**i. Abstract**

OGC API standards define modular API building blocks to spatially enable Web APIs in a consistent way. The [OpenAPI specification](file:///C:\Users\carln\AppData\Local\Temp\18-058.html#OpenAPI) 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](file:///C:\Users\carln\AppData\Local\Temp\18-058.html#OAFeat-1) with the ability to use coordinate reference system identifiers other than the defaults defined in the core.

|  |  |
| --- | --- |
| Caution | This is a DRAFT version of the second part of the OGC API - Features standards. This draft is not complete and there are open issues that are still under discussion. |

**ii. Keywords**

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

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

**v. Submitters**

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

|  |  |
| --- | --- |
| **Name** | **Affiliation** |
| Clemens Portele *(editor)* | interactive instruments GmbH |
| Panagiotis (Peter) A. Vretanos *(editor)* | CubeWerx Inc. |

**1. Scope**

This document specifies an extension to the [OGC API - Features - Part 1: Core](file:///C:\Users\carln\AppData\Local\Temp\18-058.html#OAFeat-1) standard that defines the behavior of a server that supports multiple Coordinates Reference Systems (CRS).

This document assumes that each supported CRS can be referenced by a unique resource identifier (i.e. a URI) such as (give an example).

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
* How a server can declare the CRS along with an optional axis order used to encode feature resources in response to a request.

**2. Conformance**

This standard defines one requirements / conformance class [Coordinate Reference Systems by Reference](file:///C:\Users\carln\AppData\Local\Temp\18-058.html#rc_crs). The standardization target is "Web APIs".

Conformance with this standard shall be checked using all the relevant tests specified in [Annex A](file:///C:\Users\carln\AppData\Local\Temp\18-058.html#ats) 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.

**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.

* Portele, C., Vretanos, P.: OGC 17-069r3, **OGC API - Features - Part 1: Core**, <http://docs.opengeospatial.org/is/17-069r3/17-069r3.html>
* Reed C., OGC 08-038r7, **Revision to Axis Order Policy and Recommendations**, <https://portal.opengeospatial.org/files/?artifact_id=76024>
* van den Brink, L., Portele, C., Vretanos, P.: OGC 10-100r3, Geography Markup Language (GML) Simple Features Profile\*, <http://portal.opengeospatial.org/files/?artifact_id=42729>
* Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S., Schaub, T.: IETF RFC 7946, **The GeoJSON Format**, <https://tools.ietf.org/rfc/rfc7946.txt>
* W3C: **HTML5**, W3C Recommendation, <http://www.w3.org/TR/html5/>

**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](file:///C:\Users\carln\AppData\Local\Temp\18-058.html#OAFeat-1).

**4.1. 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. |

**4.2. coordinate reference system (CRS)**

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

**4.3. coordinate system**

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

**4.4. feature**

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

|  |  |
| --- | --- |
| Note | If you are unfamiliar with the term 'feature', the explanations in the [W3C/OGC Spatial Data on the Web Best Practice document](file:///C:\Users\carln\AppData\Local\Temp\18-058.html#SDWBP) may help, in particular the section on [Spatial Things, Features and Geometry](https://www.w3.org/TR/sdw-bp/#spatial-things-features-and-geometry). |

**4.5. feature collection; collection**

a set of **features** from a **dataset**

|  |  |
| --- | --- |
| Note | In this specification, 'collection' is used as a synonym for 'feature collection'. This is done to make, for example, URI path expressions shorter and easier to understand for those that are not geo-experts. |

**4.6. spatial feature collection; spatial collection**

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

**5. Conventions and background**

See [OGC API - Features - Part 1: Core](file:///C:\Users\carln\AppData\Local\Temp\18-058.html#OAFeat-1), Clauses 5 and 6.

