OGC® DOCUMENT: 20-024

External identifier of this OGC® document: https://www.opengis.net/doc/is/ogcapi-common-2/1.0

Open Geospatial Consortium

OGC API - COMMON - PART 2: GEOSPATIAL DATA

STANDARD
Implementation

APPROVED

Version: 1.0
Submission Date: 2026-01-07
Approval Date: 2026-10-15
Publication Date: 2026-11-12
Editor: Charles Heazel, Joan Masó, Jérôme Jacovella-St-Louis

Notice: This document is an OGC Member approved international standard. This document is available on a royalty free, non-discriminatory basis. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

License Agreement

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

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

Copyright notice

Copyright © 2026 Open Geospatial Consortium
To obtain additional rights of use, visit https://www.ogc.org/legal

Note

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

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

CONTENTS

I. ABSTRACT v

II. KEYWORDS v

III. PREFACE vi

IV. SECURITY CONSIDERATIONS vii

V. SUBMITTING ORGANIZATIONS viii

VI. SUBMITTERS viii

VII. ACKNOWLEDGEMENTS viii

1. SCOPE 2

2. CONFORMANCE 4
   2.1. Summary of conformance URIs 5

3. NORMATIVE REFERENCES 7

4. TERMS, DEFINITIONS AND ABBREVIATED TERMS 10
   4.22. Abbreviated terms 14

5. CONVENTIONS 17
   5.1. Identifiers 17
   5.2. Link relations 17
   5.3. Compact URIs (CURIEs) 19
   5.4. Coordinate Reference Systems 20
   5.5. Temporal instants and intervals 20
   5.6. API definition 21

6. OVERVIEW 25
   6.1. Collections 25
   6.2. Data access mechanisms 25
   6.3. Representation 27
   6.4. UML Class Diagram 27
   6.5. Collection Description Properties 29

7. REQUIREMENTS CLASS “COLLECTIONS” 34
   7.1. List of Collections Resource (/collections) 35

7.2. Collection Description Resource (/collections/{collectionId}) 38
7.3. OGC API — Records compliance 48

8. REQUIREMENTS CLASS “UNIFORM ADDITIONAL DIMENSIONS COLLECTIONS” 51
8.1. Uniform description of all dimensions 51
8.2. Describing gridded dimensions 54

9. REQUIREMENTS CLASSES FOR ENCODINGS 59
9.1. Overview 59
9.2. Requirements Class “HTML” 59
9.3. Requirements Class “JSON” 61

10. MEDIA TYPES 64

ANNEX A CONFORMANCE CLASS ABSTRACT TEST SUITE 66
A.1. Introduction 66
A.2. Conformance Class Collections 66
A.3. Conformance Class Uniform Additional Dimensions Collections 71
A.4. Conformance Class JSON 72
A.5. Conformance Class HTML 73

ANNEX B CLASS MODEL AS TABLES 76
B.1. OGC API — Common package 76

ANNEX C GLOSSARY 100

ANNEX D BACKUS-NAUR FORMS 103
D.1. BNF for URI 103
D.2. BNF for Date-Time 104

ANNEX E REVISION HISTORY 107

BIBLIOGRAPHY 109

I ABSTRACT

The OGC API — Common Standard is a multi-part standard which defines a set of documents whose requirements classes can be used to build resource and purpose-specific Web API specifications. This document, OGC API — Common — Part 2: Geospatial Data Standard (Common API-2: GeoData) is one of those parts. This part specifies how to list and describe available collections of geospatial data within a Web API deployment.

The OGC Standards baseline has been extended to include Resource Oriented Architectures and Web APIs. In the course of developing OGC Web API standards, some practices proved to be common across multiple OGC Web API standards. These common requirements classes are documented in the OGC API — Common Multi-Part Standard, and serve as reusable components. Standards developers can specify dependencies on these requirements classes in the construction of OGC Web API Standards, and implementors of Web APIs can also indicate conformance to these requirements classes directly. The result is a modular suite of coherent API standards which can be adapted by a system designer for the unique requirements of their system.

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, Common

III PREFACE

This document defines the *OGC API — Common — Part 2: Geospatial Data* Standard. Suggested additions, changes and comments on this standard are welcome and encouraged. Such suggestions may be submitted as an issue on the *OGC API — Common* GitHub repository.

IV SECURITY CONSIDERATIONS

Clients should avoid blindly processing HTML markup code in the attribution property of collection descriptions specified by this Standard. In particular, care should be taken to avoid running any <script> code, including inline event handlers (e.g., onclick) and dangerous href values like javascript:. All untrusted input should be sanitized to prevent script execution and to mitigate XSS vulnerabilities.

See also the security considerations described in OGC API — Common — Part 1: Core.

V SUBMITTING ORGANIZATIONS

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

- Heazeltech
- Ecere Corporation
- Universitat Autònoma de Barcelona (CREAF)
- UK Met Office
- CubeWerx Inc.

VI SUBMITTERS

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

Name	Affiliation
Chuck Heazel *(editor)*	Heazeltech
Jérôme Jacovella-St-Louis *(editor)*	Ecere Corporation
Joan Masó *(editor)*	Universitat Autònoma de Barcelona (CREAF)
Chris Little	UK Met Office
Carl Reed	Carl Reed & Associates
Panagiotis (Peter) A. Vretanos	CubeWerx Inc.

VII ACKNOWLEDGEMENTS

A large portion of this document has been derived from OGC API — Features — Part 1: Core and generalized to any data access mechanism. Since the content was actually

embedded within this document, OGC API — Features is not cited as a normative reference.

Some of the work leading to the creation of this OGC Standard was undertaken as part of GeoConnections 2020-2021 and 2025-2027 projects GNS20-21IFP02 (*Modular OGC API Workflows*) and GNS25-27IFP2 (Advancing and Implementing OGC Standards for GeoDataCubes, DGGSs, Data Stores for Digital Twins and Cartographic Symbology). Financial support provided by GeoConnections, a national collaborative initiative led by Natural Resources Canada. GeoConnections supports the modernization of the Canadian Geospatial Data Infrastructure (CGDI). The CGDI is the collection of geospatial data, standards, policies, applications, and governance that facilitate its access, use, integration, and preservation.

1 SCOPE

1 SCOPE

Spatial data is rarely considered as a single entity. Since Feature Collections, Coverages, Data Sets are all aggregations of Spatial or Temporal Resources, an OGC Web API would also expose its holdings as aggregates of spatial resources called *collections*.

A single API may provide access to a large number of collections. The purpose of the OGC API — Common — Part 2: Geospatial Data (Common API-2: GeoData) Standard is to specify Web resources for listing and describing available collections.

Common API-2: GeoData does not specify the nature of the geospatial data contained in a collection. Rather, this Standard specifies a basic capability which should be applicable to any geospatial resource type or access mechanisms. Additional OGC Web API Standards extend this foundation to define specific data access mechanisms.

2 CONFORMANCE

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 Standardization Target for this Standard is Web APIs.

OGC API — Common — Part 2: Geospatial Data defines API components intended for re-use by other OGC Web API standards. For the purpose of conformance, the applicable API components are identified by Conformance Classes.

Typically, the OGC API — Common — Part 2: Geospatial Data will only be implemented through reference to these Conformance Classes by other standards.

OGC API — Common — Part 2: Geospatial Data identifies four conformance classes. Each conformance class is defined by one requirements class.

The tests in Annex A are organized by Requirements Class. Therefore, an implementation of the *Collections* conformance class must pass all tests specified in Annex A for the *Collections* requirements class.

The requirements classes defined in Part 2: Geospatial Data are:

- Collections
- Uniform Additional Dimensions Collections
- HTML
- JSON

The Collections Requirements Class defines a common means to list and describe collections of spatial resources, but does not mandate a specific encoding or format for representing resources.

The Uniform Additional Dimensions Collections Requirements Class defines a consistent way to describe the domain of multidimensional collections of data, including additional dimensions beyond space and time, as well as regular and irregular grids.

The *HTML* and *JSON* requirements classes specify representations for these resources in commonly used encodings for spatial data on the web.

Common API-2: GeoData builds on API components defined in the OGC API - Common - Part 1: Core (API-Core) Standard.

Each requirements class in the Common API-2: GeoData Standard identifies any API-Core Conformance Classes upon which it depends.

Proof of conformance with a Conformance Class includes demonstration of conformance with all dependencies of that Conformance Class.

The abstract tests in Annex A have been organized to facilitate validation of these dependencies.

These tests have been organized by requirements class.

A referencing standard only has to require conformance with a Conformance Class and all of the requirements and relevant tests are identified.

In addition, each conformance class includes one or more test.

This Standard specifies the requirements necessary to discover and describe collections of data, but does not specify how to access that data. Other OGC API Standards define data access mechanisms which can be combined with the requirements classes defined in this Standard.

2.1. Summary of conformance URIs

Table 1 — Conformance class URIs

CORRESPONDING REQUIREMENTS CLASS	CONFORMANCE CLASS URI
Collections	https://www.opengis.net/spec/ogcapi-common-2/1.0/conf/collections
Uniform Additional Dimensions Collections	https://www.opengis.net/spec/ogcapi-common-2/1.0/conf/uad-collections
HTML	https://www.opengis.net/spec/ogcapi-common-2/1.0/conf/html
JSON	https://www.opengis.net/spec/ogcapi-common-2/1.0/conf/json

3 NORMATIVE REFERENCES

3 NORMATIVE REFERENCES

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

Internet Engineering Task Force (committee): IETF RFC 7231, *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*. RFC Publisher (2014). https://www.rfc-editor.org/info/rfc7231.

Internet Engineering Task Force (committee): IETF RFC 8259, *The JavaScript Object Notation (JSON) Data Interchange Format*. RFC Publisher (2017). https://www.rfc-editor.org/info/rfc8259.

Nottingham M: IETF RFC 8288, *Web Linking*. RFC Publisher (2017). https://www.rfc-editor.org/info/rfc8288.

Klyne G, Newman C, Internet Engineering Task Force: IETF RFC 3339, *Date and Time on the Internet: Timestamps*. RFC Publisher (2002). https://www.rfc-editor.org/info/rfc3339.

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

Berners-Lee T, Fielding R, Masinter L: IETF RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*. RFC Publisher (2005). https://www.rfc-editor.org/info/rfc3986.

Gregorio J, Fielding R, Hadley M, Nottingham M, Orchard D: IETF RFC 6570, *URI Template*. RFC Publisher (2012). https://www.rfc-editor.org/info/rfc6570.

OGC Abstract Specification Topic 2: Referencing by coordinates. https://docs.ogc.org/as/18-005r4/18-005r4.html

W3C: HTML5, W3C Recommendation, https://www.w3.org/TR/html5/

Schema.org: https://schema.org/docs/schemas.html

QUDT.org. QUDT Ontologies, derived models and vocabularies 2.1. Edited by R. Hodgson. 2024. Available at https://www.qudt.org/.

Regenstrief Institute, Inc. The Unified Code for Units of Measure. Edited by G. Schadow, .C J. McDonald. 2017. Available at https://ucum.org/ucum.

Open API Initiative: OpenAPI Specification, Version 3.0. The latest patch version at the time of publication of this standard was 3.0.3, available from https://spec.openapis.org/oas/v3.0.3.

JSON Schema: A Media Type for Describing JSON Documents. Edited by A. Wright, H. Andrews, B. Hutton, G. Dennis. 2022 [viewed 2023-10-14]. Available at https://json-schema.org/draft/2020-12/json-schema-core

JSON Schema Validation: A Vocabulary for Structural Validation of JSON. Edited by A. Wright, H. Andrews, B. Hutton. 2022 [viewed 2023-10-14]. Available at https://json-schema.org/draft/2020-12/json-schema-validation

4

TERMS, DEFINITIONS AND ABBREVIATED TERMS

4 TERMS, DEFINITIONS AND ABBREVIATED TERMS

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

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

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

4.1. **collection**

(in the context of OGC APIs) set of spatiotemporal data that may be available through one or more access mechanisms defined by OGC API standard(s)

4.2. **coverage**

function which returns values from its range for any direct position within its domain

(Source: OGC 07-011r2)

4.3. **data access mechanism**

(in the context of OGC APIs) mechanism defined by an OGC API Standard to access spatiotemporal data from a collection

Such mechanism is based on Web resources which are sub-resources of that collection. The access mechanism could be as simple as the definition of a single sub-resource URL template. The access mechanism may allow Web clients to access parts and/or the totality of the data available from the collection.

4.4. dataset

collection of data, published or curated by a single agent, and available for access or download in one or more representations (DCAT)

4.5. distribution

specific representation of a dataset
A dataset might be available in multiple serializations that may differ in various ways, including natural language, media-type or format, schematic organization, temporal and spatial resolution, level of detail or profiles (which might specify any or all of the above). (DCAT)

EXAMPLE: a downloadable file, an RSS feed or an API.

4.6. extent

area covered by something. Within this document, “extent” refers to spatial extent
The size or shape that may be expressed using coordinates. (W3C/OGC Spatial Data on the Web Best Practice)

4.7. feature

abstraction of real world phenomena

(Source: ISO 19101-1:2014)

Note 1 to entry: A feature may occur as a type or an instance. Feature type or feature instance shall be used when only one is meant.

4.8. feature collection

set of features from a dataset

(Source: ISO 19168-1:2020)

4.9. **geometry**

ordered set of *n*-dimensional points in a given coordinate reference system (W3C/OGC Spatial Data on the Web Best Practice)

4.10. **OGC Web API**

Web API that implements one or more Conformance Classes from an OGC API Standard

4.11. **resource**

entity that might be identified (Dublin Core Metadata Initiative — DCMI Metadata Terms)

Note 1 to entry: The term "resource", when used in the context of an OGC Web API standard, should be understood to mean a Web Resource unless otherwise indicated.

4.12. **resource type**

definition of a type of resource

Note 1 to entry: Resource types are re-usable components that are independent of where the resource resides in the API.

4.13. **spatial resource**

resource which represents or includes spatial data
A Spatial Thing.

4.14. **spatial thing**

anything with spatial extent, (i.e. size, shape, or position) and is a combination of the real-world phenomenon and its abstraction (W3C/OGC Spatial Data on the Web Best Practice)

4.15. **temporal coordinate system**

temporal reference system based on an interval scale on which distance is measured as a multiple of a single unit of time

(Source: ISO 19108:2002)

4.16. **temporal position**

location relative to a temporal reference system

(Source: ISO 19108:2002)

4.17. **temporal reference system**

reference system against which time is measured

(Source: ISO 19108:2002)

4.18. **temporal resource**

resource which represents or includes time and date
A Temporal Thing.

4.19. **temporal thing**

anything with temporal extent, i.e. duration
Examples are: The taking of a photograph, a scheduled meeting, or a GPS time-stamped track-point. (W3C Basic Geo)

4.20. **Web API**

API using an architectural style that is founded on the technologies of the Web (W3C Data on the Web Best Practices)

4.21. **web resource**

resource that is identified by a URI

Note 1 to entry: For example, a web resource is any file or asset, such as text, images, HTML, XML, or JavaScript, that can be accessed over the web using a unique URL/URI

4.22. Abbreviated terms

API	Application Programming Interface
CORS	Cross-Origin Resource Sharing
CRS	Coordinate Reference System
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
OGC	Open Geospatial Consortium
TRS	Temporal Coordinate Reference System

URI	Uniform Resource Identifier
YAML	YAML Ain’t Markup Language

5 CONVENTIONS

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 https://www.opengis.net/spec/ogcapi-common-2/1.0.

All Requirements, Requirements Modules and Conformance Modules that appear in this document are denoted by partial URIs that are relative to this base.

Additional information about the use of Identifiers in API-Common is provided in the OGC API — Common Users Guide.

5.2. Link relations

RFC 8288 (Web Linking) is used by this standard to express relationships between resources. Link relation types from the IANA Link Relations Registry are used wherever possible. Additional link relation types are registered with the OGC Naming Authority.

The link relationships used in Common API-2: GeoData are described in Table 2. Additional relation types may be used if the implementation warrants it.

