**1. Introduction**

**i. Abstract**

The OGC standards baseline has been extended to include [Resource Oriented Architectures](https://en.wikipedia.org/wiki/Resource-oriented_architecture) and [Web APIs](https://portal.ogc.org/files/?artifact_id=71776&version=1). In the course of developing OGC Web API standards, some practices proved to be common across multiple OGC Web API standards. These common practices are documented in the OGC API - Common Multi-Part Standard. OGC API - Common standards serve as reusable building-blocks. Standards developers can use these building-blocks in the construction of OGC Web API Standards. 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.

Spatial data is rarely considered as a single entity. [Feature Collections](http://docs.opengeospatial.org/DRAFTS/20-024.html#feature-collection-definition), [Coverages](http://docs.opengeospatial.org/DRAFTS/20-024.html#coverage-definition), [Data Sets](http://docs.opengeospatial.org/DRAFTS/20-024.html#dataset-definition) - they are all aggregations of [Spatial](http://docs.opengeospatial.org/DRAFTS/20-024.html#spatial-thing-definition) or [Temporal](http://docs.opengeospatial.org/DRAFTS/20-024.html#temporal-thing-definition) Things. It stands to reason that an OGC Web API would also expose its' holdings as aggregates of spatial resources.

The purpose of the OGC API - Common - Part 2: Geospatial Data (API-GeoData) Standard is to provide a means of organizing these collections and to define operations for the discovery and selection of individual collections.

OGC API-GeoData does not specify the nature of the geospatial data that make up a collection. Rather, the standard specifies a basic capability which should be applicable to any geospatial resource type. Additional OGC Web API standards extend this foundation to define resource-specific capabilities.

SNIP, SNIP

**2. Scope**

The OGC API - Common Standard is a multi-part standard which defines a standard set of modules which can be used to build resource and mission-specific Web API standards. The OGC API - Common - Part 2: Geospatial Data Standard (API-GeoData) is one of those modules.

Geospatial resources are typically packaged into sets or collections of related resources. A single API may host a large number of collections. This API-GeoData standard provides a means of organizing these collections and defines operations for the discovery and selection of individual collections.

API-GeoData does not specify the nature of the geospatial data that make up a collection. Rather, it provides a basic capability which should be applicable to any geospatial resource type. Additional OGC Web API standards extend this foundation to define resource-specific capabilities.

**3. 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](http://docs.opengeospatial.org/DRAFTS/20-024.html#citepp) and the [OGC Compliance Testing](https://www.ogc.org/compliance/) web site.

The one Standardization Target for this standard is Web APIs.

API-GeoData defines API modules intended for re-use by other OGC Web API standards. For the purpose of conformance, the applicable API modules are identified by [Conformance Classes](http://docs.opengeospatial.org/DRAFTS/20-024.html#ctc-definition). Typically this standard will only be implemented through reference to these Conformance Classes by other standards.

This standard identifies four [conformance classes](http://docs.opengeospatial.org/DRAFTS/20-024.html#ctc-definition). Each conformance class is defined by one [requirements class](http://docs.opengeospatial.org/DRAFTS/20-024.html#requirements-class-definition). 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 API-GeoData requirements classes are:

* [Collections](http://docs.opengeospatial.org/DRAFTS/20-024.html#rc_collections-section)
* [Simple Query](http://docs.opengeospatial.org/DRAFTS/20-024.html#rc-simple-query-section)
* [HTML](http://docs.opengeospatial.org/DRAFTS/20-024.html#rc_html-section)
* [JSON](http://docs.opengeospatial.org/DRAFTS/20-024.html#rc_json-section)

The [Collections Requirements Class](http://docs.opengeospatial.org/DRAFTS/20-024.html#rc_collections-section) defines a common means to describe and access [collections](http://docs.opengeospatial.org/DRAFTS/20-024.html#collection-definition) of [spatial resources](http://docs.opengeospatial.org/DRAFTS/20-024.html#spatial-resource-definition).

The [Simple Query Requirements Class](http://docs.opengeospatial.org/DRAFTS/20-024.html#rc-simple-query-section) defines basic query parameters for the selection of individual [collections](http://docs.opengeospatial.org/DRAFTS/20-024.html#collection-definition) of [spatial resources](http://docs.opengeospatial.org/DRAFTS/20-024.html#spatial-resource-definition).

The [Collections Requirements Class](http://docs.opengeospatial.org/DRAFTS/20-024.html#rc_collections-section) does not mandate a specific encoding or format for representing resources. The [*HTML*](http://docs.opengeospatial.org/DRAFTS/20-024.html#rc_html-section) and [*JSON*](http://docs.opengeospatial.org/DRAFTS/20-024.html#rc_json-section) requirements classes specify representations for these resources in commonly used encodings for spatial data on the web.

API-GeoData builds on API modules defined in the OGC API - Common - Part 1: Core ([API-Core](http://docs.opengeospatial.org/DRAFTS/20-024.html#apicore)) Standard. Each [requirements class](http://docs.opengeospatial.org/DRAFTS/20-024.html#requirements-class-definition) in the API-GeoData Standard identifies any API-Core [Conformance Classes](http://docs.opengeospatial.org/DRAFTS/20-024.html#ctc-definition) upon which it depends.

Proof of conformance with a [Conformance Class](http://docs.opengeospatial.org/DRAFTS/20-024.html#ctc-definition) includes demonstration of conformance with all dependencies of that [Conformance Class](http://docs.opengeospatial.org/DRAFTS/20-024.html#ctc-definition). The abstract tests in [Annex A](http://docs.opengeospatial.org/DRAFTS/20-024.html#ats_section) have been organized to facilitate validation of these dependencies. These tests have been organized by [requirements class](http://docs.opengeospatial.org/DRAFTS/20-024.html#requirements-class-definition). A referencing standard only has to require conformance with a [Conformance Class](http://docs.opengeospatial.org/DRAFTS/20-024.html#ctc-definition) and all of the requirements and relevant tests are identified.

In addition, each [requirements class](http://docs.opengeospatial.org/DRAFTS/20-024.html#requirements-class-definition) is organized as one or more trees. Starting at a root test, a test script can traverse the tree to address all of the required tests, each in the appropriate context.

