

OGC API - Features - Part 2

Coordinate Reference Systems by Reference

Open Geospatial Consortium

Submission Date: 2020-07-06

Approval Date: 2020-10-27

Publication Date: 2020-11-02

External identifier of this OGC® document: <http://www.opengis.net/doc/IS/ogcapi-features-2/1.0>

Internal reference number of this OGC® document: 18-058

Version: 1.0

Category: OGC® Implementation Standard

Editors: Clements Portele, Panagiotis (Peter) A. Vretanos

OGC API - Features - Part 2: Coordinate Reference Systems by Reference

Copyright notice

Copyright © 2020 Open Geospatial Consortium

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

Warning

This document is an OGC Member approved international standard. This document is available on a royalty free, non-discriminatory basis.

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

Document type: OGC® Standard

Document subtype: Interface

Document stage: Approved

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

1. Scope	5
2. Conformance	6
3. References	7
4. Terms and Definitions	8
5. Conventions and background	9
6. Requirements Class Coordinate Reference Systems by Reference	10
6.1. Overview	10
6.2. Discovery	11
6.2.1. CRS identifier list	11
6.2.2. Storage CRS	11
6.2.3. Global list of CRS identifiers	13
6.3. Query	17
6.3.1. Parameter bbox-crs	17
6.3.2. Parameter crs	18
6.3.3. Output format considerations	19
6.3.4. Coordinate reference system information independent of the feature encoding	21
Annex A: Abstract Test Suite (Normative)	23
A.1. Discovery	23
A.2. Query	24
A.2.1. Parameter crs	24
A.2.2. Parameter bbox-crs	26
Annex B: Revision History	29
Annex C: Bibliography	30

i. Abstract

OGC API standards define modular API building blocks to spatially enable Web APIs in a consistent way. The [OpenAPI specification](#) is used to define the API building blocks.

OGC API Features provides API building blocks to create, modify and query features on the Web. OGC API Features is comprised of multiple parts, each of them is a separate standard.

This part extends the core capabilities specified in [Part 1: Core](#) with the ability to use coordinate reference system identifiers other than the defaults defined in the core.

ii. Keywords

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

OGC API, coordinate reference system identifier, CRS, feature, spatial data, openapi, crs84, wgs84, longitude, latitude

iii. Preface

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

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

iv. Submitting organizations

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

- CubeWerx Inc.
- interactive instruments GmbH
- US Army Geospatial Center (AGC)

v. Submitters

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

Name	Affiliation
Clemens Portele (<i>editor</i>)	interactive instruments GmbH
Panagiotis (Peter) A. Vretanos (<i>editor</i>)	CubeWerx Inc.

Chapter 1. Scope

This document specifies an extension to the [OGC API - Features - Part 1: Core](#) standard that defines the behavior of a server that supports the ability to present geometry valued properties in a response document in one from a list of supported Coordinates Reference Systems (CRS).

This document assumes that each supported CRS can be referenced by a uniform resource identifier (i.e., a URI) such as <http://www.opengis.net/def/crs/EPSG/0/4326>.

This document specifies:

- How, for each offered feature collection, a server advertises the list of supported CRS identifiers;
- How the coordinates of geometry valued feature properties can be accessed in one of the supported CRSs;
- How features can be accessed from the server using a bounding box specified in one of the supported CRSs; and
- How a server can declare the coordinate reference system used to present feature resources.

Chapter 2. Conformance

This standard defines one requirements class [Coordinate Reference Systems by Reference](#). The standardization target is "Web APIs".

The URI of the associated conformance class is <http://www.opengis.net/spec/ogcapi-features-2/1.0/conf/crs>.

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

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.