Table 2 — Link Relations

LINK RELATION	PURPOSE
alternate	Refers to a substitute for this context [IANA]. Refers to a representation of the current resource which is encoded using another media type (the media type is specified in the type link attribute).
https://www.opengis.net/def/rel/ogc/1.0/data-meta	Identifies general metadata for the context (dataset or collection) that is primarily intended for consumption by machines.
collection	The target IRI points to a resource which represents the collection resource for the context IRI. [IANA]

LINK RELATION	PURPOSE
https://www.opengis.net/def/rel/ogc/1.0/conformance	Refers to a resource that identifies the specifications that the link's context conforms to. [OGC]
https://www.opengis.net/def/rel/ogc/1.0/data (or data)	Indicates that the link's context is a distribution of a dataset that is an API and refers to the root resource of the dataset in an API. [OGC]
describedby	Refers to a resource providing information about the link's context. [IANA] Links to external resources which further describe the subject resource
http://www.opengis.net/def/rel/ogc/1.0/schema	Refers to a resource that lists the properties of geospatial data in the collection represented by the link's context. (OGC API — Common — Part 3) [OGC]
item	The target IRI points to a resource that is a member of the collection represented by the context IRI. [IANA]
https://www.opengis.net/def/rel/ogc/1.0/items	Refers to a resource that is comprised of members of the collection represented by the link's context. [OGC]
license	Refers to a license associated with this context. [IANA]
cite-as	Indicates that the link target is preferred over the link context for the purpose of permanent citation. [IANA]
about	Refers to a resource that is the subject of the link's context. [IANA]
author	Refers to the context's author. [IANA]
disclosure	Refers to a list of patent disclosures made with respect to material for which 'disclosure' relation is specified. [IANA]
copyright	Refers to a copyright statement that applies to the link's context. [IANA]
sponsored	Refers to a resource that is within a context that is sponsored (such as advertising or another compensation agreement). [IANA]
terms-of-service	Refers to the terms of service associated with the link's context. [IANA]
self	Conveys an identifier for the link's context. [IANA] A link to another representation of this resource.
service-desc	Identifies service description for the context that is primarily intended for consumption by machines. [IANA] API definitions are considered service descriptions.
service-doc	Identifies service documentation for the context that is primarily intended for human consumption. [IANA]

LINK RELATION	PURPOSE
service-meta	Identifies general metadata for the context that is primarily intended for consumption by machines. [IANA]

Additional information on the use of link relationships is provided in the OGC API — Common Users Guide.

5.3. Compact URIs (CURIEs)

Compact URIs (CURIEs) resolve to full URIs based on deterministic rules. Safe CURIEs (enclosed within [square brackets]) should generally be used to distinguish them from non-CURIE identifiers. The general mapping for CURIEs to URIs defined on the OGC definition server is <authority>[-<URI type>]:<element> mapping to <authority URL>/<URI type>/<authority>/0/<element>, with the default URI type being crs when not specified. For example, [OGC:CRS84] maps to https://www.opengis.net/def/crs/OGC/0/CRS84, and [ogc-rel:map] maps to `https://www.opengis.net/def/rel/ogc/0/map. Unfortunately, there are a few challenges originating from how the URIs themselves are defined which need to be considered for establishing equivalence between a CURIE and a URI.

- The /0/ is intended to refer to the latest version or for URI types where the version is not relevant (as should be the case for link relations). The CURIEs should always be interpreted as referring to the latest version, and the version should be ignored when comparing a CURIE against a URI. The definition server may or may not support resolving both the /0/ and a specific version e.g., /1.0/.
- http and https URIs should be considered as equivalent.
- While some OGC definition types use lowercase ogc (e.g., link relations /rel/), others use uppercase OGC (e.g., coordinate reference systems /crs/).

> ***WARNING***
>
> *Because multiple OGC API Standards allow the use of safe CURIEs for link relations and CRS references, OGC API — Common also allows them for compatibility reasons. However, server implementations conforming to OGC API Standards where CURIEs are not supported need to avoid them and only use full URIs for all resources relevant to those standards. Client implementations for OGC API Standards allowing the use of CURIEs need to be prepared to accept both CURIEs and full URIs.*

While CURIEs themselves avoid the confusion of http vs. https and version numbers, the challenges associated with mapping them to multiple URIs requires more complicated logic than comparing an identifier string to the CURIE itself and a single full URI.

5.4. Coordinate Reference Systems

As discussed in Chapter 9 of the W3C/OGC Spatial Data on the Web Best Practices document, the ability to express and share location in a consistent way is one of the most fundamental aspects of publishing geographic data. To do so, it is important to be clear about the coordinate reference system (CRS) within which the coordinates are expressed.

The Common API-2: GeoData Standard does not mandate the use of a specific coordinate reference system. However, if no CRS is specified, the default coordinate reference systems for spatial geometries are:

- [OGC:CRS84] — WGS 84 longitude and latitude without height; and
- [OGC:CRS84h] — WGS 84 longitude and latitude with ellipsoidal height.

NOTE 1:[OGC:CRS84] is the CURIE form for the resolvable URI https://www.opengis.net/def/crs/OGC/0/CRS84, [OGC:CRS84h] is the CURIE form for the resolvable URI https://www.opengis.net/def/crs/OGC/0/CRS84h.
NOTE 2:The order of coordinates for the bounding boxes and spatial intervals used within the collection description will always follow the axis order of the applicable CRS exactly.

OGC Abstract Specification Topic 2 – Referencing by Coordinates (ISO 19111) provides the conceptual and logical model for Coordinate Reference Systems.

In this Standard, coordinates are primarily used for describing the spatial extent for which data is available within a collection.

5.5. Temporal instants and intervals

API — Common adopts the Gregorian calendar and a 24 hour time keeping system for describing temporal instants and intervals. All representations of these temporal elements which are discussed in this document conform to RFC 3339.

An ABNF representation of the RFC 3339 format is provided in Annex F.

Temporal instants and intervals are measured relative to an underlying temporal reference system (TRS). This Common API-2: GeoData Standard does not mandate a specific temporal coordinate reference system. However, all dates or timestamps

discussed in this document are in the Gregorian calendar and conform to RFC 3339. In data, other temporal reference systems may be used where appropriate.

5.6. API definition

5.6.1. General remarks

The OGC Common API — 2: GeoData specifies requirements and recommendations for the development of APIs allowing to list and describe spatial resources using a standard way of doing so. In general, deployed APIs will go beyond the requirements and recommendations stated in this Standard, in particular by implementing other OGC API Standards specifying data access mechanims. A particular Web API implementation will support additional operations, parameters, and so on that are specific to that implementation.

So that developers can more easily learn how to use the deployed API, good documentation is essential. In the best case, documentation would be available both in HTML for human consumption and in a machine readable format that can be processed by software for compile-time or run-time binding. OpenAPI is one way to provide that machine readable documentation.

5.6.2. Role of OpenAPI

The OGC Common API — 2: GeoData uses OpenAPI 3.0 fragments in examples and to formally state requirements. Using OpenAPI 3.0 is not required for implementing an OGC API Standard. Other API definition languages may be used along with, or instead of, OpenAPI. However, any API definition language used should have an associated conformance class advertised through the /conformance path.

The OGC API — Common — Part 1 Standard includes a conformance class for API definitions that follow the OpenAPI specification 3.0. Alternative API definition languages are also allowed. Conformance classes for additional API definition languages will be added as the OGC API landscape continues to evolve. An API definition is required for implementations conforming to the OGC API — Part 1: Core “Landing Page” requirements class.

5.6.3. References to JSON Schema components in normative statements

Some normative statements (requirements) use a phrase that a response to a request must validate against a JSON Schema component in the OGC schema repository.

In this case, implementations of this API are free to extend those components in the following ways.

- Supporting representations in other media types beyond JSON following the conceptual model described by the JSON Schema.
- The range of values of a property may be extended (additional values) or constrained (only a subset of all possible values is allowed). An example for a constrained range of values is to explicitly specify the supported values of a string parameter or property using an *enum*.
- Additional properties may be added to the schema (unless explicitly disallowed by the JSON Schema).
- When providing an API definition, informative text, such as comments or description properties, may be changed or added.

5.6.4. Reusable OpenAPI components

Reusable components for OpenAPI definitions for an OGC API are referenced from this document. They are available from the OGC Schemas Registry at https://schemas.opengis.net/ogcapi/common/part1/1.0 and https://schemas.opengis.net/ogcapi/common/part2/1.0.

A server implementation of the OGC API – Common — Part 2 Standard can use the content in the openapi directory to generate a response for the API description using OpenAPI 3.0. The ogcapi-common-2.yaml file includes paths and components. An implementation should adapt their definition to include any additional resource paths or query parameters defined by other supported OGC API Standards. The components part includes parameters, responses and schemas that can be reused as-is. However, the implementation should select the appropriate collection description schema (collectionDesc.yaml or collectionDesc-UAD.yaml) based on whether they implement the Uniform Additional Dimensions Collections Requirements Class or not. For proper validation against the Uniform Additional Dimensions schemas, both the collections.yaml schema (describing the list of collections) and the rCollection.yaml OpenAPI response for the /collections/{collectionId} resource need to be modified to include the UAD version of the collection description.

The api directory contains JSON files that are templates with enumerated values for collections. A particular implementation of the Common API-2: GeoData should enumerate the actual resources exposed by the API deployment in the same way. The implementation can select to dynamically implement responses to /api/all-collections as a Web resource supported by the server, or instead to hardcode the api/all-collections.json file with the actual list of resource identifiers in the enumeration.

To improve performance, the whole content of this directory can be bundled into a single document by executing a tool such as swagger-cli. This document can be served for the OGC API — Common — Part 1 service-desc link from the landing page.

Additional information on the use of OpenAPI as an API definition is provided in the OGC API — Common Users Guide.

6 OVERVIEW

6 OVERVIEW

This OGC API — Common — Part 2: Geospatial Data Standard provides a way to list and describe available collections of spatiotemporal data. Web APIs specifying mechanisms to access data are defined in separate OGC API Standards. This Standard depends on the Core requirements class of OGC API — Common — Part 1: Core which specifies fundamental requirements for OGC Web APIs based on HTTP. Implementations may also conform to the Landing Page requirements class of Part 1 which specifies additional requirements for the landing page (root of the API), conformance declaration (/conformance) and an API definition.

6.1. Collections

Spatiotemporal data is rarely considered as a single entity, but as aggregations, or *collections*, of Spatio-Temporal things.

While *collection* is a common term, its specific meaning is often based on the context in which it is used. Given the focus on addressing geospatial data, a definition that reflects the unique characteristics of geospatial data collections is needed. Therefore, **for purposes of this standard** and in the context of OGC APIs, the term Collection refers to a set of spatiotemporal data that may be available through one or more access mechanisms defined by OGC API standard(s).

The purpose of the OGC API — Common — Part 2: Geospatial Data Standard is to list and describe the available geospatial data collections.

OGC Web API Standards could extend this definition to address the specific properties of the resources they describe.

6.2. Data access mechanisms

A collection of geospatial data may be accessed in more than one way. Some examples of data access mechanisms are:

- a feature-centric approach allowing to access individual features by their identifiers,
- providing access to a subset of geospatial information for an arbitrary area, time and resolution of interest, and

- provide access to data for pre-determined partitions of space using tiles or discrete global grid zones.

To illustrate a particular use case, a collection of data for land cover classification may be available either as individual features where one multi-polygon feature corresponds to low vegetation, another feature corresponds to trees and yet another feature corresponds to the built-up area. In an implementation of the OGC API — Features Standard, each of these features may be individually accessible at:

/collections/landcover/items/{itemId}.

The API deployment may also include an implementation of the access mechanism defined by the OGC API — Coverages Standard for the same classified land cover collection enabling retrieval of a subset for a given spatial area using:

/collections/landcover/coverage?subset=Lat(10:20),Lon(30:40).

The OGC API — Common — Part 2: Geospatial Data Standard describes data collections, but does not define any *access mechanisms* to the data. Access mechanisms are defined in separate OGC Web API Standards.

The *access mechanisms* are defined by the resource paths which are specified in the different OGC API Standards. The resources associated with a particular *access mechanism* for a specific collection will be under the collection resource:

/collections/{collectionId}/{accessResources}

where:

- {collectionId} is an identifier for the collection; and
- {accessResources} are resources defined for accessing data using a particular mechanism.

IMPORTANT

Editors of OGC API Standards and implementers of Web APIs will carefully avoid redefining different resources with the same path in different OGC API Standards and implementations, so that API deployments can offer multiple data access mechanisms for the same collections.

The available data access mechanisms supported for a specific collection are typically advertised by including links using specific link relation types defined by the various OGC API Standards in the `links` array of the collection description.

Access mechanisms should not be confused with *representations*.

6.3. Representation

Data access mechanisms may support one or more representation (data formats / encodings) for each resource they define. A representation is typically negotiated between the server and client using HTTP content negotiation (the Accept: request header). In the land cover example above, the client could request either individual features or a feature collection as GeoJSON (Accept: application/geo+json) or as GeoPackage (Accept: application/geopackage+sqlite3), and the coverage request could be requested as GeoTIFF (Accept: image/tiff; application=geotiff) or also as GeoPackage (application/geopackage+sqlite3) leveraging the GeoPackage Extension for Tiled Gridded Coverage Data. Alternative formats supported by both the server and client could also be negotiated successfully.

Some media types for encodings such as JSON are very generic. When no suitable media type has been registered for a representation using a specific schema, a mechanism to negotiate a particular profile may also be used for example using a profile query parameter or an Accept-Profile: request header as described in the W3C Content Negotiation by Profile Working Draft, in combination with the regular HTTP content negotiation headers. The concept of profile may also be used to control other aspects of the representation besides the schema.

6.4. UML Class Diagram

The following UML class diagram represents the model of common resources offered by Web APIs implementing one or more OGC API Standards. In particular, the diagram describes the resources presented in OGC API Common — Part 1: the Landing page (*LandingPage* class), the conformance page (*ConformanceClasses*) and the exception report (*Exception*). The diagram also describes the two resources specified in this Part 2: the collections page (*Collections*) and the collection description (*CollectionDescription*), as well as its associated classes. While these common resources are precisely described in the requirements to follow, they are presented here as a UML diagram for convenience. The transformation of this UML diagram into a list of tables, one for each class, can be found in Annex B. In case of discrepancy between this UML diagram, the UML tables, and the text of the requirements, the requirements should be considered correct.

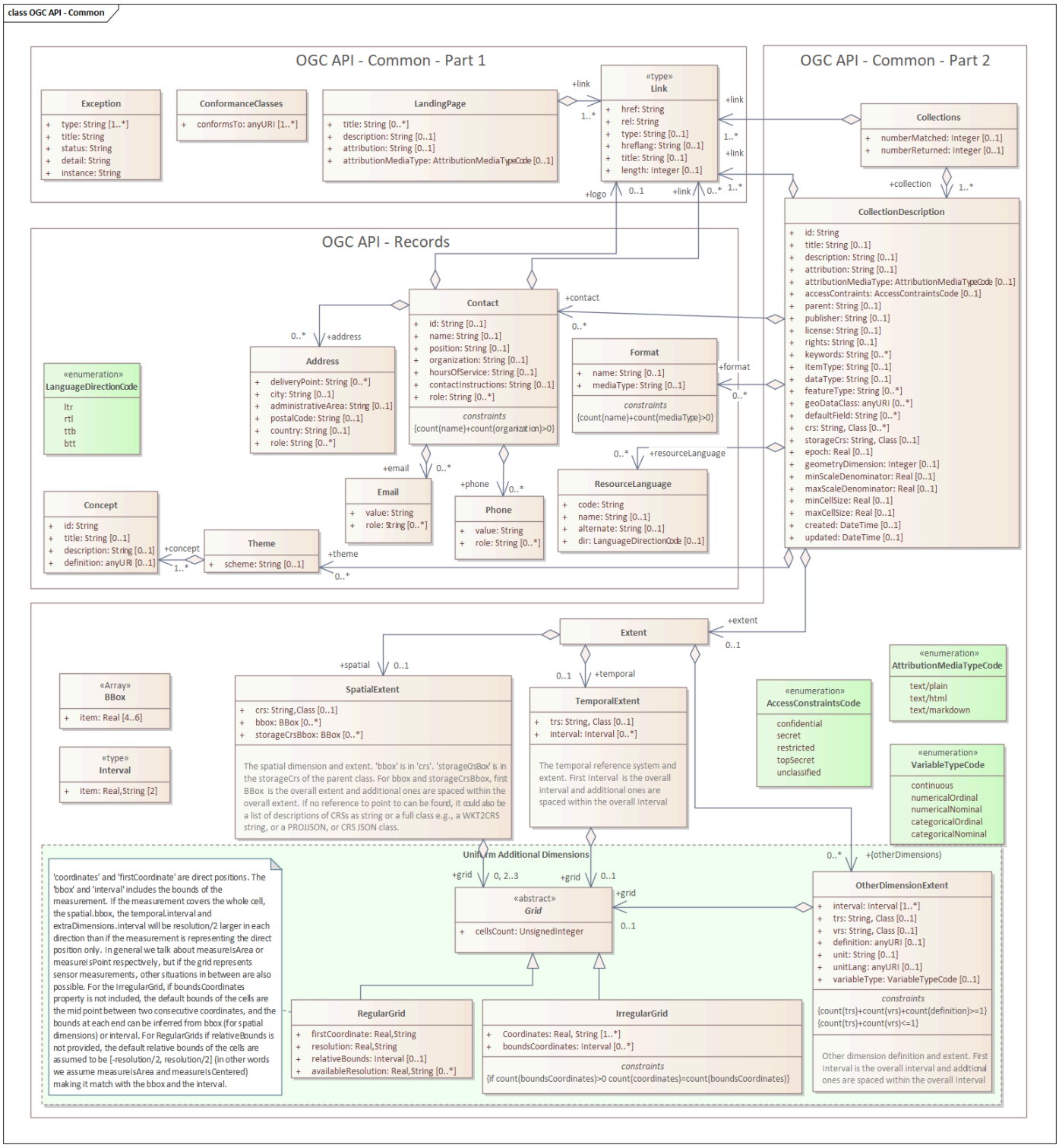


Figure 1 — UML class diagram describing the common resources

> ***IMPORTANT***
>
> *While the UML class model attributes use the singular for elements with multiplicity, the JSON encoding and schema use the plural versions, as seen in the table in the next section.*

6.5. Collection Description Properties

The following table lists all properties defined as part of the collection description. Note that only the id and links are mandatory. The "rel": "self" link is the only mandatory link, and usually at least one link tied to a specific data access mechanism (e.g., "rel": "items" or "rel": "[ogc-rel:map]") should be included. This set of properties is neither a minimal subset, nor a comprehensive union of all properties defined by all OGC API Standards, but a list of properties identified as being in use by two or more OGC API Standards, without being tied to any particular data access mechanism.

