

Open Geospatial Consortium

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: <yyyy-mm-dd>

External identifier of this OGC® document: <[http://www.opengis.net/doc/\[doc-type\]/\[standard\]/\[m.n\]](http://www.opengis.net/doc/[doc-type]/[standard]/[m.n])>

Internal reference number of this OGC® document: YY-**nnnr**x

Version: **n.n.n**

Category: OGC® Implementation Specification

Editor: Pekka Latvala

OGC Table Joining Service 2.0 - Part 1: Core

Copyright notice

Copyright © 2018 Open Geospatial Consortium

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

Warning

This document is not an OGC Standard. This document is distributed for review and comment. This document is subject to change without notice and may not be referred to as an OGC Standard.

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: Draft
Document language: English

License Agreement

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

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

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

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

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

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

Contents

1. Scope.....	6
2. Conformance.....	7
3. References.....	8
4. Terms and Definitions.....	8
5. Conventions	8
5.1 Identifiers	8
6. Requirement Class “core”	9
6.1 Overview	9
6.2 Operation /attributedatasets.....	9
6.2.1 Request.....	9
6.2.2 Response	9
6.2.3 Schema for the /attributedatasets response	9
6.3 Operation /attributedatasets/{attributedatasetid}	11
6.3.1 Request.....	11
6.3.2 Response	12
6.3.3 Schema for the /attributedatasets/{attributedatasetid} response.....	12
6.4 Operation /attributedatasets/{attributedatasetid}/data	17
6.4.1 Request.....	17
6.4.2 Response	18
6.4.3 Schema for the /attributedatasets/{attributedatasetid} }/data response	18
6.5 Operation /spatialdatasets.....	19
6.5.1 Request.....	19
6.5.2 Response	19
6.5.3 Schema for the /spatialdatasets response	20

6.6	Operation /spatialdatasets/{spatialdatasetid}	20
6.6.1	Request.....	21
6.6.2	Response	21
6.6.3	Schema for the /spatialdatasets/{spatialdatasetid} response	21
6.7	Operation /spatialdatasets/{spatialdatasetid}/key	22
6.7.1	Request.....	22
6.7.2	Response	22
6.7.3	Schema for the /spatialdatasets/{spatialdatasetid}/key response.....	23
6.8	Operation /joinabilities	23
6.8.1	Request.....	23
6.8.2	Response	23
6.8.3	Schema for the /joinabilities response	24
6.9	Operation /joindata/{spatialdatasetid}?attributedataseturl.....	24
6.9.1	Request.....	25
6.9.2	Response	25
6.9.3	Schema for the /joindata/{spatialdatasetid}?attributedataseturl response ..	25
7.	Media Types for any data encoding(s).....	27
A.1	Conformance class: AAAA (repeat as necessary)	28

i. Abstract

This document is a specification of the core module of the Table Joining Service (TJS) 2.0 standard. The TJS allows attribute datasets that don't contain any spatial information to be joined with spatial datasets. The joining between the datasets can be executed via shared identifiers.

The TJS specification is a multi-part document that can be extended by specifying additional modules to the core. Some potential modules are the data maintenance module for handling inserts, updates and deletes and the identifier mapping module.

The document specifies the core functionalities that all TJS implementations must support. The core functionalities are executed via HTTP GET queries and they are restricted to displaying the metadata and data values from different data sets, displaying service's join output capabilities and executing the joins between the attribute datasets and the spatial datasets. Table 1 contains an overview of the operations that are specified in the TJS core module.

Table 1 Overview of operations in the TJS core module

Path	HTTP Method	Description
/attributedatasets	GET	Returns metadata on all attribute datasets available on the server
/attributedatasets/{attributedatasetid}	GET	Returns detailed metadata on a specific attribute dataset
/attributedatasets/{attributedatasetid}/data	GET	Returns the data values of a specific attribute dataset
/spatialdatasets	GET	Returns metadata on all spatial datasets available on the server
/spatialdatasets/{spatialdatasetid}	GET	Returns the metadata on a specific spatial dataset
/spatialdatasets/{spatialdatasetid}/key	GET	Returns the key values of a specific spatial dataset
/joinabilities	GET	Returns metadata on server's join abilities
/joindata/{spatialdatasetid}?attributedataseturl	GET	Executes a join between a specific spatial dataset and attribute dataset located in the

		address of attributeddataseturl query parameter
--	--	--

ii. Keywords

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

ogcdoc, OGC document, table joining service, TJS, API, openapi

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 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):

- National Land Survey of Finland

v. Submitters

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

Name	Affiliation
Pekka Latvala	National Land Survey of Finland

1. Scope

The Table Joining Service 2.0 core module specifies a RESTful service interface that contains operations for displaying metadata and data on attribute datasets, metadata and key values on spatial datasets, metadata on server's join capabilities and an operation for executing joins between attribute datasets and spatial datasets.