- Open Geospatial Consortium (OGC). **OGC API - Features - Part 1: Core 1.0** [online]. Edited by C. Portele, P. Vretanos, C. Heazel. 2019 [viewed 2020-05-24]. Available at <http://www.opengis.net/doc/IS/ogcapi-features-1/1.0>

Chapter 4. Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r9], 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 in addition to the terms defined in [OGC API - Features - Part 1: Core](#).

coordinate

one of a sequence of numbers designating the position of a point [ISO 19111:2019, definition 3.1.5]

NOTE

In a spatial coordinate reference system, the coordinate numbers are qualified by units.

coordinate reference system (CRS)

coordinate system that is related to an object by a datum [ISO 19111:2019, definition 3.1.9]

coordinate system

set of mathematical rules for specifying how **coordinates** are to be assigned to points [ISO 19111:2019, definition 3.1.11]

feature

abstraction of real world phenomena [ISO 19101-1:2014]

NOTE

More details about the term 'feature' may be found in the [W3C/OGC Spatial Data on the Web Best Practice](#) in the section 'Spatial Things, Features and Geometry'.

feature collection; collection

a set of **features** from a dataset

spatial feature collection; spatial collection

a **feature collection** that includes one or more geometry-valued properties

Chapter 5. Conventions and background

See [OGC API - Features - Part 1: Core](#), Clauses 5 and 6.

Chapter 6. Requirements Class Coordinate Reference Systems by Reference

6.1. Overview

Requirements Class	
http://www.opengis.net/spec/ogcapi-features-2/1.0/req/crs	
Target type	Web API
Dependency	OGC API - Features - Part 1: Core, Requirements Class 'core'

The [OGC API - Features - Part 1: Core](#) standard defines support for only two coordinate reference systems:

- WGS 84 longitude, latitude
- WGS 84 longitude, latitude, ellipsoidal height

This extension defines the behavior of a server that supports additional coordinate reference systems.

Requirement 1	<code>/req/crs/crs-uri</code>
Each CRS supported by a server SHALL be referenceable by a uniform resource identifier (i.e. a URI).	

Recommendation 1	<code>/rec/crs/crs-format-model</code>
Servers that implement this extension SHOULD be able to recognize and generate CRS identifiers with the following format model: http://www.opengis.net/def/crs/{authority}/{version}/{code} In this format model, the token <code>{authority}</code> is a placeholder for a value that designates to authority responsible for the definition of this CRS. Typical values include "EPSG" and "OGC". The token <code>{version}</code> is a placeholder for the specific version of the CRS definition or <code>0</code> for un-versioned CRS definitions. The token <code>{code}</code> is a placeholder for the authority's code for the CRS.	

For more information, see [section 6.2 in OGC Name Type Specification, Part 1](#).

Note that while the EPSG register itself is versioned, the registered items are not versioned and the "version" is always "0" in URIs of the authority "EPSG".

6.2. Discovery

6.2.1. CRS identifier list

Requirement 2	/req/crs/fc-md-crs-list
A	The crs property in the collection object of a spatial feature collection SHALL contain the identifiers for the list of CRSs supported by the server for that collection.
B	This list SHALL include the default(s) defined in OGC API - Features - Part 1: Core .

The list has to include the default CRS—that is the CRS used unless something else is explicitly requested—is defined in **OGC API - Features - Part 1: Core** as:

- <http://www.opengis.net/def/crs/OGC/1.3/CRS84> (for coordinates without height)
- <http://www.opengis.net/def/crs/OGC/0/CRS84h> (for coordinates with height)

6.2.2. Storage CRS

The storage CRS for a spatial feature collection is the CRS identifier that may be used to retrieve features from that collection without the need to apply a CRS transformation.

Note that coordinates referenced to a dynamic coordinate reference system are ambiguous, if the coordinate epoch is unknown. It is therefore recommended to also provide the coordinate epoch when the storage CRS is dynamic, such as an ITRF realization or WGS 84. For more information on dynamic coordinate reference systems and coordinate epoch, please see **OGC Abstract Specification Topic 2: Referencing by coordinates**.

Requirement 3	/req/crs/fc-md-storageCrs
If all features in a spatial feature collection are stored using a particular CRS then the property storageCrs SHALL be specified in the collection object of the spatial feature collection to indicate the identifier for this storage CRS.	

Recommendation 2	/rec/crs/fc-md-coordinateEpoch
If the storage CRS of the spatial feature collection is a dynamic coordinate reference system, the property storageCrsCoordinateEpoch in the collection object of the spatial feature collection SHOULD provide the coordinate epoch of the coordinates.	

This document does not provide a mechanism to associate different coordinate epochs with feature geometries in a collection. If data with different coordinate epochs is merged in a collection, one option is to perform point motion operations (PMO) to convert all geometries to the same coordinate epoch. See **OGC Abstract Specification Topic 2** for more information.