Several of the properties originate from the 2D Tile Matrix Set and TileSet Metadata, which itself was largely derived from the Vector Tiles Pilot TileSet Metadata Engineering Report. Other properties originate from OGC API — Records (core records properties), as a collection can be thought as a record resource as part of a Local Resource Catalog.

This collection description remains extensible (it does not set "additionalProperties": false), and does not preclude the addition of new properties by new OGC API standards. However, to ensure consistency across the OGC API Standards, new OGC API Standards should consider reusing these properties when their functionality is relevant, and avoid defining new conflicting properties (with same name but different semantics).

Table 3 — JSON Collection Description Properties

NAME (AS ENCODED IN JSON)	JSON TYPE	DEFINITION
id	String	***(required)*** Identifier of the collection (sometimes referred as collectionId). It is used in the Web API as path parameter in /collections/{collectionId}
title	String	A short, human-readable summary of the collection
description	String	A human-readable explanation about data in the collection
extent	Extent	The extent where and when data from this collection is available
links	Array of Link	***(required)*** Links to this or other resources related to this collection

NAME (AS ENCODED IN JSON)	JSON TYPE	DEFINITION
attribution	String	For the collection that can contain markup text whose format may be indicated in the attributionMediaType property. That format can be either plain text (text/plain), HTML (text/html) or CommonMark (text/markdown)
attributionMediaType	AttributionMediaTypeCode (text/plain, text/html, text/markdown)	Media type for the markup language of the attribution: It can be either plain text, HTML or CommonMark
accessContraints	AccessContraintsCode (unclassified, restricted, confidential, secret, topSecret)	Restrictions on the availability of the collection that the user needs to be aware of before using or redistributing the data
parent	String	An id of a collection that is the parent of this collection in a hierarchical relation. For example a rivers' collection can have a topographic collection as parent
publisher	String	Organization or individual responsible for making the data available
contacts	Array of Contact	A list of contacts qualified by their role(s) in association to the collection
license	String	The legal provisions under which the data of this collection is made available
resourceLanguages	Array of ResourceLanguage	The list of languages in which the data of this collection is available
rights	String	A statement that concerns all rights not addressed by the license such as a copyright statement
keywords	String	The topic or topics of the resource. Typically represented using free-form keywords, tags, key phrases, or classification codes
themes	Array of Theme	A knowledge organization system used to classify the resource
itemType	String	Indicator about the type of the items in the collection if the collection has an accessible /collections/{collectionId}/items endpoint (e.g., feature or record)
dataType	String	Nature of the data in the collection. For example, this could be set to map, vector or coverage. It is not an indication of the available data access mechanism, but a hint regarding the internal representation of the data.
featureTypes	String	The list of feature (or coverage) type(s) which are present inside the collection. This is useful for associating the relevant portrayal rules of a style.

NAME (AS ENCODED IN JSON)	JSON TYPE	DEFINITION
formats	Array of Format	A list of formats in which the data of this collection is distributed
geoDataClasses	Array of URI	One or more classes of geospatial data, identified by a URI, which implies compatibility with a particular schema (fields/properties, as defined in OGC API — Common — Part 3: Schemas) for one or more collections, each with particular semantic definitions and data types. This is useful for example to determine compatibility with styles or processes, or to identify a particular collection within a dataset based on semantics.
defaultFields	Array of String	A sorted list of field names. For a features data access mechanism , it is the list of returnable properties included by default and complementing the geometry of the features. For a coverage data access mechanism, they are names of fields whose values are included by default. Note that an API may provide a mechanism (e.g., a query parameter called properties or exclude-properties) to override this default list in customized responses.
crs	Array of String (URI, CURIE or WKT2CRS) or Object (PROJJSON or CRS JSON)	The list of coordinate reference systems supported by the API; if present, the list should include the default coordinate reference system (usually listed first). Each item should be a URI to a registered CRS if one is available, or a full description of the CRS otherwise using e.g., a WKT2CRS string, or a PROJJSON, or CRS JSON object.
storageCrs	String (URI, CURIE or WKT2CRS) or Object (PROJJSON or CRS JSON)	The native coordinate reference system (i.e., the most efficient CRS in which to request the data, possibly how the data is stored on the server); this is the default output coordinate reference system for Maps and Coverages. This should be a URI to a registered CRS if one is available, or a full description of the CRS otherwise using e.g., a WKT2CRS string, or a PROJJSON, or CRS JSON object.
epoch	Real	Epoch of the native (storage) Coordinate Reference System (CRS)
geometry Dimension	Integer	Number of spatial dimensions of the primary geometry of individual elements of the data: 0 for points, 1 for curves, 2 for surfaces, 3 for solids and unspecified when mixed or unknown. Not to be confused with the dimensions of the domain which are defined by the extent element.
minScale Denominator	Real	Minimum scale denominator for usage of the collection
maxScale Denominator	Real	Maximum scale denominator for usage of the collection

NAME (AS ENCODED IN JSON)	JSON TYPE	DEFINITION
minCellSize	Real	Minimum cell size for usage of the collection, in degrees of latitude [1] for data referenced to Earth
maxCellSize	Real	Maximum cell size for usage of the collection, in degrees of latitude [1] for data referenced to Earth
created	String (date-time format)	Timestamp indicating when the data in the collection was first produced
updated	String (date-time format)	Timestamp of the last change/revision to the data in the collection

[1] To convert to meters, the cell size is simply converted from degrees to radians and multiplied by the equatorial radius in meters. For example, for the WGS84 ellipsoid, metersCellSize = cellSize * Pi / 180 * 6,378,137 m = cellSize * ~111,319.4908 meters / degree. The cell size is always in degrees to facilitate discovery and comparison between collections with different native CRS (storageCrs).

NOTE: Refer to Annex B for the full model class tables describing the custom data types of these properties.

7

REQUIREMENTS CLASS "COLLECTIONS"

7 REQUIREMENTS CLASS "COLLECTIONS"

REQUIREMENTS CLASS 1: COLLECTIONS	
IDENTIFIER	https://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.1: https://www.opengis.net/spec/ogcapi-common-2/1.0/conf/collections
PREREQUISITE	https://www.opengis.net/spec/ogcapi-common-1/1.0/req/core
NORMATIVE STATEMENTS	Requirement 1: /req/collections/collections-list-op Requirement 2: /req/collections/collections-list-success Requirement 3: /req/collections/collections-list-links Requirement 4: /req/collections/collections-list-collections Requirement 5: /req/collections/description-op Requirement 6: /req/collections/description-success Requirement 7: /req/collections/description-links Requirement 8: /req/collections/description-extent Requirement 9: /req/collections/description-extent-multi

The Collections Requirements Class consistently describes across the OGC API Standards suite the resources and operations used to list and describe geospatial collection resources exposed through a deployed Web API. This class neither specifies how to organize your data into collections nor how to provide access to the data described in the collections. That level of detail is reserved for OGC Web API Standards that define access mechanisms through specific resources (see Access mechanisms Section).

RECOMMENDATION 1	
IDENTIFIER	/rec/collections/rec-part1-landing-page
A	An implementation of the /Collections Requirements Class SHOULD also implement the Landing Page conformance class which covers a landing page, a conformance page and an API definition as defined in OGC API — Common Part 1: Core.

The two resources defined by this Requirements Class are summarized in Table 4 and their operations are defined in the following requirements.

Table 4 — Collection Resources

RESOURCE	URI	HTTP METHOD	DESCRIPTION
List of Collections	/collections	GET	List of available Collections including some descriptions of each one
Collection description	/collections/{collectionId}	GET	Description about a specific collection of geospatial data with links to distribution

7.1. List of Collections Resource (/collections)

Implementations of OGC API Standards typically enable deployers to organize their geospatial data into collections. Information about those collections is accessed through the /collections path, which is linked to using the https://www.opengis.net/def/rel/ogc/1.0/data link relation (or simply data for compatibility with OGC API — Features).

7.1.1. Operation

REQUIREMENT 1	
IDENTIFIER	/req/collections/collections-list-op
INCLUDED IN	Requirements class 1: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections
A	The API SHALL support the HTTP GET operation at the path /collections.
B	If the deployed API instance has a landing page (or an equivalent mechanism to expose root resources), this landing page or equivalent SHALL link to this resource listing collections (/collections) using the link relation type https://www.opengis.net/def/rel/ogc/1.0/data or data.

7.1.2. Response

REQUIREMENT 2	
IDENTIFIER	/req/collections/collections-list-success
INCLUDED IN	Requirements class 1: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections

REQUIREMENT 2	
A	A successful execution of the operation SHALL be responded with an HTTP status code 200.
B	The content of the response SHALL validate against the JSON schema collections.yaml.

The collections response returned by this operation is based on the collections.yaml JSON schema. An example of a response listing collections are provided in Clause 7.1.2.1.

This Collections schema is further constrained by the following requirements and recommendations.

To support hypermedia navigation, the links property must be populated with sufficient hyperlinks to navigate through the whole dataset.

REQUIREMENT 3	
IDENTIFIER	/req/collections/collections-list-links
INCLUDED IN	Requirements class 1: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections
A	A 200-response SHALL include the following links in the links property of the response: • A link to this response document (relation: self), • A link to the response document in every other media type supported by the API (relation: alternate).
B	All links SHALL include the rel property.
C	All links where rel is self or alternate SHALL include the type link parameter.

Additional information may be available to assist in understanding and using this dataset. Links to those resources should be provided as well.

RECOMMENDATION 2	
IDENTIFIER	/rec/collections/collections-list-describedby
A	If external descriptions exist that provide additional information about the structure or semantics for the resource, a 200-response SHOULD include links to each of those resources in the links property of the response (relation: describedby).
B	The type link parameter SHOULD be provided for each link. This applies to resources that describe the whole dataset.

The collections property of the Collections response provides a description of each individual collection distributed by the Web API deployment.

REQUIREMENT 4	
IDENTIFIER	/req/collections/collections-list-collections
INCLUDED IN	Requirements class 1: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections
A	For each spatial resource collection accessible through an implementation of this API, metadata describing that collection SHALL be provided in the collections property of the list of Collections response.

The array items included in the collection property are described in the Collection Resource section of the Collections Requirements Class.

NOTE: The Collections Requirements Class does not define any query parameters for use against the /collections resource. Implementers who plan supporting the offering of large numbers of collections from the same API endpoint may consider extending their API implementations with capabilities defined in the OGC API — Records Standard (with collections acting as the local resources of a local resource catalog), as well as candidate extensions to the OGC API — Common Standard that introduce capabilities for hierarchical collections, searching, filtering, and sorting collections.

7.1.2.1. Collections Response Example

The following JSON example response lists available collections for a dataset with a single "buildings" feature collection.

There is a link to the resource listing collections itself (link relation type: self).

Representations of this resource in other formats are referenced (link relation type: alternate).

An additional link is to the logical schema of the collection (link relation type: [ogc-rel: schema]).

The reference systems for describing the spatiotemporal extent do not need to be specified, as they default to [OGC:CRS84] and Gregorian calendar / Coordinated Universal Time (UTC). This spatiotemporal extent always needs to be specified (or default) this way for data referenced to Earth and a time scale for which the Gregorian calendar is suitable. For the spatial extent, the crs specified in the spatial property of the extent only describes the CRS of the bbox and is independent of the native / storage reference system of the data itself, which can be specified separately in the storageCrs property. For explicitly specifying the Gregorian calendar, the URI https://www.opengis.net/def/crs/OGC/0/GregorianDateTime can be used, or alternatively https://www.opengis.net/def/uom/ISO-8601/0/Gregorian for compatibility with OGC API — Features.

```
{
  "links": [
    { "href": "https://data.example.org/collections.json",
      "rel": "self", "type": "application/json", "title": "This document" },
    { "href": "https://data.example.org/collections.html",
      "rel": "alternate", "type": "text/html", "title": "This document as HTML" }
  ],
  "collections": [
    {
      "id": "buildings",
      "title": "Buildings",
      "description": "Buildings in the city of Bonn.",
      "attribution": "Copyright © 2025 _City of Bonn_ ![logo](https://example.org/cityOfBonn.png)",
      "attributionMediaType": "text/markdown",
      "extent": {
        "spatial": { "bbox": [ [ 7.01, 50.63, 7.22, 50.78 ] ] },
        "temporal": { "interval": [ [ "2010-02-15T12:34:56Z", "2018-03-18T12:11:00Z" ] ] }
      },
      "links": [
        { "href": "https://data.example.org/collections/buildings",
          "rel": "self", "type": "application/json" },
        { "href": "https://data.example.org/collections/buildings/schema",
          "rel": "[ogc-rel:schema]", "type": "application/schema+json",
          "title": "Logical schema for buildings" }
      ]
    }
  ]
}
```


Listing 1 — Example /collections **response listing collections**

7.1.3. Error situations

See OGC API — Common — Part 1: Core Section 8.2 “HTTP Status Codes” for general guidance.

7.2. Collection Description Resource (/collections/{collectionId})

Each resource collection is described by a set of metadata. That metadata can be accessed directly using the /collections/{collectionId} path and as an entry in the collections property of the /collections resource.

7.2.1. Operation

REQUIREMENT 5	
IDENTIFIER	/req/collections/description-op
INCLUDED IN	Requirements class 1: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections
A	The Implementation SHALL support the HTTP GET operation at the path /collections/{collectionId}, where the parameter collectionId can be any values of the id property in the response to the list of collections (JSONPath: $.collections[*].id in /collections).

7.2.2. Response

REQUIREMENT 6	
IDENTIFIER	/req/collections/description-success
INCLUDED IN	Requirements class 1: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections
A	A successful execution of the operation SHALL be reported as a response with a HTTP status code 200.
B	The content of a *Collection* description resource SHALL be based upon the JSON schema collectionDesc.yaml.
C	The content of the *Collection* description resource response SHALL be consistent with the content for this collection in the /collections response, with the exception of links which can contain additional links in the response for the individual *Collection* description resource. That is, if a property is included in the response to the list of collections at /collections, the value for that property of the same collection (the collection whose id value corresponds to the {collectionId}) SHALL be identical for response for the *Collection* description resource (/collections/{collectionId}).

The collection description response returned by this operation is based on the collectionDesc.yaml JSON schema.

An example of a response describing a collection are provided in Clause 7.2.2.5.

Most of the properties of the Collection resource are self-explanatory. However, a few properties require additional explanation.

