Open Geospatial Consortium

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: <2020-08-18>

External identifier of this OGC® document: http://www.opengis.net/doc/EG/ogcapi-common/1.0

Internal reference number of this OGC® document: 20-071

Version: n.n

Category: OGC® Engineering Guidance

Editor: Charles Heazel

OGC API-Common Users Guide

Warning

This document defines an the OGC Best Practices on the implementation and use of the OGC API-Common standard. This document is not an OGC Standard and may not be referred to as an OGC Standard. It is subject to change without notice. However, this document is an official position of the OGC membership on this particular technology topic.

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

Document type:	OGC® Engineering Guidance
Document subtype:	
Document stage:	Draft
Document language:	English

License Agreement

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

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

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

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

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

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

Table of Contents

i. Abstract

<Insert Abstract Text here>

ii. Keywords

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

ogcdoc, OGC document, <tags separated by commas>

iii. Preface

NOTE

Insert Preface Text here. Give OGC specific commentary: describe the technical content, reason for document, history of the document and precursors, and plans for future work. > Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

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

iv. Submitting organizations

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

Organization name(s)

v. Submitters

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

Name Affiliation

Chapter 1. Scope

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

Chapter 2. Conformance

This Best Practice defines XXXX.

Requirements for N target types are considered: * AAAA * BBBB

Conformance with this Best Practice shall be checked using all the relevant tests specified in Annex A (normative) of this document.

In order to conform to this OGC® Best Practice, a software implementation shall choose to implement: * Any one of the conformance levels specified in Annex A (normative). * Any one of the Distributed Computing Platform profiles specified in Annexes TBD through TBD (normative).

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

Chapter 3. References

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

Insert References here. If there are no references, state “There are no normative references”.

References are to follow the Springer LNCS style, with the exception that optional information may be appended to references: DOIs are added after the date and web resource references may include an access date at the end of the reference in parentheses. See examples from Springer and OGC below.

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

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

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

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

NOTE

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

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

ISO / TC 211: ISO 19115-1:2014 Geographic information — Metadata — Part 1: Fundamentals (2014)

ISO / TC 211: ISO 19157:2013 Geographic information — Data quality (2013)

ISO / TC 211: ISO 19139:2007 Geographic information — Metadata — XML schema implementation (2007)

ISO / TC 211: ISO 19115-3: Geographic information — Metadata — Part 3: XML schemas (2016)

OGC: OGC 15-097 OGC Geospatial User Feedback Standard. Conceptual Model (2016)

OGC: OGC 12-019, OGC City Geography Markup Language (CityGML) Encoding Standard (2012)

OGC: OGC 14-005r3, OGC IndoorGML (2014)

Chapter 4. Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r8], 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 Best Practice.

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

4.1. term name

text of the definition

Chapter 5. Conventions

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

5.1. Identifiers

The normative provisions in this document are denoted by the URI

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

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

Chapter 6. Users Guide

The OGC API Users Guide document contains information useful to a developer or user of OGC API-Common. This information in not normative. That is, it is not mandatory. However, it may prove essentiall to fully understand the normative text in the OGC API - Common Standards.

6.1. Architecture

The following "architecture" captures the core concepts which inform all OGC API standards.

6.1.1. Datasets

Web APIs implement a Resource Oriented approach to Web-based distributed computing. A coherent set of APIs must be based on a common understading of the Resources to be shared. For OGC APIs, those resources would be identified using an abstract concept of a geospatial dataset. Such a dataset would have the following characteristics:

- A dataset can be a vector feature collection, a coverage/imagery, or a collection of datasets (therefore potentially a mix of any of these things)
- A dataset has associated metadata, including some essential information:
  - Information about what type of data set it is (vector features (and what type of vector features if limited to one type, e.g. polygons,lines, points), coverages (values) / imagery (pixels), or sub-datasets-- more than one of those things)
  - A textual identifier (e.g. which figures in the resource path)
  - A title (short name / description)
  - Access point for the dataset (could be hosted locally or remote)
  - Geospatial & temporal extent
  - Resolution/scale
  - Units/Range/Bit-Depth/Channels/Dimensions etc. for imagery/coverages
  - A description of queryables, if applicable
- Keywords/Tags, and longer descriptions are also a commonly useful piece of metadata information
- Any other ISO 19115 metadata fields can also be associated with the dataset, but are nowhere near as essential to discovering and using geospatial data as those mentioned above. Meta data containing at minimum those essential elements can always be retrieved in ISO 19115 and potentially other formats.

