this *Coverages API*. Each collection accessible as a coverage needs to include a link to the `/collections/{resourceId}/coverage` resource using the `[ogc-rel:coverage]` link relation type, which is used to retrieve the coverage data.

Each collection must also include a link to the `/collections/{collectionId}/schema` resource, using the `[ogc-rel:coverage-schema]` relation type, providing a schema of the fields of the coverage supporting a JSON Schema representation. This schema provides equivalent functionality to the Coverage Implementation Schema (CIS) *range type*, including for each field (the measured or observed property, such as a satellite imagery band, or sea surface temperature) an identifier (the property key), a `title`, a `description`, and a JSON Schema `type`. Semantic annotations may also be included providing additional useful information.

The `extent` property of the collection description is also restricted to use a uniform schema for additional dimensions beyond spatial and temporal. As with the `temporal` dimension, and similar to the `bbox` of the `spatial` dimension, each additional dimension must include an `interval` property to specify at minimum the overall envelope of the dimension, and optionally sub-intervals where data is available, if the overall interval is sparsely populated. A `grid` property for these dimensions also allows describing the `cellsCount` of the grid, as well as a cell `resolution` for regular grids, or a list of individual `coordinates` for irregular grids using. This extent information provides equivalent information to the CIS *domain set* for most use cases. For point cloud data, a separate detailed description of the domain is not practical, and a subset of the adomain is instead retrieved together with the data, as with the common LAS and LASZip point cloud encodings. In this case, only the overall interval, and optionally detailed sub-intervals where data is available, can be provided instead within the collection extent.

Requirements Class	
http://www.opengis.net/spec/ogcapi-coverages-1/1.0/req/core	
Target type	Web API
Dependency	http://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections

7.2. Requirements

7.2.1. Extended collection list response (`/collections`)

Requirement 1	<code>/req/core/collection-list</code>
A	For every collection accessible using the <i>Coverages API</i> , the content of the elements of <code>collections</code> for the <code>/collections</code> resource SHALL include in its <code>links</code> array a link to a coverage resource at <code>/collections/{collectionId}/coverage</code> using the link relation type <code>[ogc-rel:coverage]</code> .
B	At least one collection SHALL be accessible as a coverage.

7.2.2. Extended collection description response (`/collections/{collectionId}`)

Requirement 2	<code>/req/core/collection-description</code>
A	For every collection accessible using the <i>Coverages API</i> , the collection description resource SHALL include in its <code>links</code> array a link to a coverage resource at <code>/collections/{collectionId}/coverage</code> using the link relation type <code>[ogc-rel:coverage]</code> .
B	Each of these collection resources SHALL also include in its <code>links</code> array a link to a coverage schema at <code>/collections/{collectionId}/schema</code> using the link relation type <code>[ogc-rel:coverage-schema]</code> .
C	Each of these collection resources SHALL use the uniform additional dimensions schema to describe its domain as part of the <code>extent</code> property of that collection. This implies using an <code>interval</code> array property to describe the overall envelope and optionally sub-intervals for sparsely populated data. The first element of the array is the overall envelope, while any additional elements detail where data is available within that overall interval. Each of these elements are themselves an array of start and end bound values.
D	Each additional dimension SHALL include in a <code>crs</code> , <code>vrs</code> , or <code>trs</code> property a URI or safe CURIE corresponding to the CRS associated with that dimension.
E	The name (JSON dictionary key) of each additional dimension SHALL correspond to the axis abbreviation for the CRS of that axis, to valid axis names for use with query parameters defined in optional requirements classes such as <code>subset</code> and <code>scale-axes</code> , as well as to axis names in the coverage response, if applicable.

F	If a particular dimension of the coverage is gridded, the description of that dimension in the extent SHALL describe this grid using the grid property, providing a cellsCount , as well as a resolution for regular grid or a set of coordinates for irregular grids.
G	For data referenced in relation to planet Earth, the coordinates in the bbox of the spatial dimensions of the extent SHALL be provided either as WGS 84 longitude/latitude (http://www.opengis.net/def/crs/OGC/1.3/CRS84) or as WGS 84 longitude/latitude/ellipsoidal height (http://www.opengis.net/def/crs/OGC/0/CRS84h).
H	The storageCRS property SHALL be set to the native CRS of the coverage, also correspondin to the default output CRS (used when the crs parameter is not specified and/or the CRS requirements class is not supported).

NOTE A schema for this extended collection resource is available at <https://schemas.opengis.net/ogcapi/tiles/part1/1.0/openapi/schemas/common-geodata/collectionInfo.yaml>.

NOTE A property indicating for which planetary body a CRS is applicable is being considered as part of CRS definitions, which will facilitate validating this requirement.

TODO: Discuss the ability to include a **Well Known Text** definition and/or **PROJ-JSON** definition of the CRS instead of a URI.

7.2.3. Collection schema response (**/collections/{collectionId}/schema**)

Requirement 3	/req/core/coverage-schema
A	The API SHALL support the HTTP GET operation at the path /collections/{collectionId}/schema to retrieve the schema of the coverage fields.
B	A successful execution of the operation SHALL be reported as a response with a HTTP status code 200 .
C	The content of that response SHALL include a list of the fields for the coverage, including an identifier (the property key), a title , a type and an optional description .

NOTE A meta-schema for JSON Schema used as part of the *2D Tile Matrix Set and Tileset Metadata standard* suitable for this resource is available at <https://schemas.opengis.net/ogcapi/tiles/part1/1.0/openapi/schemas/tms/propertiesSchema.yaml> . This resource should also match the draft *OGC API - Features - Part 5: Schemas*.

7.2.4. Coverage data retrieval (`/collections/{collectionId}/coverage`)

Requirement 4	<code>/req/core/get-coverage</code>
A	The implementation SHALL support the HTTP GET operation at the path <code>/collections/{collectionId}/coverage</code> to retrieve the coverage.
B	A successful execution of the operation SHALL be reported as a response with a HTTP status code <code>200</code> .
C	The content of that response SHALL be the coverage data, along with the supported self-description capabilities of the negotiated content type.
D	The response SHALL be encoded using the format(s) negotiated through the HTTP protocol.
E	If no format is negotiated, then the response SHALL be encoded using the format associated with the media type described in the link object which links to this resource, contained within the coverage (collection) resource.
Permission 1	<code>/per/core/cis-metadata</code>
A	The implementation MAY omit the CIS domain-specific metadata by default as part of the coverage response.