Requirement 4	/req/crs/fc-md-storageCrs-valid-value
The value of the storageCrs property SHALL be one of the CRS identifiers from the list of supported CRS identifiers found in the collection object using the crs property.	

The following schema fragment extends the collection object to add the **storageCrs** and **storageCrsCoordinateEpoch** properties.

```

type: object
required:
  - id
  - links
properties:
  id:
    description: identifier of the collection used, for example, in URIs
    type: string
    example: address
  title:
    description: human readable title of the collection
    type: string
    example: address
  description:
    description: a description of the features in the collection
    type: string
    example: An address.
  links:
    type: array
    items:
      $ref: link.yaml
  example:
    - href: http://data.example.com/buildings
      rel: item
    - href: http://example.com/concepts/buildings.html
      rel: describedby
      type: text/html
  extent:
    $ref: extent.yaml
  itemType:
    description: indicator about the type of the items in the collection (the default
value is 'feature').
    type: string
    default: feature
  crs:
    description: the list of CRS identifiers supported by the service
    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>

storageCrs:

description: the CRS identifier, from the list of supported CRS identifiers, that may be used to retrieve features from a collection without the need to apply a CRS transformation

type: string

format: uri

storageCrsCoordinateEpoch:

description: point in time at which coordinates in the spatial feature collection are referenced to the dynamic coordinate reference system in 'storageCrs', that may be used to retrieve features from a collection without the need to apply a change of coordinate epoch. It is expressed as a decimal year in the Gregorian calendar

type: number

example: '2017-03-25 in the Gregorian calendar is epoch 2017.23'

6.2.3. Global list of CRS identifiers

To prevent unnecessary duplication of lists of supported CRS identifiers in the [collection object](#), a global list of supported CRS identifiers may be provided as part of the collections object.

This global list of CRS identifiers is not automatically inherited by each collection offered by the service. Rather the global list of CRS identifiers must be explicitly referenced in the [crs](#) property of the collection object using a [JSON Pointer \(RFC 6901\)](#).

Requirement 5	/req/crs/fc-md-crs-list-global
If the crs property in the collection object of a spatial feature collection includes a JSON Pointer to the global list of CRS identifiers (#/crs), then all CRS identifiers in the global list SHALL be valid for the referencing collection.	

Note that only a local JSON Pointer within the same document is supported.

The following schema fragment extends the collections object to add the [crs](#) property which contains the global list of CRS identifiers.

```
allOf:
- $ref:
'http://schemas.opengis.net/ogcapi/features/part1/1.0/openapi/schemas/collections.yaml'
,
- type: object
  properties:
    crs:
      description: a global list of CRS identifiers that are supported by spatial
feature collections offered by the service
      type: array
      items:
        type: string
        format: uri
```

The following example illustrates the use of a global list of CRS identifiers.

```
{
  "links": [
    { "href": "http://data.example.org/collections.json",
      "rel": "self", "type": "application/json", "title": "this document" },
    { "href": "http://data.example.org/collections.html",
      "rel": "alternate", "type": "text/html", "title": "this document as HTML" },
    { "href": "http://schemas.example.org/1.0/buildings.xsd",
      "rel": "describedby", "type": "application/xml", "title": "GML application
schema for Acme Corporation building data" },
    { "href": "http://download.example.org/buildings.gpkg",
      "rel": "enclosure", "type": "application/geopackage+sqlite3", "title": "Bulk
download (GeoPackage)", "length": 472546 }
  ],
  "crs": [
    "http://www.opengis.net/def/crs/OGC/1.3/CRS84",
    "http://www.opengis.net/def/crs/EPSG/0/4326",
    "http://www.opengis.net/def/crs/EPSG/0/3857",
    "http://www.opengis.net/def/crs/EPSG/0/3395"
  ],
  "collections": [
    {
      "id": "bonn_buildings",
      "title": "Bonn Buildings",
      "description": "Buildings in the city of Bonn.",
      "extent": {
        "spatial": {
          "bbox": [ [ 7.01, 50.63, 7.22, 50.78 ] ]
        },
        "temporal": {
          "interval": [ [ "2010-02-15T12:34:56Z", null ] ]
        }
      },
      "links": [
        { "href": "http://data.example.org/collections/bonn_buildings/items",
          "rel": "items", "type": "application/geo+json",
          "title": "Bonn Buildings" },
        { "href": "https://creativecommons.org/publicdomain/zero/1.0/",
          "rel": "license", "type": "text/html",
          "title": "CC0-1.0" },
        { "href": "https://creativecommons.org/publicdomain/zero/1.0/rdf",
          "rel": "license", "type": "application/rdf+xml",
          "title": "CC0-1.0" }
      ],
      "crs": [
        "#/crs",
        "http://www.opengis.net/def/crs/EPSG/0/4258",
        "http://www.opengis.net/def/crs/EPSG/0/25831",
        "http://www.opengis.net/def/crs/EPSG/0/25832"
      ]
    }
  ]
}
```



