

FUSION REGISTRY WEB SERVICES API

FUSION REGISTRY
VERSION 10

Web Services API

CONTENTS

1	Overview	4
2	Web Service Definitions	6
2.1	Web Application Definition Language (WADL)	6
2.2	Swagger	6
3	SDMX API	7
3.1	Submit Content (POST)	7
3.2	Query Structures via REST API (GET)	8
3.3	Delete Structures via REST API (DELETE).....	14
3.4	Query Data via REST API.....	17
3.5	Query Data Series via POST API.....	23
3.6	Query Schemas via REST API (GET)	25
3.7	Query Data Registrations via REST API.....	27
3.8	Query Reference Metadata via REST API	29
4	Fusion Registry Web Services	31
4.1	User Details	31
4.2	Transaction History	31
4.3	Data Validation.....	33
4.4	Validators	37
4.5	Data Authoring (Excel Report Template)	37
4.6	Data Transformation	40
4.7	Data Submission.....	43
4.8	Check Import Progress	45
4.9	Historical Import Status	46
4.10	Deleting Data.....	47
4.11	Rolling back Data.....	50
4.12	Data Sources	51
4.13	Structure Search Engine.....	55
4.14	Data Search Engine (Beta).....	56
4.15	Reverse Engineer Structure Mapping	57
5	Annex 1: HTTP Headers.....	59
5.1	Compression	59
5.2	Authentication	59
5.3	Language	59
5.4	Caching.....	59
6	Annex 2: Security	60
6.1	X509 Certificate Authentication.....	60

6.2	Basic Authentication	60
6.3	Reverse Proxy.....	60
7	Annex 3 – Metrics and Data Availability	61
7.1	Include Metrics.....	61
7.2	Detail = structureOnly	63

Version History

Version #	Implemented By	Revision Date	Reason
20190318	Matt Nelson	18 March 2019	Data web service new parameter includeAnnotations
20190326	Matt Nelson	26 March 2019	Added Fail-On-Error documentation to the data transformation service
20190404	Phil Lazarou	4 April 2019	Added DatasetAction to Data Submission HTTP Header
20190523	Phil Lazarou	23 May 2019	Data Authoring (Excel Report Template) – is now no longer secure by default
20190529	Matt Nelson	29 May 2019	Data Transformation Service, added support for: zip output, include unmapped data, include metrics
20190618	Matt Nelson	6 June 2019	Support Plus Operators in Structure Queries to select multiple agencies, ids, and versions
20190729	Matt Nelson	29 July 2019	Support Inc-Metrics, Inc-Valid, Inc-Invalid, and Zip for Data Validation
20190828	Phil Lazarou	28 August 2019	Sender-Id removed from Data Validation
20190903	Phil Lazarou	3 September 2019	Data Transformation Service, added support for: Receiver-Id, Dataset-Action, Dataflow-Id
20191014	Phil Lazarou	14 October 2019	Updated Data Validation section
20191129	Matt Nelson	11 November 2019	Availability query
20200116	Phil Lazarou	16 January 2020	Added Item Validity parameters to REST Structure Query
20200130	Phil Lazarou	30 January 2020	Added sdmx-csv parameters for data request



1 Overview

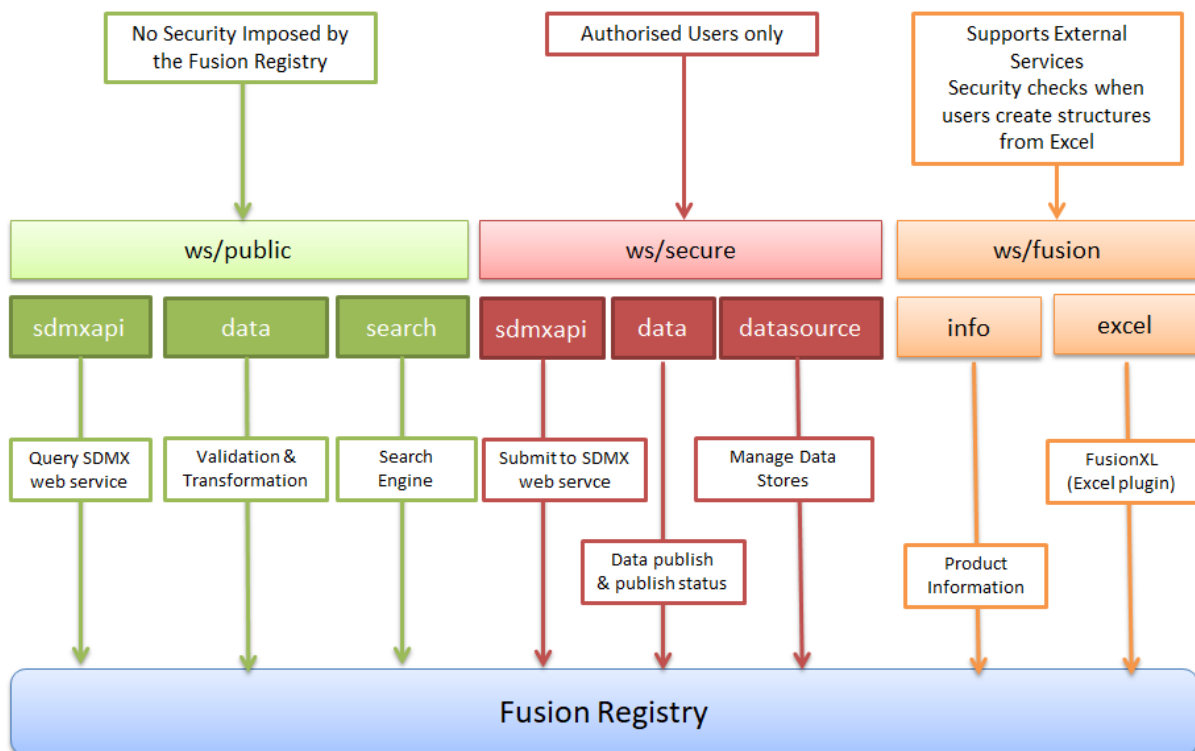
The Fusion Registry hosts a number of web services, this document does not document all the web service APIs exposed by the Fusion Registry, as most web services are made available in order to support the Web User Interface of the product. This document discusses web services which may be useful for external tools and applications, these include:

1. SDMX web services for queries and submissions
2. Data Validation and Transformation web services
3. Search
4. Data Publishing (upload data to Registry managed database)
5. Manage Data Stores (retrieve information about data stores, delete data)
6. Web services used to support the dissemination services (FusionJS and FusionXL)

The web services of the Fusion Registry are split into 4 entry points these are:

1. **ws/public** – these web services have no security imposed on them.
2. **ws/secure** – these web services are secured to only allow users with the correct access levels to communicate with them.
3. **ws/fusion** – these web services support the external tools, FusionJS JavaScript library for data dissemination, and FusionXL Excel plugin. Most services are public however the Fusion Registry performs security checks when FusionXL is used to save structures. *Whilst it is important to know the entry point and purpose of these web services, they are not discussed in this document.*
4. **ws/registry** – these web services are used to support the Fusion Registry User Interface. *Whilst it is important to know the entry point and purpose of these web services, they are not discussed in this document.*

A high level diagram of the Fusion Registry web services (pertinent to this document) are shown in the image below.



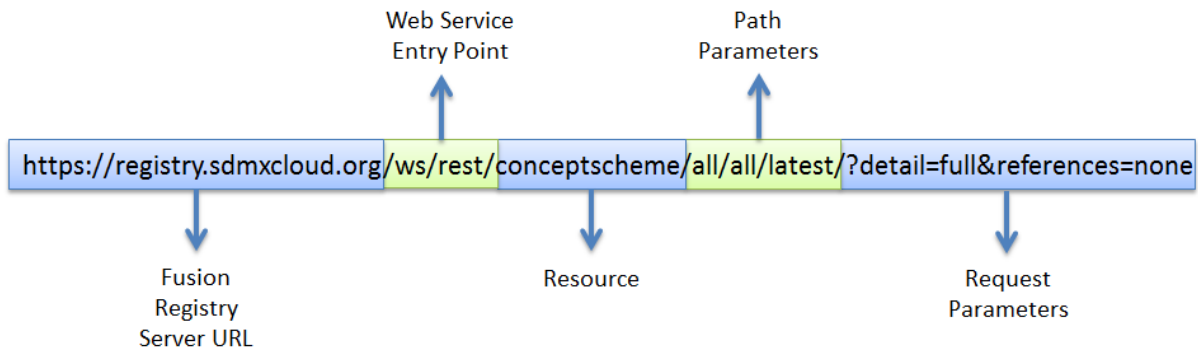
This document groups the web services into the following categories:

1. **SDMX Web Services** – these include web services to retrieve and submit structures, data, and reference metadata. These web services comply with the SDMX Specification, supporting RESTful GET, and HTTP POST. The SDMX web services API supports multiple response formats for queries. These web services are accessed via *ws/public/sdmxapi* and *ws/secure/sdmxapi*. ([SECTION 3 - SDMX API](#))
2. **Fusion Registry Web Services** - these include additional web services to support tasks which are beyond the remit of the SDMX specification. This includes data validation, transformation, searching, information retrieval, and data management. The response format for these services is exclusively JSON. ([SECTION 4 - FUSION REGISTRY WEB SERVICES](#))

Each web service documented by defining the following (where applicable):

1. **Web service entry point** – the URL defining the entry point to the web service
2. **Resource** – the resource that is being requested
3. **Path Parameters** –additional URL path to define the required result
4. **Request Parameters** – optional parameters used to further define the result (such as level of detail, format, further restrictions)
5. **HTTP Header Parameters** – Not part of the URL, but passed in on the http request. Used to pass further information to the server such as response language or response format.