Chapter 8. Requirements Class "Subsetting"

8.1. Overview

The "Subsetting" Requirements Class specifies the `subset`, `bbox` and `datetime` parameters for coverage resource, allowing to retrieve only a subset of the coverage along one or more dimensions. With the `subset` and `datetime` parameters, a subset operation may be a trimming operation (preserving dimensionality) or slicing operation (reducing dimensions), depending on whether a single coordinate or an interval is used. Negotiating a response in a format with limited multidimensionality support, such as image-based format like GeoTIFF, also implies a reduced dimensionality.

8.2. Subsetting Examples

See also [examples of subsetting requests](#) in an annex.

Requirements Class	
http://www.opengis.net/spec/ogcapi-coverages-1/1.0/req/subsetting	
Target type	Web API
Dependency	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/req/core

8.3. Requirements

8.3.1. Parameter `bbox`

Requirement 5	<code>/req/subset/rc-bbox-definition</code>
A	The coverage resource SHALL support a <code>bbox</code> parameter representing a spatial area to subset.
B	The bounding box SHALL be provided as four or six numbers, depending on whether the coordinate reference system includes a vertical axis (height or depth): <ul style="list-style-type: none">• 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)

C	If the bounding box consists of four numbers, the coordinate reference system of the values SHALL be interpreted as WGS 84 longitude/latitude (http://www.opengis.net/def/crs/OGC/1.3/CRS84) unless a different coordinate reference system is specified in a parameter <code>bbox-crs</code> .
D	If the bounding box consists of six numbers, the coordinate reference system of the values SHALL be interpreted as WGS 84 longitude/latitude/ellipsoidal height (http://www.opengis.net/def/crs/OGC/0/CRS84h) unless a different coordinate reference system is specified in a parameter <code>bbox-crs</code> .
E	Only the portion of the coverage within the specified bounding box SHALL be part of the response, performing a trim subsetting operation.
F	If a <code>bbox</code> parameter is specified requesting a coverage without any spatial dimension, the parameter SHALL either be ignored, or a 4xx client error generated.
G	For a bounding box specified in a geographic CRS, a bounding box where the first longitude value is larger than the second longitude value SHALL be interpreted as crossing the anti-meridian.

The bounding box for WGS 84 longitude/latitude is, in most cases, the sequence of minimum longitude, minimum latitude, maximum longitude and maximum latitude. However, in cases where the box spans the anti-meridian (180th meridian) the first value (west-most box edge) is larger than the third value (east-most box edge).

Example 1. The bounding box of the New Zealand Exclusive Economic Zone

The bounding box of the New Zealand Exclusive Economic Zone in WGS84 (from 160.6°E to 170°W and from 55.95°S to 25.89°S) would be represented in JSON as [160.6, -55.95, -170, -25.89] and in a query as `bbox=160.6,-55.95,-170,-25.89`.

Note that the server should return an error if a latitude value of 160.0 is used.

If the vertical axis is included, the third and the sixth number are the bottom and the top of the 3-dimensional bounding box.

8.3.2. Parameter `bbox-crs`

Requirement 6	<code>/req/subsetting/bbox-crs</code>
---------------	---------------------------------------

A	The coverage resource SHALL support a <code>bbox-crs</code> parameter specifying the CRS used for the <code>bbox</code> parameter.
B	For Earth centric data, the implementation SHALL support http://www.opengis.net/def/crs/OGC/1.3/CRS84 as a value.
C	If the <code>bbox-crs</code> is not indicated http://www.opengis.net/def/crs/OGC/1.3/CRS84 SHALL be assumed.
D	The native CRS (<code>storageCRS</code>) SHALL be supported as a value. Other conformance classes may allow additional values (see <code>crs</code> parameter definition).
E	The CRS expressed as URIs or as safe CURIEs SHALL be supported.
F	If the <code>bbox</code> parameter is not used, the <code>bbox-crs</code> SHALL be ignored.

8.3.3. Parameter `datetime`

Requirement 7	<code>/req/subsetting/datetime</code>
A	<p>The implementation SHALL support a <code>datetime</code> parameter expressed corresponding to either a date-time instant or a time interval, conforming to the following syntax (using ABNF):</p> <pre style="background-color: #f0f0f0; padding: 10px;"> 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 </pre> <p>The syntax of <code>date-time</code> is specified by RFC 3339, 5.6.</p>
B	Only the portions of the coverage within the specified interval SHALL be part of coverage response, performing a trim operation for an interval or a slicing operation for an instant.
C	The implementation SHALL support a double-dot (<code>..</code>) or an empty string for the start/end as indicating an unbounded or half-bounded interval (only having a start or end).

D	If a <code>datetime</code> parameter is specified requesting a coverage without any temporal dimension, the parameter SHALL either be ignored, or a 4xx client error generated.
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.

Example 2. A date-time

February 12, 2018, 23:20:52 GMT:
`datetime=2018-02-12T23:20:52Z`

For resources with a temporal property that is a timestamp (like `lastUpdate`), a date-time value would match all resources where the temporal property is identical.

For resources with a temporal property that is a date or a time interval, a date-time value would match all resources where the timestamp is on that day or within the time interval.

Example 3. 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`

8.3.4. Parameter `subset`

Requirement 8	<code>/req/subsetting/subset</code>
----------------------	-------------------------------------

A	<p>The operation SHALL support a parameter subset with the following characteristics (using an Extended Backus Naur Form (EBNF) fragment):</p> <pre style="background-color: #f0f0f0; padding: 10px;"> 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).</pre>
B	<p>The axis name SHALL correspond to one of the axis of the Coordinate Reference System (CRS) of the target resource or else return a 400 status code.</p>
C	<p>If the <i>intervalOrPoint</i> values fall entirely outside the range of valid values defined for the identified axis, a 204 status code SHALL be returned</p>
D	<p>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.</p>
E	<p>Only that part of the coverage that falls within the bounds of the subset expression SHALL be returned.</p>
F	<p>If a lower limit of the subset expression is populated with an asterix "*", then the minimum extent of the resource along that axis SHALL be selected.</p>
G	<p>If a upper limit of the subset expression is populated with an asterix "*", then the maximum extent of the resource along that axis SHALL be selected.</p>

NOTE

When the *intervalOrPoint* 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 coverage resulting from the subset. For subsetting on the range set, and for coverage media types with no geo-referencing mechanisms (e.g. PNG), NO_DATA values or transparency should be used. If a georeferencing mechanism is available within the negotiated media type, the service could decide whether to use NO_DATA values or simply return the properly geo-referenced values within the domain set.