6.1.2. Processes

Processes take one or more datasets as input, add parameters, and produce one or more datasets as output. This ties the processes together with the data delivery services on both ends.

I suggested so far 3 kinds of processes which can all run on a server where the data lives:

1. The complex process built as a container or executable, as typical of WPS
2. Process description languages such as WCPS
3. Pre-defined named processes such as 'vectorization', 'buffering', 'rasterization' or 'rendering of a styled map'

All of these kinds of processes could share aspects such as taking in an OGC API dataset as input and their output being usable as an OGC API dataset, for direct access and/or asynchronous delivery, and support multiple data partitioning/access mechanisms, estimates/billing elements, and so on.

6.1.3. Server-side rendering

Highlighting here that the rendering of a styled maps based on multiple source data sets and a style as a parameter, outputting a styled 'rendered map' imagery dataset as a result, fits perfectly well the description of a process (3).

6.1.4. Data partitioning

For all of data access / exchange mechanisms, e.g. to retrieve coverages/vector features directly, for a process to access its input, or to retrieve the output from a process, or throughout a daisy chain of processes, there are a variety of ways in which data can be partitioned for efficient access. The most efficient way most likely depends on the overall workflow and the implementations on both ends. I am suggesting most of the OGC API should be agnostic of this and support many such ways, so that both ends of such connection or a workflow manager can negotiate the best approach. Examples of different ways to partition/access data include:

- Bounding boxes
- Tiles
- n-dimensional sub-setting
- DGGS cells
- Single point value

This is the overall diagram of these ideas that I presented a couple weeks back at the OWS Common telecon and got the chance to explain in more details in person to some of you:

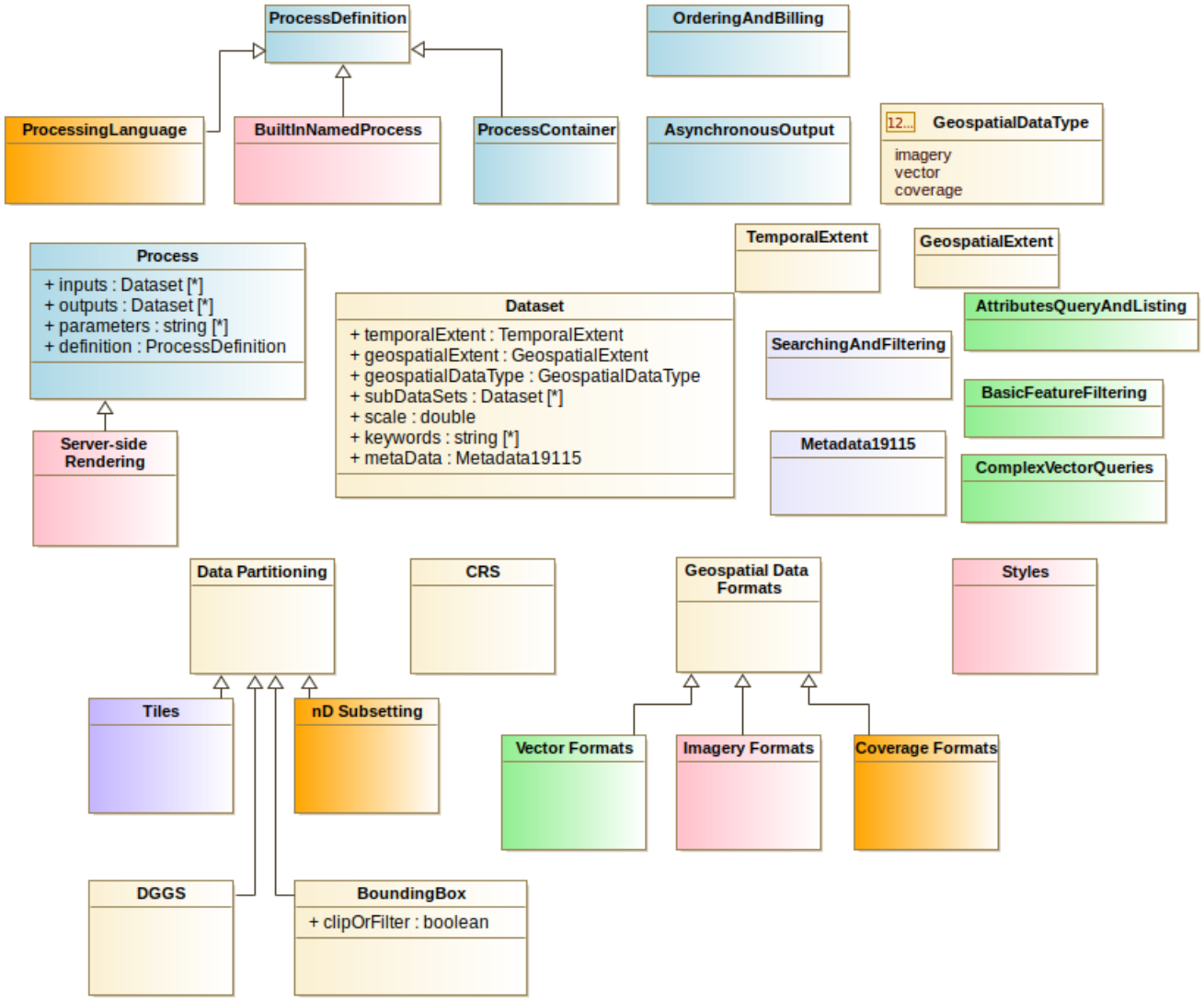

Figure 1. Dataset Concept

You might notice that this is color-coded based on where these API blocks originated from in classic services:

- Orange: Coverages (WCS)
- Red: Server-side rendering (WMS)
- Blue: Processes (WPS)
- Green: Vector features (WFS)
- Very light blue: Catalogs (CSW)
- Violet: Tiles (WMTS)

6.1.5. Resource Paths

The resources exposed through OGC APIs are accessed through standardized URL templates. These "paths" and their associated resources are described in Table 1. This table documents the paths and resources as of December 2019. This information will be updated as these standards mature.

Table 1. OGC Dataset Paths

Path Template	Resource
Common	
/collections	Metadata describing the spatial collections available from this API.
/collections/{collectionId}	Metadata describing the collection which has the unique identifier `{collectionId}`
Features	
/collections/{collectionId}	The Feature Collection resource identified by the `{collectionId}` parameter.
/collections/{collectionId}/items	The individual Features in a Feature Collection
Coverages	
/collections/{collectionId}/coverage	A general description of the coverage identified by {collectionId} including the coverage's envelope.
/collections/{collectionid}/coverage/description	returns the whole coverage description consisting of domainset, rangetype, and metadata (but not the rangeset)
/collections/{collectionid}/coverage/domainset	returns the coverage's domain set definition
/collections/{collectionid}/coverage/rangetype	returns the coverage's range type information (i.e., a description of the data semantics)
/collections/{collectionid}/coverage/metadata	returns the coverage's metadata (may be empty)
/collections/{collectionid}/coverage/rangeset	returns the coverage's range set, i.e., the actual values in the coverage's Native Format (see format encoding for ways to retrieve inspecific formats)
/collections/{collectionid}/coverage/all	returns all of the above namely the coverage's domainset, rangetype, meatadata, and rangeset comparable to a GetCoverage response
Maps and Styles	
Note: A map is associated with a resource. {resource} is a place holder for a path segment appropriate for a resource type.	
/collections/{collectionId}/{resource}/map	A data structure with specific information necessary to get a fragment of the map representing the resource collection.
/collections/{collectionId}/{resource}/map/{styleId}	A map representing the geospatial resource identified by /collections/{collectionId}/{resource}.
Tiles	
Note: A tile is associated with a resource. {resource} is a place holder for a path segment appropriate for a resource type.	