NOTE

The structure and organization of a collection of spatial resources is very much dependent on the nature of that resource and the expected access patterns. This is information which cannot be specified in a common manner. This Standard specifies the requirements necessary to discover and understand a generic collection and its' contents. Requirements governing a specific type of resource are specified in resource-specific OGC API standards.

**5. Terms, Definitions and Abbreviated Terms**

**5.1. Terms and Definitions**

This document uses the terms defined in Sub-clause 5 of [OGC API - Common - Part 1: Core](http://docs.opengeospatial.org/DRAFTS/20-024.html#apicore) (OGC 19-072), which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word "shall" (not "must") is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

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

* **Collection**  
  A geospatial resource that may be available as one or more [sub-resource](http://docs.opengeospatial.org/DRAFTS/20-024.html#resource-definition) [distributions](http://docs.opengeospatial.org/DRAFTS/20-024.html#distribution-definition) that conform to one or more OGC API standards. (OGC 20-024)
* **Coverage**  
  feature that acts as a function to return values from its range for any direct position within its spatiotemporal domain, as defined in OGC Abstract Topic 6 [(OGC 09-146r6)](http://docs.opengeospatial.org/is/09-146r6/09-146r6.html)
* **Dataset**  
  A collection of data, published or curated by a single agent, and available for access or download in one or more representations. ([DCAT](http://docs.opengeospatial.org/DRAFTS/20-024.html#DCAT))
* **Distribution**  
  A specific representation of a [dataset](http://docs.opengeospatial.org/DRAFTS/20-024.html#dataset-definition). A [dataset](http://docs.opengeospatial.org/DRAFTS/20-024.html#dataset-definition) 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](http://docs.opengeospatial.org/DRAFTS/20-024.html#DCAT))  
    
  EXAMPLE: a downloadable file, an RSS feed or an API.
* **Extent**  
  The area covered by something. Within this document, ”extent” refers to spatial extent: The size or shape that may be expresses using coordinates. ([W3C/OGC Spatial Data on the Web Best Practice](http://docs.opengeospatial.org/DRAFTS/20-024.html#SDWBP))
* **Feature**  
  abstraction of real world phenomena ([ISO 19101-1:2014](http://docs.opengeospatial.org/DRAFTS/20-024.html#iso19101))

|  |  |
| --- | --- |
| **NOTE:** | More details about the term feature may be found in the [W3C/OGC Spatial Data on the Web Best Practice](http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#SDWBP) in the section [Spatial Things, Features and Geometry](https://www.w3.org/TR/sdw-bp/#spatial-things-features-and-geometry). |

* **Feature Collection**  
  a set of **Features** from a **dataset**
* **Geometry**  
  An ordered set of *n*-dimensional points in a given coordinate reference system. ([W3C/OGC Spatial Data on the Web Best Practice](http://docs.opengeospatial.org/DRAFTS/20-024.html#SDWBP))
* **OGC Web API**  
  A [Web API](http://docs.opengeospatial.org/DRAFTS/20-024.html#webapi-definition) that implements one or more [Conformance Classes](http://docs.opengeospatial.org/DRAFTS/20-024.html#ctc-definition) from an OGC API Standard.
* **Resource**  
  entity that might be identified ([Dublin Core Metadata Initiative - DCMI Metadata Terms](http://docs.opengeospatial.org/DRAFTS/20-024.html#iso15836-2))

|  |  |
| --- | --- |
| **NOTE:** | The term "resource", when used in the context of an OGC Web API standard, should be understood to mean a [Web Resource](http://docs.opengeospatial.org/DRAFTS/20-024.html#web-resource-definition) unless otherwise indicated. |

* **Resource Type**  
  the definition of a type of [resource](http://docs.opengeospatial.org/DRAFTS/20-024.html#resource-definition). Resource types are re-usable components which are independent of where the resource resides in the API.

|  |  |
| --- | --- |
| **NOTE:** | Resource types are re-usable components that are independent of where the resource resides in the API." |

* **Spatial Resource**  
  the [resources](http://docs.opengeospatial.org/DRAFTS/20-024.html#resource-definition) which we usually think of as Geospatial Data. A [Spatial Thing](http://docs.opengeospatial.org/DRAFTS/20-024.html#spatial-thing-definition). ([OGC 19-072](http://docs.opengeospatial.org/DRAFTS/20-024.html#apicore))
* **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](http://docs.opengeospatial.org/DRAFTS/20-024.html#SDWBP))
* **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. ([ISO 19108](http://docs.opengeospatial.org/DRAFTS/20-024.html#iso19108))
* **Temporal Position**  
  location relative to a temporal reference system ([ISO 19108](http://docs.opengeospatial.org/DRAFTS/20-024.html#iso19108))
* **Temporal Reference System**  
  reference system against which time is measured ([ISO 19108](http://docs.opengeospatial.org/DRAFTS/20-024.html#iso19108))
* **Temporal Resource**  
  the [resources](http://docs.opengeospatial.org/DRAFTS/20-024.html#resource-definition) which we usually think of as time and date focused data. A [Temporal Thing](http://docs.opengeospatial.org/DRAFTS/20-024.html#temporal-thing-definition). ([OGC 19-072](http://docs.opengeospatial.org/DRAFTS/20-024.html#apicore))
* **Temporal Thing**  
  Anything with temporal extent, i.e. duration. Examples are the taking of a photograph, a scheduled meeting, a GPS time-stamped track-point. ([W3C Basic Geo](http://docs.opengeospatial.org/DRAFTS/20-024.html#W3C-BASIC-GEO))
* **Web API**  
  API using an architectural style that is founded on the technologies of the Web. ([W3C Data on the Web Best Practices](http://docs.opengeospatial.org/DRAFTS/20-024.html#DWBP))
* **Web Resource**  
  a [resource](http://docs.opengeospatial.org/DRAFTS/20-024.html#resource-definition) that is identified by an URI.

<<SNIP,SNIP>>

**6.3. Geometry**

**6.3.1. Geospatial Geometry**

Standardized concepts for geospatial characteristics are needed in order to share geographic information between applications. Concepts for shape (geometry) are key. These concepts are standardized in [*ISO 19107*](http://docs.opengeospatial.org/DRAFTS/20-024.html#iso19107).