8.3.5. Parameter `subset-crs`

Requirement 9	<code>/req/subsetting/subset-crs</code>
A	The coverage retrieval operation SHALL support a parameter <code>subset-crs</code> with the characteristics identifying the CRS in which the <code>subset</code> parameter is specified with a URI or safe CURIE.
B	For Earth centric data, http://www.opengis.net/def/crs/OGC/1.3/CRS84 as a value SHALL be supported.
C	If the <code>subset-crs</code> is not indicated, http://www.opengis.net/def/crs/OGC/1.3/CRS84 SHALL be assumed.
D	The native CRS (<code>storageCRS</code>) SHALL be supported as a value. Other requirements classes may allow additional values (see <code>crs</code> parameter definition).
E	CRSs expressed as URIs or as safe CURIEs SHALL be supported.
F	If no <code>subset</code> parameter referring to an axis of the CRS is used, the <code>subset-crs</code> SHALL be ignored.

8.3.6. Permission for Slicing Sparse Dimensions

Permission 2	<code>/per/subsetting/slice-sparse-dimension</code>

A	<p>The empty portions in a coverage resulting from a slice operation on an axis (e.g. time), combined with a trimming operations on other axes (e.g. latitude and longitude) which would either be empty or not cover the full extent of the trim operation MAY be filled with data values from the same trim operation combined with a slicing operation on a different value of the slicing axis which would return non-empty values. For example, the closest or last previous time for which data is available for a certain geospatial extent may be returned. An Earth Observation use case for this permission is to allow retrieving a slice of the last available imagery on or before a certain date, taking into account that a certain geographic area may only be observed every few days.</p>
B	<p>This permission applies to both explicit slice operations using subset, as well as to implicit slicing from requesting an output format only supporting a lower dimensionality than the data (e.g. requesting a 2D image from a 3D coverage as PNG or GeoTIFF).</p>
C	<p>A query parameter defined by a custom or standardized extension MAY be made available to enable, disable or alter that behavior.</p>

Chapter 9. Requirements Class "Scaling"

9.1. Overview

The "Scaling" Requirements Class defines the `scale-factor`, `scale-axes` and `scale-size` parameters for retrieving coverage data at a specific resolution.

9.2. Scaling Examples

See also [examples of scaling requests](#) in an annex.

Requirements Class	
http://www.opengis.net/spec/ogcapi-coverages-1/1.0/req/scaling	
Target type	Web API
Dependency	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/req/core

9.3. Requirements

9.3.1. Parameter `scale-factor`

Requirement 10	<code>/req/coverage-scaling/factor</code>
A	The operation SHALL support a <code>scale-factor</code> numeric parameter for the coverage resource.
B	For each axis, the returned coverage SHALL contain the number of original sampled values, divided by the specified factor.

9.3.2. Parameter `scale-axes` parameters

Requirement 11	<code>/req/coverage-scaling/axes</code>
-----------------------	--

A	<p>The operation SHALL support a parameter <code>scale-axes</code> for the coverage resource with the following characteristics (using an Extended Backus Naur Form (EBNF) fragment):</p> <pre style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> ScalingSpec: "scale- axes"=axisName({factor})[,axisName({factor})]* axisName: {text} Where: {factor} is an integer or floating-point number, and {axisName} refers to a valid axis for the coverage</pre>
B	<p>The implementation SHALL support the name of axes as per the extent of the collection description, which should also correspond to those defined to the abbreviations defined by the CRS (as defined by the <code><gml:axisAbbrev></code> tags), and to the <code>axisLabels</code> defined in the <i>domain set</i> for a CIS representation, and return a 400 error for an unrecognized axis name.</p>
C	<p>For each axis specified, the returned coverage SHALL contain the number of original sampled values, divided by the <i>factor</i> specified for that axis, and the original number of sample values for unspecified axes.</p>

9.3.3. Parameter `scale-size`

Requirement 12	<code>/req/coverage-scaling/size</code>
A	<p>The operation SHALL support a parameter <code>scale-size</code> for the coverage resource with the following characteristics (using an Extended Backus Naur Form (EBNF) fragment):</p> <pre style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> ScalingSpec: "scale- size"=axisName({number})[,axisName({number})]* axisName: {text} Where: {number} is an integer or floating-point number, and {axisName} refers to a valid axis for the coverage</pre>

B	The implementation SHALL support the name of axes as per the extent of the collection description, which should also correspond to those defined to the abbreviations defined by the CRS (as defined by the <code><gml:axisAbbrev></code> tags), and to the <code>axisLabels</code> defined in the <i>domain set</i> for a CIS representation, and return a 400 error for an unrecognized axis name.
C	The returned coverage SHALL contain exactly the specified number of sample values along each axis which is specified by the <code>scale-size</code> parameter, and the original number of sample values for unspecified axes.
D	When <code>scale-size</code> is used together with subsetting, the requested number of samples SHALL be mapped to the subset dimensions specified, even if those dimensions fall partially outside the extent of the coverage.

9.3.4. Downsampling permission

Permission 3	<code>/per/coverage-scaling/downsampling</code>
A	In the case where a response to a request without a <code>scale-factor</code> , <code>scale-axes</code> or <code>scale-size</code> parameters would be larger than an advertised server limit, an implementation MAY automatically downsample the coverage to a suitable resolution instead of returning a 4xx client error.

NOTE

A client retrieving a coverage from an implementation advertising support for this "Scaling" requirements class should explicitly use `scale-factor=1` if it wants to ensure retrieving the coverage in its native resolution and prefers receiving an error instead of a downsampled version of the data.

