**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/standard/tjs/2.0>

Internal reference number of this OGC® document:    YY-nnnrx

Version: n.n.n

Category: OGC® Implementation Specification

Editor: Pekka Latvala

OGC Table Joining Service 2.0 - Part 1: Core

**Copyright notice**

Copyright © 2019 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.

Document type:    OGC® Standard

Document subtype:    Interface

Document stage:    Draft

Document language:  English

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.

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](#_Toc9423737)

[2. Conformance 6](#_Toc9423738)

[3. References 7](#_Toc9423739)

[4. Terms and Definitions 7](#_Toc9423740)

[5. Conventions 8](#_Toc9423741)

[5.1 Identifiers 8](#_Toc9423742)

[6. Requirement Class “core” 8](#_Toc9423743)

[6.1 Overview 8](#_Toc9423744)

[6.2 Operation Set *Discovery Operations* 9](#_Toc9423745)

[6.2.1 GET operation at path / 10](#_Toc9423746)

[6.2.2 GET operation at path /api 12](#_Toc9423747)

[6.2.3 GET operation at path /conformance 13](#_Toc9423748)

[6.3 Operation Set *Data Joining Operations* 14](#_Toc9423749)

[6.3.1 GET operation at path /spatialdatasets 14](#_Toc9423750)

[6.3.2 GET operation at path /spatialdatasets/{spatialdatasetid} 18](#_Toc9423751)

[6.3.3 GET operation at path /spatialdatasets/{spatialdatasetid}/keys 22](#_Toc9423752)

[6.3.4 GET operation at path /spatialdatasets/{spatialdatasetid}/keys/{key} 25](#_Toc9423753)

[6.3.5 POST operation at path /joindata/{spatialdatasetid}/csv 27](#_Toc9423754)

[6.3.6 GET operation at path /joins 35](#_Toc9423755)

[6.3.7 GET operation at path /joins/{joinid} 38](#_Toc9423756)

[6.3.8 POST Operation at path /joins/{joinid}/csv 42](#_Toc9423757)

[6.3.9 DELETE Operation at path /joins/{joinid} 49](#_Toc9423758)

[6.4 Operation Set *File Joining Operations* 50](#_Toc9423759)

[6.4.1 POST operation at path /joinfiles/geojson/csv 50](#_Toc9423760)

[7. Media Types for any data encoding(s) 54](#_Toc9423761)

[*7.1* Operation sets: *discovery operations and data joining operations* 54](#_Toc9423762)

[*7.2* Outputs for the joined data in operation set: *data joining operations* 54](#_Toc9423763)

[*7.3* Operation set: *file joining operations* 55](#_Toc9423764)

[A.1 Conformance class: AAAA (repeat as necessary) 56](#_Toc9423765)

Abstract

This document is a specification for the core module of the Table Joining Service (TJS) 2.0 standard. The TJS core module specifies a service interface that allows non-spatial attribute data to be joined with spatial datasets via common identifiers that are available in both datasets. The TJS 2.0 standard supports also operations for viewing metadata on spatial datasets that are available on the server, operations for accessing, updating and deleting the created joins and operation for joining attribute data directly with inputted spatial data files.

Keywords

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

ogcdoc, OGC document, standard, TJS, API, openapi

Preface

This standard is the result of the work that was executed to renew the OGC implementation standard: OpenGIS® Georeferenced Table Joining Service (TJS) (document nr. 10-070r2), specified in 2010.

This document defines the core module of the TJS 2.0 standard. The TJS 2.0 specification is a multi-part document that can be extended by specifying extension modules to the core.

This document does not suggest any updates to the OGC Abstract Specification

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.*

Submitting organizations

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

* National Land Survey of Finland

Submitters

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

|  |  |
| --- | --- |
| Name | Affiliation |
| Pekka Latvala | National Land Survey of Finland |

# Scope

This OpenGIS® standard defines the core module for the Table Joining Service 2.0 specification. The TJS core module specifies a RESTful service interface that contains 3 operation sets: *discovery operations, data joining operations* and *file joining operations*.

The operation set *discovery operations* contains operations for obtaining general information on the TJS implementation. It includes operations for accessing the API landing page, the API definition file and information on the service’s conformance to the TJS 2.0 standard.

The operation set *data joining* *operations* contains functionalities for displaying metadata and key values on the spatial datasets that are available on the server, joining attribute data from csv files with these spatial datasets and accessing, updating and deleting the created joins.

The operation set *file joining* *operations* can be used for joining attribute data directly with inputted spatial data files. The core module contains file joining support between GeoJSON spatial data files and csv attribute data files.

The core module doesn’t contain any functionalities for inserting, updating or deleting the spatial datasets on the server. These functionalities can be defined in possible extension modules. The support for other attribute data formats than csv in the *data joining operations* and *file joining operations*, together with the support for other spatial data formats than GeoJSON in the *file joining operations*can be also defined in potential extension modules.

# Conformance

This standard defines 1 requirement class: “core”.

Requirements for 1 standardization target types are considered:

* Web services

This standard defines three operations classes *discovery operations, data joining* *operations and file joining* *operations.*

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 site[[1]](#footnote-1).

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

1. Any one of the conformance levels specified in Annex B (normative).
2. 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.

# 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.

IETF RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1. 1999

IETF RFC 3986, Uniform Resource Identifier (URI): Generic Syntax. 2005

IETF RFC 6266 Use of the Content-Disposition Header Field in the Hypertext Transfer Protocol (HTTP). 2011

IETF RFC 7578 Returning Values from Forms: multipart/form-data. 2015

IETF RFC 7946 The GeoJSON Format. 2016

The OpenAPI specification 3.0.1

# 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.

attribute dataset

Dataset that contains attribute information that can be joined with a spatial dataset through common identifiers.

spatial dataset

Dataset that contains geometry information.

# 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.

## Identifiers

The normative provisions in this specification are denoted by the URI

http://www.opengis.net/spec/tjs/2.0

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

# Requirement Class “core”

## Overview

The Table Joining Service requirement class “core” contains 3 operation sets: *discovery operations, data joining operations and file joining operations.* The Table 1 contains an overview of the operations specified in the core module.

The operation set *discovery operations* contains functionalities for accessing the API landing page, the API definition file and the information on the service’s conformance to the TJS 2.0 specification. All TJS 2.0 implementations SHALL support all operations in the operation set *discovery operations.*

All TJS 2.0 implementations SHALL support at least one of the operation sets: *data joining operations* and *file joining* operations. If a server supports a particular operation set it SHALL implement all mandatory operations that belong to it.