7.2.2.1. Attribution

A collection description can include an attribution property allowing clients to display a short attribution for data being visualized. For example, this is often shown at the bottom of a map in a 2D interface. This attribution, providing a way to identify the source of the geographic information, can contain markup text whose format may be indicated in the

attributionMediaType property. That format can be either plain text (text/plain), HTML (text/html) or CommonMark (text/markdown). If the 'attributionMediaType' indicates something other than text/plain, the attribution element string should be interpreted by a markup parser selected based on that media type to be presented to the user (e.g., text/markdown will be parsed by a library supporting CommonMark). By allowing markup, the attribution string can import images (e.g., organization logos) and format the text (e.g., the name of the organization in italics). See the example collection response for an example of the use of markup in the attribution element.

Note that in order to support attribution in different languages, a server may take into consideration the Accept-Language: request header, for example a client would make a request specifying Accept-Language: fr,en;q=0.8 to prioritize a French representation, but fall back to English if a French representation is not available. This language negotiation mechanism is part of the HTTP standard and affects all resources, and the language of all properties inside these. For example, in a collection description, this language negotiation will apply to titles, descriptions, keywords, links, etc. (not only the attribution).

The following IANA link relations can also be considered to link (using the links property — see Clause 7.2.2.3) to resources providing additional information regarding the origin, ownership, funding and restrictions associated with the data: about, author, copyright, cite-as, sponsored, disclosure, license, terms-of-service.

7.2.2.2. Item Type

In some collections of geospatial data, the members (items) that make up that collection can be individually accessed by a client. In this case, the itemType property in the Collection resource identifies the type of the items accessible from that collection.

RECOMMENDATION 3	
IDENTIFIER	/rec/collections/description-item-type
A	If the members (items) that make up a collection can be individually accessed by a client, then the itemType key SHOULD be included in the Collection resource to indicate the type of the items (e.g. feature or record).

7.2.2.3. Links

To support hypermedia navigation, the links property must be populated with sufficient hyperlinks to navigate through the whole dataset.

REQUIREMENT 7	
IDENTIFIER	/req/collections/description-links

REQUIREMENT 7	
INCLUDED IN	Requirements class 1: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections
A	A 200-response SHALL include the following links in the links property of the response: • A link to this response document (relation: self), • A link to the response document in every other media type supported by the API (relation: alternate).
B	All links SHALL include the rel property.
C	All links where rel is self or alternate SHALL include the type link parameter.

Additional information may be available to assist in understanding and using this dataset. Links to those resources should be provided as well.

RECOMMENDATION 4	
IDENTIFIER	/rec/collections/collection-describedby
A	If external schemas or descriptions exist that provide additional information about the structure or semantics of the collection, a 200-response SHOULD include links to each of those resources in the links property of the response (relation: describedby).
B	The type link parameter SHOULD be provided for each link.

The http://www.opengis.net/def/rel/ogc/1.0/schema link relation should be used to reference a logical schema (see OGC API — Common — Part 3: Schemas).

7.2.2.4. Extent

The extent property defines a spatiotemporal envelope that encompasses the geospatial data in the collection. Since not all collections are nicely clustered around a single place in space and time, the extent property provides flexibility in how that surface can be defined.

- The Spatial bounding boxes (bbox, and optionally storageCrsBbox) provide a set of rectangular bounding boxes which use coordinates to envelope portions of the collection. Typically only the first element would be populated. Additional boxes may be useful, for example, when the collection is clustered in multiple, widely separated locations.
- Temporal Interval provides a set of temporal periods. Typically only the first temporal period would be populated. However, like bbox, additional periods can be added if the collection does not form a single temporal cluster.

The temporal reference system (trs) specified in the temporal property of the extent defines not only the reference system of the interval, but also the reference system for the primary temporal dimension of the data.

For the spatial extent, the crs specified in the spatial property of the extent defines only the coordinate reference system of the bbox, minCellSize and maxCellSize. For data referenced to Earth, this crs will always be [OGC:CRS84] or [OGC:CRS84h]. The native CRS of the data is specified separately in the storageCrs property of the collection description. For some access mechanisms, this native CRS corresponds to the default output CRS. These access mechanisms may additionally support other output CRSs which are specified in the crs property of the collection description.

For the storageCrs as well as for the crs list of supported output CRSs, implementors of Web APIs based on OGC API standards are strongly recommended to reference existing CRS definitions in the form of URIs (or the equivalent CURIEs) such as the ones provided by the EPSG Geodetic Parameter Dataset. However, there are situations where the required CRS has not been registered with an authority.

For these cases, where a URI does not exist, a Well-Known Text string describing the coordinate reference system, or a JSON object encoding of the CRS (using the PROJJSON schema or a potential "CRS JSON" successor if eventually standardized by the OGC) can be specified.

Such alternate CRS definitions can also be used for the crs of the bbox for data in an engineering CRS or otherwise not georeferenced to Earth.

The following tables lists all properties defined as part of the collection's extent. While additional properties beyond the spatial and temporal defined here are left open by this "Collections" requirements class, the Uniform Additional Dimensions Collections requirements class defines a specific uniform schema for properties representing additional dimensions, while also defining the grid property which is also applicable to spatiotemporal extents.

Table 5 — JSON Collection's Extent Properties

NAME (AS ENCODED IN JSON)	JSON TYPE	DEFINITION
spatial	SpatialExtent	The spatial extent of the data in the collection.
temporal	TemporalExtent	The temporal extent of the data in the collection.
(any additional property)	OtherDimension Extent	*(if implementing Uniform Additional Dimensions Collections)* The extent for any additional dimensions beyond those described in temporal and spatial.

Table 6 — JSON SpatialExtent Properties

NAME (AS ENCODED IN JSON)	JSON TYPE	DEFINITION
bbox	Array of (at least one) Array of (4 or 6) Real	One or more bounding boxes that describe the spatial extent of the dataset. The first bounding box describes the overall spatial extent of the data. All subsequent bounding boxes describe more precise bounding boxes, e.g., to identify clusters of data. Clients only interested in the overall spatial extent will only need to access the first item in each array. For data referenced to Earth, this will be either: • [<West longitude>, <South latitude>, <East longitude>, <North latitude>] for a CRS84 bounding box, or • [<West longitude>, <South latitude>, <lowest ellipsoidal height in meters>, <East longitude>, <North latitude>, <highest ellipsoidal height in meters>] for a 3-dimensional CRS84h bounding box.
storageCrsBbox	Array of (at least one) Array of (4 or 6) Real	One or more bounding boxes that describe the spatial extent of the dataset in the storage (native) CRS (storageCrs property). The first bounding box describes the overall spatial extent of the data. All subsequent bounding boxes describe more precise bounding boxes, e.g., to identify clusters of data. Clients only interested in the overall spatial extent will only need to access the first item in each array.
crs	String (URI, CURIE or WKT2CRS) or Object (PROJJSON or CRS JSON)	Coordinate reference system **of the coordinates of the** bbox **property**. The default reference system is WGS 84 longitude/latitude. WGS 84 longitude/latitude/ellipsoidal height for coordinates with height. For data not referenced to Earth, another CRS may be specified. This should be a URI to a registered CRS if one is available, or a full description of the CRS otherwise using e.g., a WKT2CRS string, or a PROJJSON, or CRS JSON object.
grid	Array of 2 or 3 (RegularGrid or IrregularGrid)	*(if implementing Uniform Additional Dimensions Collections)* Grid description providing information about the limited availability of data within the collection organized as a grid (regular or irregular) along each spatial dimension. All units within the spatial grid object are specified in the storageCrs, defaulting to CRS84 if not specified.

Table 7 — JSON TemporalExtent Properties

NAME (AS ENCODED IN JSON)	JSON TYPE	DEFINITION
interval	Array of (at least 1) Array of 2 (Real or String)	One or more time intervals that describe the temporal extent of the dataset. The first time interval describes the overall temporal extent of the data. All subsequent time intervals describe more precise time intervals, e.g., to identify clusters of data. Clients only interested in the overall extent will only need to access the

NAME (AS ENCODED IN JSON)	JSON TYPE	DEFINITION
		first item in each array. The end temporal instant of the interval (second value of each array) includes the thickness of the last time component specified. For example, the interval ["2026-07-25T00:00:00Z", "2026-07-25T23:59:59Z"] ends at the very end of the day, including the full thickness of the last second. Conversely, specifying ["2026-07-25T00:00:00Z", "2026-07-26T00:00:00Z"] would include the whole first second of the next day. Note that RFC 3339 explicitly disallows hour 24, but if such a value is encountered for an end time, no thickness should be assumed in that particular case.
trs	String (URI, CURIE or WKT2CRS) or Object (PROJJSON or CRS JSON)	Coordinate reference system of the coordinates in the temporal extent (property interval). The default reference system is the Gregorian calendar. For data for which the Gregorian calendar is not suitable, such as geological time scale, another temporal reference system may be used. This should be a URI to a registered TRS if one is available, or a full description of the TRS otherwise using e.g., a WKT2CRS string, or a PROJJSON, or CRS JSON object.
grid	RegularGrid or IrregularGrid	*(if implementing Uniform Additional Dimensions Collections)* Grid description providing information about the limited availability of data within the collection organized as a grid (regular or irregular) along the temporal dimension.

In the following example, the native CRS of the data is in the Mercator CRS (EPSG:3395). The data is referenced to Earth, therefore the bounding box for discovery purpose needs to be specified in OGC:CRS84. The CRSs available for output (which the client could potentially select using a `crs` query parameter defined by the access mechanism) include both EPSG:3395 and OGC:CRS84.

```
"extent":
{
  "spatial": {
    "crs": "[OGC:CRS84]",
    "bbox": [ -180, -85.0840590500295, 180, 85.0840590500295 ],
    "storageCrsBbox": [ -20037508.342789, -20037508.342684, 20037508.342789, 20037508.342684 ]
  }
},
"storageCrs": "[EPSG:3395]",
"crs": [ "[EPSG:3395]", "[OGC:CRS84]" ]
```

Listing 2 — Example illustrating the CRS-related properties of the collection description

WARNING

While some OGC API Standards explicitly support the use of CURIEs for properties of the collection description such as `storageCrs` and the `crs` array of supported output CRSs, other OGC API Standards do not. For

servers implementing those OGC API Standards that do not allow the use of CURIEs, full URIs therefore need to be used.

REQUIREMENT 8	
IDENTIFIER	/req/collections/description-extent
INCLUDED IN	Requirements class 1: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections
A	When an extent property is provided for a collection, that extent's boundaries SHALL encompass the spatial and temporal extent of the data in the collection. The temporal extent may use null values to indicate a half-bounded or unbounded time interval. If the data contains spatial or temporal information defined in multiple properties, the Implementation needs to determine how this extent is derived from those properties.
B	For a collection whose native CRS (storageCrs) differs from the crs specified in the spatial property (which is always [OGC:CRS84] for data referenced to Earth), the bounds of the spatial dimensions in that native CRS SHALL be described in the storageCrsBbox of the spatial property (unlike the other dimensions which use the interval property).
C	The bounds of the spatial dimension in the CRS defined in the crs property of the spatial property (which defaults to [OGC:CRS84] if not present) SHALL be described in the bbox of the spatial property in that CRS (always [OGC:CRS84] for data referenced to Earth).

REQUIREMENT 9	
IDENTIFIER	/req/collections/description-extent-multi
INCLUDED IN	Requirements class 1: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections
A	If the extent property includes a member spatial, all data in the collection SHALL be inside the extent described by the first bounding box in the bbox array.
B	If the extent property includes a member spatial and the bbox array has more than one item, individual components (e.g., a feature or scene) of the collection SHALL be inside the extent described by one of the other bounding boxes in the bbox array.
C	If the extent property includes a member temporal, all data in the collection SHALL be inside the extent described by the first time interval in the interval array.
D	If the extent property includes a member temporal and the interval array has more than one item, individual components of the collection SHALL be inside the extent described by one of the other time intervals in the interval array.

NOTE: As a consequence, the first bounding box is a union of all subsequent bounding boxes, and the first temporal interval is a union of all subsequent temporal intervals.

RECOMMENDATION 5	
IDENTIFIER	/rec/collections/description-extent-single
A	While the spatial and temporal extents support multiple bounding boxes (bbox array) and time intervals (interval array) for more accurately describing sparse datasets, implementations SHOULD provide only a single bounding box or time interval for datasets with a relatively homogeneous spatiotemporal distribution.

PERMISSION 1	
IDENTIFIER	/per/collections/description-extent-extensions
A	This requirements class only specifies requirements for spatial and temporal extents. However, the extent object MAY be extended with additional members to represent other extents, such as thermal or pressure ranges. See the Uniform additional dimension conformance class for a well-defined way of describing additional dimensions.

RECOMMENDATION 6	
IDENTIFIER	/rec/collections/description-extent-storage-crs-bbox
A	If the native CRS of the data, as specified in the storageCrs property, is not [OGC:CRS84] or [OGC:CRS84h], the spatial extent of the data in that native CRS SHOULD be specified in the storageCrsBbox of the spatial property.

PERMISSION 2	
IDENTIFIER	/per/collections/description-reference-systems
A	For data not referenced to the Earth (e.g., the Moon, Mars), the spatial extent's bbox property MAY be specified in a CRS other than [OGC:CRS84] or [OGC:CRS84h], with this CRS indicated in the crs property of the spatial property.
B	For data for which the Gregorian calendar is not suitable, such as geological time scale, another temporal reference system MAY be specified in the trs property of the temporal property.

7.2.2.5. Collection Object Examples

This Collection Description example response in JSON is for a single "buildings" collection.

The basic descriptive information includes:

- “id”: an identifier for this collection;
- “title”: human-readable title for this collection;
- “description”: longer text describing this collection; and
- “attribution”: markup providing attribution (owner, producer, logo, etc.) of this collection.

The response includes links to:

- the response itself (link relation type: self),
- representations of this response in other formats are referenced using (link relation type: alternate),
- an additional link is to a logical schema for the collection (link relation type: [ogc-rel:schema]).

Finally, this response includes both spatial and temporal extents.

The reference systems for describing the spatiotemporal extent do not need to be specified, as they default to [OGC:CRS84] (longitude, latitude) and Gregorian calendar / Coordinated Universal Time (UTC) and should always be specified (or default) this way for data referenced to Earth and a time scale for which the Gregorian calendar is suitable. For the spatial extent, the crs specified in the spatial property of the extent only describes the CRS of the bbox and is independent of the native / storage reference system of the data itself, which can be specified separately in the storageCrs property. For explicitly specifying the Gregorian calendar, the URI https://www.opengis.net/def/crs/OGC/0/GregorianDateTime can be used, or alternatively https://www.opengis.net/def/uom/ISO-8601/0/Gregorian for compatibility with OGC API — Features.

```
{
  "id": "buildings",
  "title": "Buildings",
  "description": "Buildings in the city of Bonn.",
  "attribution": "Copyright © 2025 _City of Bonn_ ![logo](https://example.org/cityOfBonn.png)",
  "attributionMediaType": "text/markdown",
  "extent": {
    "spatial": { "bbox": [ [ 7.01, 50.63, 7.22, 50.78 ] ] },
    "temporal": { "interval": [ [ "2010-02-15T12:34:56Z", "2018-03-18T12:11:00Z" ] ] }
  },
  "links": [
    { "href": "https://data.example.org/collections/buildings",
      "rel": "self", "type": "application/json" },
    { "href": "https://data.example.org/collections/buildings/schema",
      "rel": "[ogc-rel:schema]", "type": "application/schema+json",
      "title": "Logical schema for buildings" }
  ]
}
```

Listing 3 — Example /collections/{collectionId} **response describing a collections**

7.2.2.6. Collection-specific conformance declaration

A client might be able to guess, from the API-wide conformance declaration, which conformance classes are applicable specifically to a particular collection based on the data access mechanism(s) present within the links. For instance, if the collection links to a coverage endpoint, the client can guess that the Coverages conformance classes are applicable, while the Features classes are not. However, this may not be precise enough, for example in the case of scaling or subsetting being supported by one collection, but not by another. This permission suggests a way to declare the exact list of conformance classes that are applicable specifically to a particular collection.

PERMISSION 3	
IDENTIFIER	/per/collections/conformance
A	For an API deployment serving multiple data collections, the list of conformance classes supported by a specific collection MAY be linked to using the link relation https://www.opengis.net/def/rel/ogc/1.0/conformance. The content retrieved from that link would follow the same schema (confClasses.yaml) as the API-wide conformance declaration linked to from the landing page. When present, this collection-specific conformance declaration needs to explicitly list all conformance classes applicable to that collection.