Chapter 10. Requirements Class "Field Selection"

10.1. Overview

The "Field Selection" Requirements Class defines the `properties` parameter for selecting a subset of the fields to retrieve from the coverage resource (`/collections/{collection_id}/coverage`). The available fields for the coverage, for example individual imagery bands, are defined in the schema resource (corresponding to the CIS *range type*). The order of the fields as listed in the value of the parameter also determines the order in which those fields will be ordered in the response, if the negotiated format has such a concept of field order.

10.2. Field Selection Examples

See also [examples of field selection requests](#) in an annex.

Requirements Class	
http://www.opengis.net/spec/ogcapi-coverages-1/1.0/req/fieldselection	
Target type	Web API
Dependency	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/req/core

10.3. Requirements

10.3.1. Parameter `properties`

Requirement 13	<code>/req/fieldselection/properties</code>
A	<p>The operation SHALL support a parameter <code>properties</code> with the following characteristics (using an Extended Backus Naur Form (EBNF) fragment):</p> <pre>propertiesSpec: "properties"=field[,field]* field: {fieldName} {fieldIndex} "*" fieldName: {text} fieldIndex: {number}</pre> <p>Where:</p> <ul style="list-style-type: none"><code>{number}</code> is an integer number, and<code>{text}</code> is some general ASCII text.

B	The implementation SHALL support selecting a field using the identifier corresponding to the top-level property keys of the coverage fields schema, and return 400 status code for an unrecognized selected field.
C	The implementation SHALL support selecting a field using an index corresponding to an <code>x-ogc-field-index</code> property in the coverage fields schema, if present, and return a 400 status code for an unrecognized field index.
D	The coverage fields schema SHALL not contain any potential conflict between a numeric band name and a band index.
E	The implementation SHALL interpret an <code>*</code> (asterisk) at the beginning or at the end of the comma-separated list as all other fields before or after the listed fields, respectively, as per the numeric <code>x-ogc-field-index</code> order.
F	Only the selected fields of the coverage SHALL be returned from coverage request.
G	If the coverage response can self-describes its list of fields (as with the CIS <i>range type</i>), the field description SHALL be adjusted accordingly to the new selected fields.
H	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.

Chapter 11. Requirement Class "Coordinate Reference System"

11.1. Overview

The "Coordinate Reference System" requirement class specifies a `crs` parameter allowing a client to select an output CRS (equivalent to `OUTPUTCRS` in WCS 2.0).

As defined in this standard, a coverage is conceptually associated with native CRS (also called a *storage CRS*) which is indicated in the collection description. This native CRS is the default output CRS if no `crs` parameter is specified, and the only supported output CRS without support for this requirements class. The `Content-Crs:` header (see [Requirements Class "Core"](#)) specifies the CRS used in the response to avoid any confusion as to the CRS of a response.

Requirements Class Coordinate Reference System	
http://www.opengis.net/spec/ogcapi-coverages-1/1.0/req/crs	
Target type	Web API
Dependency	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/req/core

11.2. Requirements

11.2.1. Parameter `crs`

A successful GET response is described in the Coverages API core class (<http://www.opengis.net/spec/ogcapi-coverages-1/1.0/conf/core>).

Requirement 14	<code>/req/crs/crs-definition</code>
A	<p>The coverage retrieval operation SHALL support a parameter <code>crs</code> with the characteristics defined in the OpenAPI Specification 3.0 fragment</p> <pre>crs: name: crs in: query description: A coordinate reference system of the coverage response. A list of all supported CRS values can be found under the collection metadata. required: false schema: type: string example: http://www.opengis.net/def/crs/OGC/1.3/CRS84</pre>

B	Any of the CRSs listed in the collection (or collections) description SHALL be supported. If the list of supported CRS is not present, only http://www.opengis.net/def/crs/OGC/1.3/CRS84 SHALL be supported.
C	If the spatial subsetting requirements class is supported, the <code>bbox-crs</code> and the <code>subset-crs</code> SHALL additionally support value specified in the <code>crs</code> parameter.
D	CRS expressed as URIs or as safe CURIEs SHALL be supported.
E	The content of that response SHALL be consistent with the requested CRS.
F	If the parameter value for <code>crs</code> is not valid for requested coverage, the status code of the response SHALL be 400.

NOTE When no `crs` parameters are specified, please refer to the Coverages API core conformance class to know about the default `crs`.

NOTE A CURIE `{authority}[-{objectType}]:{id}` would map to the following OGC URI: <http://www.opengis.net/def/{objectType}/{authority}/0/{id}>. If `-{objectType}` is missing, the default object type is `crs`.

See how to determine the native (storage) CRS for the coverage in [\[rc_core\]](#) respectively.

NOTE The default CRS of the BBOX is <http://www.opengis.net/def/crs/OGC/1.3/CRS84> but the default CRS of the coverage is the native (storage) CRS

Permission 4	<code>/per/crs/crs-curie</code>
A	Un-safe CURIE without square brackets MAY be supported.

NOTE This makes the notation compatible with WCS.

Chapter 12. Requirements Class "Coverage Tiles"

12.1. Overview

The "Coverage Tiles" Requirements Class defines how to combine the *OGC API - Tiles* building blocks with the *Coverages API* to request coverage tiles. Coverages parameters such as for subsetting additional dimensions not covered by the Tiles 2D Tile Matrix Set (such as time or elevation), or for range subsetting, can be used together with the path for requesting individual tiles.

12.2. Coverage Tiles Examples

See also [examples of coverage tile requests](#) in an annex.