The following shows an example of an HTTP REST GET



The Annexes to this document describe how to make use of HTTP request and response headers to support caching, compression, locale, and authentication. The Annex also discusses how to make public certain parts of the API, whilst keeping other entry points private.

2 Web Service Definitions

Each web service is defined in both a WADL and Swagger 2.0. Both are machine-readable definition of a web service, in terms of service resources, parameters supported request and responses.

A WADL is in XML format, and Swagger in JSON format.

2.1 Web Application Definition Language (WADL)

The WADL is available in the following location:

http(s)://[server]/wadl/registry/service.wadl

If a reverse proxy is being used to map alternative URLs to Fusion Registry web services, then the public WADL should be used, as this will contain the mapped web service entry points.

http(s)://[server]/wadl/public/service.wadl

2.2 Swagger

The Swagger is available in the following location:

http(s)://[server]/swagger/registry/service.json

If a reverse proxy is being used to map alternative URLs to Fusion Registry web services, then the public SWAGGER should be used, as this will contain the mapped web service entry points.

http(s)://[server]/swagger/public/service.json

3 SDMX API

3.1 Submit Content (POST)

Web Service Entry Point	/ws/secure/sdmxapi/rest
Access	Secure: Admin + Agency, Data Provider Users
HTTP Method	HTTP POST
Accepts	<ul style="list-style-type: none"> SDMX Structure Document (all SDMX versions) Registry Interface Document (all SDMX versions) <ul style="list-style-type: none"> SubmitStructureRequest SubmitRegistrationRequest QueryRegistrationRequest Generic Metadata (SDMX v2.1) Excel Document (conforming to FusionXL format)
Compression	Zippped content can be submitted
Content Type	<ol style="list-style-type: none"> multipart/form-data (if attaching file) – the attached file must be in field name of uploadFile application/text or application/xml (if submitting data in the body of the POST)
Response Format	Multiple formats supported. Can be specified in the <i>Accept</i> HTTP Header, or the <i>format</i> parameter of the URL request.
Error Response	SDMX-ML v2.1 Error Response Document

3.1.1 HTTP Headers

The HTTP headers can be used to specify response format.

HTTP Header	Purpose	Allowed Values
Accept	To define the response format.	application/vnd.sdmx.structure+xml;version=1.0 application/vnd.sdmx.structure+xml;version=2.0 application/vnd.sdmx.structure+xml;version=2.1

3.1.2 Supported Submissions

3.1.2.1 Structural Metadata

Structural Metadata (Codelists, Concept Schemes, Data Structures etc) can be submitted to the Fusion Registry in SDMX-ML and SDMX-EDI formats, as well as Excel, and for certain items such as codelists CSV is supported.

SDMX-ML documents may conform to the Structure message, or a RegistryInterfaceDocument with a SubmitStructure message inside. In addition to the actions of **Append**, **Replace**, and **Delete**, the Fusion Registry supports **FullReplace** which can be used to replace all the structures stored in the Fusion Registry with those sent in the submission.

3.1.2.2 Data

Datasets can **NOT** be sent to this web service. Datasets must be submitted to the separate data web service documented in section 4.7

3.1.2.3 Reference Metadata

Reference Metadata can be submitted to this web service. The supported format is SDMX-ML, GenericMetadata document.

3.2 Query Structures via REST API (GET)

The Structure REST API conforms to the SDMX Web Service Guidelines which can be found at <http://sdmx.org>.

In addition to the SDMX specification, the Fusion Registry supports some additional Accept header values and query parameters.

Both the SDMX and extended query parameters are included in this document.

3.2.1 Overview

Web Service Entry Point	/ws/public/sdmxapi/rest
Access	Public
HTTP Method	HTTP GET
Response Format	Multiple formats supported. Can be specified in the <i>Accept</i> HTTP Header, or the <i>format</i> parameter of the URL request.
Error Response	SDMX-ML v2.1 Error Response Document

3.2.2 HTTP Headers

The HTTP headers can be used to specify response format. This can also be defined in the request parameter.

HTTP Header	Purpose	Allowed Values
Accept	To define the response format	<p>SDMX Formats application/vnd.sdmx.structure;version=edi application/vnd.sdmx.structure+xml;version=1.0 application/vnd.sdmx.structure+xml;version=2.0 application/vnd.sdmx.structure+xml;version=2.1</p> <p>JSON Format application/vnd.sdmx.json</p> <p>RDF Formats application/vnd.rdf+xml application/vnd.rdf+json application/vnd.rdf+turtle</p> <p>Excel Format application/vnd.xlsx</p>
Accept-Language	<p>This optional header can be used to set the locale to return any multilingual text in (names and descriptions). If the text does not exist in the specified locale, then the default rules will be applied to find the next best appropriate locale.</p> <p>The corresponding locale parameter can be used to override this HTTP Header</p>	Accept-Language : en (English) Accept-Language : fr (French) Accept-Language : * (all languages – no filter) Accept-Language : all (all languages – no filter)

3.2.3 Resource

The resource is used to determine which structure type is being queried. The resources identified in the following table are supported.

Resource	Purpose
datastructure	Returns all data structure definitions that match the subsequent path parameters
metadatastructure	Returns all metadata structure definitions that match the subsequent path parameters
categoryscheme	Returns all category schemes that match the subsequent path parameters
conceptscheme	Returns all concept schemes that match the subsequent path parameters
codelist	Returns all codelists that match the subsequent path parameters
hierarchicalcodelist	Returns all hierarchical codelists that match the subsequent path parameters
organisationscheme	Returns all organisation schemes that match the subsequent path parameters
agencyscheme	Returns all agency schemes that match the subsequent path parameters
dataproviderscheme	Returns all data provider schemes that match the subsequent path parameters
dataconsumerscheme	Returns all data consumer schemes that match the subsequent path parameters
organisationunitscheme	Returns all organisation unit schemes that match the subsequent path parameters
dataflow	Returns all dataflows that match the subsequent path parameters
metadataflow	Returns all metadata flows that match the subsequent path parameters
reportingtaxonomy	Returns all reporting taxonomies that match the subsequent path parameters
provisionagreement	Returns all provision agreements that match the subsequent path parameters
structureset	Returns all structure sets that match the subsequent path parameters
process	Returns all processes that match the subsequent path parameters
categorisation	Returns all categorisations that match the subsequent path parameters
contentconstraint	Returns all content constraints that match the subsequent path parameters
attachmentconstraint	Returns all attachment constraints that match the subsequent path parameters
structure	Returns All SDMX structures that match the subsequent path parameters

3.2.4 Path Parameters

The path parameters are used to further define the attributes of the request structure(s). All the path parameters are optional. If the path parameters have a default value, it will be used in the absence of the parameter.

Parameter	Purpose	Allowed Values
agencyID	The agency which owns the structure(s) The plus operator can be used to select multiple agencies	all – default. any agency Or any string compliant with the SDMX common:NCNameIDType <i>Examples</i> BIS BIS+ECB
structureID	The id of the structure(s) to be returned The plus operator can be used to select multiple structures	all – default. all structure ids Or any string compliant with the SDMX common:NCNameIDType <i>Examples</i> CL_COUNTRY CL_COUNTRY+ CL_REF_AREA
version	The version of the structure(s) to be returned The plus operator can be used to select multiple versions	latest – default. latest version all – all versions Or a specific version number <i>Examples</i> 1.0 1.0+1.1+1.2.1
itemID	If the resource is to an item scheme (Codelist, Concept Scheme, Category Scheme), the item inside the scheme	String <i>Examples</i> M

	can be identified by this parameter	M+F
	The plus operator can be used to select multiple items	

3.2.5 Request Parameters

The request parameters are all optional and can be used to define the response detail, format, and any additional structures which reference, or are referenced by those identified in the query path.