The operation set *data joining operations* contains functionalities for accessing metadata and key values on the spatial datasets that are available on the server, joining csv files with the spatial datasets and accessing, updating and deleting the joins on the server.

The operation set *file joining operations* contains a functionality for joining attribute data from an inputted csv file to an inputted GeoJSON file.

Table 1: Overview of operations in the TJS 2.0 core module

|  |  |  |
| --- | --- | --- |
| **Path** | **HTTP method** | **Description** |
| **Discovery operations** | | |
| / | GET | API landing page |
| /api | GET | API definition file |
| /conformance | GET | API conformance declaration |
| **Data joining operations** | | |
| /spatialdatasets | GET | Returns metadata on all spatial datasets available on the server |
| /spatialdatasets/{spatialdatasetid} | GET | Returns metadata on a specific spatial dataset |
| /spatialdatasets/{spatialdatasetid}/keys | GET | Returns the key values of a specific spatial dataset |
| /spatialdatasets/{spatialdatasetid}/keys/{key} | GET | Returns a specific key value of a specific spatial dataset |
| /joindata/{spatialdatasetid}/csv | POST | Creates a new join by joining attribute data from a csv file with a specific spatial dataset |
| /joins | GET | Returns a list of all joins available on the server |
| /joins/{joinid} | GET | Returns metadata on a specific join |
| /joins/{joinid}/csv | POST | Updates fully a specific join with a data from a csv file |
| /joins/{joinid} | DELETE | Deletes a specific join |
| **File joining operations** | | |
| /joinfiles/geojson/csv | POST | Joins attribute data from a csv file with a GeoJSON file |
|  | | |

## Operation Set *Discovery Operations*

The operation set *discovery operations* contains three operations that provide general information about the TJS implementation. All TJS 2.0 implementations SHALL support all operations in this operation set.

The recommended response format for all operations defined in the *discovery operations* operation set is JSON.

### GET operation at path /

The HTTP GET operation at service root path / returns the API landing page. The landing page response document contains links to the API definition, conformance information and to the metadata on spatial dataset and created joins that are available on the server.

#### Request

|  |
| --- |
| 1. The server SHALL support the HTTP GET operation at the service root path /   http://www.opengis.net/spec/tjs/2.0//req/core/landing-page-op |

#### Response

|  |
| --- |
| 1. A successful execution of the operation SHALL be reported as a response with a HTTP status code 200.   The server SHALL return an API landing page document that is based on the schema presented in Chapter 6.2.1.4. The recommended response format is JSON. The response document SHALL contains at least links to the following resources:   * self (link ‘rel’ property value: ‘*self’’)* * /api (link ‘rel’ property value: ‘*service’*) * /conformance (link ‘rel’ property value: ‘*conformance’*) * /spatialdatasets (link ‘rel’ property value: ‘*spatialdatasets’*) * /joins (link ‘rel’ property value: ‘*joins’*)   The links SHALL contain the property values *href, rel, title and type.*  http://www.opengis.net/spec/tjs/2.0/req/core/landing-page-success |

#### Errors

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. 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.   The response body SHALL contain an exception report message. The recommended format is JSON.  **Structure of the exception report message:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response. | | message | Details of the exception. The message shall contain a value ‘Internal server error’ |   http://www.opengis.net/spec/tjs/2.0/req/core/landing-page-error |

#### Response schema

schema:

$ref: '#/components/schemas/LandingPageResponseObject'

LandingPageResponseObject:

required:

- links

type: object

properties:

links:

type: array

items:

$ref: '#/components/schemas/Link'

#### Response schema for Link object

Link:

required:

- href

type: object

properties:

href:

type: string

rel:

type: string

type:

type: string

hreflang:

type: string

title:

type: string

### GET operation at path /api

The HTTP GET operation at path /api returns the service’s API definition.

#### Request

|  |
| --- |
| 1. The server SHALL support the HTTP GET operation at the path /api   http://www.opengis.net/spec/tjs/2.0/req/core/api-op |

#### Response

|  |
| --- |
| 1. A successful execution of the operation SHALL be reported as a response with a HTTP status code 200.   The server SHALL return an API definition document. The recommended format for the API definition is OpenAPI version 3.0.1.  http://www.opengis.net/spec/tjs/2.0/req/core/api-op-success |

#### Errors

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. 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.   The response body SHALL contain an exception report message. The recommended format is JSON.  **Structure of the exception report message:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response. | | message | Details of the exception. The message shall contain a value ‘Internal server error’ |   http://www.opengis.net/spec/tjs/2.0/req/core/api-op-error |

### GET operation at path /conformance

The HTTP GET operation at path /conformance returns a list of TJS 2.0 conformance classes that the server supports.

The conformance class values, defined in the TJS 2.0 core module are:

* Data joining from CSV files
  + “http://www.opengis.net/spec/tjs/2.0/req/core/data-joining-csv”
* Join updating from CSV files:
  + “http://www.opengis.net/spec/tjs/2.0/req/core/data-joining-csv-update”
* Join deleting
  + “http://www.opengis.net/spec/tjs/2.0/req/core/data-joining-delete”
* File joining
  + “http://www.opengis.net/spec/tjs/2.0/req/core/file-joining-geojson-csv”

#### Request

|  |
| --- |
| 1. The server SHALL support the HTTP GET operation at the path /conformance   http://www.opengis.net/spec/tjs/2.0/req/core/conformance-op |

#### Response

|  |
| --- |
| 1. A successful execution of the operation SHALL be reported as a response with a HTTP status code 200.   The server SHALL return a list of TJS 2.0 conformance class names that the service implements. The response document is based on the schema presented in Chapter 6.2.3.4.  http://www.opengis.net/spec/tjs/2.0/req/core/conformance-op-success |

#### Errors

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. 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.   The response body SHALL contain an exception report message. The recommended format is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response. | | message | Details of the exception. The message shall contain a value ‘Internal server error’ |   http://www.opengis.net/spec/tjs/2.0/req/core/conformance-op-error |

#### Response schema

schema:

$ref: '#/components/schemas/ConformanceResponseObject'

ConformanceResponseObject:

required:

- conformsTo

type: object

properties:

conformsTo:

type: array

items:

type: string

## Operation Set *Data Joining Operations*

The operation set *data joining operations* contains operations for:

* Retrieving metadata and key values on spatial datasets available on the server
* Joining attribute data from CSV files with spatial datasets available on the server
* Accessing, updating and deleting the created joins.