Requirements Class	
http://www.opengis.net/spec/ogcapi-coverages-1/1.0/req/coverage-tiles	
Target type	Web API
Dependency	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/req/core
Dependency	http://www.opengis.net/spec/ogcapi-tiles-1/1.0/conf/geodata-tilesets

12.3. Requirements

12.3.1. Coverage tilesets (`/collections/{collectionId}/coverage/tiles`)

The capability to request tiles is defined in the following Requirement:

Requirement 15	<code>/req/coverage-tiles/tilesets</code>
A	The coverage SHALL have an associated list of at least one coverage tilesets accessible at <code>.../{collectionId}/coverage/tiles</code> , conforming to OGC API - Tiles - Part 1: Core
B	The coverage's collection description document at <code>.../{collectionId}</code> SHALL include a link with relation type http://www.opengis.net/def/rel/ogc/1.0/tilesets-coverage linking to that resource listing available coverage tilesets
C	This list of coverage tilesets SHALL include a link to the full descriptions of these tilesets accessible at <code>.../{collectionId}/coverage/tiles/{tileMatrixSetId}</code>

D	The tiles making up the tilesets SHALL be accessible from ... <code>/{collectionId}/coverage/tiles/{tileMatrixSetId}/{tileMatrix}/ {tileRow}/{tileCol}</code>
E	The responses for the list of tilesets, tileset metadata and tiles SHALL comply to all requirements of the OGC API - Tiles specifications
F	The response to a request for an individual tile SHALL return a subset of the coverage trimmed on the axes defined by the 2D Tile Matrix Set to cover the exact geospatial extent of the tile.
G	The response to a request for an individual tile SHALL be scaled down to a number of cells corresponding to the tile's pixel size, plus one if the coverage cells do not fully span the distance between them.

As an example, considering that the *WorldCRS84Quad* Tile Matrix Set has tile pixel sizes defined as 256 x 256, a request for tile

`.../coverage/tiles/WorldCRS84Quad/1/0/0` of a point coverage is equivalent to:

`.../coverage?subset=Lat(0:90),Lon(-180:-90)&scale-size=257,257`

and returns a tile containing one or more slices of 257 x 257 cells, while for an area coverage it is equivalent to:

`.../coverage?subset=Lat(0:90),Lon(-180:-90)&scale-size=256,256`

and returns a tile containing one or more slices of 256 x 256 cells.

Chapter 13. Requirements Class "Scenes"

13.1. Overview

The *Scenes* Requirements Class defines the ability to present separate components of a coverage as individual scenes, in addition to an overall coverage for the collection as a whole. A list of scenes and a resource associated to each individual scene, including a description of its domain, are available, along with the ability to retrieve a coverage for each scene. The native CRS of the coverage for each scene and/or the resolution of the coverage data may differ from that of the overall collection. For example, the native CRS of an individual scene may be a particular UTM zone CRS based on its geospatial extent, whereas a global coverage for the collection is natively available in a WGS84 / EPSG:4326 geographic CRS.

The API may also integrate additional functionality related to scenes, such as for advanced filtering capabilities when listing scenes, as defined in *OGC API - Records Local Resource Catalogues* and/or *SpatioTemporal Asset Catalog (STAC)*, the ability to create new scenes, update and delete existing scenes, or the ability to use scene-level metadata as part of requesting coverage data with a filtering extension.

Requirements Class	
http://www.opengis.net/spec/ogcapi-coverages-1/1.0/req/scenes	
Target type	Web API
Dependency	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/req/core

13.2. Requirements

13.2.1. List of scenes (`/collections/{collectionId}/scenes`)

Requirement 16	<code>/req/scenes/scene-list</code>
A	At least one collection available from the target implementation SHALL consist of multiple scenes.
B	For multi-scenes collections, a GET operation on the <code>/collections/{collectionId}/scenes</code> resource SHALL return a successful response with a HTTP status code <code>200</code> .
C	The JSON response to the GET <code>/collections/{collectionId}/scenes</code> operation SHALL include a <code>scenes</code> property being a JSON array of individual scenes
D	Each element of the array of scenes SHALL include at minimum an <code>id</code> .

Recommendation 1	/rec/scenes/scene-list-filtering
A	A GET operation on <code>/collections/{collectionId}/scenes</code> SHOULD support additional query parameters to filter (<code>filter=</code> , <code>bbox=</code> , <code>datetime=</code> , <code>q=</code>) and/or page (<code>limit=</code>) the list of results, as defined by the <i>OGC API - Records</i> Local Resource Catalogues requirements class and/or SpatioTemporal Asset Catalog (STAC), where the resources in this case are the scenes.
B	Scene resources SHOULD support the relevant record fields as defined in other specifications such as <i>STAC</i> and its extensions, and/or <i>OGC API - Records Core properties</i> .

Permission 5	/rec/scenes/scene-list-post
A	The <code>/collections/{collectionId}/scenes</code> resource MAY additionally support a POST operation, for authenticated clients, allowing to create new scenes, also resulting in a modification to the overall collection coverage.

13.2.2. Individual scene (`/collections/{collectionId}/scenes/{sceneId}`)

Requirement 17	/req/scenes/scene-request
A	A GET operation on <code>/collections/{collectionId}/scenes/{sceneId}</code> SHALL return a successful response with a HTTP status code <code>200</code> .
B	The JSON response to the GET <code>/collections/{collectionId}/scenes/{sceneId}</code> operation SHALL be the object associated with the requested scene.
C	The scene object response SHALL include an <code>id</code> consistent with the requested id and the id in the <code>/scenes</code> response.
D	The scene object response SHALL include a <code>links</code> property including a <code>self</code> link to the scene object, a link to the coverage request end-point (link relation <code>[ogc-rel:coverage]</code>) and a link to the coverage schema (<code>[ogc-rel:coverage-schema]</code>) at <code>/collections/{collectionId}/schema</code> supporting a JSON Schema description of the coverage fields (<i>range type</i>).

E	The scene object response SHALL include an extent property as defined in the Uniform Additional Dimension extent schema.
F	The scene object response SHALL include a crs property defining the list of available output CRSs, and a storageCRS defining the native CRS, which is also the default output CRS for the scene.

Permission 6	/rec/scenes/scene-replace-update
A	The /collections/{collectionId}/scenes/{sceneId} resource MAY additionally support a PUT operation, for authenticated clients, allowing to update an existing scene, also resulting in a modification of the overall coverage.
B	The /collections/{collectionId}/scenes/{sceneId} resource MAY additionally support a DELETE operation, for authenticated clients, allowing to delete an existing scene, also resulting in a modification of the overall coverage.