```

    ]
  },
  {
    "id": "tor_buildings",
    "title": "Toronto Buildings",
    "description": "Buildings in the city of Toronto.",
    "extent": {
      "spatial": {
        "bbox": [ [ -79.62, 43.58, -79.12, 43.87 ] ]
      },
      "temporal": {
        "interval": [ [ "2010-02-15T12:34:56Z", null ] ]
      }
    },
    "links": [
      { "href": "http://data.example.org/collections/tor_buildings/items",
        "rel": "items", "type": "application/geo+json",
        "title": "Toronto Buildings" },
      { "href": "https://creativecommons.org/publicdomain/zero/1.0/",
        "rel": "license", "type": "text/html",
        "title": "CC0-1.0" },
      { "href": "https://creativecommons.org/publicdomain/zero/1.0/rdf",
        "rel": "license", "type": "application/rdf+xml",
        "title": "CC0-1.0" }
    ],
    "crs": [
      "#/crs"
    ]
  },
  {
    "id": "dc_buildings",
    "title": "Washington DC Buildings",
    "description": "Buildings in the city of Washington DC.",
    "extent": {
      "spatial": {
        "bbox": [ [ -77.12, 38.80, -76.89, 39.01 ] ]
      },
      "temporal": {
        "interval": [ [ "2010-02-15T12:34:56Z", null ] ]
      }
    },
    "links": [
      { "href": "http://data.example.org/collections/dc_buildings/items",
        "rel": "items", "type": "application/geo+json",
        "title": "DC Buildings" },
      { "href": "https://creativecommons.org/publicdomain/zero/1.0/",
        "rel": "license", "type": "text/html",
        "title": "CC0-1.0" },
      { "href": "https://creativecommons.org/publicdomain/zero/1.0/rdf",
        "rel": "license", "type": "application/rdf+xml",
        "title": "CC0-1.0" }
    ]
  }
]

```

```

    ],
    "crs": [
        "http://www.opengis.net/def/crs/OGC/1.3/CRS84"
    ]
}
]
}

```

In the above example, the **bonn_buildings** collection is offered in all the CRSs specified in the global list plus three other CRSs.

The **tor_buildings** collection is offered in the CRSs specified in the global list.

The **dc_buildings** collection is only offered in the default CRS (i.e., WGS 84 longitude, latitude).

6.3. Query

6.3.1. Parameter **bbox-crs**

The **bbox-crs** parameter may be used to assert the CRS used for the coordinate values of the **bbox** parameter.

Requirement 7	/req/crs/fc-bbox-crs-definition
Each GET request on a 'features' resource SHALL support a query parameter bbox-crs with the following characteristics:	
<pre> name: bbox-crs in: query required: false schema: type: string format: uri style: form explode: false </pre>	

Requirement 8	/req/crs/fc-bbox-crs-valid-value
A	If the value of the bbox-crs parameter is not one of the CRS identifiers from the list of supported CRS identifiers, then the server SHALL respond with the HTTP status code 400.
B	The list of supported CRS identifiers is found in the collection object using the crs property.

As usual, it is good practice to include a message about the reason for the error in the response.

Requirement 9	/req/crs/fc-bbox-crs-valid-defaultValue
If the bbox-crs parameter is not specified then the coordinate values of the bbox parameter SHALL be assumed to be in the default CRS specified in OGC API - Feature - Part 1: Core ; that is http://www.opengis.net/def/crs/OGC/1.3/CRS84 for coordinates without height and http://www.opengis.net/def/crs/OGC/0/CRS84h for coordinates with height.	

Requirement 10	/req/crs/fc-bbox-crs-action
If the bbox-crs parameter is specified, then the values of the bbox parameter SHALL be assumed to be in the specified CRS and the server SHALL perform the necessary internal transformations to properly fetch data from within the specified bounding box.	

The following fragment illustrates the use of the **bbox-crs** parameter (reserved characters have to be encoded):

Example 2. Specifying a bounding box in one of the supported coordinate reference systems