The geospatial geometry used in the OGC API - Common Standards is documented in the [GML Simple Features Profile](http://docs.opengeospatial.org/DRAFTS/20-024.html#gmlsf) Standard. This Profile defines a subset of the ISO 19107 geometry which is aligned with the OGC [Simple Features for SQL](http://docs.opengeospatial.org/DRAFTS/20-024.html#sfsql) Standard. That geometry includes: Point, Curve (LineString), Surface (Polygon), MultiPoint, MultiCurve, and MultiSurface.

**6.3.2. Temporal Geometry**

Standardized concepts for temporal characteristics are also needed in order to share date and time information between applications. OGC API Common adopts the Gregorian calendar and a 24 hour time keeping system for its temporal geometry. All representations of that geometry which are discussed in this document conform to [RFC 3339](http://docs.opengeospatial.org/DRAFTS/20-024.html#rfc3339).

An [ABNF](http://docs.opengeospatial.org/DRAFTS/20-024.html#rfc5234) representation of the RFC 3339 format is provided in [Annex F](http://docs.opengeospatial.org/DRAFTS/20-024.html#date-time-bnf-annex).

**6.4. Coordinate Reference Systems**

As discussed in Chapter 9 of the [W3C/OGC Spatial Data on the Web Best Practices document](http://docs.opengeospatial.org/DRAFTS/20-024.html#SDWBP), 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.

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

* CRS84 - WGS 84 longitude and latitude without height
* CRS84h - WGS 84 longitude and latitude with ellipsoidal height

Temporal geometry is measured relative to an underlying temporal reference system (TRS). This API-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](http://docs.opengeospatial.org/DRAFTS/20-024.html#rfc3339). In data, other temporal reference systems may be used where appropriate.

OGC Topic Volume 2 – Referencing by Coordinates ([ISO 19111](http://docs.opengeospatial.org/DRAFTS/20-024.html" \l "iso19111)) provides the conceptual model for Coordinate Reference Systems.

**6.5. API definition**

**6.5.1. General remarks**

This OGC Standard specifies requirements and recommendations for the development of APIs that share spatial resources using a standard way of doing so. In general, deployed APIs will go beyond the requirements and recommendations stated in this Standard. They will support additional operations, parameters, and so on that are specific to the API or the software tool used to implement the API.

So that developers can more easily learn how to use the 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 run-time binding. OpenAPI is one way to provide that machine readable documentation.

**6.5.2. Role of OpenAPI**

This OGC API Standard uses OpenAPI 3.0 fragments in examples and to formally state requirements. Using OpenAPI 3.0 is not required for implementing an OGC API. 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.

This standard includes a [conformance class](http://docs.opengeospatial.org/DRAFTS/20-024.html#rc_oas30-section) for API definitions that follow the [OpenAPI specification 3.0](http://docs.opengeospatial.org/DRAFTS/20-024.html#openapi). 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.

**6.5.3. References to OpenAPI components in normative statements**

Some normative statements (requirements, recommendations and permissions) use a phrase that a component in the API definition of the server must be "based upon" a schema or parameter component in the OGC schema repository.

In this case, the following changes to the pre-defined OpenAPI component are permitted:

* If the server supports an XML encoding, xml properties may be added to the relevant OpenAPI schema components.
* The range of values of a parameter or 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 definition of a Response Object.
* Informative text, such as comments or description properties, may be changed or added.

For OGC API definitions that do not conform to the [OpenAPI Specification 3.0](http://docs.opengeospatial.org/DRAFTS/20-024.html#openapi), the normative statement should be interpreted in the context of the API definition language used.

**6.5.4. 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 <http://schemas.opengis.net/ogcapi/common/part1/1.0> and <http://schemas.opengis.net/ogcapi/common/part2/1.0>

Additional information on the use of OpenAPI as an API definition is provided in the [OGC API - Common Users Guide](http://docs.opengeospatial.org/DRAFTS/20-071.html#openapi-section).

**7. Overview**

This API - Common - Part 2: Geospatial Data Standard provides a starting point for the discovery and access of geospatial resources available via a Web API. While typically accessible on a landing page (see API - Common - Part 1), this is not required. The Requirements Classes defined in this Standard are designed to be re-usable modules. To be deployed in accordance with the requirements of the API developer.

**7.1. Collections**

Spatial data is rarely considered as a single entity. [Feature Collections](http://docs.opengeospatial.org/DRAFTS/20-024.html#feature-collection-definition), [Coverages](http://docs.opengeospatial.org/DRAFTS/20-024.html#coverage-definition), [Data Sets](http://docs.opengeospatial.org/DRAFTS/20-024.html#dataset-definition) - these are all aggregations of [Spatial](http://docs.opengeospatial.org/DRAFTS/20-024.html#spatial-thing-definition) or [Temporal](http://docs.opengeospatial.org/DRAFTS/20-024.html#temporal-thing-definition) Things. It stands to reason that an OGC Web API would also expose its holdings as aggregates of spatial/temporal resources.

The purpose of the API-GeoData Standard is to provide a common connection between the API and the set of Geospatial data collections that are accessible through that API. That connection includes metadata which describes the hosted resources, common parameters for selecting subsets of the hosted resources, and URI templates for identifying the above.

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**, a collection is defined as follows:

* [Collection](http://docs.opengeospatial.org/DRAFTS/20-024.html#collection-definition): A geospatial resource that may be available as one or more sub-resource distributions that conform to one or more OGC API standards.

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

**7.2. Views**

A collection of geospatial data may be represented in more than one way. For example, a point cloud may be represented as a collection of Features, as a coverage, or as a color-coded map. Each is a different representation of the same data. However the access methods and returned data structures are very different. OGC Web API standards refer to these representations as views.

Views should not be confused with encodings. HTTP content negotiation allows a client to negotiate the encoding (XML, JSON, etc.) to be used for the returned data. Regardless of the encoding, the underlying data model is the same. A view, on the other hand, is both a data model and a set of access mechanisms. A view is an addressable resource in its own right and must be treated as such.

The API-GeoData Standard does not define any views. These are defined in separate OGC Web API Standards. What is important to understand is how these view-specific standards extend the API-Geodata Standard.

The URI for a view of a collection follows the URI template:

/collections/{collectionId}/{viewId}

Where:

* collectionId = an identifier for the collection
* viewId = an identifier for the type of view.

So the URIs for the point cloud described above could be:

* For Features: /collections/mycollection/items
* For Coverages: /collections/mycollection/coverage
* For Maps: /collections/mycollection/maps

The view identifiers are maintained as a controlled vocabulary by the OGC.