The data joining is executed through common keys that are shared between the spatial dataset and the attribute dataset.

The recommended response format for all operations defined in the *data joining operations* operation set is JSON.

### GET operation at path /spatialdatasets

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

#### Request

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. If the server implements the data joining operation set it SHALL support the HTTP GET operation at path /spatialdatasets   The HTTP GET request at path /spatialdatasets MAY support the following query parameters:  **Query parameters:**   |  |  |  | | --- | --- | --- | | **Name** | **Type** | **Description** | | organization | String | Filters spatial datasets by organization | | startDate | String | Filters spatial datasets by start date. Format: “yyyy-mm-dd” | | endDate | String | Filters spatial datasets by end date. Format: “yyyy-mm-dd”  Format: “yyyy-nn-dd”  Format: “yyyy-nn-dd” |   http://www.opengis.net/spec/tjs/2.0/req/core/spatialdatasets-op |

#### Response

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. A successful execution of the operation SHALL be reported as a response with a HTTP status code 200.   The recommended response format for the operation is JSON. The response schema is presented in Chapter 6.3.1.4. The response’s *links* property shall include the following links (Chapter 6.2.1.4)*.*   * A link to this document (link rel: self) * Links to this document in other supported media types (link rel: alternate)   The links SHALL contain the parameters *href, rel, title* and *type.*  The response’s spatialDatasets array shall contain metadata on all spatial datasets that are available on the server*.*  **Description of properties of the spatialDatasetObject property:**   |  |  | | --- | --- | | **Name** | **Description** | | date | Date that applies to the spatial dataset. Format: “yyyy-mm-dd” | | description | Description of the spatial dataset | | documentation | Link to the spatial dataset’s documentation | | links | Link to this spatial dataset’s different representations. The links SHALL include the properties *href, rel* and *type*. The value of the *rel* property SHALL be ‘spatialdataset’. | | organization | Name of the organization that has produced the spatial dataset | | spatialDatasetId | Unique identifier for the spatial dataset. The identifier is used in other operations for indicating the spatial dataset in question | | spatialDatasetURI | URI that identifies uniquely the spatial dataset | | title | Title of the spatial dataset |   http://www.opengis.net/spec/tjs/2.0/req/core/spatialdatasets-op-success |

#### Errors

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. If an incorrect request is made to the server, it SHALL be reported as a response with a HTTP status code 400.   The response body SHALL contain an exception report message. The recommended format is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. Possible values are ‘*InvalidParameterValue’* and ‘*MissingParameterValue’* | | locator | Name of the request parameter that caused the exception |   http://www.opengis.net/spec/tjs/2.0/req/core/spatialdatasets-op-bad-request |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. If spatial datasets are not found on the server, it SHALL be reported as a response with a HTTP status code 404.   The response body SHALL contain an exception report message. The recommended format is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. The message shall contain a value ‘Spatial datasets not found’ |   http://www.opengis.net/spec/tjs/2.0/req/core/spatialdatasets-op-not-found |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. 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.   The response body SHALL contain an exception report message. The recommended format is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. The message shall contain a value ‘Internal server error’ |     http://www.opengis.net/spec/tjs/2.0/req/core/spatialdatasets-op-error |

#### Response schema

schema:

$ref: '#/components/schemas/SpatialDatasetsResponseObject'

SpatialDatasetsResponseObject:

required:

- links

- spatialDatasets

type: object

properties:

links:

type: array

items:

$ref: '#/components/schemas/Link'

spatialDatasets:

type: array

items:

$ref: '#/components/schemas/SpatialDatasetsObject'

SpatialDatasetsObject:

required:

- date

- description

- links

- organization

- spatialDatasetId

- spatialDatasetURI

- title

type: object

properties:

date:

type: string

description:

type: string

documentation:

type: string

links:

type: array

items:

$ref: '#/components/schemas/Link'

organization:

type: string

spatialDatasetId:

type: integer

format: int32

spatialDatasetURI:

type: string

title:

type: string

### GET operation at path /spatialdatasets/{spatialdatasetid}

The HTTP GET operation at path /spatialdatasets/{spatialdatasetid} returns metadata on a specific spatial dataset available on the server.

#### Request

|  |
| --- |
| 1. If the server implements the data joining operation set it SHALL support the HTTP GET operation at path /spatialdatasets/{spatialdatasetid}   http://www.opengis.net/spec/tjs/2.0/req/core/spatialdatasets-spatialdatasetid-op |

#### Response

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. A successful execution of the operation shall be reported as a response with a HTTP status code 200.   The recommended response format for the operation is JSON. The response schema is presented in Chapter 6.3.2.4. The response’s *links* property shall include the following links (Chapter 6.2.1.4)*.*   * A link to this document (link rel: self) * Links to this document in other supported media types (link rel: alternate)   The links SHALL contain the parameters *href, rel, title* and *type.*  The response property: *spatialDataset* SHALL contain metadata on the requested spatial dataset*.*  **Description of properties of the response’s spatialDataset property:**   |  |  | | --- | --- | | **Name** | **Description** | | date | Date that applies to the spatial dataset. Format: “yyyy-mm-dd” | | description | Description of the spatial dataset | | documentation | Link to the spatial dataset’s documentation | | keys | Links to the different representations of this spatial dataset’s key values. The links SHALL include the properties *href, rel* and *type*. The value of the *rel* property SHALL be ‘keys’. | | organization | Name of the organization that has produced the spatial dataset | | spatialDatasetId | Unique identifier for the spatial dataset | | spatialDatasetURI | URI that identifies uniquely the spatial dataset | | title | Title of the spatial dataset |   http://www.opengis.net/spec/tjs/2.0/req/core/spatialdatasets-spatialdatasetid--op-success |

#### Errors

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. If an incorrect request is made to the server, it SHALL be reported as a response with a HTTP status code 400.   The response body SHALL contain an exception report message. The recommended format is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. Possible values are ‘*InvalidParameterValue’* and ‘*MissingParameterValue’* | | locator | Name of the request parameter that caused the exception |   http://www.opengis.net/spec/tjs/2.0/req/core/spatialdatasets-spatialdatasetid-op-bad-request |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. If the spatial dataset is not found on the server, it SHALL be reported as a response with a HTTP status code 404.   The response body SHALL contain an exception report message. The recommended format is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. The message shall contain a value ‘Spatial dataset {spatialdatasetid} not found’ |   http://www.opengis.net/spec/tjs/2.0/req/core/spatialdatasets-spatialdatasetid-op-not-found |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. 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.   The response body SHALL contain an exception report message. The recommended format is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response. | | message | Details of the exception. The message shall contain a value ‘Internal server error’ |   http://www.opengis.net/spec/tjs/2.0/req/core/spatialdatasets-spatialdatasetid-op-error |