**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, Conformance Class 'core'](file:///C:\Users\carln\AppData\Local\Temp\18-058.html#OAFeat-1) |

The [OGC API - Features - Part 1: Core](file:///C:\Users\carln\AppData\Local\Temp\18-058.html#OAFeat-1) standard defines support for only two coordinate reference systems:

* WGS 84 longitude/latitude
* WGS 84 longitude/latitude plus ellipsoidal height

This extensions defines the behaviour of a server that supports additional coordinate reference systems.

|  |  |  |  |
| --- | --- | --- | --- |
| Caution | [Open issue 295](https://github.com/opengeospatial/ogcapi-features/issues/295) This draft is specific to Features and extends the resources 'feature collection', 'features' and 'feature'. However, in general the approach specified in this draft standard should also be applicable to other resource types and feedback would be welcome how such reuse could be improved. | | |
| **Requirement 1** | | **/req/crs/crs-uri** |
| Each coordinate reference system supported by a server shall be referenceable by a unique resource identifier (i.e. a URI). | | |

|  |  |
| --- | --- |
| **Recommendation 1** | **/rec/crs/crs-format-model** |
| 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}](http://www.opengis.net/def/crs/%7bauthority%7d/%7bversion%7d/%7bcode%7d)  In this format model, the token {authority} is a placeholder for a code that designates to the authority responsible for the definition of this CRS. Typical values include "EPSG" and "OGC".  The token {version} is a placeholder for the specific version of the CRS definition or 0 for the latest version or if the version is unknown.  The token {code} is a placeholder for the authority’s code for the CRS. | |

**6.2. Discovery**

**6.2.1. CRS identifier list**

|  |  |
| --- | --- |
| **Requirement 2** | **/req/crs/fc-md-crs-list** |
| For each spatial feature collection offered by a server in a CRS or CRSs other than the defaults defined in [OGC API - Features - Part 1: Core](file:///C:\Users\carln\AppData\Local\Temp\18-058.html#OAFeat-1), the crs property in the collection metadata shall contain the list of additional CRS identifiers supported by the service for this collection. | |

The default CRS — that is the CRS used unless something else is explicitly requested — is defined in [OGC API - Features - Part 1: Core](file:///C:\Users\carln\AppData\Local\Temp\18-058.html#OAFeat-1) 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)

The list of supported CRS identifiers in the collection metadata may include these defaults but this is not a requirement.

**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.

|  |  |  |  |
| --- | --- | --- | --- |
| Caution | [Open issue 264](https://github.com/opengeospatial/ogcapi-features/issues/264) To align with the new version of ISO 19111 / OGC Abstract Specification Topic 2, if the storage CRS is a dynamic CRS then stating the [coordinate epoch](http://docs.opengeospatial.org/as/18-005r4/18-005r4.html#68) of the data (if known) should be possible. In this case, it should also be recommended to use a specific realization, not a CRS using a datum ensemble. | | |
| **Requirement 3** | | **/req/crs/fc-md-storageCrs** |
| If all features in a spatial collection are stored using a specific CRS, then the property storageCRS shall be used in the collection metadata to indicate the identifier for this storage CRS. | | |

|  |  |
| --- | --- |
| **Requirement 4** | **/req/core/fc-md-storageCrs-valid-value** |
| The value of the storageCrs property shall be one of the CRS identifiers from the list of supported coordinate reference systems listed in the spatial collection metadata specified using the crs property. | |

The following schema fragment extends the collection metadata to add the storageCRS property.

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

**6.2.3. Global list of CRS identifiers**

To prevent unnecessary duplication of lists of supported CRS identifiers in the collection metadata, a global list of supported CRS identifiers may be provided as part of the collections metadata.

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 collection metadata using a JSON Pointer (RFC 6901).

|  |  |  |  |
| --- | --- | --- | --- |
| Caution | [Open issue 274](https://github.com/opengeospatial/ogcapi-features/issues/274) We are looking for feedback what option would be best to encode a reference to the global list (a JSON Pointer, a JSON Reference, or something else). What works best for clients parsing the response? | | |
| **Requirement 5** | | **/req/crs/fc-md-crs-list-global** |
| If referenced in the collection metadata, then all CRS identifiers in the global list shall be valid for the referencing collection. | | |

The following schema fragment extends the collections metadata to add the crs property which contains the global list of CRS identifiers.