The core module doesn't contain any functionalities for dataset maintenance. These functionalities can be defined in the possible additional modules. Other useful

functionalities for additional modules include identifier mapping module and a module for uploading and on-the-fly joining of various data formats, such as CSV files.

2. Conformance

This document defines 1 requirement / conformance class, “core”.

The core conformance class specifies the requirements that all TJS implementations must support.

Requirements are considered for 1 standardization target type:

- Web services

The common response format for all operations defined in the core conformance class is the JavaScript Object Notation (JSON) format. The core module does not specify any output formats for the joined datasets and the service implementations can support any formats they choose.

The core module does not specify any mandatory formats for the API definition. Some options that can be used for the API definition are different versions of the OpenAPI specification.

Conformance with this standard shall be checked using all the relevant tests specified in Annex A (normative) 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 site¹.

In order to conform to this OGCTM interface standard, a software implementation shall choose to implement:

- a) Any one of the conformance levels specified in Annex B (normative).
- b) Any one of the Distributed Computing Platform profiles specified in Annexes TBD through TBD (normative).

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

¹ www.opengeospatial.org/cite

3. References

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

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

4. Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r8], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

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

4.1

Attribute dataset

Definition...

4.2

Spatial dataset

Definition...

5. Conventions

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

5.1 Identifiers

The normative provisions in this specification are denoted by the URI

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

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

6. Requirement Class “core”

6.1 Overview

The requirement class “core” contains 8 operations that all TJS services must implement. All of these operations are executed as HTTP GET queries. The common output format for all operation responses is JSON that the servers SHALL support. Servers may support also any other output formats they choose.

6.2 Operation /attributedatasets

The HTTP GET operation at path **/attributedatasets** returns a list containing basic metadata on all attribute datasets that are available on the server.

6.2.1 Request

Req 1	<p>/req/core/attributedatasets-op</p> <p>The server SHALL support the HTTP GET operation at the path /attributedatasets</p>
--------------	---

6.2.2 Response

Req 2	<p>/req/core/attributedatasets-success</p> <p>A successful execution of the operation SHALL be reported as a response with a HTTP status code 200</p> <p>For successful operation, the server SHALL return a list containing basic metadata of all attribute datasets available on the server</p>
	<p>/req/core/attributedatasets-failure</p> <p>If attribute datasets are not found on the server, it SHALL be reported as a response with a HTTP status code 404</p>
	<p>/req/core/attributedatasets-error</p> <p>If there is an error in the server during the processing of the request, it SHALL be reported as a response with a HTTP status code 500</p>

6.2.3 Schema for the /attributedatasets response

AttributeDatasetsObject

type "object"

properties

relatedSpatialDataset

\$ref "#/definitions/RelatedSpatialDatasetObject"

attributeDatasetId

```

        type    "integer"
        format  "int32"
attributeDatasetURI
        type    "string"
relatedSpatialDatasetURI
        type    "string"
organization
        type    "string"
title
        type    "string"
attributeDatasetAbstract
        type    "string"
referenceDate
        type    "string"
startDate
        type    "string"
version
        type    "string"
documentation
        type    "string"

```

RelatedSpatialDatasetObject

```

type    "object"
properties
  spatialDatasetId
    type    "integer"
    format  "int32"
  spatialDatasetURI
    type    "string"
  organization
    type    "string"
  title
    type    "string"
  metadataAbstract
    type    "string"
  referenceDate
    type    "string"
  startDate
    type    "string"
  version
    type    "string"
  documentation
    type    "string"
  spatialDatasetKeyName

```

```

        type    "string"
    spatialDatasetKeyType
        type    "string"
    spatialDatasetKeyLength
        type    "string"
    spatialDatasetKeyDecimals
        type    "string"
    bboxNorth
        type    "string"
    bboxSouth
        type    "string"
    bboxEast
        type    "string"
    bboxWest
        type    "string"
    spatialDatasetLink
        type    "string"

```

6.3 Operation /attributedatasets/{attributedatasetid}

The HTTP GET operation at path **/attributedatasets/{attributedatasetid}** returns detailed metadata on a specific attribute dataset specified with the path parameter **attributedatasetid**.