#### Response schema

schema:

$ref: '#/components/schemas/SpatialDatasetResponseObject'

SpatialDatasetResponseObject:

required:

- links

type: object

properties:

links:

type: array

items:

$ref: '#/components/schemas/Link'

spatialDataset:

$ref: '#/components/schemas/SpatialDatasetObject'

SpatialDatasetObject:

required:

- date

- description

- organization

- spatialDatasetId

- spatialDatasetURI

- title

type: object

properties:

date:

type: string

description:

type: string

documentation:

type: string

keys:

type: array

items:

$ref: '#/components/schemas/Link'

organization:

type: string

spatialDatasetId:

type: integer

format: int32

spatialDatasetURI:

type: string

title:

type: string

### GET operation at path /spatialdatasets/{spatialdatasetid}/keys

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

#### Request

|  |
| --- |
| 1. If the server implements the data joining operation set, it SHALL support the HTTP GET operation at the path /spatialdatasets/{spatialdatasetid}/keys   http://www.opengis.net/spec/tjs/2.0/req/core/spatialdatasets-spatialdatasetid-keys-op |

#### Response

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. A successful execution of the operation SHALL be reported as a response with a HTTP status code 200.   The recommended response format for the operation is JSON. The response schema is presented in Chapter 6.3.3.4. The response’s *links* property shall include the following links (Chapter 6.2.1.4)*.*   * A link to this document (link rel: self) * Links to this document in other supported media types (link rel: alternate)   The links SHALL contain the parameters *href, rel, title* and type.  **Description of response’s keys property:**   |  |  | | --- | --- | | **Name** | **Description** | | key | Spatial dataset key value | | links | Links to the different representations of this key value. Properties *href, rel* and *type* are mandatory. The value of the *rel* property SHALL be ‘key’ | | title | Human-readable description of the key value |   http://www.opengis.net/spec/tjs/2.0/req/core/spatialdatasets-spatialdatasetid-keys-op-success |

#### Errors

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. If the spatial dataset keys are not found on the server, it SHALL be reported as a response with a HTTP status code 404.   The response body SHALL contain an exception report message. The recommended format is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. Value ‘Spatial dataset {spatialdatasetid} keys not found’ |   http://www.opengis.net/spec/tjs/2.0/req/core/spatialdatasets-spatialdatasetid-keys-op-not-found |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. 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.   The response body SHALL contain an exception report message. The recommended format is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. The message shall contain a value ‘Internal server error’ |   http://www.opengis.net/spec/tjs/2.0/req/core/spatialdatasets-spatialdatasetid-keys-op-error |

#### Response schema

schema:

type: array

items:

$ref: '#/components/schemas/SpatialDatasetKeysResponseObject'

SpatialDatasetKeysResponseObject:

required:

- keys

- links

type: object

properties:

links:

type: array

items:

$ref: '#/components/schemas/Link'

keys:

type: array

items:

$ref: '#/components/schemas/SpatialDatasetKeysObject'

SpatialDatasetKeysObject:

required:

- key

- links

type: object

properties:

key:

type: string

links:

type: array

items:

$ref: '#/components/schemas/Link'

title:

type: string

### GET operation at path /spatialdatasets/{spatialdatasetid}/keys/{key}

The HTTP GET operation at path **/spatialdatasets/{spatialdatasetid}/keys/{key}** returns a specific key value from a specific spatial dataset.

#### Request

|  |
| --- |
| 1. If the server implements the data joining operation set it SHALL support the HTTP GET operation at the path /spatialdatasets/{spatialdatasetid}/keys/{key}   http://www.opengis.net/spec/tjs/2.0/req/core/spatialdatasets-spatialdatasetid-keys-key-op |

#### Response

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. A successful execution of the operation SHALL be reported as a response with a HTTP status code 200.   The recommended response format for the operation is JSON. The response schema is presented in Chapter 6.3.4.4. The response’s *links* property shall include the following links (Chapter 6.2.1.4)*.*   * A link to this document (link rel: self) * Links to this document in other supported media types (link rel: alternate)   The links SHALL contain the parameters *href, rel, title* and type  **Description of properties of the key response property:**   |  |  | | --- | --- | | **Name** | **Description** | | key | Spatial dataset key | | title | Human-readable description of the key value |   http://www.opengis.net/spec/tjs/2.0/req/core/spatialdatasets-spatialdatasetid-keys-key-op-success |

#### Errors

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. If the spatial dataset key value is not found on the server, it SHALL be reported as a response with a HTTP status code 404.   The response body SHALL contain an exception report message. The recommended format is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. Value ‘Spatial dataset {spatialdatasetid} key {key} not found’ |   http://www.opengis.net/spec/tjs/2.0/req/core/spatialdatasets-spatialdatasetid-keys-key-op-not-found |
| 1. 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.   The response body SHALL contain an exception report message. The recommended format is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. The message shall contain a value ‘Internal server error’ |   http://www.opengis.net/spec/tjs/2.0/req/core/spatialdatasets-spatialdatasetid-keys-key-op-error |

#### Response schema

schema:

$ref: '#/components/schemas/SpatialDatasetKeyResponseObject'

SpatialDatasetKeyResponseObject:

required:

- key

- links

type: object

properties:

links:

type: array

items:

$ref: '#/components/schemas/Link'

key:

$ref: '#/components/schemas/SpatialDatasetKeyObject'

SpatialDatasetKeyObject:

required:

- key

type: object

properties:

key:

type: string

title:

type: string

### POST operation at path /joindata/{spatialdatasetid}/csv

The HTTP POST operation at path /joindata/{spatialdatasetid}/csv joins attribute data from a csv file with a spatial dataset that is available on the server. The join is executed by using shared key values in the two datasets.