Additional information on Views is provided in the [OGC API - Common Users Guide](http://docs.opengeospatial.org/DRAFTS/20-071.html#views-section).

**8. Requirements Class "Collections"**

|  |  |
| --- | --- |
| **Requirements Class** | |
| <http://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections> | |
| Target type | Web API |
| Dependency | [IETF RFC 7231, Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content](http://docs.opengeospatial.org/DRAFTS/20-024.html#rfc7231) |

|  |  |
| --- | --- |
| **Requirement 1** | **/req/collections/rc-dependency-http** |
| A | An implementation of the /Collections Requirements Class SHALL conform to [HTTP 1.1](http://docs.opengeospatial.org/DRAFTS/20-024.html#rfc7231). |
| B | If the API supports HTTPS, then the implementation SHALL also conform to [HTTP over TLS](http://docs.opengeospatial.org/DRAFTS/20-024.html#rfc2818). |

|  |  |
| --- | --- |
| **Recommendation 1** | **/rec/collections/rec-dependency-core** |
| A | An implementation of the /Collections Requirements Class SHOULD demonstrate conformance with the Core Conformance Class defined in OGC API - Common Part 1. |

|  |  |
| --- | --- |
| **Recommendation 2** | **/rec/collections/rec-dependency-landing-page** |
| A | An implementation of the /Collections Requirements Class SHOULD demonstrate conformance with the Landing Page Conformance Class defined in OGC API - Common Part 1. |

This Requirements Class describes the resources and operations used to describe and access resource collections exposed through an OGC Web API. This Class does not include any requirements about how resources are aggregated into collections nor about the aggregated resources themselves. That detail is reserved for resource-specific OGC Web API standards (see [Views Section](http://docs.opengeospatial.org/DRAFTS/20-024.html#views-introduction)).

The two resources and their operations are defined in this Requirements Class. They are summarized in [Table 2](http://docs.opengeospatial.org/DRAFTS/20-024.html#collection-resources).

| Table 2. Collection Resources | | | |
| --- | --- | --- | --- |
| **Resource** | **URI** | **HTTP Method** | **Description** |
| [Collections](http://docs.opengeospatial.org/DRAFTS/20-024.html#collections-metadata) | /collections | GET | Information which describes the set of available Collections |
| [Collection](http://docs.opengeospatial.org/DRAFTS/20-024.html#collection-description) | /collections/{collectionId} | GET | Information about a specific collection of geospatial data with links to distribution |

**8.1. Collections**

OGC APIs typically organize their Spatial Resources into collections. Information about those collections is accessed through the /collections path and the <http://www.opengis.net/def/rel/ogc/1.0/data> link relation.

**8.1.1. Operation**

|  |  |
| --- | --- |
| **Requirement 2** | **/req/collections/rc-md-op** |
| A | The API SHALL support the HTTP GET operation at the path /collections. |

**8.1.2. Response**

|  |  |
| --- | --- |
| **Requirement 3** | **/req/collections/rc-md-success** |
| A | A successful execution of the operation SHALL be reported as a response with a HTTP status code 200. |
| B | The content of that response SHALL be based upon the JSON schema [collections.yaml](http://beta.schemas.opengis.net/ogcapi/common/part2/0.1/collections/openapi/schemas/collections.yaml). |

The collections response returned by this operation is based on the [collections.yaml](http://beta.schemas.opengis.net/ogcapi/common/part2/0.1/collections/openapi/schemas/collections.yaml) JSON schema. Examples of collections responses are provided in [Collections Response Example](http://docs.opengeospatial.org/DRAFTS/20-024.html#collections-metadata-examples).

collections.yaml

type: object

required:

- links

- collections

properties:

links:

type: array

items:

$ref: link.yaml

timeStamp:

type: string

format: date-time

numberMatched:

type: integer

minimum: 0

numberReturned:

type: integer

minimum: 0

collections:

type: array

items:

$ref: collectionDesc.yaml

This [collections.yaml](http://beta.schemas.opengis.net/ogcapi/common/part2/0.1/collections/openapi/schemas/collections.yaml) JSON 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 4** | **/req/collections/rc-md-links** |
| 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 and type link parameters. |

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

|  |  |
| --- | --- |
| **Recommendation 3** | **/rec/collections/rc-md-descriptions** |
| A | If external schemas or 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 timeStamp property of the Collections response indicates when the response was generated.

|  |  |
| --- | --- |
| **Requirement 5** | **/req/collections/rc-timeStamp** |
| A | If a property timeStamp is included in the response, the value SHALL be set to the time stamp when the response was generated. |

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

|  |  |
| --- | --- |
| **Requirement 6** | **/req/collections/rc-md-items** |
| A | For each spatial resource collection accessible through this API, metadata describing that collection SHALL be provided in the collections property of the Collections response. |
| B | The content of that metadata SHALL comply with the [http://www.opengis.net/spec/ogcapi-common-2/1.0/rm/collection](http://docs.opengeospatial.org/DRAFTS/20-024.html#rm_collection) Requirements Module described in the [Collection Resource Definition](http://docs.opengeospatial.org/DRAFTS/20-024.html#collection-resource-definition-section) section of this Standard. |

The array items included in the collection property are described in the [Collection Resource](http://docs.opengeospatial.org/DRAFTS/20-024.html#collection-description) section of this Requirements Class.

This Requirements Class does not define any parameters for use against a collections resource. Implementers who wish to support filtering of the collections to be included in a result set should implement the [Simple Query](http://docs.opengeospatial.org/DRAFTS/20-024.html#rc-simple-query-section) Conformance Class for that purpose.

**8.1.3. Error situations**

See [HTTP Status Codes](http://docs.opengeospatial.org/DRAFTS/20-024.html#http-status-codes) for general guidance.

**8.2. Resource Collection**

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.