13.2.3. Scene coverage (

/collections/{collectionId}/scenes/{sceneId}/coverage)

Requirement 18	/req/scenes/scene-coverage
A	A GET operation on the /collections/{collectionId}/scenes/{sceneId}/coverage resource SHALL return a successful response for the scene coverage with HTTP status code 200 .
B	The response to a GET operation on the /collections/{collectionId}/scenes/{sceneId}/coverage resource SHALL return coverage consisting of only the data within the domain of the selected scene.
C	The response for the scene coverage SHALL default to the native (storageCRS) for the scene, which may differ to that of the overall collection.
D	If applicable, the scale-factor and scale-axes response SHALL be based on the resolutions described in the individual scene domain description (extent property), which may differ from the overall collection.

E	The response for the scene coverage SHALL otherwise follow all of the same requirements as defined in the <i>Core</i> requirements class for a collection coverage response.
---	--

13.2.4. Scene coverage tiles

([/collections/{collectionId}/scenes/{sceneId}/coverage/tiles](#))

Requirement 19	/req/scenes/scene-coverage-tiles
A	If the <i>Coverage Tiles</i> requirements class is also supported, tilesets resources at /collections/{collectionId}/scenes/{sceneId}/coverage/tiles SHALL be supported, as defined by that requirement class and <i>OGC API - Tiles</i> .

13.2.5. Scene queryables

Permission 7	/rec/scenes/scene-queryables
A	A coverage request from a multi-scene collection MAY support scene-level filtering capabilities based on scene metadata queryables directly as part of the /collections/{collectionId}/coverage request, for example allowing to include certain scene IDs, or only scenes with a maximum cloud cover, as part of a coverage response.

Chapter 14. Requirements Classes for Encodings

This standard does not mandate any particular encoding or format in which to return the `../coverage` resource. Coverages can be encoded in any suitable data format. However, it does define requirements class for encodings which are expected to be commonly supported in implementations of this standard. These requirements classes include:

- [HTML](#)
- [GeoTIFF](#)
- [netCDF](#)
- [CIS JSON](#)
- [CoverageJSON](#)
- [LAS](#)
- [LASZip](#)
- [PNG](#)
- [JPEG XL](#)
- [JPEG 2000](#)
- [\(Geo\)Zarr](#)

None of these encodings is mandatory. An implementor of this standard may choose to implement none of them, selecting different encodings instead.

The media type used to encode a response to a request shall be determined based on HTTP content negotiation, as specified in *OGC API - Common*. If not using content negotiation, the encoding must follow the media type described in the link to the resource from the collection.

For most efficient usage, a coverage should be available in binary form, for which a number of encodings requirements classes are defined below.

14.1. Media Types

A table of the media types corresponding to each encoding requirements class defined in this standard follows.

Table 4. Media Types for each encoding requirements class

Encoding	media type
HTML	text/html
GeoTIFF	image/tiff; application=geotiff
netCDF	application/x-netcdf

CIS JSON	application/json (TODO: either OGC API will adopt a mechanism for negotiation by profile, or a more specific media type will be registered)
CoverageJSON	application/prs.coverage+json
LAS	vnd.las
LASZip	vnd.laszip
PNG	image/png
JPEG XL	image/jxl
JPEG-2000	image/jp2
(Geo)Zarr	application/x-zarr (TODO: Register this media type?)

14.2. Requirements Class "HTML"

14.2.1. Overview

The HTML requirements class defines at a high level the ability to retrieve an HTML response for the coverage 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, such as embedding a static or interactive view of a coverage.

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	
http://www.opengis.net/spec/ogcapi-coverages-1/1.0/req/html	
Target type	Web API
Dependency	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/req/core
Dependency	HTML5

14.2.2. Requirements

14.2.2.1. Coverage

Requirement 20	/req/html/coverage
A	The coverage resource, as defined in the "Core" requirement class, SHALL support negotiating an HTML response using the <code>text/html</code> media type.

B	The response SHALL be a valid HTML document
C	The response SHALL represent the coverage.

14.3. Requirements Class "GeoTIFF"

14.3.1. Overview

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

14.4. Requirements Class "netCDF"

14.4.1. Overview

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

14.5. Requirements Class "CIS JSON"

14.5.1. Overview

The CIS JSON requirements class defines support for encoding a coverage response according to the [OGC Coverage Implementation Schema \(CIS\) 1.1](#).

The [Coverage Implementation Schema \(CIS\) 1.1](#) defines a logical model and physical encodings (e.g., CIS JSON) which can be used to retrieve the coverage of its components according to this [OGC API - Coverages](#) Web API standard. A CIS JSON representation is required at least for the domain set and the range type, which can both either be embedded within the collection resource, or provided as separate resources.

A high-level view of the CIS data model is provided in [Figure 1](#).