#### Request

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. If the server implements the *data joining operations* operation set, it SHALL support the HTTP POST operation at path /joindata/{spatialdatasetid}/csv   The csv file can either be uploaded to the server with the *csvFile* parameter or provided through URL link with the *csvFileURL* parameter.  The request SHALL contain the header:   * Content-Type: multipart/form-data.   If csv file is provided by upload, it SHALL contain the header   * Content-Disposition: form-data; filename="[csv file’s name]"; name="csvFile"   If sld file is provided by upload, it SHALL contain the header:   * Content-Disposition: form-data; filename="[sld file’s name]" name="sldFile";   **Request’s form data parameters:**   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Description** | **Type and values** | **Required** | | csvFile | The csv file (uploaded file) | File | Optionala | | csvFileURL | The csv file URL | URL | Optionala | | csvFileKeyColumnNumber | The number of the key column in the csv file (counting starts from 1) | Integer | Mandatory | | csvFileAttributeColumnNumberList | The numbers of the attribute columns in the csv file that will be joined with the spatial dataset (counting start from 1) | Integers separated by commas | Mandatory | | csvFileDelimiter | The delimiter character used in the csv file | string | Mandatory | | csvFileHeaderRowNumber | The row number of csv file’s header row in the csv file (counting starts from 1) | integer | Optional, (omit, if header row is not available in the csv file) | | csvFileDataStartRowNumber | The row number where the data values start in the csv file. (counting starts from 1) | integer | Optionalb | | outputFormats | List of outputs that will be included to the response document | String, (comma separated) | Optionalc | | csvFileDuplicateKeyHandlingMethod | Method for handling duplicate key values in the csv file. Value is one of the following: *first*, *last*, count, *add*, *average* | String | Optionald | | sldFile | The sld file for WMS output (uploaded file) | File | Optionale | | sldFileURL | URL, containing sld file that will be applied to WMS output | URL | Optionale | | a  One of the parameters: *csvFile* or *csvFileURL* is mandatory to be used with the operation. The *csvFile* parameter can be used for uploading a csv file to the server. The *csvFileURL* parameter can be used for providing the csv file through URL link. If both parameters are provided in the query, the server SHALL send an exception with message 'DuplicateAttributeFileInput’ | | | | | b Default value for the parameter is 2. If ‘*csvFileHeaderRowNumber’* parameter is missing, default value is 1 | | | | | c Comma separated list of the outputs that will be included to the response document. The supported output formats can be found from the API description document. In case of empty or missing parameter defaults to *geojson* | | | | | d Possible values are: *first*, *last*, count, *add*, *average*. The value *first* uses data values from the first row where the key is encountered, the value *last* takes the values from the last row where the key is encountered, the value *count* tells how many rows there were for the key in the attribute dataset*,* the value *add* adds all data values from all rows together (only for numerical values), the value *average* calculates an average from all rows (only for numerical values) | | | | | e  Styling file can be provided either by uploading it by using the *sldFile* parameter or through URL link by using the *sldFileURL* parameter | | | |   http://www.opengis.net/spec/tjs/2.0/req/core/joindata-spatialdatasetid-csv-op |

#### Response

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. A successful execution of the operation SHALL be reported with a response that has a HTTP status code 201.   The recommended response format is a JSON. The response document contains information on the execution of the data join operation, including links to the created outputs for the joined data. The response schema is presented in Chapter 6.3.5.4.  All TJS implementations SHALL support the GeoJSON output format for the joined data and the servers MAY support any other output formats they choose. Other recommended output formats to be supported are WFS and WMS outputs.  The supported output formats SHALL be listed in the API description document.  On a successful query, the response SHALL include the following links in the property *links.*   * a link to this document (link relation: self) * links to this document in other supported media types (link relation: alternate)   **Description of response elements:**   |  |  | | --- | --- | | **Name** | **Description** | | inputs | Join operation inputs | | joinId | Unique identifier for the join | | joinInformation | Information on the execution of the data join operation | | joinLocation | URL address for accessing the join information | | outputs | Links to the created outputs | | timestamp | Timestamp when the join has been executed |   **Description of inputs property:**   |  |  | | --- | --- | | **Name** | **Description** | | attributeDataset | Name or URL of the joined csv file | | spatialDataset | Links object that contains links to spatialdataset’s different representations. Properties *href, rel* and *type* are mandatory. |   **Description of joinInformation property:**   |  |  | | --- | --- | | **Name** | **Description** | | additionalAttributeKeysList | List of additional keys in the csv file that are not included among the spatial dataset keys | | duplicateAttributeHandlingMethod | The method that was used for handling duplicate keys in csv file | | duplicateAttributeKeys | List of duplicate keys in the csv file | | matchedSpatialDatasetKeys | Spatial dataset keys that were successfully matched with attribute data | | numberOfAdditionalAttributeKeys | The amount of additional attribute key values in the attribute dataset | | numberOfDuplicateAttributeKeys | The amount of attribute keys that had duplicate entries | | numberOfMatchedSpatialDatasetKeys | The amount of spatial dataset keys, to which attribute data was joined successfully | | numberOfUnmatchedSpatialDatasetKeys | The amount of spatial dataset keys, to which attribute data couldn’t be joined | | unmatchedSpatialDatasetKeys | List of spatial dataset keys, to which attribute data couldn’t be joined |   **Description of outputs property:**   |  |  | | --- | --- | | **Name** | **Description** | | format | Name of the output format | | link | Link to the output | | layerName | Layer name (For WMS and WFS outputs) | | styleName | Name of the joined layer style |   http://www.opengis.net/spec/tjs/2.0/req/core/joindata-spatialdatasetid-csv-op-success |

#### Errors

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. If an incorrect request is made to the server, it SHALL be reported as a response with a HTTP status code 400.   The response body shall contain an exception report message. The recommended format is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. Possible values are *InvalidParameterValue, MissingParameterValue*, *DuplicateAttributeFileInput* and *DuplicateStylingFileInput* | | locator | Name of the request parameter that caused the exception |   http://www.opengis.net/spec/tjs/2.0/req/core/joindata-spatialdatasetid-csv-op-bad-request |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. If spatialdataset is not found on the server, it SHALL be reported as a response with a HTTP status code 404.   The response body SHALL contain an exception report message. The recommended format is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. Value ‘Spatial dataset {spatialdatasetid} not found’ | | locator | Name of the request parameter that caused the exception |   http://www.opengis.net/spec/tjs/2.0/req/core/joindata-spatialdatasetid-csv-op-not-found |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. 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.   The response body shall contain an exception report message. The recommended format is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response. | | message | Details of the exception. The message shall contain a value ‘Internal server error’ |   http://www.opengis.net/spec/tjs/2.0/req/core/joindata-spatialdatasetid-csv-op-error |