Parameter	Purpose	Allowed Values
detail	To define which structures (if any) are output as stubs.	<p>full – default. Output full response.</p> <p>allstubs – Output all the structures as stubs.</p> <p>referencestubs – Output the full query result, and any referenced structures are returned as stubs</p> <p>referencepartial – Outputs the full query result and any referenced Codelists, Concept Schemes, Agency Schemes are returned as partial lists based on the Codes, Concepts, and Agencies used by the referencing Provision Agreements, Dataflows, Data Structures, Hierarchical Codelists. Partial Codelists are derived from Content Constraints used to define allowable content for data reporting.</p> <p><i>Example:</i> detail=allstubs detail=referencepartial</p>
references	<p>To define if additional structures are returned from the query.</p> <p>The structures can either be ones which reference, or are referenced by the structures in the query result.</p> <p>If the query result is for a specific item in an item scheme, then this parameter will identify the references for that item.</p>	<p>none – default. Do not output any additional structures</p> <p>parents –output structures the reference the structures matching the query</p> <p>parentsandsiblings – same as parents, but also include all the additional structures referenced by the parents</p> <p>children – the structures referenced by the structures in the query result</p> <p>descendants – children and their children (up to any level)</p> <p>In addition, a concrete type of resource may be used, for example: datastructure</p> <p><i>Example:</i> references=datastructure</p>
partial	<p>If set to true creates partial Codelists in the response based on the Fusion Registry Content Constraints defining allowable content.</p> <p>The pre-requisite is that the query must be for a single constrainable structure (Provision Agreements, Dataflow, or Data Structure)</p>	<p>true/false</p> <p>Note: This is deprecated as of v9.2.23, use detail=referencepartial instead</p>

	and include references.	
format	Can be used to define the response format (as an alternative to the HTTP Accept Header).	<p>sdmx (latest version) sdmx-1.0 sdmx-2.0 sdmx-2.1 sdmx-edi sdmx-edi-lenient sdmx-json rdf-turtle rdf-json rdf-xml xlsx</p> <p><i>Example:</i> format=sdmx-edi</p>
includeMetadata	<p>If set to true, then the response structures will contain additional Annotations if there are reference metadata attached.</p> <p>The annotation will have the AnnotationType of 'METADATA' and the AnnotationURI will provide a URI to the MetadataSet.</p>	<p>true/false</p> <p><i>Example:</i> includeMetadata=true</p>
locale	<p>This optional parameter can be used to set the locale to return any multilingual text in (names and descriptions). If the text does not exist in the specified locale, then the default rules will be applied to find the next best appropriate locale.</p> <p>This takes priority over the Accept-Language HTTP header</p>	<p>Any locale</p> <p><i>Example:</i> locale=fr</p>
includeMetrics	Enriches structures with information about data that is available. See Annex 3 – Metrics and Data Availability	<p>true/false/only</p> <p>IncludeMetrics=only is used to indicate that only maintainable structures with metrics should be returned, and maintainables which have no metrics are removed from the query response.</p> <p><i>Example:</i> includeMetrics=true includeMetrics=only</p>
saveAs	<p>If provided the HTTP Header 'Content-Disposition' will be set to attachment with the filename being set to the value provided.</p> <p>This will result in the response</p>	<p>String</p> <p><i>Example:</i> saveAs=myDownload</p>



	<p>being saved to a file.</p> <p>The file extension is not required as the Fusion Registry will determine the extension based on the response format.</p>	
prettyPrint	If the you are requesting XML, and you would like the response XML to be formatted, then you can pass true	String prettyPrint=true
validFrom	For structures with a defined validFrom or validTo value, returns only those structures which have a validFrom value before the specified date	Single time period. Conforms to the ISO-8601 standard but SDMX date time formats may also be used <i>Example:</i> validFrom=1960-12-31
validTo	For structures with a defined validFrom or validTo value, returns only those structures which have a validTo value after the specified date	Single time period. Conforms to the ISO-8601 standard but SDMX date time formats may also be used <i>Example:</i> validTo=1960-12-31
validOn	Returns structures where the items listed are applicable for the specified date. This parameter is only applicable for those structures which support Item Validity.	Single time period. Conforms to the ISO-8601 standard but SDMX date time formats may also be used <i>Example:</i> validOn=1960-12-31
labels	<p>For sdmx-csv format only</p> <p>For each element of the CSV, will return either the ID only or the ID and Name separated by a colon</p>	<p>Id – default both</p> <p><i>Example:</i> labels=both</p>
timeFormat	<p>For sdmx-csv format only</p> <p>Normalized TIME_PERIOD values are converted to the most granular ISO 8601 representation taking into account the highest frequency of the data in the message</p>	<p>original – default normalized</p> <p><i>Example:</i> timeFormat=normalized</p>

3.2.6 Examples

3.2.6.1 All concept schemes in SDMX v2.0 format formatted

<https://registry.sdmxcloud.org/ws/public/sdmxapi/conceptscheme/all/all/latest/?format=sdmx+2.0&detail=full&references=none&prettyPrint=true>

3.2.6.2 All structures saved to a file

<https://registry.sdmxcloud.org/ws/public/sdmxapi/rest/structure/all/all/latest?saveAs=fullexport>

3.2.6.3 *Any concept with Id OBS_CONF and all the data structures that reference it*
[https://registry.sdmxcloud.org/
ws/public/sdmxapi/rest/conceptscheme/all/all/all/OBS_CONF?references=datastructure](https://registry.sdmxcloud.org/ws/public/sdmxapi/rest/conceptscheme/all/all/all/OBS_CONF?references=datastructure)

3.3 Delete Structures via REST API (DELETE)

The Fusion Registry supports the ability to change delete structures using a RESTful URL, when the method is set to DELETE as oppose to GET.

The delete API is identical to the REST GET API for identifying which structures are to be included for deletion. The references query parameter can be used to include further structures.

3.3.1 Overview

Web Service Entry Point	/ws/secure/sdmxapi/rest
Access	Secure: Admin + Agency Users
HTTP Method	HTTP DELETE
Response Format	SDMX 2.1 RegistryInterface Document
Error Response	SDMX-ML v2.1 Error Response Document if request did not match any structures or if there was a series error. SubmitStructureResponse with StatusMessage Failure if a matched structure could not be deleted due to Registry validation failure

3.3.2 Resource

The resource is used to determine which structure type is being deleted. The resources identified in the following table are supported.

Resource	Purpose
datastructure	Returns all data structure definitions that match the subsequent path parameters
metadatastructure	Returns all metadata structure definitions that match the subsequent path parameters
categoryscheme	Returns all category schemes that match the subsequent path parameters
conceptscheme	Returns all concept schemes that match the subsequent path parameters
codelist	Returns all codelists that match the subsequent path parameters
hierarchicalcodelist	Returns all hierarchical codelists that match the subsequent path parameters
organisationscheme	Returns all organisation schemes that match the subsequent path parameters
agencyscheme	Returns all agency schemes that match the subsequent path parameters
dataproviderscheme	Returns all data provider schemes that match the subsequent path parameters
dataconsumerscheme	Returns all data consumer schemes that match the subsequent path parameters
organisationunitscheme	Returns all organisation unit schemes that match the subsequent path parameters
dataflow	Returns all dataflows that match the subsequent path parameters
metadataflow	Returns all metadata flows that match the subsequent path parameters
reportingtaxonomy	Returns all reporting taxonomies that match the subsequent path parameters
provisionagreement	Returns all provision agreements that match the subsequent path parameters
structureset	Returns all structure sets that match the subsequent path parameters
process	Returns all processes that match the subsequent path parameters
categorisation	Returns all categorisations that match the subsequent path parameters
contentconstraint	Returns all content constraints that match the subsequent path parameters
attachmentconstraint	Returns all attachment constraints that match the subsequent path parameters
structure	Returns All SDMX structures that match the subsequent path parameters

3.3.3 Path Parameters

The path parameters are used to further define the attributes of the structure(s). All the path parameters are optional. If the path parameters have a default value, it will be used in the absence of the parameter.

Parameter	Purpose	Allowed Values
agencyID	The agency which owns the structure(s)	all – default. any agency Or any string compliant with the SDMX common:NCNameIDType
structureID	The id of the structure(s) to be returned	all – default. all structure ids Or any string compliant with the SDMX common:NCNameIDType
version	The version of the structure(s) to be returned	latest – default. latest version all – all versions Or a specific version number
itemID	If the resource is to an item scheme (Codelist, Concept Scheme, Category Scheme), the item inside the scheme can be identified by this parameter	String

3.3.4 Request Parameters

The request parameters are all optional and can be used to further include structures for deletion

Parameter	Purpose	Allowed Values
references	<p>To define if additional structures are returned from the query.</p> <p>The structures can either be ones which reference, or are referenced by the structures in the query result.</p> <p>If the query result is for a specific item in an item scheme, then this parameter will identify the references for that item.</p>	<p>none – default. Do not output any additional structures</p> <p>parents –output structures the reference the structures matching the query</p> <p>parentsandsiblings – same as parents, but also include all the additional structures referenced by the parents</p> <p>children – the structures referenced by the structures in the query result</p> <p>descendants – children and their children (up to any level)</p> <p>In addition, a concrete type of resource may be used, for example: datastructure</p> <p><i>Example:</i> references=datastructure</p>

3.3.5 Examples

3.3.5.1 Delete all concept schemes

<https://registry.sdmxcloud.org/ws/secure/sdmxapi/rest/conceptscheme/all/all/all>

3.3.5.2 Delete any concept with Id OBS_CONF and all the data structures that reference it

https://registry.sdmxcloud.org/ws/secure/sdmxapi/rest/conceptscheme/all/all/all/OBS_CONF?references=datastructure

3.4 Query Data via REST API

The Data REST API conforms to the SDMX Web Service Guidelines which can be found at <http://sdmx.org>.