Attribute datasets may contain 4 types of attributes:

- Nominal
 - Nominal attributes have names as attribute values (i.e. Helsinki)
- Ordinal
 - Ordinal attributes have values that contain ranking information (i.e. low, medium and high categories)
- Count
 - Count attributes are numeric and their values are cumulative total for a region (i.e. population on a region)
- Measure
 - Measure attributes are numeric and their values are measurements on a region (i.e. rainfall on a region)

6.3.1 Request

Req 3	/req/core/req-attributedatasets-attributedatasetid -op The server SHALL support the HTTP GET operation at the path /attributedatasets/{attributedatasetid}
--------------	--

6.3.2 Response

Req 4	/req/core/req-attributedatasets-attributedatasetid-success A successful execution of the operation SHALL be reported as a response with a HTTP status code 200 For successful operation, the server SHALL return the metadata on a specific attribute dataset available on the server
	/req/core/req-attributedatasets-attributedatasetid-failure If the specific attribute dataset is not found on the server, it SHALL be reported as a response with a HTTP status code 404
	/req/core/req-attributedatasets-attributedatasetid-error If there is an error in the server during the processing of the request, it SHALL be reported as a response with a HTTP status code 500

6.3.3 Schema for the /attributedatasets/{attributedatasetid}response

AttributeDatasetObject

type "object"

properties

relatedSpatialDataset

\$ref "#/definitions/RelatedSpatialDatasetObject"

attributeDatasetId

type "integer"

format "int32"

attributeDatasetURI

type "string"

relatedSpatialDatasetURI

type "string"

organization

type "string"

title

type "string"

attributeDatasetAbstract

type "string"

referenceDate

type "string"

startDate

type "string"

version

type "string"

documentation

type "string"

```
columnset
  $ref "#/definitions/ColumnsetObject"
```

RelatedSpatialDatasetObject

```
type "object"
properties
  spatialDatasetId
    type "integer"
    format "int32"
  spatialDatasetURI
    type "string"
  organization
    type "string"
  title
    type "string"
  metadataAbstract
    type "string"
  referenceDate
    type "string"
  startDate
    type "string"
  version
    type "string"
  documentation
    type "string"
  spatialDatasetKeyName
    type "string"
  spatialDatasetKeyType
    type "string"
  spatialDatasetKeyLength
    type "string"
  spatialDatasetKeyDecimals
    type "string"
  bboxNorth
    type "string"
  bboxSouth
    type "string"
  bboxEast
    type "string"
  bboxWest
    type "string"
  spatialDatasetLink
    type "string"
```

```

ColumnsetObject
type "object"
properties
  frameworkKey
    $ref "#/definitions/ColumnsetFrameworkKeyObject"
  attributes
    type "array"
    items
      $ref "#/definitions/ColumnsetAttributeColumnObject"

```

```

ColumnsetFrameworkKeyObject
type "object"
properties
  complete
    type "boolean"
    default false
  relationship
    type "string"
  type
    type "string"
  name
    type "string"

```

```

ColumnsetAttributeColumnObject
type "object"
properties
  type
    type "string"
  name
    type "string"
  purpose
    type "string"
  title
    type "string"
  columnAbstract
    type "string"
  documentation
    type "string"
  count
    $ref "#/definitions/ColumnsetAttributeColumnValuesCountObject"
  nominal

```

\$ref "#/definitions/ColumnsetAttributeColumnValuesNominalObject"

ColumnsetAttributeColumnValuesCountObject

type "object"

properties

exceptions

\$ref

"#/definitions/ColumnsetAttributeColumnValuesCountExceptionsObject"

uom

\$ref

"#/definitions/ColumnsetAttributeColumnValuesCountUOMObject"

uncertainty

\$ref

"#/definitions/ColumnsetAttributeColumnValuesCountUncertaintyObject"