#### Response schema

schema:

$ref: '#/components/schemas/JoinDataResponseObject'

JoinDataResponseObject:

required:

- join

- links

type: object

properties:

links:

type: array

items:

$ref: '#/components/schemas/Link'

join:

$ref: '#/components/schemas/JoinDataObject'

JoinDataObject:

required:

- inputs

- joinId

- outputs

- timestamp

type: object

properties:

joinId:

type: integer

format: int32

timestamp:

type: string

format: date-time

inputs:

$ref: '#/components/schemas/JoinInputsObject'

outputs:

type: array

items:

$ref: '#/components/schemas/OutputObject'

joinInformation:

$ref: '#/components/schemas/JoinInformationObject'

JoinInputsObject:

required:

- attributeDataset

- spatialDataset

type: object

properties:

attributeDataset:

type: string

spatialDataset:

type: array

items:

$ref: '#/components/schemas/Link'

OutputObject:

required:

- format

- link

type: object

properties:

format:

type: string

layerName:

type: string

link:

type: string

styleName:

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

additionalAttributeKeys:

type: array

items:

type: string

duplicateAttributeKeys:

type: array

items:

type: string

duplicateAttributeKeyHandlingMethod:

type: string

numberOfDuplicateAttributeKeys:

type: integer

format: int32

### GET operation at path /joins

The HTTP GET operation at path /joins returns an array that contains metadata on all joins that are available on the server.

#### Request

|  |
| --- |
| 1. If the server implements the data joining operation set it SHALL support the HTTP GET operation at the path /joins   http://www.opengis.net/spec/tjs/2.0/req/core/joins-op |

#### Response

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. A successful execution of the operation shall be reported as a response with a HTTP status code 200.   On a successful query, the response SHALL include the following links in the property *links.*   * a link to this document (link relation: self) * links to this document in other supported media types (link relation: alternate)   On a successful query, the server SHALL return metadata on all joins that are available on the server*.* The recommended response format for the operation is JSON.  The response schema of a successful is presented in Chapter 6.3.6.4.  **Description of elements in join response element:**   |  |  | | --- | --- | | **Name** | **Description** | | joinId | Unique identifier for the join. The identifier is used in other operations for indicating the join in question | | joinedLayerName | Name of the created layer for the join | | links | Link to this spatial dataset item representations. Properties *href, rel* and *type* are mandatory | | timestamp | Timestamp when the join has been executed |   http://www.opengis.net/spec/tjs/2.0/req/core/joins-op-success |

#### Errors

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. If joins are not found on the server, it SHALL be reported as a response with a HTTP status code 404.   The response body SHALL contain an exception report message. The recommended format is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. Value ‘Joins not found’ |   http://www.opengis.net/spec/tjs/2.0/req/core/joins-op-not-found |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. 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.   The response body SHALL contain an exception report message. The recommended format is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. The message shall contain a value ‘Internal server error’ |   http://www.opengis.net/spec/tjs/2.0/req/core/joins-op-error |

#### Response schema

schema:

$ref: '#/components/schemas/JoinsResponseObject'

JoinsResponseObject:

required:

- joins

- links

type: object

properties:

links:

type: array

items:

$ref: '#/components/schemas/Link'

joins:

type: array

items:

$ref: '#/components/schemas/JoinsObject'

JoinsObject:

required:

- joinId

- joinTimestamp

- joinedLayerName

- links

type: object

properties:

joinId:

type: integer

format: int32

joinedLayerName:

type: string

joinTimestamp:

type: string

format: date-time

links:

type: array

items:

$ref: '#/components/schemas/Link'

### GET operation at path /joins/{joinid}

The HTTP GET operation at path /joins/{joinid} returns metadata on a specific join that is available in the server.

#### Request

|  |
| --- |
| 1. If the server implements the data joining operation set it SHALL support the HTTP GET operation at the path /joins/{joinid}   http://www.opengis.net/spec/tjs/2.0/req/core/joins-joinid-op |

#### Response

|  |
| --- |
| 1. A successful execution of the operation SHALL be reported as a response with a HTTP status code 200.   The response schema of a successful is presented in Chapter 6.3.7.4.  http://www.opengis.net/spec/tjs/2.0/req/core/joins-joinid-op-success |

#### Errors

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. If spatial datasets are not found on the server, it SHALL be reported as a response with a HTTP status code 404.   The response body shall contain an exception report message. The recommended format is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. Value ‘Join {joinid} not found’ |   http://www.opengis.net/spec/tjs/2.0/req/core/joins-joinid-op-not-found |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. 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.   The response body SHALL contain an exception report message. The recommended format is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. The message shall contain a value ‘Internal server error’ |   http://www.opengis.net/spec/tjs/2.0/req/core/joins-joinid-op-error |

#### Response schema

schema:

$ref: '#/components/schemas/JoinResponseObject'

JoinResponseObject:

required:

- join

- links

type: object

properties:

links:

type: array

items:

$ref: '#/components/schemas/Link'

join:

$ref: '#/components/schemas/JoinDataObject'

JoinDataObject:

required:

- inputs

- joinId

- outputs

- timestamp

type: object

properties:

joinId:

type: integer

format: int32

timestamp:

type: string

format: date-time

inputs:

$ref: '#/components/schemas/JoinInputsObject'

outputs:

type: array

items:

$ref: '#/components/schemas/OutputObject'

joinInformation:

$ref: '#/components/schemas/JoinInformationObject'

JoinInputsObject:

required:

- attributeDataset

- spatialDataset

type: object

properties:

attributeDataset:

type: string

spatialDataset:

type: array

items:

$ref: '#/components/schemas/Link'

OutputObject:

required:

- format

- link

type: object

properties:

format:

type: string

layerName:

type: string

link:

type: string

styleName:

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

additionalAttributeKeys:

type: array

items:

type: string

duplicateAttributeKeys:

type: array

items:

type: string

duplicateAttributeKeyHandlingMethod:

type: string

numberOfDuplicateAttributeKeys:

type: integer

format: int32

### POST Operation at path /joins/{joinid}/csv

The HTTP POST operation at path /joins/{joinid}/csv updates fully the specific join with a data from a csv file.