In addition to the SDMX specification, the Fusion Registry supports some additional Accept header values and query parameters.

Both the SDMX and extended query parameters are included in this document.

3.4.1 Overview

URL Entry Point	/ws/public/sdmxapi/rest/data
Access	Public
HTTP Method	HTTP GET
Response Format	Multiple supported. Can be defined in the Accept HTTP Header, or the format parameter of the URL request.
Error Response	SDMX-ML v2.1 Error Response Document

3.4.2 HTTP Headers

HTTP Header	Purpose	Allowed Values
Accept	To define the response format	<p>SDMX Formats</p> <pre>application/vnd.sdmx.genericdata+xml;version=1.0 application/vnd.sdmx.genericdata+xml;version=2.0 application/vnd.sdmx.genericdata+xml;version=2.1 application/vnd.sdmx.structurespecificdata+xml;version=1.0 application/vnd.sdmx.structurespecificdata+xml;version=2.0 application/vnd.sdmx.structurespecificdata+xml;version=2.1 application/vnd.sdmx.edi application/vnd.sdmx.json application/vnd.sdmx.data+json;version=1.0.0-wd</pre> <p>RDF Formats</p> <pre>application/vnd.rdf+xml application/vnd.rdf+json application/vnd.rdf+turtle</pre> <p>CSV/Excel Formats</p> <pre>application/vnd.xlsx application/vnd.csv</pre>
Accept-Language	<p>This optional header can be used to set the locale to return any multilingual text in (names and descriptions). If the text does not exist in the specified locale, then the default rules will be applied to find the next best appropriate locale.</p> <p>The corresponding locale parameter can be used to override this HTTP Header</p>	<pre>Accept-Language : en (English) Accept-Language : fr (French) Accept-Language : * (all languages – no filter) Accept-Language : all (all languages – no filter)</pre>

3.4.3 Resource

The resource is used to determine which structure type is being queried. The resources identified in the following table are supported.

Resource	Purpose
data	Returns a dataset containing data that matches the subsequent path parameters. If the query parameter includeHistory=true , then multiple datasets may be returned

3.4.4 Path Parameters

The path parameters are used to further define the attributes of the request structure(s). All the path parameters are optional. If the path parameters have a default value, it will be used in the absence of the parameter.

Parameter	Purpose	Allowed Values
dataflowRef	<p>(required)</p> <p>A string identifying the dataflow.</p> <p>The returned dataset will belong to the dataflow identified by this parameter.</p> <p>This path parameter is mandatory.</p>	<p>The syntax is agency id, artefact id, version, separated by a ",". For example: AGENCY_ID, FLOW_ID, VERSION</p> <p>In case the string only contains one out of these 3 elements, it is considered to be the flow id, i.e. all, FLOW_ID, latest</p> <p>In case the string only contains two out of these 3 elements, they are considered to be the agency id and the flow id, i.e. AGENCY_ID, FLOW_ID, latest</p>
seriesKey	<p>Used to identify the series to return in the result dataset.</p> <p>The key parameter can contain zero to many code id selections for each Dimension of the Dataflow's Data Structure Definition.</p> <p>The absence of this path parameter is interpreted as all series.</p>	<p>The syntax is code selections for the first dimension followed by the '.' separator followed by code selections for the second dimension, and so on.</p> <p>Multiple code selections are concatenated with the '+' separator.</p> <p>The dimension may be wild-carded by omitting code selections for the dimension.</p> <p>The all keyword can be used to indicate all data for each dimension.</p> <p>For example, if the following series key identifies the bilateral exchange rates for the daily US dollar exchange rate against the euro, D.USD.EUR.SP00.A, then the following series key can be used to retrieve the data for all currencies against the euro: D..EUR.SP00.A.</p> <p>The OR operator is supported using the + character. For example, the following series key can be used to retrieve the exchange rates against the euro for both D.USD+JPY.EUR.SP00.A.</p>

providerRef	(optional) Identifies the data provider that provided the data. A single dataflow may have data provided by multiple data providers. The absence of this path parameter is interpreted as all data providers.	The syntax is agency id, provider id, separated by a ",". For example: AGENCY_ID, PROVIDER_ID In case the string only contains one out of these 2 elements, it is considered to be the provider id, i.e. all, PROVIDER_ID
--------------------	---	--

3.4.5 Request Parameters

Parameter	Purpose	Allowed Values																
startPeriod	Only return observations if the observation time is after the time specified.	Single time period (not a time range). Conforms to the ISO-8601 standard.																
endPeriod	Only return observations if the observation time is before the time specified.	Supported time periods include																
updatedAfter	Only return data if it was updated after the time specified.	<table border="1"> <thead> <tr> <th>Period</th> <th>Example</th> </tr> </thead> <tbody> <tr> <td>Annual</td> <td>2009</td> </tr> <tr> <td>Semester</td> <td>2009-S1</td> </tr> <tr> <td>Trimester</td> <td>2009-T1</td> </tr> <tr> <td>Quarterly</td> <td>2009-Q1</td> </tr> <tr> <td>Monthly</td> <td>2009-01</td> </tr> <tr> <td>Daily</td> <td>2009-01-31</td> </tr> <tr> <td>Date Time</td> <td>2009--01T20:22:00</td> </tr> </tbody> </table>	Period	Example	Annual	2009	Semester	2009-S1	Trimester	2009-T1	Quarterly	2009-Q1	Monthly	2009-01	Daily	2009-01-31	Date Time	2009--01T20:22:00
Period	Example																	
Annual	2009																	
Semester	2009-S1																	
Trimester	2009-T1																	
Quarterly	2009-Q1																	
Monthly	2009-01																	
Daily	2009-01-31																	
Date Time	2009--01T20:22:00																	
firstNObservations	The first 'n' observations to return for each matched series.	Positive Integer																
lastNObservations	The last 'n' observations to return for each matched series.	Positive Integer																
dimensionAtObservation	To define if the observations are grouped or not. This is only supported if the response format is SDMX JSON.	TIME_PERIOD (default) – observations iterate over time. AllDimensions – no grouping.																
detail	Specifies the amount of information to be returned.	full (default) – return the complete dataset dataonly – exclude attributes and groups serieskeyonly – only include series, no observations. nodata – returns the groups and series, including attributes and annotations, without observations structureonly – (sdmx-json format only) returns only the structure part of the json document, no data is included. This is useful for knowing which Dimension values are valid for a dataset. When used in combination with metrics, a client application can know which codes remain valid selections based on current query state.																
format	Used to define the response format (as an alternative to the HTTP Accept Header).	SDMX Formats sdmx-compact-[version] (default) sdmx-generic-[version] sdmx-json sdmx-edi Where [version] is one of the following																

- 1.0
- 2.0
- 2.1 (default)

RDF Formats

rdf-xml
 rdf-json
 rdf-turtle

Excel Format

excel (use code ids and determine observation cells from Primary Measure)

excel-[resolve]-[obsCellFormat]

Where:

[resolve] used to indicate if code names are used instead of code ids, or if both ids and names should be included.

Valid values are:

id, name, both

[obsCellFormat] used to specify the type of the formatting to be used on Observation Cells in the Excel workbook. If not specified then the default behaviour will be to try and determine the cell type from the primary measure of the DSD. If specified as string, integer or double then this value will be applied to the Observation Cells regardless of observation value.

Valid values are:

primarymeasure, string, integer, double

CSV Format

csv
 or
 csv-ts
 or
 csv-sdmx
 or
 csv-[keyseparator]-[delimiter]-[decode]-[time]

The first format listed above will output the CSV as flat with the format decided by the Registry. The second deprecated format will output it based around time (on x or y axis).

Where:

		<p>[keyseparator] character used to separate component parts of the series key. E.g ':' would output a key like A:UK:EMPLOYMENT</p> <p>[delimiter] the CSV delimiter, can be one of the following strings: comma, tab, semicolon, space</p> <p>[decode] if true will output code names, if false will output code ids</p> <p>[time] whether time is output on the x axis or y axis. Allowed values are either x or y.</p>
includeHistory	Returns a dataset for every data import over time. Each dataset includes a validFrom and validTo date. Supports the use case of viewing how an observation value is revised over time.	true/false
includeMetadata	<p>If set to true, then the response will contain additional Annotations if there are reference metadata attached.</p> <p>The annotation will have the an AnnotationType of 'METADATA' and the AnnotationURI will provide a URI to the MetadataSet.</p>	<p>true/false</p> <p><i>Example:</i> includeMetadata=true</p>
includeMetrics	<p>For sdmx-json format only</p> <p>If true then the dataset will contain metrics about the number of series and observations that will be returned with the query. The structure section of the dataset will include codes which may not necessarily appear in the data, but remain valid future selections. The size of the cube for each code selection is also provided.</p> <p>See Annex 3 – Metrics and Data Availability</p>	<p><i>Example:</i> Include full metrics includeMetrics=true</p> <p><i>Example:</i> Include metrics, exclude the Time Dimension includeMetrics=excludeTime</p> <p><i>Example:</i> Do not include metrics includeMetrics=false</p>
includeAnnotations (Since v9.5.1)	If true then the series and observation will include any annotations stored against them. Default	<p><i>Example:</i> Include annotations on Series / Observations includeAnnotations=true</p>

	value is false.	
saveAs	<p>This will result in the response being saved to a file.</p> <p>The file extension is not required as the Fusion Registry will determine the extension based on the response format.</p>	
locale	<p>This optional parameter can be used to set the locale to return any multilingual text in (names and descriptions). If the text does not exist in the specified locale, then the default rules will be applied to find the next best appropriate locale.</p> <p>since v9.2.13 For CSV formats, the locale will also be used to format the observation value, if the value is a number</p>	<p>Any locale</p> <p><i>Example:</i> locale=fr</p>
serieslimit obslimit	<p>Optional Parameter used to limit the number of series/obs that can be returned from a Data Query.</p> <p>The response size is determined before the query is executed, and if the limit is exceeded an Error response is returned (XML or JSON depending on response format). The Error response code is 130 with a HTTP response code of 413 (request entity too large).</p>	
prettyPrint	<p>Set this to true to have the returned XML formatted. This will result in a larger file, but should be more readable</p>	prettyPrint=true