7.2.3. Error Situations

See OGC API — Common — Part 1: Core Section 8.2 “HTTP Status Codes” for general guidance.

If the parameter `collectionId` does not exist on the server, the status code of the response will be `404`.

7.3. OGC API — Records compliance

When implementing the “Collections” requirement class, the `/collections` end-point can be considered a *Local Resource Catalog*, where the local resources are the collections being cataloged.

This requirement class is consistent with the requirements of OGC API — Records — Part 1: Core “Local Resources Catalog” (Deployment). An implementation may gain additional interoperability by conforming to that requirement class from OGC API — Records as well.

In this case, the implementation should verify and declare conformance with these five conformance classes defined in Records corresponding to the following URIs:

Table 8

OGC API — RECORDS REQUIREMENTS CLASS	URI
Record Core	https://www.opengis.net/spec/ogcapi-records-1/1.0/conf/record-core
Record Collection	https://www.opengis.net/spec/ogcapi-records-1/1.0/conf/record-collection
Crawlable Catalog	https://www.opengis.net/spec/ogcapi-records-1/1.0/conf/crawlable-catalog
Local Resources Catalog	https://www.opengis.net/spec/ogcapi-records-1/1.0/conf/local-resources-catalog
Autodiscovery	https://www.opengis.net/spec/ogcapi-records-1/1.0/conf/autodiscovery

The additional requirements include linking to the /collections end-point with an https://www.opengis.net/def/rel/ogc/1.0/ogc-catalog link relation from the landing page.

8

REQUIREMENTS CLASS “UNIFORM ADDITIONAL DIMENSIONS COLLECTIONS”

8 REQUIREMENTS CLASS "UNIFORM ADDITIONAL DIMENSIONS COLLECTIONS"

REQUIREMENTS CLASS 2: UNIFORM ADDITIONAL DIMENSIONS COLLECTIONS	
IDENTIFIER	https://www.opengis.net/spec/ogcapi-common-2/1.0/req/uad-collections
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.2: https://www.opengis.net/spec/ogcapi-common-2/1.0/conf/uad-collections
PREREQUISITE	Requirements class 1: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections
NORMATIVE STATEMENTS	Requirement 10: /req/uad-collections/extent-uad-definition Requirement 11: /req/uad-collections/grid-description

The Collections Requirements Class defines a Collection resource which supports both geospatial and temporal dimensions. However, the domain of some datasets cannot be fully described with only spatiotemporal dimensions. The Uniform Additional Dimensions Collections Requirements Class extends the Collection resource to support an unlimited number of dimensions defined in a uniform manner, so that clients supporting this requirements class can interpret these dimensions in a generic manner.

8.1. Uniform description of all dimensions

A Uniform Additional Dimensions Collection is an extension of the Collection resource.

This requirements class supports the definition of additional dimensions beyond spatial and temporal in a specific way which is consistent with the spatial and temporal dimensions.

Each dimension is identified by using a semantic definition property. An interval providing the lower and upper bound for the coordinates along that dimension also needs to be included, as well as a unit of measure where applicable.

As an alternative — or in addition to — the definition, a temporal reference system (trs) or a vertical reference system (vrs) can be specified. The trs or vrs property can be populated with either a URI or CURIE.

The following table lists all properties defined as part of a uniform additional dimension's extent.

Table 9 — JSON OtherDimensionExtent Properties

NAME (AS ENCODED IN JSON)	JSON TYPE	DEFINITION
definition	String (URI)	***(at least one of definition, trs, vrs is required)*** A URI for the definition of the measured or observed property corresponding to this dimension
unit	String	The unit of measure in which the interval and/or grid values are expressed
trs	String (URI, CURIE or WKT2CRS) or object (PROJJSON or CRS JSON)	***(at least one of definition, trs, vrs is required)*** Temporal Coordinate Reference System (e.g. as defined by Features for 'temporal'). This should be a URI to a registered TRS if one is available, or a full description of the TRS otherwise using e.g., a WKT2CRS string, or a PROJJSON, or CRS JSON object.
interval	Array of (at least 1) Array of 2 (Real or String)	***(required)*** One or more intervals that describe the extent for this dimension of the dataset. The value null is supported and indicates an unbounded or half-bounded interval. The first interval describes the overall extent of the data for this dimension. All subsequent intervals describe more precise intervals, e.g., to identify clusters of data. Clients only interested in the overall extent will only need to access the first item (a pair of lower and upper bound values).
grid	RegularGrid or IrregularGrid	Grid description providing information about the limited availability of data within the collection organized as a grid (regular or irregular) along the temporal dimension.
unitLang	String	The language (or vocabulary) in which the unit is expressed (defaults to "UCUM" if not specified).
variableType	VariableType Code (continuous, numericalOrdinal, numericalNominal, categoricalOrdinal, categoricalNominal)	The class of variable, distinguishing between continuous vs. discrete, numerical vs. categorical, and ordinal vs. nominal. Continuous variables (continuous) represent an infinite spectrum of values (such as spatial coordinates or temporal instants), while discrete variables (numericalOrdinal, numericalNominal, categoricalOrdinal, and categoricalNominal) represent a finite or countable set of distinct values. Nominal variables are unordered, while ordinal variables imply a specific order. Numerical variables are identified with integer or real numbers, while categorical variables are identified with textual strings. The variable type informs correct interpretation and interpolation method.
vrs	String (URI, CURIE or WKT2CRS) or	***(at least one of definition, trs, vrs is required)*** Vertical Coordinate Reference System (e.g. as defined in EDR for 'vertical'). This should be a URI to a registered VRS if one is

NAME (AS ENCODED IN JSON)	JSON TYPE	DEFINITION
	object (PROJJSON or CRS JSON)	available, or a full description of the VRS otherwise using e.g., a WKT2CRS string, or a PROJJSON, or CRS JSON object. **NOTE:** This property is included mainly for compatibility with OGC API — Environmental Data Retrieval (EDR). A true vertical CRS such as height above the ellipsoid would normally be the third dimension of the spatial extent, while a dimension representing a parameter such as pressure would more accurately be described as a parametric CRS, and would generally be described using the semantic definition and unit properties.

For additional dimensions beyond spatial and temporal, the reference system is defined as a combination of the definition and unit, or alternatively the trs or vrs. In addition to specifying the reference system of the interval, this also specifies the reference system of that dimension for the data itself. Unlike the spatial CRS, the collection description in this standard only supports defining a single reference system for the temporal dimension and each additional dimension.

To validate against this Uniform Additional Dimensions Collections Requirements Class, the collectionDesc-UAD.yaml collection description YAML JSON Schema including extent-UAD.yaml should be used instead of collectionDesc.yaml

REQUIREMENT 10	
IDENTIFIER	/req/uad-collections/extent-uad-definition
INCLUDED IN	Requirements class 2: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/uad-collections
A	Any additional dimension property added to the extent of a collection description SHALL contain an interval property consisting of an array of one or more interval(s), where the first element describes the overall interval where data is available for the collection, whereas any additional elements describe intervals encompassing clusters of data availability within the overall interval.
B	Each interval element SHALL be described as an array of two values, with the first being the lower bound and the second the upper bound.
C	An unbounded or half-bounded interval SHALL be described using a null value for its lower and/or upper bound.
D	Any additional dimension SHALL specify a URI as either the value of a definition to indicate the semantic concept for the variable (from any semantic definition vocabulary, such as QUDT) associated with the dimension, the value of a trs property indicating a temporal reference system or the value of a vrs property indicating the vertical reference system.

REQUIREMENT 10	
E	For additional dimension defined using definition, where a particular unit is used, that dimension SHALL contain a unit property expressing the unit of measure, where the language for defining the unit is UCUM (Unified Code for Units of Measure), unless specified otherwise in a unitLang property (using values such as "UCUM" or "QUDT").

8.2. Describing gridded dimensions

For gridded data, whether using a regular or an irregular grid, this requirements class defines a grid property which is used to describe such grids.

While the interval specified for the dimension (or bbox for spatial dimensions) describes the region occupied by all cells, including the region of validity or bounds of all measurements, this grid property describes in more detail each individual measurement.

Common types of grids are often called *Value-is-Area* or *Value-is-Point*, but especially in the case of sensor measurements, other situations in-between are also possible.

8.2.1. Regular grids

A regular grid has data measurements separated by a constant distance (the resolution of the grid), while for an irregular grid the distance between data measurements varies. For both cases, the number of cells (cellsCount) is specified. A cell is a region within which a particular measurement is valid.

The bounds are specified as relativeBounds which are relative to points starting from a firstCoordinate separated by an equal distance of resolution units. The lower value for the interval for a particular dimension (or bbox in the case of spatial dimensions) corresponds to the firstCoordinate plus the first value of the relativeBounds, while the upper value corresponds to the firstCoordinate plus cellsCount times the resolution plus the second value of the relativeBounds.

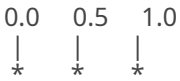
The following table lists all properties defined as part of a regular grid.

Table 10 — JSON RegularGrid Properties

NAME (AS ENCODED IN JSON)	JSON TYPE	DEFINITION
cellsCount	Integer (positive)	***(required)*** Number of samples available along the dimension
resolution	Real or String	***(required)*** Resolution of regularly gridded data along the dimension in the collection

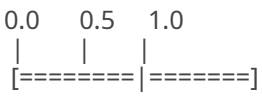
NAME (AS ENCODED IN JSON)	JSON TYPE	DEFINITION
firstCoordinate	Real or String	***(required)*** First coordinate where a regular grid begins, with subsequent coordinates adding resolution unit at each step
relativeBounds	Array of 2 (Real or String)	Distance in units from coordinate to the lower and upper bounds of each cell for regular grids, describing the geometry of the cells
available Resolutions	Array of (Real or String)	List of resolutions made available by the API. If not specified, there are no limit other than the limits indicated in minCellSize and maxCellSize of the collection description (though the API might still allow requesting data beyond these).

For example, a regular grid where each measure is associated with infinitely small points would have relativeBounds of [0, 0]. With a cellsCount of 3, a resolution of 0.5, and a firstCoordinate at 0, the corresponding interval would be [0, 1].



Listing 4 — Diagram illustrating a *Value-is-Point* regular grid

When each measure is associated with an area filling entire cells, there could be multiple ways to define the same grid based on a preference of how the coordinate points are defined. Consider the following regular grids, with a cellsCount of 2, a resolution of 0.5, and an interval of [0, 1].



Listing 5 — Diagram illustrating a *Value-is-Area* regular grid

One way to describe this grid would be with the firstCoordinate at 0 and the relativeBounds as [0, 0.5].

Another equivalent way to describe the same grid would be a firstCoordinate at 0.25 and the relativeBounds as [-0.25, 0.25].

If the relativeBounds property is not provided, the default relative bounds of the cells are assumed to be [-resolution/2, resolution/2]. In other words, the cell, or the region of validity of the measurement, is centered on each repeating point starting from the firstCoordinate.

For a *Value-is-Area* grid, the interval (or bbox for the spatial dimensions) will be resolution/2 larger in each direction compared to the same number of cells and same resolution for a *Value-is-Point* grid.

8.2.2. Irregular grids

Irregular grids are described by explicitly listing a sequence of coordinates (one for each cell) along each irregularly gridded dimension(s).

Bounds for each coordinate can also be individually specified in a parallel boundsCoordinates array.

If the boundsCoordinates property is not included, the default bounds of the cells are assumed to lie in the middle of the two values specified in coordinates, and for the two extremities, the lower (for the first cell) and upper bounds (for the last cell) are assumed to be at the same distance from the specified coordinate as to their immediate neighbor.

The following table lists all properties defined as part of a regular grid.

Table 11 — JSON IrregularGrid Properties

NAME (AS ENCODED IN JSON)	JSON TYPE	DEFINITION
cellsCount	Integer (positive)	***(required)*** Number of samples available along the dimension
coordinates	Array of (Real or String)	***(required)*** List of coordinates along the dimension for which data organized as an irregular grid in the collection is available.
bounds Coordinates	Array of Array of 2 (Real or String)	For irregular grids, an ordered list of the coordinates in absolute units of the lower and upper bounds of the dimension for each cell.

8.2.3. Requirements for describing grids

REQUIREMENT 11	
IDENTIFIER	/req/uad-collections/grid-description
INCLUDED IN	Requirements class 2: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/uad-collections
A	If data is organized as a regular or irregular grid for the temporal and any additional dimension, that dimension SHALL contain a grid object property describing the grid of that dimension.
B	If data is organized as a regular or irregular grid for the spatial dimensions, the spatial object SHALL contain a grid array property, where each element is an object describing the grid of that dimension.

REQUIREMENT 11

C	If data is organized as a regular or irregular grid for a dimension, the grid object(s) for that dimension SHALL indicate the number of coordinates along the dimension in a cellsCount property. • For values representing the whole area of contiguous cells spanning *resolution* units along the dimension, the *cellsCount* will be (*upperBound* — *lowerBound*) / *resolution*. • For values representing infinitely small point cells spaced by *resolution* units along the dimension, the *cellsCount* will be (*upperBound* — *lowerBound*) / *resolution* + 1.
D	For regularly gridded dimensions, the grid object SHALL contain a resolution property indicating the resolution of the grid in the unit of the unit property or the unit associated with the trs, vrs, or storageCrs in the case of the spatial dimensions (defaulting to [OGC:CRS84] if none is provided). The regular grid can be generated by incrementally adding the resolution starting from the lower bound of the overall interval.
E	For irregularly gridded (or categorical) dimensions, the grid object SHALL contain a coordinates property indicating the list of valid coordinates where data is available along that dimension.

PERMISSION 4

IDENTIFIER	/per/uad-collections/rec-categorical-dimension-interval
A	For a categorical dimension, the first and the last categories listed in coordinates MAY be selected as the lower and upper bounds of the mandatory overall interval, or null MAY be used for both the lower and upper bounds.

9

REQUIREMENTS CLASSES FOR ENCODINGS

9 REQUIREMENTS CLASSES FOR ENCODINGS

9.1. Overview

This clause specifies two requirements classes for encodings to be used with the Collections and Collection resources. These encodings are common representations for accessing spatial data on the web:

- HTML
- JSON

Neither of these encodings is mandatory. An implementation of the Collections requirements class may implement either, both, or neither of them.

9.2. Requirements Class “HTML”

Best practices for sharing geospatial data on the Web (W3C Best Practice) recommend supporting an HTML representation of API resources. For consistency with the Web, this should be done in a way that enables users and search engines to access all of the data they are authorized to access.

Providing an HTML representation for the landing page of a dataset, for listing available collections, and for describing detailed metadata about a particular collection allows users, using only their Web browser as a client, to evaluate the usefulness of data available from the API deployment before proceeding to access this data.

Additionally, an HTML representation of resources facilitate information discovery from Web search engines.

The HTML encoding should present all of the same information available in other representations such as the machine-readable JSON, so that users see this information in their browser in a way that naturally blends in with the rest of the institutional site, and ideally following Web Content Accessibility Guidelines.

Links that correspond to the *Link* class from the UML class diagram will be included as <a/> elements within the HTML <body/>, and could also be returned as HTTP response Link: headers.

REQUIREMENTS CLASS 3: HTML

IDENTIFIER	https://www.opengis.net/spec/ogcapi-common-2/1.0/req/html
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.4: https://www.opengis.net/spec/ogcapi-common-2/1.0/conf/html
PREREQUISITES	Requirements class 1: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections Schema.org HTML Living Standard
NORMATIVE STATEMENTS	Requirement 12: /req/html/definition Requirement 13: /req/html/content

REQUIREMENT 12

IDENTIFIER	/req/html/definition
INCLUDED IN	Requirements class 3: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/html
A	200-responses of the server SHALL support the text/html media type for the list of Collections and Collection description resources.

REQUIREMENT 13

IDENTIFIER	/req/html/content
INCLUDED IN	Requirements class 3: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/html
A	Every 200-response of the API Implementation that has the media type “text/html” SHALL be a HTML 5 document that includes the following information in the HTML body: • All information identified in the schemas of the Response Object in the HTML <body/>, and • All links in HTML <a/> elements in the HTML <body/>.

RECOMMENDATION 7

IDENTIFIER	/rec/html/schema-org
A	A 200-response for the /collections, /collections/{collectionId} and /collections/{collectionId}/schema resources with the media type text/html, SHOULD include Schema.org annotations.