#### Request

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. If the server implements the data joining operation set it MAY support the HTTP POST operation at the path /joins/{joinid}/csv   The csv file can either be uploaded to the server with the *csvFile* parameter or provided through URL link with the *csvFileURL* parameter.  The request SHALL contain the header:   * Content-Type: multipart/form-data.   If csv file is provided by upload, it SHALL contain the header   * Content-Disposition: form-data; filename="[csv file’s name]"; name="csvFile"   If sld file is provided by upload, it SHALL contain the header:   * Content-Disposition: form-data; filename="[sld file’s name]" name="sldFile";   **Request’s form data parameters:**   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Description** | **Type and values** | **Required** | | csvFile | The csv file (uploaded file) | File | Optionala | | csvFileURL | The csv file URL | URL | Optionala | | csvFileKeyColumnNumber | The number of the key column in the csv file (counting starts from 1) | Integer | Mandatory | | csvFileAttributeColumnNumberList | The numbers of the attribute columns in the csv file that will be joined with the spatial dataset (counting start from 1) | Integers separated by commas | Mandatory | | csvFileDelimiter | The delimiter character used in the csv file | string | Mandatory | | csvFileHeaderRowNumber | The row number of csv file’s header row in the csv file (counting starts from 1) | integer | Optional, (omit, if header row is not available in the csv file) | | csvFileDataStartRowNumber | The row number where the data values start in the csv file. (counting starts from 1). | integer | Optionalb | | outputFormats | List of outputs that will be included to the response document | String, (comma separated) | Optionalc | | csvFileDuplicateKeyHandlingMethod | Method for handling duplicate key values in the csv file. Value is one of the following: *first*, *last*, *count*, *add*, *average* | String | Optionald | | sldFile | The sld file for WMS output (uploaded file) | File | Optionale | | sldFileURL | URL, containing sld file that will be applied to WMS output | URL | Optionale | | a  One of the parameters: *csvFile* or *csvFileURL* is mandatory to be used with the operation. The *csvFile* parameter can be used for uploading a csv file to the server. The *csvFileURL* parameter can be used for providing the csv file through URL link | | | | | b Default value for the parameter is 2. If ‘*csvFileHeaderRowNumber’* parameter is missing, default value is 1 | | | | | c Comma separated list of the outputs that will be included to the response document. The supported output formats can be found from the OpenAPI description document. In case of empty or missing parameter defaults to *geojson* | | | | | d Possible values are: *first*, *last*, *add*, *average*. The value *first* uses data values from the first row where the key is encountered, the value *last* takes the values from the last row where the key is encountered, the value *count* tells how many rows there were for the key in the attribute dataset, the value *add* adds all data values from all rows together (only for numerical values), the value *average* calculates an average from all rows (only for numerical values). | | | | | e  Styling file can be provided either by uploading it by using the *sldFile* parameter or through URL link by using the *sldFileURL* parameter | | | |   http://www.opengis.net/spec/tjs/2.0/req/core/joins-joinid-csv-op |

#### Response

|  |
| --- |
| 1. A successful execution of the operation SHALL be reported as a response with a HTTP status code 200.   The recommended response format for the operation is a JSON. The response document contains information on the execution of the data join operation, including links to the created outputs for the joined data.  The response of the successful operation is similar to response in Chapter 6.3.8.4.  All TJS implementations SHALL support the GeoJSON output format for the joined data and the servers MAY support any other output formats they choose. Other recommended output formats to be supported output formats are WFS and WMS outputs.  The supported output formats SHALL be listed in the API description document.  On a successful query, the response SHALL include the following links in the property links.  • a link to this document (link relation: self)  • links to this document in other supported media types (link relation: alternate)  http://www.opengis.net/spec/tjs/2.0/req/core/joins-joinid-csv-op-success |

#### Errors

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. If an incorrect request is made to the server, it SHALL be reported as a response with a HTTP status code 400.   The response body SHALL contain an exception report message. The recommended format for the response is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. Possible values are *InvalidParameterValue* and *MissingParameterValue* | | locator | Name of the request parameter that caused the exception |   http://www.opengis.net/spec/tjs/2.0/req/core/joins-joinid-csv-post-bad-request |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. If join is not found on the server, it SHALL be reported as a response with a HTTP status code 404.   The response body shall contain an exception report message. The recommended format for the response is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. Value ‘Join {joinid}not found’ |     http://www.opengis.net/spec/tjs/2.0/req/core/joins-joinid-csv-not-found |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. 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.   The response body shall contain an exception report message. The recommended format for the response is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. The message shall contain a value ‘Internal server error’ |   http://www.opengis.net/spec/tjs/2.0/req/core/joins-joinid-csv-op-error |

#### Response schema

schema:

$ref: '#/components/schemas/JoinDataResponseObject'

JoinDataResponseObject:

required:

- join

- links

type: object

properties:

links:

type: array

items:

$ref: '#/components/schemas/Link'

join:

$ref: '#/components/schemas/JoinDataObject'

JoinDataObject:

required:

- inputs

- joinId

- outputs

- timestamp

type: object

properties:

joinId:

type: integer

format: int32

timestamp:

type: string

format: date-time

inputs:

$ref: '#/components/schemas/JoinInputsObject'

outputs:

type: array

items:

$ref: '#/components/schemas/OutputObject'

joinInformation:

$ref: '#/components/schemas/JoinInformationObject'

JoinInputsObject:

required:

- attributeDataset

- spatialDataset

type: object

properties:

attributeDataset:

type: string

spatialDataset:

type: array

items:

$ref: '#/components/schemas/Link'

OutputObject:

required:

- format

- link

type: object

properties:

format:

type: string

layerName:

type: string

link:

type: string

styleName:

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

additionalAttributeKeys:

type: array

items:

type: string

duplicateAttributeKeys:

type: array

items:

type: string

duplicateAttributeKeyHandlingMethod:

type: string

numberOfDuplicateAttributeKeys:

type: integer

format: int32

### DELETE Operation at path /joins/{joinid}

The HTTP DELETE operation at path /joins/{joinid}deletes the specific join from the server.

#### Request

|  |
| --- |
| 1. If the server implements the data joining operation set it MAY support the HTTP DELETE operation at the path /joins/{joinid}   http://www.opengis.net/spec/tjs/2.0/req/core/joins-joinid-delete-op |

#### Response

|  |
| --- |
| 1. A successful execution of the operation shall be reported as a response with a HTTP status code 204. The response body SHALL be empty.   http://www.opengis.net/spec/tjs/2.0/req/core/joins-joinid-delete-op-success |

#### Errors

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. If join is not found on the server, it shall be reported as a response with a HTTP status code 404.   The response body shall contain an exception report message. The recommended format for the response is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. Value ‘Join {joinid}not found’ |   http://www.opengis.net/spec/tjs/2.0/req/core/joins-joinid-delete-op-not-found |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. 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.   The response body shall contain an exception report message. The recommended format for the response is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. The message shall contain a value ‘Internal server error’ |   http://www.opengis.net/spec/tjs/2.0/req/core/joins-joinid-delete-op-error |

#### Response structure

The response of the operation is empty.