3.4.6 Examples

3.4.6.1 All World Bank Education data

https://registry.sdmxcloud.org/ws/public/sdmxapi/rest/data/WB,WDI_EDUCATION,1.0/all

3.4.6.2 World Bank Education data for New Zealand and Australia

https://registry.sdmxcloud.org/ws/public/sdmxapi/rest/data/WB,WDI_EDUCATION,1.0/..NZL+AUS

3.4.6.3 ONS Gross Domestic Product data since 2008 include historical revisions

<https://registry.sdmxcloud.org/ws/public/sdmxapi/rest/data/ONS,GDP,1.0/ALL?startPeriod=2008&includeHistory=true>

3.5 Query Data Series via POST API

This API allows data to be obtained for specific or wildcarded series keys from a POST request. The input to the service is a JSON Object defining the Series to be retrieved, the output is the dataset in the requested format.

3.5.1 Overview

URL Entry Point	/ws/public/sdmxapi/rest/data
Access	Public
HTTP Method	HTTP POST
Response Format	Multiple supported. Can be defined in the Accept HTTP Header, or the format parameter of the URL request.
Content Type	<ol style="list-style-type: none"> multipart/form-data (if attaching file) – the attached file must be in field name of uploadFile application/json if submitting data in the body of the POST
Error Response	SDMX-ML v2.1 Error Response Document
Available from version	v9.3

3.5.2 HTTP Headers

HTTP Header	Purpose	Allowed Values
Accept	To define the response format	<p>SDMX Formats</p> <p>application/vnd.sdmx.genericdata+xml;version=1.0 application/vnd.sdmx.genericdata+xml;version=2.0 application/vnd.sdmx.genericdata+xml;version=2.1 application/vnd.sdmx.structurespecificdata+xml;version=1.0 application/vnd.sdmx.structurespecificdata+xml;version=2.0 application/vnd.sdmx.structurespecificdata+xml;version=2.1 application/vnd.sdmx.edi application/vnd.sdmx.json application/vnd.sdmx.data+json;version=1.0.0-wd</p> <p>RDF Formats</p> <p>application/vnd.rdf+xml application/vnd.rdf+json application/vnd.rdf+turtle</p> <p>CSV/Excel Formats</p> <p>application/vnd.xlsx application/vnd.csv</p>
Accept-	This optional header can be	Accept-Language : en (English)

Language	<p>used to set the locale to return any multilingual text in (names and descriptions). If the text does not exist in the specified locale, then the default rules will be applied to find the next best appropriate locale.</p> <p>The corresponding locale parameter can be used to override this HTTP Header</p>	<p>Accept-Language : fr (French) Accept-Language : * (all languages – no filter) Accept-Language : all (all languages – no filter)</p>
-----------------	--	--

3.5.3 POST Request

```
{
  "id": "QueryId - Optional",
  "series": ["A:UK:EMP:M", " A:UK:EMP:F"],
  "startPeriod ": "2008",
  "endPeriod": "2009-Q2",
  "lastNObservations": 1,
  "firstNObservations": 2,
  "detail" : "full",
  "dataflow": {
    "agencyId" : "ONS",
    "id" : "EMPLY",
    "version" : "1.0"
  }
}
```

The POSTed content should be a valid JSON Object which contains an Array of series to obtain. A Series consists of a colon separated key, where each value represents a dimension value in the same sequence as the dimensions are defined in the corresponding Data Structure Definition (DSD). For example in the above example the DSD defines employment status and has dimensions FREQ, COUNTRY, EMPLOYMENT_STATUS, SEX, with the corresponding values being A (Annual), UK (United Kingdom), EMP (Employed), M (Male). Please note, the above example is based on a theoretical DSD / Code values.

To wildcard series, simply remove the code value from the series, for example A::EMP:M will return series for all reported codes in the COUNTRY Dimension, where all the other Dimension values match.

The **start** and **end** periods, **first** and **lastN** periods, and **detail** are optional parameters. These behave as documented in the Request Parameters for the previous section (Data Query via HTTP GET).

The dataflow object defines which Dataflow the series are for, and is required.

3.5.4 POST Response

The response is the SDMX dataset in the requested format.

3.6 Query Schemas via REST API (GET)

The Schema REST API conforms to the SDMX Web Service Guidelines which can be found at <http://sdmx.org>.

In addition to the SDMX specification, the Fusion Registry supports some additional Accept header values and query parameters.

Both the SDMX and extended query parameters are included in this document.

3.6.1 Overview

Web Service Entry Point	/ws/public/sdmxapi/rest/schema
Access	Public
HTTP Method	HTTP GET
Response Format	Multiple formats supported. Can be specified in the <i>Accept</i> HTTP Header, or the <i>format</i> parameter of the URL request.
Error Response	SDMX-ML v2.1 Error Response Document

3.6.2 HTTP Headers

The HTTP headers can be used to specify response format. This can also be defined in the request parameter.

HTTP Header	Purpose	Allowed Values
Accept	To define the response format	SDMX Formats application/vnd.sdmx.structure+xml;version=1.0 application/vnd.sdmx.structure+xml;version=2.0 application/vnd.sdmx.structure+xml;version=2.1

3.6.3 Resource

The resource is used to determine which structure type is being queried. The resources identified in the following table are supported.

Resource	Purpose
datastructure	Returns a schema where constraints attached to the DSD will be applied when generating the schema.
dataflow	Returns a schema where constraints attached to the dataflow and the DSD will be applied when generating the schema.
provisionagreement	Returns a schema where constraints attached to the provision agreement, dataflow and the DSD will be applied when generating the schema.

3.6.4 Path Parameters

The path parameters are used to further define the attributes of the request structure(s). All the path parameters are optional. If the path parameters have a default value, it will be used in the absence of the parameter.

Parameter	Purpose	Allowed Values
agencyID	The agency which owns the structure(s)	all – default. any agency Or any string compliant with the SDMX common:NCNameIDType
structureID	The id of the structure(s) to be returned	all – default. all structure ids Or any string compliant with the SDMX

		common:NCNameIDType
version	The version of the structure(s) to be returned	latest – default. latest version all – all versions Or a specific version number
itemID	If the resource is to an item scheme (Codelist, Concept Scheme, Category Scheme), the item inside the scheme can be identified by this parameter	String

3.6.5 Request Parameters

The request parameters are all optional and can be used to define the response detail, format, and any additional structures which reference, or are referenced by those identified in the query path.

Parameter	Purpose	Allowed Values
dimensionAtObservation	The ID of the dimension to be attached at the observation level.	TIME_PERIOD (default) – observations iterate over time. AllDimensions – no grouping.
explicitMeasure	For cross-sectional data validation, indicates whether observations are strongly typed.	true/false
format	Can be used to define the response format (as an alternative to the HTTP Accept Header).	sdmx (latest version) sdmx-1.0 sdmx-2.0 sdmx-2.1 <i>Example:</i> format=sdmx-2.1
saveAs	If provided the HTTP Header 'Content-Disposition' will be set to attachment with the filename being set to the value provided. This will result in the response being saved to a file. The file extension is not required as the Fusion Registry will determine the extension based on the response format.	String <i>Example:</i> saveAs=myDownload

3.6.6 Examples

3.6.6.1 Schema for Data Structure, DSD1, for the World Bank in SDMX v2.0 format

<https://registry.sdmxcloud.org/ws/public/sdmxapi/rest/schema/datastructure/WB/DSD1/1.0/?format=sdmx-2.0>

3.6.6.2 Schema for Data Structure, DSD2, for the World Bank in SDMX v2.1 format saved to a file

<https://registry.sdmxcloud.org/ws/public/sdmxapi/rest/schema/datastructure/WB/DSD2/1.0/?format=sdmx-2.1&saveAs=myDownload>

3.7 Query Data Registrations via REST API

This is not part of the SDMX REST specification, as there is no specification for data registration queries via REST. All these response messages conform to the SDMX specification for registration query responses.

The SDMX Specification allows for Registrations to be queried for using HTTP POST as described in section 3.1.