**8.2.1. Operation**

|  |  |
| --- | --- |
| **Requirement 7** | **/req/collections/src-md-op** |
| A | The API SHALL support the HTTP GET operation at the path /collections/{collectionId}. |
| B | The parameter collectionId is each id property in the collections response (JSONPath: $.collections[\*].id). |

**8.2.2. Response**

|  |  |
| --- | --- |
| **Requirement 8** | **/req/collections/src-md-success** |
| A | A successful execution of the operation SHALL be reported as a response with a HTTP status code 200. |
| B | The content of that response SHALL comply with the requirements in the [http://www.opengis.net/spec/ogcapi-common-2/1.0/rm/collection](http://docs.opengeospatial.org/DRAFTS/20-024.html#rm_collection) Requirements Module described in section [Collection Resource Definition](http://docs.opengeospatial.org/DRAFTS/20-024.html#collection-resource-definition-section) of this Standard. |
| C | The content of that response SHALL be consistent with the content for this collection in the /collections response. That is, the values for id, title, description and extent SHALL be identical. |

**8.2.3. Error Situations**

See [HTTP Status Codes](http://docs.opengeospatial.org/DRAFTS/20-024.html#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 (see [Table 5](http://docs.opengeospatial.org/DRAFTS/20-024.html#status-codes)).

**8.2.4. Collection Resource Definition**

|  |  |
| --- | --- |
| **Requirements Module** | |
| <http://www.opengis.net/spec/ogcapi-common-2/1.0/rm/collection> | |
| Target type | Web Resource |

|  |  |
| --- | --- |
| **Requirement 9** | **/req/collections/collection-definition** |
| A | The content of a Collection resource SHALL be based upon the JSON schema [collectionDesc.yaml](http://beta.schemas.opengis.net/ogcapi/common/part2/0.1/collections/openapi/schemas/collectionDesc.yaml). |

Collection Resource Schema

type: object

required:

- id

- links

properties:

id:

type: string

example: address

title:

type: string

example: address

description:

type: string

example: An address

attribution:

type: string

title: attribution for the collection

links:

type: array

items:

$ref: link.yaml

extent:

$ref: extent.yaml

itemType:

description: An indicator about the type of the items in the collection

type: string

crs:

description: the list of coordinate reference systems supported by the API; the first item is the default coordinate reference system

type: array

items:

type: string

default:

- http://www.opengis.net/def/crs/OGC/1.3/CRS84

example:

- http://www.opengis.net/def/crs/OGC/1.3/CRS84

- http://www.opengis.net/def/crs/EPSG/0/4326

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

**Attribution**

The attribution element is a special type of string property. Specifically, the attribution element can contain markup text. Markup allows a text string to import images and format text. The capabilities are only limited by the markup language used. See the example [collection response](http://docs.opengeospatial.org/DRAFTS/20-024.html#collection-description-examples) for an example of the use of markup in the attribution element.

**Item Type**

In some Geospatial collections, 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 4** | **/rec/collections/rc-md-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). |

**Links**

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

|  |  |
| --- | --- |
| **Requirement 10** | **/req/collections/rc-md-items-links** |
| 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 and type properties. |

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

|  |  |
| --- | --- |
| **Recommendation 5** | **/rec/collections/rc-md-items-descriptions** |
| 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. |

**Extent**

|  |  |
| --- | --- |
| **Requirements Module** | |
| <http://www.opengis.net/spec/ogcapi-common-2/1.0/rm/extent> | |
| Target type | Web Resource |

The extent property defines a spatial-temporal surface 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.

* Spatial Bounding Box (Bbox) provides a set of rectangular bounding boxes which use geographic 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.

|  |  |
| --- | --- |
| **Requirement 11** | **/req/collections/rc-md-extent** |
| A | For each spatial collection resource, the extent property, if provided, SHALL define boundaries that encompass the spatial and temporal properties of all of the resources in this collection. The temporal extent may use null values to indicate an open time interval. |
| B | If a spatial resource has multiple properties with spatial or temporal information, it is the decision of the API implementation whether only a single spatial or temporal geometry property is used to determine the extent or all relevant geometries. |

|  |  |
| --- | --- |
| **Recommendation 6** | **/rec/collections/rc-md-extent** |
| A | If an extent contains multiple spatial boundaries (multiple bbox, etc.), then the extent SHOULD include in the first bbox a boundary which represents the union of all of the other boundaries. |
| B | If an extent contains multiple temporal intervals, then the extent SHOULD include as the first interval an interval which represents the union of all of the other intervals. |

|  |  |
| --- | --- |
| **Recommendation 7** | **/rec/collections/rc-md-extent-single** |
| A | While the spatial and temporal extents support multiple bounding boxes (bbox array) and time intervals (interval array) for advanced use cases, implementations SHOULD provide only a single bounding box or time interval unless the use of multiple values is important for the use of the dataset and agents using the API are known to be support multiple bounding boxes or time intervals. |

|  |  |
| --- | --- |
| **Permission 1** | **/per/collections/rc-md-extent-extensions** |
| A | API-Common 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. |
| B | API-Common only supports   * Spatial extents in CRS84 or CRS84h and * Temporal extents in the Gregorian calendar   These are the only *enum* values in [extent.json](http://beta.schemas.opengis.net/ogcapi/common/part2/0.1/collections/openapi/schemas/extent.json)). |
| C | Extensions MAY add additional reference systems to the extent object. |

**9. Requirements Class "Simple Query"**

|  |  |
| --- | --- |
| **Requirements Class** | |
| <http://www.opengis.net/spec/ogcapi-common-2/1.0/req/simple-query> | |
| Target type | Web API |
| Dependency | [Collections Requirements Class](http://docs.opengeospatial.org/DRAFTS/20-024.html#rc_collections) |
| Dependency | [RFC 3339 (Date and Time on the Internet: Timestamps)](http://docs.opengeospatial.org/DRAFTS/20-024.html#rfc3339) |

This Requirements Class describes query parameters that can be used to discover and select resource collections exposed through an OGC Web API.

Implementers of this Requirements Class must also implement the /Collections Requirements Class.

|  |  |
| --- | --- |
| **Requirement 12** | **/req/simple-query/rc-dependency-collections** |
| A | An implementation of the Simple-Query Requirements Class SHALL demonstrate conformance with the /collections Conformance Class. |

**9.1. Parameter Requirements**

Query parameters are used in URIs to limit the resources which are returned on a GET request. The OGC API - Common - Part 2: Geospatial Data Standard identifies three query parameters for use in OGC API standards:

* [bbox](http://docs.opengeospatial.org/DRAFTS/20-024.html#bbox-parameter-requirements): Bounding Box
* [datetime](http://docs.opengeospatial.org/DRAFTS/20-024.html#parameter-datetime-requirements): Date and Time
* [limit](http://docs.opengeospatial.org/DRAFTS/20-024.html#limit-parameter-requirements): Response resource count limit

The behavior generated by these parameters is specific to the operation and resource upon which they are applied. Those behaviors are described for each resource type and operation in the [Target Resource Requirements](http://docs.opengeospatial.org/DRAFTS/20-024.html#target-resource-requirements) section.