Path Template	Resource
/collections/{collectionId}/{resource}/tile/{styleId}/{tileMatrixSetId}/{tileMatrixId}/{tileRow}/{tileCol}	Tile representation of real-world elements at a given resolution restricted by the selected Tile Matrix Set. {styleId} is optional.
/collections/{collectionId}/{resource}/tiles	An enumeration of the Tiles and Styles that are available at some TileMatrixSetId
/tileMatrixSet/{tileMatrixSetId}	A description of the TileMatrixSet identified by the {tileMatrixSetId} identifier.
Processes	
/process	Lists the processes this API offers.
/processes/{process-id}	Returns a detailed description of a process.
/processes/{process-id}/jobs	Returns the running and finished jobs for a process (GET), Executes a process, i.e. creates a new job. Inputs and outputs will have to be specified in a JSON document that needs to be send in the POST body. (POST)
/processes/{process-id}/jobs/{job-id}	Returns the status of a job of a process.
/processes/{process-id}/jobs/{job-id}/results	Returns the result of a job of a process.
Records (Catalog)	
/collections/{collectionId}/tbd	TBD

6.2. Modular APIs

This section of the Best Practice discusses OGC principals and practices regarging modular APIs.

6.3. Link Relations

provides a discussion of link relations, the governing policies and standards, as well as how they should be used.

Links between elements of an OGC API are encoded using the link relation schema provided in Figure 1.

Link Schema

```
type: object
required:
  - href
  - rel
properties:
  href:
    type: string
    example: http://data.example.com/buildings/123
  rel:
    type: string
    example: alternate
  type:
    type: string
    example: application/geo+json
  hreflang:
    type: string
    example: en
  title:
    type: string
    example: Trierer Strasse 70, 53115 Bonn
  length:
    type: integer
```

6.3.1. Href Attribute

The `href` attribute is mandatory.

"The Locator Attribute (href) supplies the data that allows an application to find a remote resource or resource fragment. The value of this attribute is an IRI which servers as the Uniform Resource Identifier for the remote resource." (W3C XLink Version 1.1)

6.3.2. Rel Attribute

The `rel` attribute is mandatory.

"In the simplest case, a link relation type identifies the semantics of a link. For example, a link with the relation type "copyright" indicates that the current link context has a copyright resource at the link target.

Link relation types can also be used to indicate that the target resource has particular attributes, or exhibits particular behaviours; for example, a "service" link implies that the link target can be used as part of a defined protocol (in this case, a service description).

Relation types are not to be confused with media types [RFC2046]; they do not identify the format of the representation that results when the link is dereferenced. Rather, they only describe how the current context is related to another resource.

Relation types SHOULD NOT infer any additional semantics based upon the presence or absence of

another link relation type, or its own cardinality of occurrence. An exception to this is the combination of the "alternate" and "stylesheet" registered relation types, which has special meaning in HTML for historical reasons.

There are two kinds of relation types: registered and extension." (RFC8288)

Relation types are discussed in more detail in Relation Types.

6.3.3. Type Attribute

The `type` attribute is optional

"The "type" attribute, when present, is a hint indicating what the media type of the result of dereferencing the link should be. Note that this is only a hint; for example, it does not override the Content-Type header field of a HTTP response obtained by actually following the link. The type attribute MUST NOT appear more than once in a given link-value; occurrences after the first MUST be ignored by parsers." (RFC8288)

6.3.4. Hreflang Attribute

The `hreflang` attribute is optional.

"The "hreflang" attribute, when present, is a hint indicating what the language of the result of dereferencing the link should be. Note that this is only a hint; for example, it does not override the Content-Language header field of a HTTP response obtained by actually following the link. Multiple hreflang attributes on a single link-value indicate that multiple languages are available from the indicated resource." (RFC8288)

6.3.5. Title Attribute

The `title` attribute is optional.

"The "title" attribute, when present, is used to label the destination of a link such that it can be used as a human-readable identifier (e.g., a menu entry) in the language indicated by the Content-Language header field (if present). The title attribute MUST NOT appear more than once in a given link; occurrences after the first MUST be ignored by parsers." (RFC8288)

6.3.6. Length Attribute

The `length` attribute does not appear to be defined in the normative standards.

6.4. Relation Types

There are two kinds of relation types; registered and extension.