3.7.1 Overview

URL Entry Point	/ws/public/sdmxapi/rest/registration
Access	Public
HTTP Method	HTTP GET
Response Format	Multiple supported. Can be defined in the Accept HTTP Header, or the format parameter of the URL request.
Error Response	SDMX-ML v2.1 Error Response Document.

3.7.2 HTTP Headers

HTTP Header	Purpose	Allowed Values
Accept	To define the response format	SDMX Formats application/vnd.sdmx.structure+xml;version=2.0 application/vnd.sdmx.structure+xml;version=2.1

3.7.3 Resource

The resource is used to determine which structure type is being queried. The resources identified in the following table are supported.

Resource	Purpose
registration	Defines the that required resource is a SDMX Registration (or registrations)

3.7.4 Path Parameters

The path parameters are used to identify one or more structures, any registrations which directly or indirectly reference the identified structure(s) will be returned.

Parameter	Purpose	Allowed Values
structureType	Defines the structure type	dataprotider datastructure dataflow provisionagreement registration
agencyID	(optional) The Id of the agency that the structure is maintained by	String
structureId	(optional) The Id of the structure	String
version	(optional) The version of the structure	String

3.7.5 Request Parameters

Parameter	Purpose	Allowed Values
updatedBeforeDate	Returns any registrations which have been updated before the	Single time period (not a time range). Conforms to the ISO-8601 standard.

	given date.																	
updatedAfterDate	Returns any registrations which have been updated after the given date.	Supported time periods include																
		<table border="1"> <thead> <tr> <th>Period</th> <th>Example</th> </tr> </thead> <tbody> <tr> <td>Annual</td> <td>2009</td> </tr> <tr> <td>Semester</td> <td>2009-S1</td> </tr> <tr> <td>Trimester</td> <td>2009-T1</td> </tr> <tr> <td>Quarterly</td> <td>2009-Q1</td> </tr> <tr> <td>Monthly</td> <td>2009-01</td> </tr> <tr> <td>Daily</td> <td>2009-01-31</td> </tr> <tr> <td>Date Time</td> <td>2009--01T20:22:</td> </tr> </tbody> </table>	Period	Example	Annual	2009	Semester	2009-S1	Trimester	2009-T1	Quarterly	2009-Q1	Monthly	2009-01	Daily	2009-01-31	Date Time	2009--01T20:22:
Period	Example																	
Annual	2009																	
Semester	2009-S1																	
Trimester	2009-T1																	
Quarterly	2009-Q1																	
Monthly	2009-01																	
Daily	2009-01-31																	
Date Time	2009--01T20:22:																	
saveAs	<p>This will result in the response being saved to a file.</p> <p>The file extension is not required as the Fusion Registry will determine the extension based on the response format.</p>																	

3.7.6 Examples

3.7.6.1 All Registrations

<https://registry.sdmxcloud.org/ws/public/sdmxapi/rest/registration/registration>

3.7.6.2 All Registrations updated after 1st January 2016

<https://registry.sdmxcloud.org/ws/public/sdmxapi/rest/registration/registration?updatedAfterDate=2016-01-01>

3.7.6.3 All Registrations for Data Provider PROVIDER_1 owned by the Agency WB

https://registry.sdmxcloud.org/ws/public/sdmxapi/rest/registration/dataprovider/WB/PROVIDER_1

3.7.6.4 All Registrations for Data Structure BOP maintained by the IMF

<https://registry.sdmxcloud.org/ws/public/sdmxapi/rest/registration/datastructure/IMF/BOP>

3.7.6.5 All Registrations for specific Dataflow between 2014 and 2015

https://registry.sdmxcloud.org/ws/public/sdmxapi/rest/registration/dataflow/ESTAT/NAMAIN_T0101_A/1.6?updatedAfterDate=2014&updatedBeforeDate=2015

3.8 Query Reference Metadata via REST API

The SDMX speciation defines how to retrieve a Metadata Set if the Metadataflow identifiers are known, and metadata key. This is only useful for applications that already know what metadata exists, and what they wish to retrieve.

The Fusion Registry implementation provides the means to ask the question ‘what reference metadata do you have for this structure (or data) query. This is achieved by creating a valid structure or data query, as described in the previous sections, and prefixing the resource path parameter with the resource parameter ‘metadata’.

For example the following 2 queries show the difference between a structure query, and a metadata query for the same structures:

`/ws/public/sdmxapi/rest/codelist/ECB/all` = All codelists
`/ws/public/sdmxapi/rest/metadata/codelist/ECB/all` = Metadata for all codelists

The following 2 queries show the difference between a data query, and a metadata query for the same data:

`/ws/public/sdmxapi/rest/data/ECB,EXR/A...` = Exchange Rates (Annual data)
`/ws/public/sdmxapi/rest/metadata/data/ECB,EXR/A...` = Metadata for Exchange Rates (Annual data)

3.8.1 Overview

URL Entry Point	<code>/ws/public/sdmxapi/rest/metadata</code>
Access	Public
HTTP Method	HTTP GET
Response Format	Multiple supported. Can be defined in the <i>Accept</i> HTTP Header, or the <i>format</i> parameter of the URL request.
Error Response	SDMX-ML v2.1 Error Response Document

3.8.2 HTTP Headers

HTTP Header	Purpose	Allowed Values
Accept	To define the response format.	SDMX Formats application/vnd.sdmx application/vnd.sdmx.json

3.8.3 Resource

The resource is used to define that is reference metadata.

Resource	Purpose
metadata	To define that the required output is Reference Metadata.

3.8.4 Path Parameters

The subsequent path parameters can be either that of a valid data query, or structure query. The returned result includes all the Reference Metadata that is attached to the Data or Structures defined by the query, for example:

An SDMX structure query or data query can be post fixed to the entry point. For example:

3.8.4.1 Reference Metadata for OECD Population Dataflow

<http://registry.sdmxcloud.org/ws/public/sdmxapi/rest/metadata/dataflow/OECD/POPULATION/1.0>

3.8.4.2 Reference Metadata for all Dataflows

<http://registry.sdmxcloud.org/ws/public/sdmxapi/rest/metadata/dataflow/all/all/all>

3.8.4.3 Reference Metadata for Population Dataset (Annual Data for the UK)

<http://registry.sdmxcloud.org/ws/public/sdmxapi/rest/metadata/data/OECD,POPULATION,1.0/A.UK../>

In addition the path parameter 'set' can be used followed by the Id of a specific Metadata set, for example

3.8.4.4 Specific Metadata Set

<http://registry.sdmxcloud.org/ws/public/sdmxapi/rest/metadata/set/b53a14bb-4abb-450a-a715-93987056ead2>

3.8.5 Request Parameters

Parameter	Purpose	Allowed Values
match	<p>Defines if the metadata must match the query target exactly, or if it can be attached to child structures. This parameter is only relevant for Structure queries. For Data queries any metadata against Series, Codes, Concepts, Dimensions, Attributes, and Dataflows that are relevant to the dataset, will be returned.</p> <p>exact (default) match will only return metadata if it is attached to the specific structure that is being referenced, i.e the Codelist or the Code, or Concept.</p> <p>children match will check for metadata against the matched item, and any child items (recursive check). A query for metadata against a Codelist match=children will include all child codes. A query for a Code match=children will include any child Codes.</p> <p>all for a Structure query all will check for metadata for the Maintainable container regardless of the matched item, for example if a Code is in the target, all will return metadata for the entire Codelist (and child structures) that the Code is a member of.</p>	
format	To define the response format	sdmx sdmx-json
saveAs	<p>This will result in the response being saved to a file.</p> <p>The file extension is not required as the Fusion Registry will determine the extension based on the response format.</p>	
Locale	This optional parameter can be used to set the locale to return any multilingual text in (names and descriptions). If the text does not exist in the specified locale, then the default rules will be applied to find the next best appropriate locale.	Any locale <i>Example:</i> locale=fr

4 Fusion Registry Web Services

4.1 User Details

4.1.1 Obtaining Data Transactions

Returns information about the currently authenticated user on the session.

URL Entry Point	/ws/public/currentuser/details
Access	Public
HTTP Method	GET
Response Format	application/json
Error Response	{ "Error" : ["Error Message"] }
Since	v9.3.11

4.1.2 Server Response

The response details if the user is authenticated and if so returns their username and name

```
{
  "authenticated": true,
  "username": "mnelson",
  "name": "Matt Nelson"
}
```

4.2 Transaction History

4.2.1 Obtaining Data Transactions

Returns all the data transactions against the Registry. This may be filtered by parameter.

URL Entry Point	/ws/registry/tx/getDataTransactions
Access	Public
HTTP Method	GET
Response Format	application/json
Error Response	{ "Error" : ["Error Message"] }

4.2.2 Request Parameters

All of these parameters are optional.