```
...&bbox=32507317%2C5224265%2C33427450%2C5603836&bbox-  
crs=http%3A%2F%2Fwww.opengis.net%2Fdef%2Fcrs%2FEPSG%2F0%2F25832
```

6.3.2. Parameter crs

Requirement 12	/req/crs/fc-crs-definition
Each GET request on a 'features' or 'feature' resource SHALL support a query parameter named crs with the following characteristics:	
<pre>name: crs in: query required: false schema: type: string format: uri style: form explode: false</pre>	
Requirement 13	/req/crs/fc-crs-valid-value

A	If the value of the crs parameter is not one of the CRS identifiers from the list of supported CRS identifiers, then the server SHALL respond with the HTTP status code 400.
B	The list of supported CRS identifiers is found in the collection object using the crs property.

As usual, it is good practice to include a message about the reason for the error in the response.

Requirement 14	/req/crs/fc-crs-default-value
If the crs parameter is not specified the geometry coordinates SHALL be presented in the default CRS specified in OGC API - Feature - Part 1: Core ; that is http://www.opengis.net/def/crs/OGC/1.3/CRS84 for coordinates without ellipsoidal height and http://www.opengis.net/def/crs/OGC/0/CRS84h for coordinates with ellipsoidal height.	

Requirement 15	/req/crs/fc-crs-action
If the crs parameter is specified, then the coordinates of all geometry-valued properties in the response document SHALL be presented in the requested CRS.	

Permission 1	/per/crs/fc-crs-action
Notwithstanding the requirement /req/crs/crs-action , if the requested feature representation is subject to any limitations for supporting coordinate reference systems, the Web API MAY return a response with a status code 400 .	

For example, OGC KML only supports the default CRS (WGS84 longitude and latitude, optionally with ellipsoidal height).

The following fragment illustrates the use of the **crs** parameter:

Example 3. Retrieving features from a collection in one of the supported CRSs

```
.../collections/buildings/items?crs=http%3A%2F%2Fwww.opengis.net%2Fdef%2Fcrs%2FEPSG%2F0%2F26703&...
```

6.3.3. Output format considerations

OGC API - Features - Part 1: Core defines three conformance classes related to the output formats:

- GML/XML
- GeoJSON/JSON

- HTML

6.3.3.1. Collections and Collection resources

This document specifies extensions to the Collections resource (the global list of coordinate reference systems) and the Collection resource (the storage CRS including the associated coordinate epoch).

How these extensions are reflected in each encoding is not fully specified by this standard, except for JSON-based or YAML-based encodings where the extensions are fully specified by the OpenAPI schema components.

For HTML, the requirement <http://www.opengis.net/spec/ogcapi-features-1/1.0/req/html/content> applies and the additional information has to be included in the body of the HTML document.

For XML, the content model of the of the complex types `core:CollectionsType` and `core:CollectionType` would have to be extended with additional information. This document does not specify the details for such extensions due to a lack of demand.

6.3.3.2. Features and Feature resources

GML has full CRS support and no further conventions are imposed by this standard.

NOTE

The CRS model in GML is based on ISO 19111:2007, but GML geometries reference CRSs by their URI identifier in the `srsName` attribute. These can resolve to a CRS that is defined based on the CRS model specified by ISO 19111:2019 (same as [OGC Abstract Specification Topic 2: Referencing by coordinates](#)), or a future edition.

HTML does not have any provisions for spatial geometries and coordinate reference systems. However, note that schema.org that is embedded in HTML only supports WGS 84 in the axis order latitude/longitude, so any coordinates in schema.org markup will have to be in that coordinate reference system, independent of the requested coordinate reference system.

GeoJSON normatively supports WGS 84 (without height: [CRS84](#); with height: [CRS84h](#)), but the "prior arrangement" provision allows other coordinate systems to be used.

Requirement 16	/req/crs/geojson
Servers that implement this extension plus the GeoJSON requirements class and clients that use this extension SHALL be subject to the prior arrangement provision in the second paragraph of section 4 of the GeoJSON standard .	

An explicit request by a client with a query parameter `crs` establishes a prior arrangement. It is the responsibility of the client that submits the request to handle the coordinates in the response correctly. In particular, clients should not make the GeoJSON document available to others unless they are aware of the prior arrangement.

This standard does not specify any standardized approach to encoding coordinate reference system information in a GeoJSON document.

The first paragraph of section 4 in GeoJSON also states:

An OPTIONAL third-position element SHALL be the height in meters above or below the WGS 84 reference ellipsoid. In the absence of elevation values, applications sensitive to height or depth SHOULD interpret positions as being at local ground or sea level.

If the requested coordinate reference system includes the vertical axis, the third-position element has to be interpreted according to that coordinate reference system, not as if it would be relative to the WGS 84 reference ellipsoid.

6.3.4. Coordinate reference system information independent of the feature encoding

Because of the inconsistent provision of CRS metadata in geospatial encodings and the continued confusion caused by the axis order of coordinates, this standard defines a mechanism for a server to clearly and unambiguously assert the CRS and axis order being used in a response document independent of the requested output format.

Requirement 17	/req/crs/ogc-crs-header
An HTTP header named Content-Crs SHALL be used to assert the coordinate reference system used in the body of a response.	

Requirement 18	/req/crs/ogc-crs-header-value
The value of the Content-Crs header SHALL be a URI identifying the coordinate reference system used in the response document according to the following grammar for CRS-header.	
CRS-header = "Content-Crs" ":" CRS-value	
CRS-value = "<" URI-reference ">"	

NOTE	The header is consistent with the draft "content negotiation by coordinate reference system" specification.
------	---

The following example illustrates the Content-Crs header in a response.

Example 4. HTTP header declaring the CRS and axis order used in the body of the response