9.3. Requirements Class “JSON”

JSON is a text syntax that facilitates structured data interchange between programming languages. JSON is commonly used for Web-based software-to-software interchanges. Most Web developers are comfortable with using a JSON-based format, so supporting JSON is recommended for machine-to-machine interactions.

REQUIREMENTS CLASS 4: JSON	
IDENTIFIER	https://www.opengis.net/spec/ogcapi-common-2/1.0/req/json
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.3: https://www.opengis.net/spec/ogcapi-common-2/1.0/conf/json
PREREQUISITES	Requirements class 1: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections IETF RFC 8259: The JavaScript Object Notation (JSON) Data Interchange Format JSON Schema
NORMATIVE STATEMENTS	Requirement 14: /req/json/definition Requirement 15: /req/json/content

REQUIREMENT 14	
IDENTIFIER	/req/json/definition
INCLUDED IN	Requirements class 4: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/json
A	200-responses of the server SHALL support the application/json media type for the list of Collections and Collection description resources.

REQUIREMENT 15	
IDENTIFIER	/req/json/content
INCLUDED IN	Requirements class 4: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/json
A	Every 200-response with the media type application/json SHALL include, or link to, a payload encoded according to the JSON Interchange Format.

REQUIREMENT 15	
B	The schema of all responses with the media type application/json SHALL conform with the JSON Schema specified for that resource.

JSON Schema for the Collections and Collection responses are available at collections.yaml and collectionDesc.yaml.

These are generic schemas that do not include any application schema information about specific resource types or their properties.

10

MEDIA TYPES

10 MEDIA TYPES

The media type for lists of collections and collection description as JSON is application/json.

The media type for HTML for all API resources would be text/html.

See also OGC API — Common — Part 1: Core Media Types section for additional media types such as Problem Details and OpenAPI API definitions.

Other media types used in OGC Standards can be found on OGC Media Type Register.

See also the IANA media type register.

A

ANNEX A (NORMATIVE) CONFORMANCE CLASS ABSTRACT TEST SUITE

ANNEX A
(NORMATIVE)
CONFORMANCE CLASS ABSTRACT TEST SUITE

A.1. Introduction

OGC Web APIs are not Web Services in the traditional sense. Rather, they define the behavior and content of a set of Resources exposed through a Web Application Programming Interface (Web API). Therefore, an API may expose resources in addition to those defined by the standard. A test engine must be able to traverse the API, identify and validate test points, and ignore resource paths which are not to be tested.

The Conformance Classes addressed by this Abstract Test Suite are the:

- Collections Conformance Class
- Uniform Additional Dimensions Collections Conformance Class
- HTML Conformance Class
- JSON Conformance Class

A.2. Conformance Class Collections

CONFORMANCE CLASS A.1	
IDENTIFIER	https://www.opengis.net/spec/ogcapi-common-2/1.0/conf/collections
REQUIREMENTS CLASS	Requirements class 1: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections
TARGET TYPE	Web API
CONFORMANCE TESTS	Abstract test A.1: /conf/collections/collections-list-op Abstract test A.2: /conf/collections/collections-list-success Abstract test A.3: /conf/collections/collections-list-links

CONFORMANCE CLASS A.1
Abstract test A.4: /conf/collections/collections-list-collections Abstract test A.5: /conf/collections/description-op Abstract test A.6: /conf/collections/description-success Abstract test A.7: /conf/collections/description-links Abstract test A.8: /conf/collections/description-extent Abstract test A.9: /conf/collections/description-extent-multi

The Collections Conformance Class has a dependency on OGC API — Common — Part 1: Core's "Core" conformance class, implying a dependency on HTTP and optionally HTTPS protocols:

https://www.opengis.net/spec/ogcapi-common-1/1.0/conf/core

Conformance with the Collections Conformance Class is demonstrated by execution, in order, of all abstract tests it consists of:

A.2.1. List of Collections (/collections) Tests

ABSTRACT TEST A.1	
IDENTIFIER	/conf/collections/collections-list-op
REQUIREMENT	Requirement 1: /req/collections/collections-list-op
TEST PURPOSE	Validate that information about the list of Collections can be retrieved from the expected location.
TEST METHOD	1. Issue an HTTP GET request without query parameters to the URL {root}/collections 2. Validate that a document was returned with a status code 200 3. Validate the contents of the returned document using test /conf/collections/collections-list-success.

ABSTRACT TEST A.2	
IDENTIFIER	/conf/collections/collections-list-success
REQUIREMENT	Requirement 2: /req/collections/collections-list-success
TEST PURPOSE	Validate that the list of Collections content complies with the required structure and contents.
TEST METHOD	1. Validate the list of Collections resource for all supported media types using the resources and tests identified in Table A.1

The list of Collections content may be retrieved in a number of different formats. The following table identifies the applicable schema document for each format and the test to be used to validate against that schema. All supported formats should be exercised.

Table A.1 — Schema and Tests for Collections content

FORMAT	SCHEMA DOCUMENT	TEST ID
HTML	collections.yaml	/conf/html/content
JSON	collections.yaml	/conf/json/content

ABSTRACT TEST A.3	
IDENTIFIER	/conf/collections/collections-list-links
REQUIREMENT	Requirement 3: /req/collections/collections-list-links
TEST PURPOSE	Validate that the required links are included in the Collections document.
TEST METHOD	Verify that the response document includes: 1. a link to this response document (relation: self), 2. a link to the response document in every other media type supported by the server (relation: alternate). Verify that all links include the rel and type link parameters.

ABSTRACT TEST A.4	
IDENTIFIER	/conf/collections/collections-list-collections
REQUIREMENT	Requirement 4: /req/collections/collections-list-collections
TEST PURPOSE	Validate that each collection accessible through the API Implementation is described in the Collections document.
TEST METHOD	1. Verify that the Collections document includes a collections property. 2. Verify that the collections property is an array. 3. Verify that there is an entry in the collections property for each resource collection accessible through the API. 4. Verify that each entry in the collections array is valid according to /conf/collections/description-success.

A.2.2. Collection Description (/collections/{collectionId}) Tests

ABSTRACT TEST A.5	
IDENTIFIER	/conf/collections/description-op
REQUIREMENT	Requirement 5: /req/collections/description-op
TEST PURPOSE	Validate that the Collection description content can be retrieved from the expected location.
TEST METHOD	For every Collection described in the list of Collections resource at /collections, issue an HTTP GET request to the URL /collections/{collectionId} where {collectionId} is the id property for the collection. 1. Validate that a Collection description was returned with a status code 200 2. Validate the contents of the returned document using test /conf/collections/description-success.

ABSTRACT TEST A.6	
IDENTIFIER	/conf/collections/description-success
REQUIREMENT	Requirement 6: /req/collections/description-success
TEST PURPOSE	Validate that a Collection document complies with the required structure, contents and values.
TEST METHOD	For each Collection description (/collections/{collectionId}) resource, validate: 1. That the Collection description resource includes an id property. 2. Validate the content of the Collection resource for all supported media types using the resources and tests identified in Table A.2 3. Verify that the content of the response is consistent with the content for the corresponding collection in the response to the /collections resource. That is, the values for id, title, description and extent are identical.

The Collection description content may be retrieved in a number of different formats. The following table identifies the applicable schema document for each format and the test to be used to validate against that schema. All supported formats should be exercised.

Table A.2 — Schema and Tests for Collection content

FORMAT	SCHEMA DOCUMENT	TEST ID
HTML	collectionDesc.yaml	/conf/html/content
JSON	collectionDesc.yaml	/conf/json/content

ABSTRACT TEST A.7	
IDENTIFIER	/conf/collections/description-links
REQUIREMENT	Requirement 7: /req/collections/description-links
TEST PURPOSE	Validate that a Collection document includes all required links.
TEST METHOD	1. Verify that the Collection document includes a links property. 2. Verify that the links property includes an item which refers back to the Collection document (relation: self). 3. Verify that the links property includes an item for each supported encoding of this Collection document and that each of these items includes a href to an appropriate resource (relation: alternate). 4. Verify that all links include the rel and type link parameters.

ABSTRACT TEST A.8	
IDENTIFIER	/conf/collections/description-extent
REQUIREMENT	Requirement 8: /req/collections/description-extent
TEST PURPOSE	Validate the extent property if it is present
TEST METHOD	If the extent property is present, then: 1. Verify that the extent provides bounding boxes that include all spatial geometries in the collection. 2. Verify that the extent provides time intervals that include all temporal geometries in the collection. NOTE:A temporal extent of null indicates a half-bounded or unbounded time interval.

ABSTRACT TEST A.9	
IDENTIFIER	/conf/collections/description-extent-multi
REQUIREMENT	Requirement 9: /req/collections/description-extent-multi
TEST PURPOSE	Validate the consistency of multi-elements extents property if applicable
TEST METHOD	If the extent property is present, then: 1. Verify that for spatial properties having more than a single bounding box, all bounding boxes are included within the first bounding box. 2. Verify that for temporal properties (as well as for additional properties if conforming to Uniform Additional Dimensions Collections) having more than a single interval, all intervals are within the first interval. NOTE: A temporal extent of null indicates a half-bounded or unbounded time interval.

A.3. Conformance Class Uniform Additional Dimensions Collections

CONFORMANCE CLASS A.2	
IDENTIFIER	https://www.opengis.net/spec/ogcapi-common-2/1.0/conf/uad-collections
REQUIREMENTS CLASS	Requirements class 2: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/uad-collections
TARGET TYPE	Web API
CONFORMANCE TESTS	Abstract test A.10: /conf/uad-collections/extent-uad-definition Abstract test A.11: /conf/uad-collections/grid-description

A.3.1. Abstract Test for Uniform Additional Dimensions Collections Definition

ABSTRACT TEST A.10	
IDENTIFIER	/conf/uad-collections/extent-uad-definition
REQUIREMENT	Requirement 10: /req/uad-collections/extent-uad-definition

ABSTRACT TEST A.10	
TEST PURPOSE	To verify that the collection descriptions follow the uniform schema for describing data collections with additional dimensions beyond spatial and temporal
TEST METHOD	For each additional dimension beyond spatial and temporal: 1. Validate that the dimension property includes an interval property, 2. Validate that the dimension property includes a trs, vrs or definition, 3. Validate that the dimension property includes a unit if the trs, vrs or definition does not imply a specific unit.
DESCRIPTION	**NOTE:** The first two aspects of this validation can be performed by using the collectionDesc-UAD.yaml collection description YAML JSON Schema including extent-UAD.yaml schema instead of collectionDesc.yaml.

ABSTRACT TEST A.11	
IDENTIFIER	/conf/uad-collections/grid-description
REQUIREMENT	Requirement 11: /req/uad-collections/grid-description
TEST PURPOSE	Validate that the grid of gridded data is described
TEST METHOD	For each dimension of the extents, including spatial and temporal as well as additional dimensions: 1. If the data is known or described as being organized according to a specific regular or irregular grid, validate that the dimension property includes a grid property. 2. If the data is known or described as being organized according to a regular grid, validate that the grid property validates against regularGrid.yaml 3. If the data is known or described as being organized according to an irregular grid, validate that the grid property validates against irregularGrid.yaml
DESCRIPTION	**NOTE:** The grid property can be included whether an Implementation conforms to Uniform Additional Dimensions or not, but grid description is mandatory for gridded data when conforming to this Conformance Class.

A.4. Conformance Class JSON

CONFORMANCE CLASS A.3	
IDENTIFIER	https://www.opengis.net/spec/ogcapi-common-2/1.0/conf/json
REQUIREMENTS CLASS	Requirements class 4: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/json
TARGET TYPE	Web API
CONFORMANCE TESTS	Abstract test A.12: /conf/json/definition Abstract test A.13: /conf/json/content

A.4.1. JSON Definition

ABSTRACT TEST A.12	
IDENTIFIER	/conf/json/definition
REQUIREMENT	Requirement 14: /req/json/definition
TEST PURPOSE	Verify support for JSON
TEST METHOD	1. A resource is requested with response media type of application/json 2. All 200-responses SHALL support the media type application/json.

A.4.2. JSON Content

ABSTRACT TEST A.13	
IDENTIFIER	/conf/json/content
REQUIREMENT	Requirement 15: /req/json/content
TEST PURPOSE	Verify the content of a JSON document given an input document and schema.
TEST METHOD	1. Validate that the document is a JSON document. 2. Validate the document against the schema using a JSON Schema validator.

A.5. Conformance Class HTML

CONFORMANCE CLASS A.4	
IDENTIFIER	https://www.opengis.net/spec/ogcapi-common-2/1.0/conf/html
REQUIREMENTS CLASS	Requirements class 3: https://www.opengis.net/spec/ogcapi-common-2/1.0/req/html
TARGET TYPE	Web API
CONFORMANCE TESTS	Abstract test A.14: /conf/html/definition Abstract test A.15: /conf/html/content

A.5.1. HTML Definition

ABSTRACT TEST A.14	
IDENTIFIER	/conf/html/definition
REQUIREMENT	Requirement 12: /req/html/definition
TEST PURPOSE	Verify support for HTML
TEST METHOD	Verify that every 200-response of every operation of the API Implementation where HTML was requested is of media type text/html

A.5.2. HTML Content

ABSTRACT TEST A.15	
IDENTIFIER	/conf/html/content
REQUIREMENT	Requirement 13: /req/html/content
TEST PURPOSE	Verify the content of an HTML document given an input document and schema.
TEST METHOD	1. Validate that the document is an HTML 5 document 2. Manually inspect the document against the schema.

B

ANNEX B (INFORMATIVE) CLASS MODEL AS TABLES

ANNEX B (INFORMATIVE) CLASS MODEL AS TABLES

B.1. OGC API — Common package

B.1.1. OGC API — Common overview

B.1.2. Defining tables

Table B.1 — Elements of “OGC API — Common::” ()

NAME	
DEFINITION	
STEREOTYPE	interface
ABSTRACT	False
ASSOCIATIONS	(none)
ATTRIBUTES	(none)
CONSTRAINTS	(none)

Table B.2 — Elements of “OGC API — Common::OGC API — Common — Part 1” ()

NAME	OGC API — Common — Part 1
DEFINITION	
STEREOTYPE	interface

ABSTRACT	False
ASSOCIATIONS	(none)
ATTRIBUTES	(none)
CONSTRAINTS	(none)

Table B.3 — Elements of “OGC API — Common::OGC API — Common — Part 2” ()

NAME	OGC API — Common — Part 2
DEFINITION	
STEREOTYPE	interface
ABSTRACT	False
ASSOCIATIONS	(none)
ATTRIBUTES	(none)
CONSTRAINTS	(none)

Table B.4 — Elements of “OGC API — Common::OGC API — Records” ()

NAME	OGC API — Records
DEFINITION	
STEREOTYPE	interface
ABSTRACT	False
ASSOCIATIONS	(none)
ATTRIBUTES	(none)
CONSTRAINTS	(none)

Table B.5 — Elements of “OGC API — Common::Uniform Additional Dimensions” ()

NAME	Uniform Additional Dimensions
DEFINITION	
STEREOTYPE	interface
ABSTRACT	False
ASSOCIATIONS	(none)
ATTRIBUTES	(none)
CONSTRAINTS	(none)

Table B.6 — Elements of “OGC API — Common::Address” ()

NAME	Address		
DEFINITION	Postal address of the resource		
STEREOTYPE	interface		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	deliveryPoint [*]	String	Address lines for the location (e.g. street name and door number)
	city [0..1]	String	City for the location
	administrativeArea [0..1]	String	State or province of the location
	postalCode [0..1]	String	ZIP or other postal code
	country [0..1]	String	Country of the physical address. ISO 3166-1 is recommended
	role [*]	String	The type of postal address (e.g. office, home, etc.)
CONSTRAINTS	(none)		

Table B.7 — Elements of “OGC API — Common::BBox” (Array)

NAME	BBox		
DEFINITION	A sorted list of 4 (2D) or 6 (3D) coordinates		
STEREOTYPE	Array		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	item [4..6]	Real	A sorted list of 4 (2D) or 6 (3D) coordinates
CONSTRAINTS	(none)		

Table B.8 — Elements of “OGC API — Common::CollectionDescription” ()