ColumnsetAttributeColumnValuesCountExceptionsObject

type "object"

properties

identifier

type "string"

title

type "string"

exceptionAbstract

type "string"

documentation

type "string"

color

type "string"

ColumnsetAttributeColumnValuesCountUOMObject

type "object"

properties

reference

type "string"

shortForm

type "string"

longForm

type "string"

ColumnsetAttributeColumnValuesCountUncertaintyObject

```
type "object"
properties
  gaussian
    type "string"
```

ColumnsetAttributeColumnValuesNominalObject

```
type "object"
properties
  classes
    $ref
    "#/definitions/ColumnsetAttributeColumnValuesNominalClassesObject"
  exceptions
    $ref
    "#/definitions/ColumnsetAttributeColumnValuesCountExceptionsObject"
```

ColumnsetAttributeColumnValuesNominalClassesObject

```
type "object"
properties
  title
    type "string"
  abstract_
    type "string"
  documentation
    type "string"
  value
    $ref
    "#/definitions/ColumnsetAttributeColumnValuesNominalClassesValueObject"
```

ColumnsetAttributeColumnValuesCountExceptionsObject

```
type "object"
properties
  identifier
    type "string"
  title
    type "string"
  exceptionAbstract
    type "string"
  documentation
```



```

        type    "string"
color
        type    "string"

```

ColumnsetAttributeColumnValuesNominalClassesValueObject

```

type    "object"
properties
  identifier
    type    "string"
  title
    type    "string"
  abstract_
    type    "string"
  documentation
    type    "string"
  color
    type    "string"

```

6.4 Operation /attributedatasets/{attributedatasetid}/data

The HTTP GET operation at path **/attributedatasets/{attributedatasetid}/data** returns the data values of a specific attribute dataset specified with the path parameter attributedatasetid.

The operation MAY contain a query parameter **attributes** that can be used for selecting specific attributes for the response. If the attributes parameter is not used or available, the server SHALL return all attributes that are related to the particular attribute dataset.

6.4.1 Request

Req 5

/req/core/req-attributedatasets-attributedatasetid-data-op

The server SHALL support the HTTP GET operation at the path /attributedatasets/{attributedatasetid}/data

Query parameters:

Name	Description	Type and values	Required
attributes	The names of the attributes that are requested from the	String type. The value contains the names of the requested attributes in a comma-	Optional

		attribute dataset	delimited format	
--	--	----------------------	---------------------	--

6.4.2 Response

Req 6	/req/core/req-attributedatasets-attributedatasetid-data-success A successful execution of the operation SHALL be reported as a response with a HTTP status code 200 For successful operation, the server SHALL return a data of a specific attribute dataset available on the server
	/req/core/req-attributedatasets-attributedatasetid-data-failure If the attribute dataset data are not found on the server, it SHALL be reported as a response with a HTTP status code 404
	/req/core/req-attributedatasets-attributedatasetid-data-error If there is an error in the server during the processing of the request, it SHALL be reported as a response with a HTTP status code 500

6.4.3 Schema for the /attributedatasets/{attributedatasetid}/data response

RowsetObject

type "object"

properties

datasetUri

type "string"

attributes

type "array"

items

\$ref "#/definitions/RowsetAttributeObject"

rowset

type "array"

items

\$ref "#/definitions/RowsetRowObject"

RowsetAttributeObject

type "object"

properties

attributeName

type "string"

```

RowsetRowObject
type  "object"
properties
  k
      type  "string"
  v
      type  "array"
      items
          type  "string"

```

6.5 Operation /spatialdatasets

The HTTP GET operation at path **/spatialdatasets** returns a list containing metadata on all spatial datasets that are available on the server.

6.5.1 Request

Req 7	/req/core/req-spatialdatasets-op The server SHALL support the HTTP GET operation at the path /spatialdatasets
--------------	--

6.5.2 Response

Req 8	/req/core/req-spatialdatasets-success A successful execution of the operation SHALL be reported as a response with a HTTP status code 200 The server SHALL return a metadata on all spatial dataset available in the server
	/req/core/req-spatialdatasets-failure If the spatial datasets are not found on the server, it SHALL be reported as a response with a HTTP status code 404
	/req/core/req-spatialdatasets-error If there is an error in the server during the processing of the request, it SHALL be reported as a response with a HTTP status code 500