```
$ curl -i
"https://example.com/api/v1/collections/poi/items/1?crs=http%3A%2F%2Fwww.opengis.n
et%2Fdef%2Fcrs%2FEPSG%2F0%2F3395"

HTTP/1.1 200 OK
Date: Sun, 24 May 2020 15:30:56 GMT
Content-Type: application/geo+json
Content-Language: en
Content-Crs: <http://www.opengis.net/def/crs/EPSG/0/3395>
Link:
<https://example.com/api/v1/collections/poi/items/1?crs=http%3A%2F%2Fwww.opengis.n
et%2Fdef%2Fcrs%2FEPSG%2F0%2F3395&f=json>; rel="self"; title="This document";
type="application/geo+json"
Link:
<https://example.com/api/v1/collections/poi/items/1?crs=http%3A%2F%2Fwww.opengis.n
et%2Fdef%2Fcrs%2FEPSG%2F0%2F3395&f=html>; rel="alternate"; title="This document as
HTML"; type="text/html"
Link: <https://example.com/api/v1/collections/poi>; rel="collection";
title="The collection the feature belongs to"
Vary: Accept-Language,Accept-Encoding
Content-Length: 1064

...
```

Annex A: Abstract Test Suite (Normative)

Conformance Class	
http://www.opengis.net/spec/ogcapi-features-2/1.0/conf/crs	
Target type	Web API
Requirements class	Requirements Class 'Coordinate Reference Systems by Reference'
Dependency	OGC API - Features - Part 1: Core, Conformance Class 'core'

A.1. Discovery

Abstract Test 1	/conf/crs/crs-uri
Test Purpose	Verify that each CRS identifier is a valid value
Requirement	/req/crs/crs-uri , /req/crs/fc-md-crs-list A , /req/crs/fc-md-storageCrs , /req/crs/fc-md-crs-list-global
Test Method	<p>For each string value in a crs or storageCrs property in the collections and collection objects in the paths /collections and /collections/{collectionId}, validate that the string conforms to the generic URI syntax as specified by RFC 3986, section 3. In addition, accept a single value of #/crs in each collection object at path /collections, if the collections object has a crs property.</p> <ol style="list-style-type: none">1. For http-URIs (starting with http:) validate that the string conforms to the syntax specified by RFC 7230, section 2.7.1.2. For https-URIs (starting with https:) validate that the string conforms to the syntax specified by RFC 7230, section 2.7.2.3. For URNs (starting with urn:) validate that the string conforms to the syntax specified by RFC 8141, section 2.4. For OGC URNs (starting with urn:ogc:def:crs:) and OGC http-URIs (starting with http://www.opengis.net/def/crs/) validate that the string conforms to the syntax specified by OGC Name Type Specification - definitions - part 1 – basic name.

Abstract Test 2	/conf/crs/default-crs
Test Purpose	Verify that the list of supported CRSs includes the default CRS.
Requirement	/req/crs/fc-md-crs-list B

Test Method	For each string value in a crs property in a collection object (for each path /collections and /collections/{collectionId}) validate that either http://www.opengis.net/def/crs/OGC/1.3/CRS84 or http://www.opengis.net/def/crs/OGC/1.3/CRS84h is included in the array, if the collection has a spatial extent, i.e., is a spatial feature collection.
-------------	--

Abstract Test 3	/conf/crs/storageCrs
Test Purpose	Verify that the storage CRS identifier is a valid value
Requirement	/req/crs/fc-md-storageCrs-valid-value
Test Method	For each collection object that includes a storageCrs property in the paths /collections and /collections/{collectionId} , validate that the string is also found in the crs property of the collection or, in case the crs property includes a value #/crs , in the global list of CRSs.

A.2. Query

A.2.1. Parameter crs

Abstract Test 4	/conf/crs/crs-parameter
Test Purpose	Verify that the parameter crs has been implemented correctly