Registered relation types are registered in the IANA register at https://www.iana.org/assignments/link-relations/link-relations.xhtml. Registered relation types are used in OGC API standards whenever appropriate.

Extension relation types are those which are not registered with IANA. These extension types are in

the form of "--- a URI [RFC3986] that uniquely identifies the relation type. Although the URI can point to a resource that contains a definition of the semantics of the relation type, clients SHOULD NOT automatically access that resource to avoid overburdening its server.

The URI used for an extension relation type SHOULD be under the control of the person or party defining it or be delegated to them.

When extension relation types are compared, they MUST be compared as strings (after converting to URIs if serialised in a different format) in a case-insensitive fashion, character by character. Because of this, all-lowercase URIs SHOULD be used for extension relations.

Note that while extension relation types are required to be URIs, a serialisation of links can specify that they are expressed in another form, as long as they can be converted to URIs." (RFC8288)

Extension relation types used in OGC API Standards are registered at https://github.com/opengeospatial/NamingAuthority/blob/master/registers/linkrelations.csv

6.5. What is a Collection

<OGC API> Collection
A geospatial data resource that may be available as one or more sub-resource distributions that conform to one or more OGC API Standards.

6.5.1. Rationale

Dataset
collection of data

NOTE	Published or curated by a single agent, and available for access or download in one or more serializations or formats.
NOTE	This definition was adopted as the OGC definition of "dataset" by the OAB (see OAB issue 1434)

A dataset is a restriction on the general concept of a collection

1. A dataset is a collection of data
2. A dataset is published or curated by a single agent.
3. A dataset is available in one or more serializations or formats (distributions).
4. A distribution is synonymous to the REST concept of a representation.

<OGC API> Collection (proposed)
A geospatial data resource that may be available as one or more sub-resource distributions that conform to one or more OGC API Standards.

An OGC API Collection is a restriction on the general concept of a collection:

1. An OGC API Collection may only contain data. It does not encompass non-data resources (but

the Collection resource may still have related auxiliary resources, such as schemas, or processes, as sub-resources).

2. Geospatial datasets include Spatial Things and/or Temporal Things as data items (but not all data items in the dataset have to have a spatial or temporal extent).
3. An OGC API Collection has one or more sub-resource distributions.
4. A distribution provides access to a representation of a dataset.

Conclusions

1. Any subset of a dataset can be an OGC API Collection.
2. To generalize the OGC API Collection concept so that a dataset or dataset series could also be a OGC API Collection, a different definition without "sub-resource distribution" would be needed. Maybe change "as one or more sub-resource distributions" to "in one or more representations"?
3. Such a change, however, would again require a mechanism to identify how datasets are related to the resources in a Web API that conforms to Common Part 2. How to identify that an OGC API Collection represents a dataset? How to identify that an API provides access to a single dataset?

6.5.2. Key Concepts

- **Collection**
  A body of resources that belong or are used together. An aggregate, set, or group of related resources. (Derived from Websters)

- **Distribution**
  specific representation of a dataset. (DCAT)

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

- **Sub-resource Distribution**
  a subset of the distribution of a dataset

- **Feature Collection**
  a set of **Features** from a **dataset**

- **Media Type**
  The data format of a representation. (Fielding 2000)

- **Resource**

  1. Any information that can be named. (Fielding 2000)
  2. A conceptual mapping to a set of entities. (Fielding 2000)
  3. A resource R is a temporally varying membership function $M_R(t)$, which for time t maps to a set of entities, or values, which are equivalent. (Fielding 2000)

- **Representation**

  1. A format matching one of an evolving set of standard data types, selected dynamically based on the capabilities or desires of the recipient and the nature of the resource. (Fielding 2000)

2. The current or intended state of a resource. (Fielding 2000)

- **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)

- **Temporal Thing**
  Anything with temporal extent, i.e. duration. e.g. the taking of a photograph, a scheduled meeting, a GPS time-stamped track-point. (W3C Basic Geo)

6.5.3. OAPI Collections and OGC Resource Types

NOTE	Jeff's analysis goes here.