Use of these query parameters with any specific operation is optional. Developers of API-GeoData servers should document their supported parameters in the API definition as describe in [API-Core](http://docs.opengeospatial.org/DRAFTS/20-024.html#apicore).

**9.1.1. Parameter bbox**

|  |  |
| --- | --- |
| **Requirements Module** | |
| <http://www.opengis.net/spec/ogcapi-common-2/1.0/rm/bbox> | |
| Target type | Web API Query Parameter |

The bbox parameter is used to select resources based on the geospatial footprint or extent.

The bbox parameter is defined as follows:

|  |  |
| --- | --- |
| **Requirement 13** | **/req/collections/rc-bbox-definition** |
| A | The bbox parameter SHALL possess the following characteristics (using an OpenAPI Specification 3.0 fragment):  name: bbox  in: query  required: false  schema:  type: array  oneOf:  - minItems: 4  maxItems: 4  - minItems: 6  maxItems: 6  items:  type: number  style: form  explode: false |
| B | The bounding box SHALL be provided as four or six numbers, depending on whether the coordinate reference system includes a vertical axis (height or depth):   * Lower left corner, coordinate axis 1 * Lower left corner, coordinate axis 2 * Minimum value, coordinate axis 3 (optional) * Upper right corner, coordinate axis 1 * Upper right corner, coordinate axis 2 * Maximum value, coordinate axis 3 (optional) |
| C | If the bounding box consists of four numbers, the coordinate reference system of the values SHALL be interpreted as WGS 84 longitude/latitude (<http://www.opengis.net/def/crs/OGC/1.3/CRS84>) unless a different coordinate reference system is specified in a parameter bbox-crs. |
| D | If the bounding box consists of six numbers, the coordinate reference system of the values SHALL be interpreted as WGS 84 longitude/latitude/ellipsoidal height (<http://www.opengis.net/def/crs/OGC/0/CRS84h>) unless a different coordinate reference system is specified in a parameter bbox-crs. |

While the processing of the bbox parameter is specific to the resource and operation for which it is applied, there is a general set of requirements which all implementations must address.

|  |  |
| --- | --- |
| **Requirement 14** | **/req/collections/rc-bbox-response** |
| A | If the bbox parameter is provided by the client and supported by the server, then only resources that have a spatial geometry that intersects the bounding box SHALL be part of the result set. |
| B | If a resource has multiple spatial geometry properties, it is the decision of the server whether only a single spatial geometry property is used to determine the extent or all relevant geometries. |
| C | The bbox parameter SHALL also match all resources in the collection that are not associated with a spatial geometry. |

"Intersects" means that a coordinate that is part of the spatial geometry of the resource falls within the area specified in the parameter bbox. This includes the boundaries of the geometries. For curves the boundary includes the start and end position. For surfaces the boundary includes the outer and inner rings.

In case of a degenerate bounding box, the resulting geometry is used. For example, if the lower left corner is the same as the upper right corner, all resources match where the geometry intersects with this point.

This standard does not specify requirements for the parameter bbox-crs. Those requirements will be specified in a later version of this standard.

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

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

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

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

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

A template for the definition of the parameter in YAML according to OpenAPI 3.0 is available at [bbox.yaml](http://beta.schemas.opengis.net/ogcapi/common/part2/0.1/collections/openapi/parameters/bbox.yaml).

**9.1.2. Parameter datetime**

|  |  |
| --- | --- |
| **Requirements Module** | |
| <http://www.opengis.net/spec/ogcapi-common-2/1.0/rm/datetime> | |
| Target type | Web API Query Parameter |

The datetime parameter selects resources based on their temporal extent. The definition of temporal extent is specific to the resource type being filtered.

The datetime parameter is defined as follows:

|  |  |
| --- | --- |
| **Requirement 15** | **/req/collections/rc-datetime-definition** |
| A | The datetime parameter SHALL have the following characteristics (using an OpenAPI Specification 3.0 fragment):  name: datetime  in: query  required: false  schema:  type: string  style: form  explode: false |
| B | Temporal geometries are either a date-time value or a time interval. The parameter value SHALL conform to the following syntax (using [ABNF](https://tools.ietf.org/html/rfc5234)):  interval-closed = date-time "/" date-time  interval-open-start = [".."] "/" date-time  interval-open-end = date-time "/" [".."]  interval = interval-closed / interval-open-start / interval-open-end  datetime = date-time / interval |
| C | The syntax of date-time is specified by [RFC 3339, 5.6](https://tools.ietf.org/html/rfc3339#section-5.6). |
| D | Open ranges in time intervals at the start or end are supported using a double-dot (..) or an empty string for the start/end.. |

While the processing of the datetime parameter is specific to the resource and operation for which it is applied, there is a general set of requirements which all implementations must address.

|  |  |
| --- | --- |
| **Requirement 16** | **/req/collections/rc-datetime-response** |
| A | If the datetime parameter is provided by the client and supported by the server, then only resources that have a temporal geometry that intersects the temporal information in the datetime parameter SHALL be part of the result set. If a resource has multiple temporal properties, it is the decision of the server whether only a single temporal property is used to determine the extent or all relevant temporal properties. |
| B | The datetime parameter SHALL match all resources in the collection that are not associated with a temporal geometry. |

"Intersects" means that the time (instant or period) specified in the parameter datetime includes a timestamp that is part of the temporal geometry of the resource (again, a time instant or period). For time periods this includes the start and end time.

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

Example 2. A date-time

February 12, 2018, 23:20:52 GMT:

datetime=2018-02-12T23%3A20%3A52Z

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

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

Example 3. Intervals

February 12, 2018, 00:00:00 GMT to March 18, 2018, 12:31:12 GMT:

datetime=2018-02-12T00%3A00%3A00Z%2F2018-03-18T12%3A31%3A12Z

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

datetime=2018-02-12T00%3A00%3A00Z%2F..

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

datetime=..%2F2018-03-18T12%3A31%3A12Z

A template for the definition of the parameter in YAML according to OpenAPI 3.0 is available at [datetime.yaml](http://beta.schemas.opengis.net/ogcapi/common/part2/0.1/collections/openapi/parameters/datetime.yaml).