Parameter	Purpose	Allowed Values
provision	The URN of the Provision Agreement.	String
flow	The URN of the Dataflow	String
provider	The URN of the Data Provider	String
url		String
dateFrom	The date from which to obtain transactions.	Date
dateTo	The highest date to retrieve transactions for.	Date
internalDS	To return transactions for data from internal data stores only.	true/false false is default
webservice	To return transactions for data from Web Services only.	true/false false is default
max		Positive Integer
user	To filter on a particular user	String
action	To filter on a particular action.	String
orderDesc	If true the returned order of transactions will be	true/false

4.2.3 Server Response

The response for all of the data transactions related to the specific Provision Agreement (if specified). The response is an Array of JSON objects.

```
[
  {
    "Date": 1505472449542,
    "Id": 111,
    "Action": "Append",
    "DataURL": "http://...",
    "ProvisionURN": "urn...",
    "DataflowURN": "urn...",
    "DataProviderURN": "urn...",
    "SeriesCount": 1,
    "DataFrom": 315532800000,
    "DataTo": 378691200000
  },
  ...
]
```



4.3 Data Validation

The data validation web service consumes a dataset (both SDMX and non-SDMX formats are supported) and returns a JSON response identifying details about the dataset, including if there are any validation errors.

The Fusion Registry also provides a HTML User Interface (UI) for data validation, which makes use of the same information. *The HTML UI is documented in the Fusion Registry User Guide.*

4.3.1 Overview

URL Entry Point	/ws/public/data/validate
Access	Public
HTTP Method	POST
Accepts	Dataset in any supported format including: <ul style="list-style-type: none"> • CSV • XLSX • SDMX Or other custom formats added via plugin framework
Compression	Zipped data can be submitted
Content Type	<ol style="list-style-type: none"> 1. multipart/form-data (if attaching file) – the attached file must be in field name of uploadFile 2. application/text or application/xml (if submitting data in the body of the POST)
Response Format	application/json
Error Response	{ "Status": "Error", "Errors": true, "Error" : "Error Message" }

4.3.2 HTTP Headers

HTTP Header	Purpose	Allowed Values
Data-Format	Used to inform the server when the data is in CSV format.	csv;delimiter=[delimiter] Where [delimiter] is either: <ul style="list-style-type: none"> • comma • tab • semicolon • space
Structure	Optional. Provides the structure to validate the data against. This is optional as this information may be present in the header of the DataSet. If provided this value will override the value in the dataset (if present).	Valid SDMX URN for Provision Agreement, Dataflow, or Data Structure Definition.
Inc-Metrics (Since v9.8)	Optional. Includes metrics on the validation. This will add extra detail to the validation report.	Boolean (true/false)
Inc-Valid (Since v9.8)	Optional. Instructs the service to include a dataset with all the valid series and observations in the response.	Boolean (true/false)

	As the result will contain a separate file for the dataset, the response format will be set to either multipart/mixed message with a boundary per file, or if the Zip header is set to true, the output will be a single zip file. The file is called ValidData with the file extension based on the output format.	
Inc-Invalid (Since v9.8)	Optional. Instructs the service to include a dataset with all the invalid series and observations in the response. As the result will contain a separate file for the dataset, the response format will be set to either multipart/mixed message with a boundary per file, or if the Zip header is set to true, the output will be a single zip file. The file is called InvalidData with the file extension based on the output format.	Boolean (true/false)
Accept (Since v9.8)	Optional. Instructs the service which data output format to output the valid or invalid datasets in. This Header is only used if Inc-Valid or Inc-Invalid are set to true.	See Accept formats for REST Data Query
Zip (Since v9.8)	Optional. Compresses the output as a zip file. If used in conjunction with Inc-Valid or Inc-Invalid the zip will contain multiple files.	Boolean (true/false)
Prior-Data-Dependent (Since v9.8)	Optional. This allows data to be validated under the assumption that other data will provide missing information. If this value is set to true, particular data validators will not be used when validating the data. These validators are "Mandatory Observations" and "Valid Calculations". Default value is false.	Boolean (true/false)

4.3.3 Validation Output

The validation output contains both human readable error descriptions, as well as machine processible locations of the errors within the dataset. The location in the dataset is described as a key or observation locator in the format; A:UK:M:2008 – where each component relates to the Dimension value, separated by a colon. If the error position is observation, the last part of the key is the observation time period.

There are 3 types of output that can be produced which share a common structure: unable to parse input (returns HTTP 400); able to parse input but references invalid data structure (returns HTTP 200); parsed input and returns output, which may have validation errors (return HTTPS 200). Below are examples of each:

Example output for a valid dataset:

```
{
  "Meta": {
    "RequestTime": 1564410081711,
    "Duration": 43
  },
}
```

```

"FileFormat": "Structure Specific (Compact) v2.1",
"Prepared": "2019-07-29T10:23:01",
"SenderId": "FR_DEMO",
"DataSetId": null,
"Status": "Complete",
"Errors": false,
"Datasets": [
  {
    "DSD": "urn:sdmx:org.sdmx.infomodel.datastructure.DataStructure=OECD:HIGH_AGLINK_2011(1.0)",
    "Dataflow": "urn:sdmx:org.sdmx.infomodel.datastructure.Dataflow=OECD:AGRIC_OUTLOOK_2011_2020(1.0)",
    "DataProvider": "urn:sdmx:org.sdmx.infomodel.base.DataProvider=METATECH:DATA_PROVIDERS(1.0).METATECH",
    "ProvisionAgreement": "urn:sdmx:org.sdmx.infomodel.registry.ProvisionAgreement=OECD:OECD_AGRIC_OUTLOOK(1.0)",
    "KeysCount": 2,
    "ObsCount": 62,
    "GroupsCount": 0,
    "Errors": false
    "ReportedPeriods": {
      "A": {
        "Name": "Annual",
        "StartPeriod": "1990",
        "EndPeriod": "2020"
      }
    }
  }
],
"PreventsConversion": false,
"PreventsPublication": false
}

```

Example output for a dataset that has been validated but errors exist:

```

{
  "Meta": {
    "RequestTime": 1564401209760,
    "Duration": 34
  },
  "InvalidData": {
    "Datasets": [
      {
        "Structure": "urn:sdmx:org.sdmx.infomodel.registry.ProvisionAgreement=OECD:OECD_AGRIC_OUTLOOK(1.0)",
        "Series": 2,
        "Observations": 61,
        "Groups": 0
      }
    ]
  },
  "ValidData": {
    "Datasets": [
      {
        "Structure": "urn:sdmx:org.sdmx.infomodel.registry.ProvisionAgreement=OECD:OECD_AGRIC_OUTLOOK(1.0)",
        "Series": 2,
        "Observations": 32,
        "Groups": 0
      }
    ]
  },
  "FileFormat": "Structure Specific (Compact) v2.1",
  "Prepared": "2019-07-29T10:23:01",
  "SenderId": "FR_DEMO",
  "DataSetId": null,
  "Status": "Complete",
  "Errors": true,
  "Datasets": [
    {
      "DSD": "urn:sdmx:org.sdmx.infomodel.datastructure.DataStructure=OECD:HIGH_AGLINK_2011(1.0)",
      "Dataflow": "urn:sdmx:org.sdmx.infomodel.datastructure.Dataflow=OECD:AGRIC_OUTLOOK_2011_2020(1.0)",
      "DataProvider": "urn:sdmx:org.sdmx.infomodel.base.DataProvider=METATECH:DATA_PROVIDERS(1.0).METATECH",
      "ProvisionAgreement": "urn:sdmx:org.sdmx.infomodel.registry.ProvisionAgreement=OECD:OECD_AGRIC_OUTLOOK(1.0)",
      "KeysCount": 3,
      "ObsCount": 93,
      "GroupsCount": 0,
      "ReportedPeriods": {
        "A": {
          "Name": "Annual",

```



```

        "StartPeriod": "1990",
        "EndPeriod": "2020"
    }
},
"Errors": true,
"ValidationReport": [
    {
        "Type": "Constraint",
        "Errors": [
            {
                "Message": "Disallowed Dimension Value: REF_AREA=AFR",
                "Dataset": 0,
                "ComponentId": " REF_AREA ",
                "ReportedValue": "AFR",
                "Position": "Series",
                "Keys": ["AFR:BT:AA"]
            }
        ]
    },
    {
        "Type": "Representation",
        "Errors": [
            {
                "Message": "Dimension 'VARIABLE' is reporting value 'AA' which is not a valid
representation in referenced Codelist 'OECD:CL_HIGH_AGLINK_2011_VARIABLE(1.0)'",
                "Dataset": 0,
                "Position": "Series",
                "ComponentId": "VARIABLE",
                "ReportedValue": "AA",
                "Keys": ["AFR:BT:AA"]
            },
            {
                "Message": "Error in Primary Measure 'OBS_VALUE': Reported value 'XXX' is not of expected
type 'Double'",
                "Dataset": 0,
                "ComponentId": " OBS_VALUE",
                "ReportedValue": "XXX",
                "Position": "Observation",
                "Keys": ["AFR:BT:IM:2010"]
            }
        ]
    },
    {
        "Type": "FormatSpecific",
        "Errors": [
            {
                "Message": "Unexpected attribute 'ASD' for element 'StructureSpecificData/DataSet/Series/Obs'",
                "Dataset": 0,
                "Position": "Dataset"
            }
        ]
    }
]
},
"PreventsConversion": false,
"PreventsPublication": true
}

```

Note the first three elements 'Meta', 'InvalidData', 'ValidData', there are present in the report if Inc-Metrics is set to true. Inc-valid and Inc-Invalid set to true enables the report to know the metrics for the invalid and valid data.