NAME	CollectionDescription		
DEFINITION	Description of a collection. It is returned by the Web API at /collections/{collectionId}		
STEREOTYPE	interface		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	id [1]	String	Identifier of the collection (sometimes referred as collectionId). It is used in the Web API as path parameter in /collections/{collectionId}
	title [0..1]	String	A short, human-readable summary of the collection
	description [0..1]	String	A human-readable explanation about data in the collection
	attribution [0..1]	String	Provides a way to identify the source of the geographic information which can contain markup text whose format may be indicated in the

			attributionMediaType property. That format can be either plain text (text/plain), HTML (text/html) or Common Mark (text/markdown)
	attributionMediaType [0..1]	AttributionMediaTypeCode	Media type for the markup language of the attribution: It can be either plain text (text/plain), HTML (text/html) or CommonMark (text/markdown).
	accessContraints [0..1]	AccessContraintsCode	Restrictions on the availability of the collection that the user needs to be aware of before using or redistributing the data.
	parent [0..1]	String	An id of a collection that is the parent of this collection, if the collection is part of a hierarchy. In this case, this property allows to identify another collection within the same dataset as its parent (by setting parent to the id of that collection), enabling clients to construct a hierarchy. For example a rivers' collection can have an topographic collection as parent.
	publisher [0..1]	String	Organization or individual responsible for making the data available
	license [0..1]	String	The legal provisions under which the data of this collection is made available
	rights [0..1]	String	A statement that concerns all rights not addressed by the license such as a copyright statement
	keywords [*]	String	The topic or topics of the resource. Typically represented using free-form keywords, tags, key phrases, or classification codes
	itemType [0..1]	String	Indicator about the type of the items in the collection if the collection has an accessible /collections/{collectionId}/items endpoint. Examples are "feature" (default) of "record".
	dataType [0..1]	String	Nature of the data in the collection. For example, this could be set to map, vector or coverage. It is not an indication of the available data access mechanism, but a hint regarding the internal representation of the data.

featureType [*]	String	The list of feature (or coverage) type(s) which are present inside the collection. If the data is exposed as items each feature type is expected to share the same property types. For a coverage type it is related to the range type. This is useful for associating the relevant portrayal rules of a style.
geoDataClass [*]	anyURI	URIs identifying a class of data contained in the geospatial data (useful for example to determine compatibility with styles or processes)
defaultField [*]	String	A sorted list of field names. For a features data access mechanism , it is the list of returnable properties included by default and complementing the geometry of the features. For a coverage data access mechanism, they are names of fields whose values are included by default. Note that an API may provide a mechanism (e.g. a query parameter called 'properties' or 'exclude-properties') to override this default list in customized responses.
crs [*]	String, Class	The list of references to coordinate reference systems supported by the API expressed as full URIs or CURIEs; the first item is typically the default coordinate reference system. If no reference to point to can be found, it could also be a list of descriptions of CRSs as string or a full class using e.g. , a WKT2CRS string, or a PROJJSON, or CRS JSON class.
storageCrs [0..1]	String, Class	The native coordinate reference system (i.e., the most efficient CRS in which to request the data, possibly how the data is stored on the server); this is the default output coordinate reference system for Maps and Coverages. If no reference to point to can be found, it could also be a description of CRSs as string or a full class using e.g., a WKT2CRS string, or a PROJJSON, or CRS JSON class.

	epoch [0..1]	Real	Epoch of the native (storage) Coordinate Reference System (CRS)
	geometryDimension [0..1]	Integer	Number of spatial dimensions of the primary geometry of individual elements of the data: 0 for points, 1 for curves, 2 for surfaces, 3 for solids and unspecified when mixed or unknown. Not to be confused with the dimensions of the domain which are defined by the extent element.
	minScaleDenominator [0..1]	Real	Minimum scale denominator for usage of the collection
	maxScaleDenominator [0..1]	Real	Maximum scale denominator for usage of the collection
	minCellSize [0..1]	Real	Minimum cell size for usage of the collection in degrees of latitude for data referenced to Earth
	maxCellSize [0..1]	Real	Maximum cell size for usage of the collection in degrees of latitude for data referenced to Earth
	created [0..1]	DateTime	Timestamp indicating when the data in the collection was first produced
	updated [0..1]	DateTime	Timestamp of the last change/revision to the data in the collection
CONSTRAINTS	(none)		

Table B.9 — Elements of “OGC API — Common::Collections” ()

NAME	Collections		
DEFINITION	A set of geospatial resources that may be available as one or more sub-resource distributions. It is returned by the Web API at /collections		
STEREOTYPE	interface		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	numberMatched [0..1]	Integer	Number of elements in the response that match the selection parameters like bbox.

	numberReturned [0..1]	Integer	Number of elements in the response. Omitted unknown or difficult to compute
CONSTRAINTS	(none)		

Table B.10 — Elements of “OGC API — Common::Concept” ()

NAME	Concept		
DEFINITION	A concept defined in a vocabulary		
STEREOTYPE	interface		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	id [1]	String	Identifier for the concept in the knowledge system
	title [0..1]	String	A human readable title for the concept
	description [0..1]	String	A human readable description for the concept
	definition [0..1]	anyURI	A URI providing further description of the concept
CONSTRAINTS	(none)		

Table B.11 — Elements of “OGC API — Common::ConformanceClasses” ()

NAME	ConformanceClasses
DEFINITION	A declaration of the conformance classes that the API conforms to. It is returned by the Web API at /conformance
STEREOTYPE	interface
ABSTRACT	False
ASSOCIATIONS	(none)

	NAME	*TYPE*	*DEFINITION*
ATTRIBUTES	conformsTo [1..*]	anyURI	A conformance class URIs from the ones defined in the OGC API standard documents
CONSTRAINTS	(none)		

Table B.12 — Elements of “OGC API — Common::Contact” ()

NAME	Contact		
DEFINITION	Contact point of the resource. It can be a person or an organization		
STEREOTYPE	interface		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	*NAME*	*TYPE*	*DEFINITION*
	id [0..1]	String	A value uniquely identifying a contact
	name [0..1]	String	The name of the responsible person
	position [0..1]	String	The name of the role or position of the responsible person taken from the organization’s formal organizational hierarchy or chart
	organization [0..1]	String	Organization/affiliation of the contact
	hoursOfService [0..1]	String	Time period when the contact can be contacted
	contactInstructions [0..1]	String	Supplemental instructions on how or when to contact can be made
	role [*]	String	The set of named duties, job functions and/or permissions associated with this contact. (e.g. developer, administrator, etc.)
CONSTRAINTS	(none)		

Table B.13 — Elements of “OGC API — Common::Coordinates” (Array)

NAME	Coordinates

DEFINITION				
STEREOTYPE	Array			
ABSTRACT	False			
ASSOCIATIONS	(none)			
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***	
	item [1..*]	Real,String	(none)	
CONSTRAINTS	(none)			

Table B.14 — Elements of “OGC API — Common::Email” ()

NAME	Email		
DEFINITION	Email address of the resource		
STEREOTYPE	interface		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	value [1]	String	The value is the email itself
	role [*]	String	The type of email (e.g. home, work, etc.)
CONSTRAINTS	(none)		

Table B.15 — Elements of “OGC API — Common::Exception” ()

NAME	Exception
DEFINITION	A schema for exceptions based on RFC 7807. It is returned by the Web API in case of an error retrieving the API resources
STEREOTYPE	interface

ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	type [1..*]	String	A URI that identifies the problem type. When this member is not present, its value is assumed to be about:blank
	title [1]	String	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem
	status [1]	String	The HTTP status code generated by the server for this occurrence of the problem
	detail [1]	String	A human-readable explanation specific to this occurrence of the problem
	instance [1]	String	A URI reference that identifies the specific occurrence of the problem. It may or may not yield further information if dereferenced
CONSTRAINTS	(none)		

Table B.16 — Elements of “OGC API — Common::Extent” ()

NAME	Extent
DEFINITION	The extent of the resource with spatial, temporal optional Uniform Additional Dimensions (UAD) schema. The first item in the arrays for bbox or interval describe the overall extent of the data. All subsequent items describe more precise extents, e.g., to identify clusters of data.
STEREOTYPE	interface
ABSTRACT	False
ASSOCIATIONS	(none)
ATTRIBUTES	(none)
CONSTRAINTS	(none)

Table B.17 — Elements of “OGC API — Common::Fields” (Array)

NAME	Fields		
DEFINITION			
STEREOTYPE	Array		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	item [*]		(none)
CONSTRAINTS	(none)		

Table B.18 — Elements of “OGC API — Common::Format” ()

NAME	Format		
DEFINITION	Format of the resource		
STEREOTYPE	interface		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	name [0..1]	String	Name of the format
	mediaType [0..1]	String	Media type of the format
CONSTRAINTS	(none)		

Table B.19 — Elements of “OGC API — Common::Grid” (abstract)

NAME	Grid
DEFINITION	Provides information about the limited availability of data within the collection organized as a grid (regular or irregular) along the dimension.

STEREOTYPE	abstract		
ABSTRACT	True		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	cellsCount [1]	UnsignedInteger	Number of samples available along the dimension for data organized as a regular or irregular grid
CONSTRAINTS	(none)		

Table B.20 — Elements of "OGC API — Common::Interval" (type)

NAME	Interval		
DEFINITION	Commonly a sorted list of a pair of numbers, but it is string to support for example for a pair of dates in ISO format		
STEREOTYPE	type		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	item [2..2]	Real,String	(none)
CONSTRAINTS	(none)		

Table B.21 — Elements of "OGC API — Common::IrregularGrid" ()

NAME	IrregularGrid
DEFINITION	Irregular grid description using syncronized lists of coordinates and its bounds. For the spatial dimensions they are expressed in the 'storageCRS' of the collectionDescription. For TemporalExtent they are based in the trs. For OtherDimensions they are in 'trs' , 'vrs' or directly in the way specified in the 'definition'
STEREOTYPE	interface
ABSTRACT	False

ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	Coordinates [1..*]	Real, String	Sorted list of coordinates along the dimension for which data organized as an irregular grid in the collection is available (e.g., 2, 10, 80, 100).
	boundsCoordinates [*]	Interval	Ordered list of the coordinates in absolute units of the lower and upper bounds of the dimension for each cell. For the spatial dimensions they are expressed in the 'storageCRS' of the collection Description. For TemporalExtent they are based in the trs. For OtherDimensions they are in 'trs' , 'vrs' or directly in the way specified in the 'definition'
CONSTRAINTS	(none)		

Table B.22 — Elements of "OGC API — Common::LandingPage" ()

NAME	LandingPage		
DEFINITION	Provides the information needed to navigate all the resources exposed through the API. It is returned by the Web API at the root node Web API endpoint.		
STEREOTYPE	interface		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	title [*]	String	The title of the API. While a title is not required, implementors are strongly advised to include one
	description [0..1]	String	A longer description of the API.
	attribution [0..1]	String	Provides a way to identify the source of the geographic information provided by the instance of the Web API. The attribution of the Web API should be short and intended for presentation to a user, for example, in a corner of a map. Parts of the text can be links to other resources if additional

			information is needed. The string can include HTML markup or markdown.
	attributionMediaType [0..1]	AttributionMediaTypeCode	The 'attribution' can contain markup text whose format may be indicated in the attributionMediaType property (e.g. , text/html for HTML or text/markdown for CommonMark). If the 'attribution MediaType' indicates something other than text/plain, the attribution element string should be interpreted by a markup parser selected based on that media type to be presented to the user (e.g., text/markdown will be parsed by a library supporting CommonMark). By allowing markup, the attribution string can import images (e.g., organization logos) and format the text (e.g., the name of the organization in italics).
CONSTRAINTS	(none)		

Table B.23 — Elements of "OGC API — Common::Link" (type)

NAME	Link		
DEFINITION	An expression of a relationship between resources.		
STEREOTYPE	type		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	href [1]	String	URI to a remote resource (or resource fragment)
	rel [1]	String	The type or semantics of the relation
	type [0..1]	String	A hint indicating what the media type of the result of dereferencing the link should be.
	hreflang [0. .1]	String	A hint indicating what the language of the result of dereferencing the link should be.
	title [0..1]	String	Used to label the destination of a link such that it can be used as a human-readable description

	length [0..1]	Integer	A hint indicating the Content-length of the result of dereferencing the link should be
CONSTRAINTS	(none)		

Table B.24 — Elements of “OGC API — Common::OtherDimensionExtent” ()

NAME	OtherDimensionExtent		
DEFINITION	Other dimension definition and extent. First Interval is the overall interval and additional ones are spaced within the overall Interval		
STEREOTYPE	interface		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	interval [1..*]	Interval	Sorted list of one or more intervals that describe the extent for this dimension of the dataset. The value null is supported and indicates an unbounded or half-bounded interval. The first interval describes the overall extent of the data for this dimension. All subsequent intervals describe more precise intervals, e.g., to identify clusters of data. Clients only interested in the overall extent will only need to access the first item (a pair of lower and upper bound values).
	trs [0..1]	String, Class	Temporal Coordinate Reference System (e.g. as defined by Features for ‘temporal’). This must be a URI to a registered TRS if one is available, or a fully description.
	vrs [0..1]	String, Class	Vertical Coordinate Reference System (e.g. as defined in EDR for ‘vertical’). This must be a URI to a registered VRS if one is available, or a fully description.
	definition [0..1]	anyURI	A URI to the definition of the measured or observed property corresponding to this dimension
	unit [0..1]	String	The unit of measure in which the interval and/or grid values are expressed
	unitLang [0..1]	anyURI	The language (or vocabulary) in which the unit is expressed (defaults to “UCUM” if not specified)
	variableType [0..1]	VariableTypeCode	Class of variable, distinguishing between continuous vs. discrete, numerical vs. categorical, and ordinal vs. nominal. Continuous variables (continuous)

			represent an infinite spectrum of values (such as spatial coordinates or temporal instants), while discrete variables (numericalOrdinal, numericalNominal, categoricalOrdinal, and categoricalNominal) represent a finite or countable set of distinct values.
CONSTRAINTS	(none)		

Table B.25 — Elements of “OGC API — Common::Phone” ()

NAME	Phone		
DEFINITION	Phone or Fax of the resource		
STEREOTYPE	interface		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	value [1]	String	The value is the phone number itself
	role [*]	String	The type of phone number (e.g. home, work, fax, etc.)
CONSTRAINTS	(none)		

Table B.26 — Elements of “OGC API — Common::RegularGrid” ()

NAME	RegularGrid		
DEFINITION	Resolution of regularly gridded data along the dimension in the collection.		
STEREOTYPE	interface		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	firstCoordinate [1]	Real, String	First coordinate where a regular grid begins, with subsequent coordinates adding resolution unit at each

			step. Commonly is a number but can be a string to support date first coordinates. For the spatial dimensions they are expressed in the 'storageCRS' of the collectionDescription. For TemporalExtent they are based in the trs. For the Other Dimensions they are in 'trs' , 'vrs' or directly in the way specified in the 'definition'
	resolution [1]	Real, String	Resolution of regularly gridded data along the dimension in the collection. Commonly is a number but can be a string to support date resolution. For the spatial dimensions they are expressed in the collectionDescription 'storageCRS' units. For TemproalExtent they are based in the trs units. For the OtherDimensions they are in 'trs' units , 'vrs' units or directly in 'unit'
	relativeBounds [0..1]	Interval	Distance in units (for the spatial dimensions they are the collectionDescription 'storageCRS' units; for TemproalExtent they are based in the trs units; for the OtherDimensions they are in 'trs' units , 'vrs' units or directly in 'unit') from coordinate to the lower and upper bounds of each cell for regular grids, describing the geometry of the cells.
	availableResolution [*]	Real, String	List of resolutions made available by the API. If it is not indicated, there are no restrictions in resolutions other that the limits indicated in the 'minCellSize' and 'maxCellSize' of the collection description. For the spatial dimensions they are expressed in the collectionDescription 'storageCRS' units. For TemproalExtent they are based in the trs units. For the OtherDimensions they are in 'trs' units , 'vrs' units or directly in 'unit'
CONSTRAINTS	(none)		

Table B.27 — Elements of "OGC API — Common::ResourceLanguage" ()

NAME	ResourceLanguage		
DEFINITION	Language of the resource.		
STEREOTYPE	interface		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	code [1]	String	The language tag as per RFC-5646

	name [0..1]	String	The untranslated name of the language
	alternate [0..1]	String	The name of the language in another well-understood language, usually English
	dir [0..1]	LanguageDirectionCode	The direction for text in this language. The default, ltr (left-to-right), represents the most common situation. However, care should be taken to set the value of dir appropriately if the language direction is not ltr. Other values supported are rtl (right-to-left), ttb (top-to-bottom), and btt (bottom-to-top)
CONSTRAINTS	(none)		

Table B.28 — Elements of "OGC API — Common::TemporalExtent" ()