**9.1.3. Parameter limit**

|  |  |
| --- | --- |
| **Requirements Module** | |
| <http://www.opengis.net/spec/ogcapi-common-2/1.0/rm/limit> | |
| Target type | Web API Query Parameter |

The limit parameter limits the number of resources that can be returned in a single response.

|  |  |
| --- | --- |
| **Requirement 17** | **/req/collections/rc-limit-definition** |
| A | The limit parameter SHALL possess the following characteristics (using an OpenAPI Specification 3.0 fragment):  name: limit  in: query  required: false  schema:  type: integer  minimum: 1  maximum: 10000  default: 10  style: form  explode: false |
| Note: | The values for minimum, maximum and default are only examples and MAY be changed. |

While the processing of the limit parameter is specific to the resource and operation for which it is applied, there is a general set of requirements which all implementations must address.

|  |  |
| --- | --- |
| **Requirement 18** | **/req/collections/rc-limit-response** |
| A | If the limit parameter is provided by the client and supported by the server, then the response SHALL not contain more resources than specified by the limit parameter. |
| B | If the API definition specifies a maximum value for the limit parameter, the response SHALL not contain more resources than this maximum value. |
| C | Only items are counted that are on the first level of the collection. Any nested objects contained within the explicitly requested items SHALL not be counted. |

The number of resources returned depends on the server and the value of the limit parameter.

* The client can request a limit to the number of resources returned.
* The server may have a default value for the limit, and a maximum limit.
* If the server has any more results available than it returns (the number it returns is less than or equal to the requested/default/maximum limit) then the server will include a link to the next set of results.

|  |  |
| --- | --- |
| **Permission 2** | **/per/collections/rc-server-limit** |
| A | If a server is configured with a maximum response size, then the server MAY page responses which exceed that threshold. |

Since many servers will place a limit on the size of their responses, clients should be prepared to handle a paged response even if they have not specified a limit parameter in their query.

The effect of the limit parameter is to divide the response into a number of pages. Each page (except for the last) contains the specified number of entities. The response contains the first page. Additional pages can be accessed through hyperlink navigation.

|  |  |
| --- | --- |
| **Recommendation 8** | **/rec/collections/rc-next-1** |
| A | A 200-response SHOULD include a link to the next "page" (relation: next), if more resources have been selected than returned in the response. |

|  |  |
| --- | --- |
| **Recommendation 9** | **/rec/collections/rc-next-2** |
| A | Dereferencing a next link SHOULD return additional resources from the set of selected resources that have not yet been returned. |

|  |  |
| --- | --- |
| **Recommendation 10** | **/rec/collections/rc-next-3** |
| A | The number of resources in a response to a next link SHOULD follow the same rules as for the response to the original query and again include a next link, if there are more resources in the selection that have not yet been returned. |

Providing prev links supports navigating back and forth between pages, but depending on the implementation approach it may be too complex to implement.

|  |  |
| --- | --- |
| **Permission 3** | **/per/collections/rc-prev** |
| A | A response to a next link MAY include a prev link to the resource that included the next link. |

**9.2. Target Resource Requirements**

The target of the parameters defined in this conformance class is the [Collection](http://docs.opengeospatial.org/DRAFTS/20-024.html#collection-description) resource described in the [Collections](http://docs.opengeospatial.org/DRAFTS/20-024.html#rc-collections-section) Requirements Class. The purpose of these parameters is to select a subset of [Collection](http://docs.opengeospatial.org/DRAFTS/20-024.html#collection-description) resources to be included in the response to a [/collections](http://docs.opengeospatial.org/DRAFTS/20-024.html#collections-metadata) request.

Three parameters are defined for use with the [Collections](http://docs.opengeospatial.org/DRAFTS/20-024.html#collections-metadata) resource. These parameters subset the set of [Collection](http://docs.opengeospatial.org/DRAFTS/20-024.html#collection-description) entries returned based on spatial, temporal, and volumetric filters. These parameters are documented in the [Parameter Requirements](http://docs.opengeospatial.org/DRAFTS/20-024.html#parameter-requirements) section.

The collections property of the Collections response provides a description of each individual collection hosted by the API. These descriptions are based on the [Resource Collection Schema](http://beta.schemas.opengis.net/ogcapi/common/part2/0.1/collections/openapi/schemas/collectionDesc.yaml). This schema is described in detail in the [Resource Collection Description](http://docs.opengeospatial.org/DRAFTS/20-024.html#collection-resource-definition-section) section of this Standard.

**9.2.1. Spatial and Temporal Filtering**

A client may select a subset of the hosted collections using the [bbox](http://docs.opengeospatial.org/DRAFTS/20-024.html#bbox-parameter-requirements) and the [datetime](http://docs.opengeospatial.org/DRAFTS/20-024.html#datetime-parameter-requirements) parameter. These parameters are evaluated against the extent element of each Collection item in the Collections response.

The requirements governing the processing of these parameters are:

|  |  |
| --- | --- |
| **Requirement 19** | **/req/collections/rc-bbox-collection-response** |
| A | The API server SHALL process the bbox parameter against the Collection resources (/collections/{collectionId}) accessible through that API. |
| B | The bbox parameter SHALL be evaluated against the geometry defined by the bbox element of the extent property of the Collection resource. |

|  |  |
| --- | --- |
| **Requirement 20** | **/req/collections/rc-datetime-collection-response** |
| A | The API server SHALL process the datetime parameter against the Collection resources (/collections/{collectionId}) accessible through that API. |
| B | The datetime parameter SHALL be evaluated against the temporal geometry defined by the interval element of the extent property of the Collection resource. |

**9.2.2. Volumetric Filtering**

The client may limit the number of collections returned in a response by using the limit parameter. When applied against the /collections resource, the [limit](http://docs.opengeospatial.org/DRAFTS/20-024.html#limit-parameter-requirements) parameter indicates the maximum number of collections which should be included in a single response.

|  |  |
| --- | --- |
| **Requirement 21** | **/req/collections/rc-limit-collection-response** |
| A | If the limit parameter is provided by the client, then the collections element of the collections response SHALL not contain more items than specified by the limit parameter. |
| B | If the API definition specifies a maximum value for the limit parameter, the collections element of the collections response SHALL not contain more items than this maximum value. |

The server also has the option of limiting the size of the Collections response.