Requirement	/req/crs/fc-crs-definition , /req/crs/fc-crs-valid-value B, /req/crs/ogc-crs-header , /req/crs/ogc-crs-header-value , /req/crs/geojson

Test Method	<p>For</p> <ul style="list-style-type: none"> • each spatial feature collection collectionId, • every GML or GeoJSON feature representation supported by the Web API, and • every CRS supported for the collection (every CRS listed in the crs property of the collection plus those in the global CRS list, if #/crs is included in the crs property) <p>send a request with CRS identifier in the parameter crs to</p> <ul style="list-style-type: none"> • /collections/{collectionId}/items and • /collections/{collectionId}/items/{featureId} (with a valid featureId for the collection). <p>Verify that</p> <ul style="list-style-type: none"> • every response is a valid Features or Feature response, • has the status code 200 and • includes a Content-Crs http header with the value of the requested CRS identifier.
-------------	--

Abstract Test 5	/conf/crs/crs-parameter-invalid
Test Purpose	Verify that invalid values in the parameter crs are reported
Requirement	/req/crs/fc-crs-valid-value
Test Method	<p>For</p> <ul style="list-style-type: none"> • each spatial feature collection collectionId <p>send a request with an unsupported CRS identifier in the parameter crs to</p> <ul style="list-style-type: none"> • /collections/{collectionId}/items and • /collections/{collectionId}/items/{featureId} (with a valid featureId for the collection). <p>Verify that the response has status code 400.</p> <p>Unsupported CRS identifiers are all strings not included in the crs property of the collection and also not included in the global CRS list, if #/crs is included in the crs property.</p>

Abstract Test 6	/conf/crs/crs-parameter-default
Test Purpose	Verify that the default value for parameter crs has been implemented correctly
Requirement	/req/crs/fc-crs-default-value , /req/crs/ogc-crs-header , /req/crs/ogc-crs-header-value
Test Method	For each spatial feature collection, send a request without the crs parameter and verify that the response includes a Content-Crs http header with the value of the default CRS identifier of the collection.

Abstract Test 7	/conf/crs/crs-parameter-transform
Test Purpose	Verify that the geometries are transformed
Requirement	/req/crs/crs-action
Test Method	<p>For every CRS identifier advertized by the Web API that is known to the test engine and for which the test engine can convert geometries between the CRS and the default CRS of the Web API ("known CRS") execute the following test. Skip the test for unknown CRSs.</p> <ol style="list-style-type: none"> 1. For each spatial feature collection collectionId, send a request with the parameter crs to /collections/{collectionId}/items and /collections/{collectionId}/items/{featureId} (with a valid featureId for the collection) for every known CRS listed. In addition, send the same request, but without the crs parameter. 2. Convert the response for the known CRS to the default CRS and verify that the responses match. Due to the use of different coordinate conversions in the test engine and by the API, there will not be an exact match and the test engine will have to allow for reasonable differences when assessing whether the geometries match.

A.2.2. Parameter **bbox-crs**

Abstract Test 8	/conf/crs/bbox-crs-parameter
------------------------	-------------------------------------

Test Purpose	Verify that the parameter <code>bbox-crs</code> has been implemented correctly
Requirement	/req/crs/fc-bbox-crs-definition , /req/crs/bbox-crs-action