NAME	TemporalExtent		
DEFINITION	The temporal reference system and extent. First Interval is the overall interval and additional ones are spaced within the overall Interval		
STEREOTYPE	interface		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	trs [0..1]	String, Class	Reference system of the temporal extent (property interval). The default reference system is the Gregorian calendar. In the Core this is the only supported temporal reference system. Extensions may support additional temporal reference systems. This must be a URI to a registered TRS if one is available, or a fully description.
	interval [*]	Interval	One or more time intervals that describe the temporal extent of the dataset. In the Core only a single time interval is supported. Extensions may support multiple intervals. The first time interval describes the overall temporal extent of the data. All subsequent time intervals describe more precise time intervals, e.g., to identify clusters of data. Clients only interested in the overall extent will only need to access the first item in each array.
CONSTRAINTS	(none)		

Table B.29 — Elements of “OGC API — Common::SpatialExtent” ()

NAME	SpatialExtent		
DEFINITION	The spatial dimension and extent. ‘bbox’ is in ‘crs’. ‘storageCrsBox’ is in the storageCrs of the parent class. For bbox and storageCrsBbox, first BBox is the overall extent and additional ones are spaced within the overall extent. If no reference to point to can be found, it could also be a list of descriptions of CRSs as string or a full class e.g., a WKT2CRS string, or a PROJJSON, or CRS JSON class.		
STEREOTYPE	interface		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	crs [0..1]	String,Class	Coordinate reference system of the coordinates of the bbox property. For data referenced to Earth only the default value is allowed. The default reference system is CRS84 or CRS84h for coordinates with height. For non-terrestrial coordinate reference system, another CRS may be specified
	bbox [*]	BBox	Sorted list of one or more bounding boxes that describe the spatial extent of the dataset. The first bounding box describes the overall spatial extent of the data. All subsequent bounding boxes describe more precise bounding boxes, e.g., to identify clusters of data. Clients only interested in the overall spatial extent will only need to access the first item in each array.
	storageCrsBbox [*]	BBox	One or more bounding boxes that describe the spatial extent of the dataset in the storage (native) CRS (storageCrs property in the collection Desc). The first bounding box describes the overall spatial extent of the data. All subsequent bounding boxes describe more precise bounding boxes, e.g., to identify clusters of data. Clients only interested in the overall spatial extent will only need to access the first item in each array.
CONSTRAINTS	(none)		

Table B.30 — Elements of “OGC API — Common::Theme” ()

NAME	Theme		
DEFINITION	A theme defined by a concept in an scheme of concepts.		
STEREOTYPE	interface		
ABSTRACT	False		
ASSOCIATIONS	(none)		
ATTRIBUTES	***NAME***	***TYPE***	***DEFINITION***
	scheme [0..1]	String	An identifier for the knowledge organization system used to classify the resource. It is recommended that the identifier be a resolvable URI. The list of schemes used in a searchable catalog can be determined by inspecting the server’s OpenAPI document or, if the server implements CQL2, by exposing a queryable (e.g. named scheme) and enumerating the list of schemes in the queryable’s schema definition.
CONSTRAINTS	(none)		

Table B.31 — Definition table of “OGC API — Common::AccessConstraintsCode” (enumeration)

NAME:	AccessConstraintsCode	
DEFINITION:	Enumeration of contraints	
STEREOTYPE:	enumeration	
ABSTRACT:	False	
ASSOCIATIONS:	(none)	
VALUES:	*Name*	*Definition*
	confidential	Confidential access constraint
	secret	secret access constraint
	restricted	Restricted access constraint
	topSecret	Top secret access constraint

	unclassified	Unclassified access constraint

Table B.32 — Definition table of “OGC API — Common::AttributionMediaTypeCode” ()

NAME:	AttributionMediaTypeCode	
DEFINITION:	Enumeration of media types for attribution	
STEREOTYPE:	interface	
ABSTRACT:	False	
ASSOCIATIONS:	(none)	
VALUES:	*Name*	*Definition*
	text/plain	Plain text
	text/html	Hypertext markup language
	text/markdown	Mark down text format

Table B.33 — Definition table of “OGC API — Common::LanguageDirectionCode” (enumeration)

NAME:	LanguageDirectionCode	
DEFINITION:	Enumeration of language directions.	
STEREOTYPE:	enumeration	
ABSTRACT:	False	
ASSOCIATIONS:	(none)	
VALUES:	*Name*	*Definition*
	ltr	left to right
	rtl	right to left
	ttb	Top to bottom

	btt	Bottom to top

Table B.34 — Definition table of “OGC API — Common::VariableTypeCode” ()

NAME:	VariableTypeCode	
DEFINITION:	Enumeration of Variable types. The class of variable, distinguishing between continuous vs. discrete, numerical vs. categorical, and ordinal vs. nominal. Continuous variables (continuous) represent an infinite spectrum of values (such as spatial coordinates or temporal instants), while discrete variables (numericalOrdinal, numericalNominal, categoricalOrdinal, and categoricalNominal) represent a finite or countable set of distinct values. Nominal variables are unordered, while ordinal variables imply a specific order. Numerical variables are identified with integer or real numbers, while categorical variables are identified with textual strings. The variable type informs correct interpretation and interpolation method.	
STEREOTYPE:	interface	
ABSTRACT:	False	
ASSOCIATIONS:	(none)	
VALUES:	*Name*	*Definition*
	continuous	Continuous variable representing an infinite spectrum of values (such as spatial coordinates or temporal instants)
	numericalOrdinal	Numerical ordinal variables are discrete variables, identified with integer or real numbers implying a specific order.
	numericalNominal	Numerical nominal variables are discrete variables, identified with unordered integer or real numbers.
	categoricalOrdinal	Categorical ordinal variables are discrete variables representing a finite or countable set of distinct values implying a specific order.
	categoricalNominal	Categorical nominal variables are discrete variables representing a finite or countable set of distinct unordered values.

C

ANNEX C (INFORMATIVE) GLOSSARY

ANNEX C (INFORMATIVE) GLOSSARY

Conformance Test Module
set of related tests, all within a single conformance test class (OGC 08-131r3)

NOTE 1: When no ambiguity is possible, the word test may be omitted. i.e. conformance test module is the same as conformance module. Conformance modules may be nested in a hierarchical way.
This term and those associated to it are included here for consistency with ISO 19105.

Conformance Test Class; Conformance Test Level
set of conformance test modules that must be applied to receive a single **certificate of conformance**. (OGC 08-131r3)

NOTE 2: When no ambiguity is possible, the word test may be left out, so conformance test class may be called a conformance class.

Executable Test Suite (ETS)
A set of code (e.g. Java and CTL) that provides runtime tests for the assertions defined by the ATS. Test data required to do the tests are part of the ETS (OGC 08-134)

Recommendation
expression in the content of a document conveying that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited (OGC 08-131r3)

NOTE 3: "Although using normative language, a recommendation is not a requirement. The usual form replaces the shall (imperative or command) of a requirement with a should (suggestive or conditional)." (ISO Directives Part 2)

Requirement
expression in the content of a document conveying criteria to be fulfilled if compliance with the document is to be claimed and from which no deviation is permitted (OGC 08-131r3)

Requirements Class
aggregate of all requirement modules that must all be satisfied to satisfy a conformance test class (OGC 08-131r3)

Requirements Module
aggregate of requirements and recommendations of a specification against a single standardization target type (OGC 08-131r3)

Standardization Target
entity to which some requirements of a standard apply (OGC 08-131r3)

NOTE 4: The standardization target is the entity which may receive a certificate of conformance for a requirements class.

D

ANNEX D (NORMATIVE) BACKUS-NAUR FORMS

ANNEX D (NORMATIVE) BACKUS-NAUR FORMS

D.1. BNF for URI

The following Augmented Backus-Naur Form (ABNF) is from Appendix A of IETF RFC 3986.

```
URI         = scheme ":" hier-part [ "?" query ] [ "#" fragment ]

hier-part     = "//" authority path-abempty
              / path-absolute
              / path-rootless
              / path-empty

URI-reference = URI / relative-ref

absolute-URI  = scheme ":" hier-part [ "?" query ]

relative-ref  = relative-part [ "?" query ] [ "#" fragment ]

relative-part = "//" authority path-abempty
              / path-absolute
              / path-noscheme
              / path-empty

scheme        = ALPHA *( ALPHA / DIGIT / "+" / "-" / "." )

authority     = [ userinfo "@" ] host [ ":" port ]
userinfo      = *( unreserved / pct-encoded / sub-delims / ":" )
host          = IP-literal / IPv4address / reg-name
port          = *DIGIT

IP-literal    = "[" ( IPv6address / IPvFuture  ) "]"

IPvFuture     = "v" 1*HEXDIG "." 1*( unreserved / sub-delims / ":" )

IPv6address   =                            6( h16 ":" ) ls32
              /                       "::" 5( h16 ":" ) ls32
              / [              h16 ] "::" 4( h16 ":" ) ls32
              / [ *1( h16 ":" ) h16 ] "::" 3( h16 ":" ) ls32
              / [ *2( h16 ":" ) h16 ] "::" 2( h16 ":" ) ls32
              / [ *3( h16 ":" ) h16 ] "::"    h16 ":"   ls32
              / [ *4( h16 ":" ) h16 ] "::"              ls32
              / [ *5( h16 ":" ) h16 ] "::"              h16
              / [ *6( h16 ":" ) h16 ] "::"

h16           = 1*4HEXDIG
ls32          = ( h16 ":" h16 ) / IPv4address
IPv4address   = dec-octet "." dec-octet "." dec-octet "."
```

```
dec-octet     = DIGIT               ; 0-9
              / %x31-39 DIGIT       ; 10-99
              / "1" 2DIGIT          ; 100-199
              / "2" %x30-34 DIGIT   ; 200-249
              / "25" %x30-35        ; 250-255

reg-name      = *( unreserved / pct-encoded / sub-delims )

path          = path-abempty    ; begins with "/" or is empty
              / path-absolute   ; begins with "/" but not "//"
              / path-noscheme   ; begins with a non-colon segment
              / path-rootless   ; begins with a segment
              / path-empty      ; zero characters

path-abempty  = *( "/" segment )
path-absolute = "/" [ segment-nz *( "/" segment ) ]
path-noscheme = segment-nz-nc *( "/" segment )
path-rootless = segment-nz *( "/" segment )
path-empty    = 0<pchar>

segment       = *pchar
segment-nz    = 1*pchar
segment-nz-nc = 1*( unreserved / pct-encoded / sub-delims / "@" )
              ; non-zero-length segment without any colon ":"

pchar         = unreserved / pct-encoded / sub-delims / ":" / "@"

query         = *( pchar / "/" / "?" )

fragment      = *( pchar / "/" / "?" )

pct-encoded   = "%" HEXDIG HEXDIG

unreserved    = ALPHA / DIGIT / "-" / "." / "_" / "~"
reserved      = gen-delims / sub-delims
gen-delims    = ":" / "/" / "?" / "#" / "[" / "]" / "@"
sub-delims    = "!" / "$" / "&" / "'" / "(" / ")"
              / "*" / "+" / "," / ";" / "="
```

Listing D.1

D.2. BNF for Date-Time

The following Augmented Backus-Naur Form (ABNF) is from IETF RFC 3339.

```
date-fullyear  = 4DIGIT
date-month     = 2DIGIT  ; 01-12
date-mday      = 2DIGIT  ; 01-28, 01-29, 01-30, 01-31 based on month/year
time-hour      = 2DIGIT  ; 00-23
time-minute    = 2DIGIT  ; 00-59
time-second    = 2DIGIT  ; 00-58, 00-59, 00-60 based on leap second rules
time-secfrac   = "." 1*DIGIT
time-numoffset = ("+" / "-") time-hour ":" time-minute
time-offset    = "Z" / time-numoffset
partial-time   = time-hour ":" time-minute ":" time-second [time-secfrac]
full-date      = date-fullyear "-" date-month "-" date-mday
```

```
full-time      = partial-time time-offset
date-time      = full-date "T" full-time
```

Listing D.2

Note that unlike ISO 8601, the local time zone offset is required by RFC 3339.

E

ANNEX E (INFORMATIVE) REVISION HISTORY

ANNEX E (INFORMATIVE) REVISION HISTORY

Table E.1

DATE	RELEASE	EDITOR	PRIMARY CLAUSES MODIFIED	DESCRIPTION
2021-10-06	0.0.9	Charles Heazel	all	SWG review draft.
2024-04-18	0.0.10	Jérôme St-Louis	all	SWG review draft.
2025-09-18	0.9.0	Jérôme St-Louis, Joan Masó	all	Version ready for OAB review
2025-12-04	1.0.0rc1	Jérôme St-Louis, Joan Masó	all	Addressing Carl's comments, document ready for RFC.
2026-02-12	1.0.0rc2	Jérôme St-Louis, Joan Masó	all	Addressing comments received during the public comments period.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Peter Baumann, Open Geospatial Consortium: OGC 07-011r2, *Topic 6.1 — Schema for Coverage Geometry and Functions – Part 1: Fundamentals*. Open Geospatial Consortium (2024). http://www.opengis.net/doc/AS/Topic-6.1/2.0.

[2] John Herring, Open Geospatial Consortium: OGC 06-103r4, *OpenGIS Implementation Specification for Geographic information — Simple feature access — Part 1: Common architecture*. Open Geospatial Consortium (2011). http://www.opengis.net/doc/is/sfa/1.2.1.

[3] Policy SWG, Open Geospatial Consortium: OGC 08-131r3, *The Specification Model — Standard for Modular specifications*. Open Geospatial Consortium (2009). https://docs.ogc.org/pol/08-131r3/08-131r3/pdf.

[4] Peter Baumann, Eric Hirschorn, Joan Masó, Open Geospatial Consortium: OGC 09-146r6, *OGC Coverage Implementation Schema*. Open Geospatial Consortium (2017). http://www.opengis.net/doc/IS/cis/1.1.0.

[5] Overell P: IETF RFC 5234, *Augmented BNF for Syntax Specifications: ABNF*. RFC Publisher (2008). https://www.rfc-editor.org/info/rfc5234.

[6] International Organization for Standardization (committee): ISO 15836-2:2019, *Information and documentation — The Dublin Core metadata element set — Part 2: DCMI Properties and classes*. International Organization for Standardization, Geneva (2019). https://www.iso.org/standard/71341.html.

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

[8] International Organization for Standardization (committee): ISO 19107:2019, *Geographic information — Spatial schema*. International Organization for Standardization, Geneva (2019). https://www.iso.org/standard/66175.html.

[9] International Organization for Standardization (committee): ISO 19108:2002, *Geographic information — Temporal schema*. International Organization for Standardization, Geneva (2002). https://www.iso.org/standard/26013.html.

[10] International Organization for Standardization (committee): ISO 19111:2019, *Geographic information — Referencing by coordinates*. International Organization for Standardization, Geneva (2019). https://www.iso.org/standard/74039.html.

[11] International Organization for Standardization (committee): ISO 19168-1:2020, *Geographic information — Geospatial API for features — Part 1: Core*. International

Organization for Standardization, Geneva (2020). https://www.iso.org/standard/32586.html.

[12] W3C: Data Catalog Vocabulary (DCAT) — Version 2, W3C Recommendation, 04 February 2020, https://www.w3.org/TR/vocab-dcat-2/

[13] W3C: Data on the Web Best Practices, W3C Recommendation, 31 January 2017, https://www.w3.org/TR/dwbp/

[14] Open Geospatial Consortium: Welcome To The OGC APIs [online, viewed 2020-03-16]. Available at https://www.ogcapi.org/.

[15] Open Geospatial Consortium: OGC Link Relations Registry, [online, viewed 2024-11-20]. Available at https://www.opengis.net/def/rel

[16] W3C/OGC: Spatial Data on the Web Best Practices, W3C Working Group Note, 28 September 2017, https://www.w3.org/TR/sdw-bp/

[17] YAML Ain't Markup Language [online, viewed 2020-03-16]. Edited by O. Ben-Kiki, C. Evans, Ingy döt Net. Available at https://yaml.org.

[18] Open Geospatial Consortium: Compliance Testing Program Policies & Procedures [online, viewed 2020-04-18]. Available at https://portal.ogc.org/files/?artifact_id=28982.

[19] IANA: Link Relation Types, https://www.iana.org/assignments/link-relations/link-relations.xml

[20] Fielding, Roy Thomas: Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California, Irvine, 2000, https://www.ics.uci.edu/fielding/pubs/dissertation/fielding_dissertation.pdf