6.6. Using OpenAPI (Informative)

The following section describes the main components of an OpenAPI document and how they should be used to construct API requests. The authoritative source is the OpenAPI 3.0 standard available here.

6.6.1. OpenAPI Document (root)

The OpenAPI document (Root) contains descriptive information about the API and serves as the root for the other parts of the document.

6.6.2. Paths

All API resources are accessed through a path. The Paths field of the OpenAPI document defines all of the paths available through this API. The Paths field is a collection of Paths Objects. Each Paths Object includes the URL or URL template for this path, any Server Objects specific to this Path, Parameters which are applicable to this Path, and an Operation Object for each of the HTTP Verbs applicable to this Path.

6.6.3. Operations

Operation Objects provide the details needed to create an HTTP request and response. Specifically, they provide definitions of the request message (including parameters) and all possible responses. In addition, they define any operation specific Server Objects or Security Requirements.

6.6.4. Parameters

Parameter Objects describe parameters which can be used in an API. These objects provide the parameter name, where it is passed (query, header, path, or cookie), and a detailed description of its' structure (if needed).

Parameter Objects can be defined at the following levels:

- Path - applicable to all operations on this path.

- Operation - only applicable to this operation. Overrides any parameters defined at the Path level which have the same name.

6.6.5. Servers

An API is not restricted to a single server. Furthermore, the set of valid servers may be different for different sections of the API. An OpenAPI Server Object describes a single server. Most important, it provides the URL to that server. Server Objects are grouped into arrays. These arrays provide a list of the servers which can be used to access a section of the API.

Example Array of Server Objects

```
servers:
  - url: https://dev.example.org/
    description: Development server
  - url: https://data.example.org/
    description: Production server
```

Server Objects can be defined on following levels:

- Root - applicable to the whole API unless overriden
- Path - applicable to all operations on this path. Servers defined at the Rool level are still valid.
- Operation - only applicable to this operation. Overrides any servers defined at the Path or Root level.

Building URLs

An OpenAPI document can describe a large number of URLs. Extracting all of the URLs is a non-trivial task. The OpenAPI objects used to construct URLs are:

- Server Objects (URL template for the root and variables to populate it)
- Paths Objects (URL template for the path and parameters)
- Operation Objects (including parameters)

They are organized as follows:

{Server Object}/{Path Object}/?{Parameters}

Server Objects may be found at the OpenAPI document, Path Object, and Operation Object level. Given this potentially large number of servers, how do you create the valid paths?

We can assume that the authors of a OAS document are not doing it for their personal enjoyment. Therefore, if a Server Object is included, there must be a reason for its' presence. So the Server Objects with the most restrictive scope are the ones we should use. Clients should look for Server Objects in the following order:

1. The Operation Object,
2. Then Path Item,

3. The root.

The first scope where a Server Object is found dictates the behavior completely.

```
IF Server Objects are supplied
    THEN save them for latter
    ELSE create a Server Object for the host of the OpenAPI document
DO FOR each Path
    IF Server Objects are included, THEN
        Use them instead of those previously identified
    IF Parameter Objects are included, THEN
        save them for latter
    DO FOR Each Operation
        IF Server Objects are included, THEN
            Use them instead of those previously identified
        IF Parameter Objects are included THEN
            IF this parameter was previously defined
                THEN replace the previous definition
                ELSE add this parameter to the set.
        DO FOR Each Server Object
            Extract all URL roots
            DO FOR each URL root
                Concatenate the URL root and Path to create a working URL
                Concatenate the working URL with the Parameters
                Save the completed URL for future use
            CONTINUE
        DONE
    DONE
DONE
```

Annex A: Revision History

Date	Release	Editor	Primary clauses modified	Description
2016-04-28	0.1	G. Editor	all	initial version

Annex B: Bibliography

NOTE

Example Bibliography (Delete this note).

The TC has approved Springer LNCS as the official document citation type.

Springer LNCS is widely used in technical and computer science journals and other publications

- For citations in the text please use square brackets and consecutive numbers: [1], [2], [3]

– Actual References:

[n] Journal: Author Surname, A.: Title. Publication Title. Volume number, Issue number, Pages Used (Year Published)

[n] Web: Author Surname, A.: Title, http://Website-Url

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