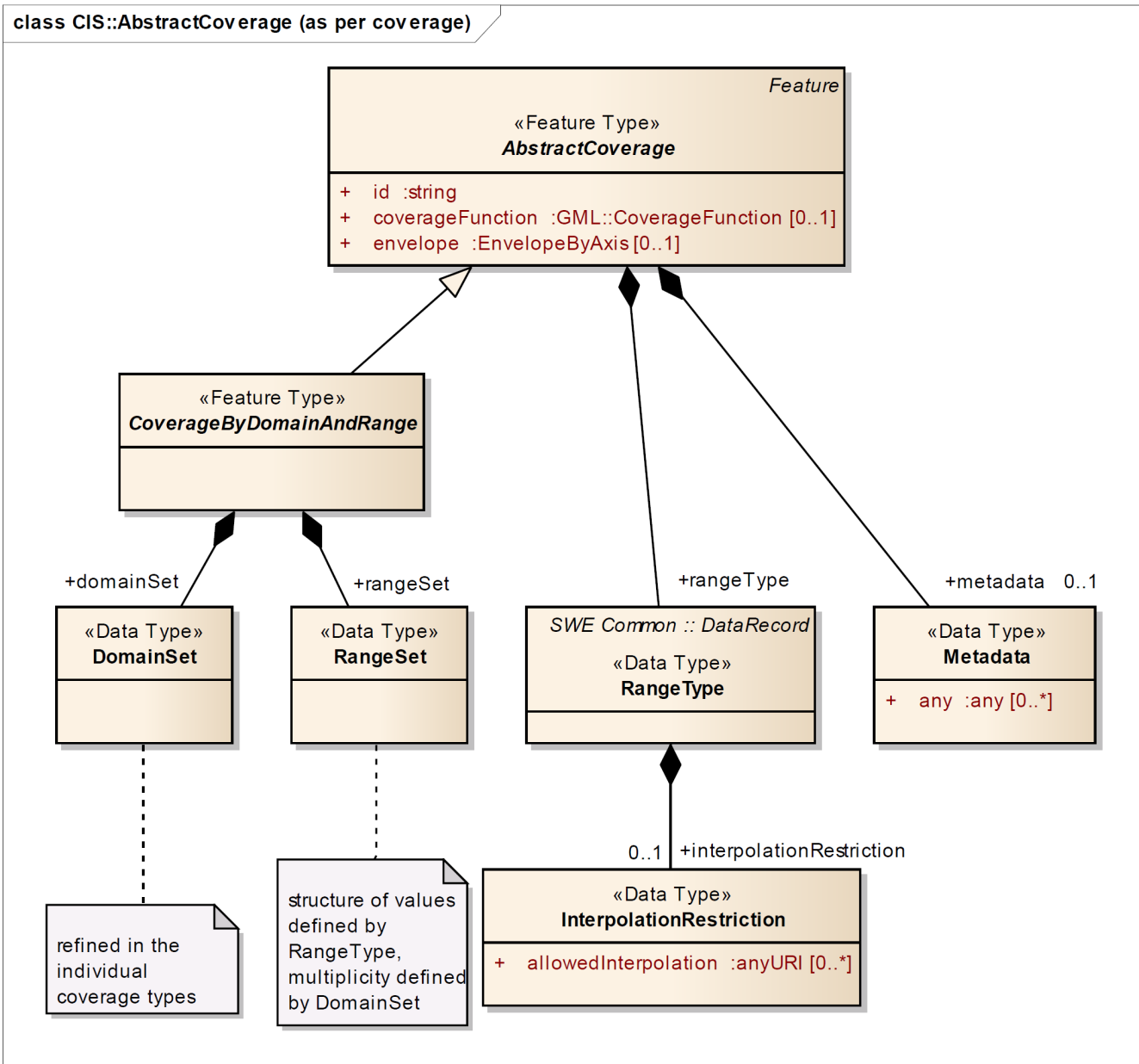


Figure 1. Abstract Coverage

The CIS JSON structures are defined using JSON Schema at <https://schemas.opengis.net/cis/1.1/json/coverage-schema.json>. A draft corrigendum is available at <https://github.com/opengeospatial/coverage-implementation-schema/blob/main/standard/schemas/1.1/json/coverage-schema.json>, with updated examples available at <https://github.com/opengeospatial/coverage-implementation-schema/tree/main/standard/schemas/1.1/json/examples>.

Specifically for the `./coverage` resource, the media type `application/json` is reserved for use with CIS JSON until a dedicated media type is registered.

The CIS Standard defines four aspects of a coverage which are described in separate properties: the domain set, the range type, the rangeset, and the domain-specific metadata. By default, all of these aspects should be returned as part of a coverage response, with the domain-specific metadata being optional if none is available.

This CIS JSON requirements class defines a permission for implementations to support a `profile` query parameter which can be used by a client to retrieve only some of these aspects of the coverage e.g., `profile=domainset,rangetype`. Information equivalent to the CIS `DomainSet` is already

included in the collection description resource `extent` property, whereas information equivalent to the CIS *RangeType* is included in the schema resource associated with the collection resource. This profiling capability also allows clients to retrieve a subset of the domainset when using `profile=domainset` together with one or more subsetting parameters (`subset`, `bbox` or `datetime`).

The CIS standard defines a number of alternative way in which a coverage encoding can be structured. In order to maximize interoperability, this requirements class specifies that the response will be a *CoverageByDomainAndRange* consisting of either a DomainSet described by a *GeneralGridCoverage* (for coverages consisting of one or more dimensions of regular and/or irregular grids) or a *DirectMultiPoint* coverage (for point clouds), excluding a *CoverageByPartitioning* defined by a PartitionSet, unless a prior arrangement (not specified by this Standard) is established with the client.

Similarly, this requirement class also specifies that the CIS JSON response will be self-contained (not linking to data available in separate resources), unless a prior arrangement (not specified by this Standard) is established with the client.

Since the description of the coverage domain and range is easily available at the `/collections/{collectionId}` and `/collections/{collectionId}/schema` resources, implementations and clients are likely to prefer keeping to single-part encodings, and would select a binary encoding such as those defined in the other encoding requirements classes for efficient access.

This requirements class also clarifies how to describe the *DomainSet* of coverages based on whether the *RangeSet* values correspond to cells whose geometry are a single point or an area, ensuring that the extent interval in the collection resource always fully cover the spatial extent described by the coverage and that this extent description is consistent with other OGC API Standards, such as OGC API - Maps (where map pixels always represent an area), which may be made available for the same collection resource.

Requirements Class	
http://www.opengis.net/spec/ogcapi-coverages-1/1.0/req/cisjson	
Target type	Web API
Dependency	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/req/core
Dependency	http://www.opengis.net/spec/CIS/1.1/req/json-coverage

14.5.2. Requirements

14.5.2.1. CIS JSON Coverage representation

Requirement 21	<code>/req/cisjson/coverage</code>
A	The coverage resource, as defined in the Core requirement class, SHALL support negotiating a CIS JSON response using the <code>application/json</code> media type.

B	The response SHALL be a CIS JSON document which validates against the CIS JSON schema coverage-schema.yaml .
C	The coverage object response SHALL be of type <i>CoverageByDomainAndRange</i> (not <i>CoverageByPartitioning</i> , the implementation can instead consider supporting the "Scenes" requirement class to present individual parts of a coverage).

14.5.2.2. Profiling CIS components to include

Recommendation 2	/rec/cisjson/profile
A	The coverage request operation SHOULD support a profile query parameter allowing to select specific one or more CIS coverage components to include (domainset, rangetype, rangeset and/or metadata).
B	If supported, the response SHOULD only contain the selected element(s). For example, <code>../coverage?profile=domainset,rangetype</code> should only return the domain set and range type for the coverage.
C	If both this capability and subsetting are supported, the subsetting SHOULD be applied to both the domainset and the rangeset components of the response.
D	The implementation SHOULD return the metadata when explicitly requested to do so with the profile query parameter.