Test Method	<p>For every CRS identifier advertized by the Web API that is known to the test engine and for which the test engine can convert geometries between the CRS and the default CRS of the Web API ("known CRS") execute the following test. Skip the test for unknown CRSs.</p> <ol style="list-style-type: none"> 1. For each spatial feature collection <code>collectionId</code> and every GML or GeoJSON feature representation supported by the Web API, send a request with the parameters <code>bbox</code> and <code>bbox-crs</code> to <code>/collections/{collectionId}/items</code> for every known CRS. Use a <code>bbox</code> value in the spatial extent of the collection, converted to the known CRS. Send the same request, but with no <code>bbox-crs</code> parameter and a <code>bbox</code> value in the default CRS. Do not include a <code>crs</code> parameter in the requests. Verify that the responses include the same features.

Abstract Test 9	/conf/crs/bbox-crs-parameter-invalid
Test Purpose	Verify that the parameter <code>bbox-crs</code> has been implemented correctly
Requirement	/req/crs/fc-bbox-crs-valid-value
Test Method	For each spatial feature collection <code>collectionId</code> , send a request with the parameters <code>bbox</code> and <code>bbox-crs</code> to <code>/collections/{collectionId}/items</code> with a value for <code>bbox-crs</code> that is not included in the list of CRSs and verify that the response has status code <code>400</code> .

Abstract Test 10	/conf/crs/bbox-crs-parameter-default
Test Purpose	Verify that the parameter <code>bbox-crs</code> has been implemented correctly
Requirement	/req/crs/fc-bbox-crs-default-value

Test Method	<p>For each spatial feature collection <code>collectionId</code> and every GML or GeoJSON feature representation supported by the Web API, send a request with the parameters <code>bbox</code> and <code>bbox-crs</code> to <code>/collections/{collectionId}/items</code> for the default CRS of the collection. Use a <code>bbox</code> value in the spatial extent of the collection. Send the same request, but with no <code>bbox-crs</code> parameter. Do not include a <code>crs</code> parameter in the requests. Verify that the responses include the same features.</p>
-------------	---

Annex B: Revision History

Date	Release	Editor	Primary clauses modified	Description
2019-11-25	1.0.0-SNAPSHOT	Panagiotis (Peter) Vretanos, Clemens Portele	all	initial version
2020-06-18	1.0.0-draft.1	Panagiotis (Peter) Vretanos, Clemens Portele	all	resolve open issues and comments from the public review

Annex C: Bibliography

- OpenAPI Initiative (OAI). **OpenAPI Specification 3.0** [online]. 2020 [viewed 2020-03-16]. The latest patch version at the time of publication of this standard was 3.0.3, available at <http://spec.openapis.org/oas/v3.0.3>
- Open Geospatial Consortium (OGC). OGC 08-038r7: **Revision to Axis Order Policy and Recommendations** [online]. Edited by C. Reed. 2017 [viewed 2020-05-24]. Available at https://portal.opengeospatial.org/files/?artifact_id=76024
- Open Geospatial Consortium (OGC). OGC 10-100r3: **Geography Markup Language (GML) Simple Features Profile** [online]. Edited by L. van den Brink, C. Portele, P. Vretanos. 2012 [viewed 2020-03-16]. Available at http://portal.opengeospatial.org/files/?artifact_id=42729
- Internet Engineering Task Force (IETF). RFC 7946: **The GeoJSON Format** [online]. Edited by H. Butler, M. Daly, A. Doyle, S. Gillies, S. Hagen, T. Schaub. 2016 [viewed 2020-03-16]. Available at <http://tools.ietf.org/rfc/rfc7946.txt>
- WHATWG. **HTML**, Living Standard [online, viewed 2020-03-16]. Available at <https://html.spec.whatwg.org/>