type: object

required:

- links

- collections

properties:

links:

type: array

items:

$ref: link.yaml

crs:

description: a list of CRS identifiers that are supported for more that one feature collection offered by the service

type: array

items:

type: string

format: uri

collections:

type: array

items:

$ref: collection.yaml

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

Example 1. Collections metadata containing 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/EPSG/0/4326",

"http://www.opengis.net/def/crs/EPSG/0/3857",

"http://www.opengis.net/def/crs/EPSG/0/3395",

"http://www.opengis.net/def/crs/EPSG/0/4267",

"http://www.opengis.net/def/crs/EPSG/0/4269",

"http://www.opengis.net/def/crs/EPSG/0/26716",

"http://www.opengis.net/def/crs/EPSG/0/26717",

"http://www.opengis.net/def/crs/EPSG/0/26718",

"http://www.opengis.net/def/crs/EPSG/0/26719",

"http://www.opengis.net/def/crs/EPSG/0/26916",

"http://www.opengis.net/def/crs/EPSG/0/26917",

"http://www.opengis.net/def/crs/EPSG/0/26918",

"http://www.opengis.net/def/crs/EPSG/0/26919",

"http://www.opengis.net/def/crs/EPSG/0/32616",

"http://www.opengis.net/def/crs/EPSG/0/32617",

"http://www.opengis.net/def/crs/EPSG/0/32618",

"http://www.opengis.net/def/crs/EPSG/0/32619",

"http://www.opengis.net/def/crs/EPSG/0/32188"

],

"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/OGC/1.3/CRS41001",

"http://www.opengis.net/def/crs/OGC/0/2246",

"http://www.opengis.net/def/crs/OGC/0/3130",

"http://www.opengis.net/def/crs/OGC/0/3634",

"http://www.opengis.net/def/crs/OGC/0/6702",

]

},

{

"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" }

]

}

]

}

In the above example, the bonn\_buildings collection is offered in all the CRSs specified in the global list plus five other CRSs. The tor\_buildings collection is offered in the CRSs specified in the global list. The dc\_buildings collection, not having a crs property, is only offered in the default CRS (i.e. WGS84 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 6** | **/req/core/fc-bbox-crs-definition** |
| Each GET request on a 'features' resource shall support a query parameter bbox-crs with the following characteristics:  name: bbox-crs  in: query  required: false  schema:  type: string  format: uri  style: form  explode: false | |

|  |  |
| --- | --- |
| **Requirement 7** | **/req/core/fc-bbox-crs-valid-value** |
| The value of the bbox-crs parameter shall be one of the CRS identifiers from the list of supported CRSs listed in the spatial collection metadata using the crs property. | |

|  |  |  |
| --- | --- | --- |
| Caution | [Open issue 294](https://github.com/opengeospatial/ogcapi-features/issues/294) The wording of the requirement may be unclear. | |
| **Requirement 8** | | **/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](file:///C:\Users\carln\AppData\Local\Temp\18-058.html#OAFeat-1); 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 9** | **/req/core/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. | |

|  |  |
| --- | --- |
| **Requirement 10** | **/req/crs/bbox-crs-exception** |
| In all cases, an invalid or unrecognized CRS value shall trigger a 400 exception with an appropriate message. | |

The following fragment illustrates the use of the bbox-crs parameter:

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

...&bbox=160.6,-155.95,-170,-25.89&bbox-crs=http://www.opengis.net/...