14.6. Requirements Class "CoverageJSON"

14.6.1. Overview

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

14.7. Requirements Class "LAS"

14.7.1. Overview

The LAS requirements class defines support for encoding a coverage response according to the [OGC LAS Community Standard](#).

This requirements class is particularly well suited for distributing point cloud coverages.

14.8. Requirements Class "LASZip"

14.8.1. Overview

The LASZip requirements class defines support for encoding a coverage response according to the [LASZip](#) compression extension of the [OGC LAS Community Standard](#).

This requirements class is particularly well suited for distributing point cloud coverages.

14.9. Requirements Class "PNG"

14.9.1. Overview

The PNG requirements class defines support for encoding a coverage 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 coverage values, and allowing the implementation to inform the client of the scale factor and offset used for that quantization.

14.10. Requirements Class "JPEG XL"

14.10.1. Overview

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

14.11. Requirements Class "JPEG 2000"

14.11.1. Overview

The JPEG 2000 requirements class defines support for encoding a coverage response according to the [JPEG 2000 ISO/IEC 15444](#), with support for georeferencing with [GMLJP2](#).

14.12. Requirements Class "(Geo)Zarr"

14.12.1. Overview

The (Geo)Zarr requirements class defines support for encoding a coverage response according to the [OGC Zarr Community Standard](#), with eventual support for [GeoZarr](#).

Chapter 15. Requirements class "OpenAPI 3.0"

15.1. Overview

The OpenAPI 3.0 Requirements Class is applicable to OGC API - Coverages as well. So an implementation of *OGC API - Coverages* which supports OpenAPI 3.0 as an API Description format must also comply with the *OGC API - Common oas30* Conformance Class.

In addition, this optional requirements class defines the `.getCoverage` operationId suffix to be used in order to facilitate the identification of coverage resources available from an API.

Requirements Class	
http://www.opengis.net/spec/ogcapi-coverages/1.0/req/oas30	
Target type	Web API
Dependency	http://www.opengis.net/spec/ogcapi-coverages-1/1.0/req/core
Dependency	http://www.opengis.net/spec/ogcapi-common-1/1.0/req/oas30

15.2. Requirements

Requirement 22	<code>/req/oas30/oas-common</code>
A	The OpenAPI api description SHALL use an operationId ending with <code>.getCoverage</code> for coverage resources.

Annex A: Conformance Class Abstract Test Suite (Normative)

NOTE

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

A.1. Conformance Class A

A.1.1. Requirement 1

Test id:	/conf/conf-class-a/req-name-1
Requirement:	/req/req-class-a/req-name-1
Test purpose:	Verify that...
Test method:	Inspect...

A.1.2. Requirement 2

Annex B: Examples (Informative)

B.1. Basic Coverage Retrieval Example

B.2. Subsetting Example

These examples return a subset of a coverage in the negotiated format, including domain set, range type (and metadata if applicable) to the extent than they can be described in that format:

- `{datasetAPI}/collections/{collectionId}/coverage?subset=Lat(40:50),Lon(10:20)` — returns a cutout (trim) from the coverage for the extent between corner coordinates (Lat: 40, Lon: 10) and (Lat: 50, Lon: 20)
- `{datasetAPI}/collections/{collectionId}/coverage?subset=time("2019-03-27")` — returns a coverage slice at the timestamp (if the coverage is has 2D spatial dimensions plus time, the result will be a 2D image with its full spatial extent)

B.3. Scaling Example

- `{datasetAPI}/collections/{collectionId}/coverage?scale-factor=1.5` — returns a coverage re-scaled so as to contain 1.5 times less sample values along all of its axes
- `{datasetAPI}/collections/{collectionId}/coverage?scale-size=Lat(300),Lon(400)` — returns the entire coverage re-scaled to 300 pixels along its latitude axis, and 400 samples along its longitude axis in the negotiated format
- `{datasetAPI}/collections/{collectionId}/coverage?scale-axes=Lat(2)` — returns a coverage re-scaled so as to contain 2 times less sample values along its latitude axis, and all original values along all its other dimensions

B.4. Field Selection (Range Subsetting) Example

These examples return a range subset of a coverage in the negotiated format, including domain set, range type (and metadata if applicable) to the extent than they can be described in that format:

- `{datasetAPI}/collections/{collectionId}/coverage?properties=B02,B03,B04` — returns only the bands with IDs B02, B03 and B04 from the coverage
- `{datasetAPI}/collections/{collectionId}/coverage?properties=3,4,5` — returns the 4th, 5th and 6th band (0-based indexing, as listed in the Range Type) from the coverage
- `{datasetAPI}/collections/{collectionId}/coverage?properties=B07,*` — returns the band B07 and all subsequent bands from the coverage

B.5. Coverage Tiles Example

Annex C: Revision History

Date	Release	Editor	Primary clauses modified	Description
2019-03-06	Template	C. Heazel	all	initial template

Annex D: Bibliography

- W3C/OGC: **Spatial Data on the Web Best Practices**, W3C Working Group Note 28 September 2017, <https://www.w3.org/TR/sdw-bp/>
- W3C: **Data on the Web Best Practices**, W3C Recommendation 31 January 2017, <https://www.w3.org/TR/dwbp/>
- W3C: **Data Catalog Vocabulary**, W3C Recommendation 16 January 2014, <https://www.w3.org/TR/vocab-dcat/>
- IANA: **Link Relation Types**, <https://www.iana.org/assignments/link-relations/link-relations.xml>
- International Telecommunication Union, **ITU-T.800 : Information technology - JPEG 2000 image coding system: Core coding system**, June, 2019, <https://www.itu.int/rec/T-REC-T.800-201906-I/en>
- W3C, **RDF 1.1 Semantics**, February 2014, <https://www.w3.org/TR/rdf11-mt/>
- OGC: **OGC 13-102r2, Name type specification – Time and index coordinate reference system definitions** (OGC Policy Document), version 1, 2014