## Operation Set *File Joining Operations*

The operation set *file joining operations* contains operation for joining attribute data from an inputted csv file with an inputted spatial data file. The response of the operation is the spatial data file that contains also the joined attribute data.

### POST operation at path /joinfiles/geojson/csv

The HTTP POST operation at path /joinfiles/geojson/csv joins a data from a csv file with a GeoJSON file.

#### Request

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. If the server supports the *file joining operation* operation set it SHALL support the HTTP POST operation at the path /joinfiles/geojson/csv   The geojson file can either be uploaded to the server with the geojson*File* form data parameter or provided through URL link with the geojson*FileURL* form data parameter.  The csv file can either be uploaded to the server with the *csvFile* form data parameter or provided through URL link with the *csvFileURL* form data parameter.  The request SHALL contain the header:   * Content-Type: multipart/form-data   If geojson file is provided by upload, it SHALL contain the header:   * Content-Disposition: form-data; filename="[sld file’s name]"; name="geojsonFile"   If csv file is provided by upload, it SHALL contain the header:   * Content-Disposition: form-data; filename="[csv file’s name]"; name="csvFile"   **Request’s form data parameters:**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Name** | **Description** | **Type and values** | | **Required** | | | geojsonFile | The GeoJSON file (uploaded file) | File | | Optionala | | | geojsonFileURL | A URL link to the GeoJSON file | URL type, not empty | | Optionala | | | geojsonFileKeyFieldPath | The path to the key field name in the GeoJSON file that contains the key values. Example: “features.properties.kunta” | String | | Mandatory | | | csvFile | The csv file (uploaded file) | File | | Optionalb | | | csvFileURL | A URL link to the csv file | URL type, not empty. | | Optionalb | | | csvFileKeyColumnNumber | The column number in the csv that contains key values. (Counting starts from 1) | | Integer | Mandatory | | | csvFileAttributeColumnNumberList | The numbers of the columns in the csv file that will be joined with a GeoJSON file. When multiple columns are joined the values shall be comma-separated. (Counting starts from 1) | | integer (multiple values are comma-separated) | | Mandatory | | csvFileDelimiter | The delimiter character used in the csv file | | String | | Mandatory | | csvFileHeaderRowNumber | The row number of the title line in the csv file. (Row counting starts from 1) | | Integer | | Optional, (omit, if header row is not available in the csv file) | | csvFileDataStartRowNumber | The row number of the data start line in the csv file. (Row counting starts from 1) | | Integer | | Optionalc | | csvFileDuplicateKeyHandlingMethod | Method for handling duplicate key values in the csv file. Value is one of the following: *first*, *last*, *count*, *add*, *average* | | String | | Optionald | | a  One of the parameters: geojsonFile or geojsonFileURL is mandatory to be used with the operation. The geojsonFile parameter can be used for uploading a geojson file to the server. The geojsonFileURL parameter can be used for providing the geojson file through URL link | | | | | | | b  One of the parameters: csvFile or csvFileURL is mandatory to be used with the operation. The csvFile parameter can be used for uploading a csv file to the server. The csvFileURL parameter can be used for providing the csv file through URL link | | | | | | | c Default value for the parameter is 2. If ‘*csvFileHeaderRowNumber’* parameter is missing, default value is 1 | | | | | |   d Possible values are: *first*, *last*, *add*, *average*. The value *first* uses data values from the first row where the key is encountered, the value *last* takes the values from the last row where the key is encountered, the value *count* tells how many rows there were for the key in the attribute dataset, the value *add* adds all data values from all rows together (only for numerical values), the value *average* calculates an average from all rows (only for numerical values)  http://www.opengis.net/spec/tjs/2.0/req/core/joinfiles-geojson-csv-op |

#### Response

|  |
| --- |
| 1. A successful execution of the operation SHALL be reported as a response with a HTTP status code 200.   The server SHALL return the GeoJSON file that includes the joined fields from the csv file.  http://www.opengis.net/spec/tjs/2.0/req/core/joinfiles-geojson-csv-op-success |

#### Errors

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. If the request is not valid, the server SHALL return a response with a HTTP status code 400.   The response body SHALL contain an exception report message. The recommended format for the response is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. Possible values are *InvalidParameterValue* and *MissingParameterValue* | | locator | Name of the request parameter that caused the exception |   http://www.opengis.net/spec/tjs/2.0/req/core/joinfiles-geojson-csv-op-bad-request |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. 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.   The response body shall contain an exception report message. The recommended format for the response is JSON.  **Exception report message elements:**   |  |  | | --- | --- | | **Name** | **Description** | | status | The HTTP status code of the response | | message | Details of the exception. The message shall contain a value ‘Internal server error’ |   http://www.opengis.net/spec/tjs/2.0/req/core/joinfiles-geojson-csv-op-error |

#### Response structure

The response of the /joinfiles/geojson/csv operation SHALL be the GeoJSON file that contains also the joined attributes from the csv file.

# Media Types for any data encoding(s)

## Operation sets: *discovery operations and data joining operations*

|  |
| --- |
| **Recommendation:** The TJS 2.0 standard recommends the use of JSON in the responses of operation sets: discovery operations and file joining operations. |

## Outputs for the joined data in operation set: *data joining operations*

|  |
| --- |
| 1. For the joined data output formats in the operation set *data joining operations*, all services SHALL support the GeoJSON data encoding format.   The services are free to support any other output formats for the joined data.  http://www.opengis.net/spec/tjs/2.0/req/core/data-joining-operations-output |

## Operation set: *file joining operations*

|  |
| --- |
| 1. For the *file joining operations* operation set, all services SHALL support the GeoJSON data encoding format.   http://www.opengis.net/spec/tjs/2.0/req/core/file-joining-operations-output |

Annex A: Conformance Class Abstract Test Suite (Normative)

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. >

1. [www.opengeospatial.org/cite](http://www.opengeospatial.org/cite) [↑](#footnote-ref-1)