6.5.3 Schema for the /spatialdatasets response

SpatialDatasetObject

type "object"

properties

spatialDatasetId

type "integer"

format "int32"

spatialDatasetURI

type "string"

organization

type "string"

title

type "string"

metadataAbstract

type "string"

referenceDate

type "string"

startDate

type "string"

version

type "string"

documentation

type "string"

spatialDatasetKeyName

type "string"

spatialDatasetKeyType

type "string"

spatialDatasetKeyLength

type "string"

spatialDatasetKeyDecimals

type "string"

bboxNorth

type "string"

bboxSouth

type "string"

bboxEast

type "string"

bboxWest

type "string"

6.6 Operation /spatialdatasets/{spatialdatasetid}

The HTTP GET operation at path **/spatialdatasets/{spatialdatasetid}** returns metadata on a specific spatial dataset specified with the path parameter **spatialdatasetid**.

6.6.1 Request

Req 9	<p>/req/core/req-spatialdatasets-spatialdatasetid-op</p> <p>The server SHALL support the HTTP GET operation at the path /spatialdatasets/{spatialdatasetid}</p>
--------------	---

6.6.2 Response

Req 10	<p>/req/core/req-spatialdatasets-spatialdatasetid-success</p> <p>A successful execution of the operation SHALL be reported as a response with a HTTP status code 200</p> <p>The server SHALL return metadata on a specific spatial dataset available in the server</p>
	<p>/req/core/req-spatialdatasets-spatialdatasetid-failure</p> <p>If the spatial dataset is not found on the server, it SHALL be reported as a response with a HTTP status code 404</p>
	<p>/req/core/req-spatialdatasets-spatialdatasetid-error</p> <p>If there is an error in the server during the processing of the request, it SHALL be reported as a response with a HTTP status code 500</p>

6.6.3 Schema for the /spatialdatasets/{spatialdatasetid} response

```
SpatialDatasetObject
type "object"
properties
  spatialDatasetId
    type "integer"
    format "int32"
  spatialDatasetURI
    type "string"
  organization
    type "string"
  title
    type "string"
  metadataAbstract
    type "string"
  referenceDate
    type "string"
  startDate
```

```

        type    "string"
    version
        type    "string"
    documentation
        type    "string"
    spatialDatasetKeyName
        type    "string"
    spatialDatasetKeyType
        type    "string"
    spatialDatasetKeyLength
        type    "string"
    spatialDatasetKeyDecimals
        type    "string"
    bboxNorth
        type    "string"
    bboxSouth
        type    "string"
    bboxEast
        type    "string"
    bboxWest
        type    "string"

```

6.7 Operation /spatialdatasets/{spatialdatasetid}/key

The HTTP GET operation at path **/spatialdatasets/{spatialdatasetid}/key** returns a list of key values of a specific spatial dataset.

6.7.1 Request

Req 11	/req/core/req-spatialdatasets-spatialdatasetid-key-op The server SHALL support the HTTP GET operation at the path /spatialdatasets/{spatialdatasetid}/key
---------------	---

6.7.2 Response

Req 12	/req/core/req-spatialdatasets-spatialdatasetid-key-success A successful execution of the operation SHALL be reported as a response with a HTTP status code 200 The server SHALL return the key values of a specific spatial dataset.
	/req/core/req-spatialdatasets-spatialdatasetid-key-failure If the spatial dataset key values are not found on the server, it SHALL

	be reported as a response with a HTTP status code 404
	/req/core/req-spatialdatasets-spatialdatasetid-key-error If there is an error in the server during the processing of the request, it SHALL be reported as a response with a HTTP status code 500

6.7.3 Schema for the /spatialdatasets/{spatialdatasetid}/key response

```

SpatialDatasetKeyObject
type "object"
properties
  key
    type "string"
  title
    type "string"

```

6.8 Operation /joinabilities

The HTTP GET operation at path /joinabilities returns a list of output formats that are supported by the server in the data join.