|  |  |
| --- | --- |
| **Permission 4** | **/per/collections/rc-md-items** |
| A | To support servers with many collections, servers MAY limit the number of items included in the collections property. |

**9.2.3. Paged Response**

If the collections response does not contain all of the collection resources available from this server, then the client should be informed of that fact.

|  |  |
| --- | --- |
| **Recommendation 11** | **/rec/collections/rc-paged-response** |
| A | If the number of items in the collections element is less than the number available through the API, then the numberMatched and numberReurned properties SHOULD be included in the Collections response. |

The numberMatched property of the Collections response indicates the number of Collection items included in the Collections response. This may be a subset of the total set of collections hosted by the API. Selection of which collections to include in a subset is controlled through the [bbox](http://docs.opengeospatial.org/DRAFTS/20-024.html#bbox-parameter-requirements), [datetime](http://docs.opengeospatial.org/DRAFTS/20-024.html#datetime-parameter-requirements) and other selection parameters provided by the client.

|  |  |
| --- | --- |
| **Requirement 22** | **/req/collections/rc-numberMatched** |
| A | If a property numberMatched is included in the response, the value SHALL be identical to the number of hosted collections that meet the selection parameters provided by the client. |
| B | A server MAY omit this information in a response, if the information about the number of matching resources is not known or difficult to compute. |

The number of collection items included in a Collections response may be a subset of the number matched. In that case, the numberReturned property of the Collections response indicates the number of collection items returned in this "page" of the Collections response.

|  |  |
| --- | --- |
| **Requirement 23** | **/req/collections/rc-numberReturned** |
| A | If a property numberReturned is included in the response, the value SHALL be identical to the number of items in the collections array in the Collections document. |
| B | A server MAY omit this information in a response, if the information about the number of resources in the response is not known or difficult to compute. |

If the Collections response contains a subset of the selected collection items (numberReturned is less than numberMatched) then the Collections response should contain links for navigating to the rest of the collection items as described in the [limit parameter](http://docs.opengeospatial.org/DRAFTS/20-024.html#limit-parameter-requirements) section.

**10. Encoding Requirements Classes**

**10.1. Overview**

This clause specifies two requirements classes for encodings to be used with the [Collections](http://docs.opengeospatial.org/DRAFTS/20-024.html#collections-metadata) and [Collection](http://docs.opengeospatial.org/DRAFTS/20-024.html#collection-description) resources. These encodings are commonly used encodings for spatial data on the web:

* [HTML](http://docs.opengeospatial.org/DRAFTS/20-024.html#rc_html-section)
* [JSON](http://docs.opengeospatial.org/DRAFTS/20-024.html#rc_json-section)

Neither of these encodings is mandatory. An implementation of the [Collections](http://docs.opengeospatial.org/DRAFTS/20-024.html#rc-collections-section) requirements class may implement either, both, or neither of them.

**10.2. Requirement Class "HTML"**

Geographic information that is only accessible in formats like GeoJSON or GML has two issues:

* The data is not discoverable using the most common mechanism for discovering information: Web search engines.
* The data cannot be viewed directly in a browser. Additional tools are required to view the data.

Therefore, sharing data on the Web should include publication in HTML. To be consistent with the Web, this should be done in a way that enables users and search engines to access all data.

This is discussed in detail in the [W3C Best Practice](http://docs.opengeospatial.org/DRAFTS/20-024.html#SDWBP). This standard therefore [recommends](http://docs.opengeospatial.org/DRAFTS/20-024.html#rec_html) supporting HTML as an encoding.

|  |  |
| --- | --- |
| **Requirements Class** | |
| <http://www.opengis.net/spec/ogcapi_common-2/1.0/req/html> | |
| Target type | Web API |
| Dependency | [HTML5](http://docs.opengeospatial.org/DRAFTS/20-024.html#html5) |
| Dependency | [Schema.org](http://docs.opengeospatial.org/DRAFTS/20-024.html#schema_org) |

|  |  |
| --- | --- |
| **Requirement 24** | **/req/html/definition** |
| A | 200-responses of the server SHALL support the text/html media type for the [Collections](http://docs.opengeospatial.org/DRAFTS/20-024.html#collections-metadata) and [Collection](http://docs.opengeospatial.org/DRAFTS/20-024.html#collection-description) resources. |

|  |  |
| --- | --- |
| **Requirement 25** | **/req/html/content** |
| A | Every 200-response of the API with the media type "text/html" SHALL be a [HTML 5 document](https://www.w3.org/TR/html5/) that includes the following information in the HTML body:   * All information identified in the schemas of the [Response Object](https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.3.md#responseObject) in the HTML <body/>, and * All links in HTML <a/> elements in the HTML <body/>. |

|  |  |
| --- | --- |
| **Recommendation 12** | **/rec/html/schema-org** |
| A | A 200-response with the media type text/html, SHOULD include [Schema.org](http://docs.opengeospatial.org/DRAFTS/20-024.html#schema_org) annotations. |

**10.3. Requirement 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** | |
| <http://www.opengis.net/spec/ogcapi_common-2/1.0/req/json> | |
| Target type | Web API |
| Dependency | [IETF RFC 8259: The JavaScript Object Notation (JSON) Data Interchange Format](http://docs.opengeospatial.org/DRAFTS/20-024.html#rfc8259) |
| Dependency | [JSON Schema](http://docs.opengeospatial.org/DRAFTS/20-024.html#jschema) |

|  |  |
| --- | --- |
| **Requirement 26** | **/req/json/definition** |
| A | 200-responses of the server SHALL support the application/json media type for the [Collections](http://docs.opengeospatial.org/DRAFTS/20-024.html#colections-metadata) and [Collection](http://docs.opengeospatial.org/DRAFTS/20-024.html#collection-description) resources. |

|  |  |
| --- | --- |
| **Requirement 27** | **/req/json/content** |
| A | Every 200-response with the media type application/json SHALL include, or link to, a payload encoded according to the [JSON Interchange Format](http://docs.opengeospatial.org/DRAFTS/20-024.html#rfc8259). |
| 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](http://beta.schemas.opengis.net/ogcapi/common/part2/0.1/collections/openapi/schemas/collections.yaml) and [collectionDesc.yaml](http://beta.schemas.opengis.net/ogcapi/common/part2/0.1/collections/openapi/schemas/collectionDesc.yaml).

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