**6.3.2. Parameter crs**

|  |  |
| --- | --- |
| **Requirement 11** | **/req/core/fc-crs-definition** |
| Each GET request on a 'features' or 'feature' resource shall support a query parameter named crs with the following characteristics:  name: crs  in: query  required: false  schema:  type: string  format: uri  style: form  explode: false | |

|  |  |
| --- | --- |
| **Requirement 12** | **/req/core/fc-crs-valid-value** |
| The value of the crs parameter shall be one of the CRS identifiers from the list of supported coordinate reference systems listed in the spatial collection metadata using the crs property. | |

|  |  |  |
| --- | --- | --- |
| Caution | [Open issue 294](https://github.com/opengeospatial/ogcapi-features/issues/294) The wording of the requirement may be unclear. | |
| **Requirement 13** | | **/req/core/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](file:///C:\Users\carln\AppData\Local\Temp\18-058.html#OAFeat-1); that is <http://www.opengis.net/def/crs/OGC/1.3/CRS84> for coordinates without height/depth and <http://www.opengis.net/def/crs/OGC/0/CRS84h> for coordinates with height/depth. | | | |

|  |  |
| --- | --- |
| **Requirement 14** | **/req/core/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 subject to any limitations placed on the response based on the requested output representation (e.g. the requested representation mandates a fixed CRS). | |

|  |  |
| --- | --- |
| **Requirement 15** | **/req/core/fc-crs-action-exception** |
| If the requested crs parameter values violates some requirement of the requested output format then the server shall raise an 400 exception with an appropriate message. | |

|  |  |
| --- | --- |
| **Requirement 16** | **/req/crs/crs-exception** |
| An invalid or unrecognized crs value shall trigger a 400 exception with an appropriate message. | |

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://www.opengis.net/def/crs/epsg/0/26703&...

|  |  |
| --- | --- |
| Caution | [Open issue 153](https://github.com/opengeospatial/ogcapi-features/issues/153) There is ongoing work to add negotiation of the CRS to the content negotiation process. However, this is still draft and not widely supported. For this reason, such a capability is considered only for a future version of this standard. |

**6.3.3. Output format considerations**

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

* GML
* GeoJSON
* HTML

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

GeoJSON normatively supports WGS84 (lon,lat) but the "prior arrangement" provision allows other coordinate systems to be used.

|  |  |
| --- | --- |
| **Requirement 17** | **/req/crs/geojson** |
| Servers that implement this extension and clients that use this extension shall be subject to the prior arrangements provision of the GeoJSON standard (see <https://tools.ietf.org/html/rfc7946#page-12>). | |

|  |  |
| --- | --- |
| Note | Need to do more work on HTML! |

HTML only supports WGS84 based on schema.org dependency. The OGC is not sure if this is an issue but schema.org annotations seem to require WGS84 (lat,lon) yet WFS core requires (lon,lat) by default.

**6.3.4. Coordinate system and axis order**

Because of the inconsistent provision of coordinate reference system 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 18** | **/req/crs/ogc-crs-header** |
| An HTTP header named OGC-CRS shall be used to assert the CRS and, optionally, the coordinate axis order used in the body of a response. | |

|  |  |
| --- | --- |
| **Requirement 19** | **/req/crs/ogc-crs-header-value** |
| The value of the OGC-CRS header shall be a URI referencing the CRS used in the response document with an optional parameter named axisOrder. | |

|  |  |
| --- | --- |
| **Requirement 20** | **/req/crs/ogc-crs-axis-order-value** |
| If the axisOrder parameter is specified, the value of the axisOrder parameter shall be an ordered list of axis names indicating the order in which coordinates are presented in a response document. | |

|  |  |
| --- | --- |
| **Requirement 21** | **/req/crs/ogc-crs-axis-names** |
| The axis names shall be taken from the CRS definition. | |

|  |  |
| --- | --- |
| **Requirement 22** | **/req/crs/ogc-crs-header-axis-action** |
| If the axisOrder parameter is not include with the OGC-CRS header, then the order of coordinates shall be assumed to be generated according to the requirements of the requested output format. | |

|  |  |
| --- | --- |
| Caution | [Open issue 153](https://github.com/opengeospatial/ogcapi-features/issues/153) The draft content negotiation by CRS specifies a Content-Crs header that could be used instead of a custom OGC-CRS header. The Content-Crs header just states the CRS, but has no option to state the axis order. A related general question, is whether the capability to state the axis order would be helpful or not. |

The following example illustrates the use of the OGC-CRS header.

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

OGC-CRS: http://www.opengis.net/def/crs/OGC/0/4326; axisOrder=lon,lat