6.8.1 Request

Req 13	/req/core/req-joinabilities-op The server SHALL support the HTTP GET operation at the path /joinabilities
---------------	--

6.8.2 Response

Req 14	/req/core/req-joinabilities-success A successful execution of the operation SHALL be reported as a response with a HTTP status code 200 The server SHALL return metadata on the output formats that are supported by the server in the data join
	/req/core/req-joinabilities-failure If the join abilities metadata are not found on the server, it SHALL be reported as a response with a HTTP status code 404

	/req/core/req-joinabilities-error If there is an error in the server during the processing of the request, it SHALL be reported as a response with a HTTP status code 500
--	--

6.8.3 Schema for the /joinabilities response

```

OutputMechanismObject
type  "object"
properties
  identifier
    type  "string"
  title
    type  "string"
  description
    type  "string"
  reference
    type  "string"

```

6.9 Operation /joindata/{spatialdatasetid}?attributedataseturl

The HTTP GET operation at path **/joindata/{spatialdatasetid}** joins an attribute dataset data with a spatial dataset specified with the **spatialdatasetid** parameter. The attribute dataset data is retrieved from the URL given by the mandatory query parameter **attributedataseturl**.

The operation response contains a list of created outputs and information on the successfulness of the join. The join successfulness information includes following fields:

- **numberOfMatchedSpatialDatasetKeys**
 - Contains the amount of spatial dataset keys, to which attribute data was successfully joined
- **matchedSpatialDatasetKeys**
 - Contains a list of spatial dataset keys, to which attribute data was successfully joined
- **numberOfUnmatchedSpatialDatasetKeys**
 - Contains the amount of spatial dataset keys, to which attribute data could not be joined
- **unmatchedSpatialDatasetKeys**
 - Contains a list of spatial dataset keys, to which attribute data could not be joined
- **numberOfAdditionalAttributeKeys**
 - Contains the amount of attribute elements that didn't have corresponding spatial features available

- AdditionalAttributeKeyList
 - Contains a list of attribute identifiers that didn't have corresponding spatial features available

6.9.1 Request

Req 15

/req/core/req-joindata-spatialdatasetid-op

The server SHALL support the HTTP GET operation at the path /joindata/{ spatialdatasetid }

Query parameters:

Name	Description	Type and values	Required
attributedataseturl	The URL to the attribute dataset data request	URL type, not empty. The value is a request from Chapter 6.4	Mandatory

6.9.2 Response

Req 16	<p>/req/core/req-joindata-spatialdatasetid-success</p> <p>A successful execution of the operation SHALL be reported as a response with a HTTP status code 200</p> <p>The server SHALL return information on the join's successfulness and links to the created outputs.</p> <p>/req/core/req-joindata-spatialdatasetid-failure</p> <p>If the spatial dataset key values are not found on the server, it SHALL be reported as a response with a HTTP status code 404</p> <p>/req/core/req-joindata-spatialdatasetid-errpr</p> <p>If there is an error in the server during the processing of the request, it SHALL be reported as a response with a HTTP status code 500</p>
---------------	---

6.9.3 Schema for the /joindata/{spatialdatasetid}?attributedataseturl response

```

JoinDataObject
type "object"
properties
  inputs
    $ref "#/definitions/InputsObject"
  outputs
    type "array"
    items
      $ref "#/definitions/OutputObject"
  joinInformation
    $ref "#/definitions/JoinInformationObject"

```

```

InputsObject
type "object"
properties
  attributeDataset
    type "string"
  spatialDataset
    type "string"

```

```

OutputObject
type "object"
properties
  format
    type "string"
  link
    type "string"

```

```

JoinInformationObject
type "object"
properties
  numberOfMatchedSpatialDatasetKeys
    type "integer"
    format "int32"
  numberOfUnmatchedSpatialDatasetKeys
    type "integer"
    format "int32"
  numberOfAdditionalAttributeKeys
    type "integer"
    format "int32"
  matchedSpatialDatasetKeys
    type "array"
    items

```

```

        type    "string"
unmatchedSpatialDatasetKeys
    type    "array"
    items
        type    "string"
additionalAttributeKeysList
    type    "array"
    items
        type    "string"

```

7. Media Types for any data encoding(s)

A section describing the MIME-types to be used is mandatory for any standard involving data encodings. If no suitable MIME type exists in <http://www.iana.org/assignments/media-types/index.html> then this section may be used to define a new MIME type for registration with IANA.

Annex A: Conformance Class Abstract Test Suite (Normative)

A.1 Conformance class: AAAA (repeat as necessary)

Annex <insert Annex number>: Revision history

Date	Release	Author	Paragraph modified	Description

Annex <insert annex number>: Bibliography

<A Bibliography, if present, shall appear as the last annex. >