Note also each Error has a Type, this is the category of error which caused the validator to fail. For a list of all validators see the following section on Validators.

The Error Position is either set to Dataset, Series, Observation, or Group.

PreventsConversion and PreventsPublication is an indication on the severity of the error. These settings on which errors prevent conversion and publication can be set in the Fusion Registry by the administrator of the system.

Example output for an invalid dataset reference:

In this case, "Errors" has a value of true, the Status states that this is an Invalid Reference and the Datasets section refers to the invalid Dataflow

```
{
  "FileFormat": "Generic v2.1",
  "MimeType": "application/xml",
  "Status": "InvalidRef",
  "Errors": true,
  "Datasets": [{"Dataflow":
"urn:sdmx:org.sdmx.infomodel.datastructure.Dataflow=BIS:INVALID_DATAFLOW(1.0)"}],
  "Error": "Unexpected '<' character in element (missing closing '>?')\r\n at [row,col {unknown-
source}]: [17,3]"
}
```

Example output for un-processible request:

When the input could not be processed, "Errors" has a status of true, the "Status" is "Error" and an Error section gives detail on the error process.

```
{
  "Status": "Error",
  "Errors": true,
  "Error": "Unexpected '<' character in element (missing closing '>?')\r\n at [row,col {unknown-
source}]: [17,3]"
}
```

4.4 Validators

Returns a list of all the data validators used by the Registry.

URL Entry Point	/ws/public/settings/validation/validators
Access	Public
HTTP Method	GET
Response Format	application/json
Error Response	{ "Error" : ["Error Message"] }

4.5 Data Authoring (Excel Report Template)

The Data Authoring service generates an Excel Report Template for a specific Data Provider. The service is not secure by default but can be made secure by setting the Registry to have Reporting Template Security enabled. When secure the authentication information must be provided. The collecting organisation must have set up a Reporting Template definition before an Excel Reporting Template can be generated.

For more information, read the Excel Report Template guide.

4.5.1 Overview – List Templates

URL Entry Point	/ws/public/reporttemplate/templates
Access	Public unless secured and then is Secure Data Provider, Agency or Admin only
HTTP Method	GET
Response Format	application/json
Error Response	{ "Error" : "Error Message" }

4.5.1.1 Server Response

The server will respond with a JSON array, each array object contains a Map of template URN to an array of data providers that can report for the template. The list of Data Providers will be filtered based on user type (data providers can-not see what other data providers can report)

```
[
  {
    "urn" : "template-urn",
    "providers" : ["provider-urn"]
  }
]
```

4.5.2 Overview – Create Excel File based on Template

URL Entry Point	/ws/secure/reporttemplate/provider/create
Access	Secure Data Provider, Agency or Admin only
HTTP Method	GET
Response Format	application/octet-stream
Error Response	Error file (error.txt)

4.5.3 Path Parameters

The path parameters are used to further define the attributes of the request structure(s). All the path parameters are optional. If the path parameters have a default value, it will be used in the absence of the parameter.

`/ {agencyId} / {templateId} / {version} / {providerAgencyId} / {providerId}`

Parameter	Purpose	Allowed Values
agencyID	The agency which owns the Reporting Template to use	Any string compliant with the SDMX common:NCNameIDType
templateId	The id of the Reporting Template to use	Any string compliant with the SDMX common:NCNameIDType
version	The version of the Reporting Template to use	latest latest version Or a specific version number
providerAgencyId	The Agency Id that the Data Provider belongs to	String
providerId	The Data Provider Id to generate the template for	String

4.5.4 Query Parameters

The query parameters are used to define the report period for which the observations are created.

Parameter	Purpose	Allowed Values										
reportPeriod	This query parameter is used to specify the report period to generate the report for. This is a required parameter.	Single time period (not a time range). Conforms to the ISO-8601 standard. Supported time periods include										
reportPeriodTo (Since v9.3)	If the template is to be created for a range of time periods this is achieved using the reportPeriodTo parameter	<table border="1"> <thead> <tr> <th>Period</th> <th>Example</th> </tr> </thead> <tbody> <tr> <td>Annual</td> <td>2009</td> </tr> <tr> <td>Semester</td> <td>2009-S1</td> </tr> <tr> <td>Trimester</td> <td>2009-T1</td> </tr> <tr> <td>Quarterly</td> <td>2009-Q1</td> </tr> </tbody> </table>	Period	Example	Annual	2009	Semester	2009-S1	Trimester	2009-T1	Quarterly	2009-Q1
Period	Example											
Annual	2009											
Semester	2009-S1											
Trimester	2009-T1											
Quarterly	2009-Q1											

		Monthly	2009-01
		Weekly	2009-W1
		Daily	2009-01-31
		Date Time	2009-- 01T20:22:00
Freq (Since v9.3)	The frequency can be derived from the passed in reportPeriod parameter. However this optional argument can be used to format the reportPeriod and reportPeriodTo to conform to another frequency. For example If reportPeriod=2009-01-01 and freq=A then the Excel template will be created for 2009	Period	Example
		Annual	A
		Semester	S
		Trimester	T
		Quarterly	Q
		Weekly	W
		Monthly	M
		Daily	D

4.5.4.1 Example

This example shows the Data Provider **ONS** who belongs to the Agency **ACY** is creating a template which is owned by **ACY** has an id of **TEMPLATE_1** and a version of **1.0**

https://myserver.org/FusionRegistry/ws/secure/reporttemplate/provider/create/ACY/TEMPLATE_1/1.0/ACY/ONS?reportPeriod=2018-Q1

4.6 Data Transformation

4.6.1 Overview

URL Entry Point	/ws/public/data/transform
Access	Public
HTTP Method	POST
Accepts	Dataset in any supported format including: <ul style="list-style-type: none">• CSV• XLSX• SDMX Or other custom formats added via plugin framework. Zipped data can be submitted.
Compression	
Content Type	<ol style="list-style-type: none">1. multipart/form-data (if attaching file) – the attached file must be in field name of uploadFile2. application/text or application/xml (if submitting data in the body of the POST)
Response Format	Defaults to SDMX Structure Specific v2.1
Error Response	{ "Error" : "Error Message" }

4.6.2 HTTP Headers

The Accept Header is used to define the output format, to transform the data to. The supported accept header values are:

SDMX Formats

```
application/vnd.sdmx.genericdata+xml;version=1.0
application/vnd.sdmx.genericdata+xml;version=2.0
application/vnd.sdmx.genericdata+xml;version=2.1
application/vnd.sdmx.structurespecificdata+xml;version=1.0
application/vnd.sdmx.structurespecificdata+xml;version=2.0
application/vnd.sdmx.structurespecificdata+xml;version=2.1
application/vnd.sdmx.edi
application/vnd.sdmx.json
```

RDF Formats

```
application/vnd.rdf+xml
application/vnd.rdf+json
application/vnd.rdf+turtle
```

CSV/Excel Formats

```
application/vnd.xlsx
application/vnd.csv
```

In addition, the following optional header parameters can be used to provide further details on the incoming dataset. If these details are not provided, the Fusion Registry will interrogate the dataset header to get the information. If the dataset is a non-SDMX format, or does not contain the required information in the header, then an error response will be returned.

HTTP Header	Purpose	Allowed Values
Data-Format	Used to inform the server when the data is in CSV format.	csv;delimiter=[delimiter] Where [delimiter] is either: <ul style="list-style-type: none"> • comma • tab • semicolon • space
Sender-Id	The SenderId is included in the validation report. If not provided, the SenderId will be taken from the header of the dataset. If the dataset does not contain a SenderId (for example a non-SDMX format) then the validation report will contain the SenderId of the Fusion Registry.	The following characters are allowed: A-z, a-z 0-9 \$, _ , -, @, \
Receiver-Id (Since v9.8.1)	The ReceiverId may be included in the validation report. If not provided, the ReceiverId will be taken from the header of the dataset if it is present. If the dataset does not contain a ReceiverId (for example a non-SDMX format) then the validation report will not contain a ReceiverId in the header.	The following characters are allowed: A-z, a-z 0-9 \$, _ , -, @, \
Structure	Provides the structure used to read the data. This is optional as this information may be present in the header of the DataSet. If provided this value will override the value in the dataset (if present).	Valid SDMX URN for Provision Agreement, Dataflow, or Data Structure Definition.
Dataset-Idx	If the loaded file contains multiple datasets, this argument can be used to indicate which dataset is transformed. If this argument is not present then all datasets will be in the output file (if the file formats permits multiple datasets).	Zero indexed integer, example: 0
Dataset-Id (Since v9.8.1)	An optional parameter which allows the user to specify the value of the DataSetID generated in the validation.	The following characters are allowed: A-z, a-z 0-9 \$, _ , -, @, \ <p>Specific variables permit the insertion of Data Structure / Data Flow values. These values are:</p> <pre> \${DATFLOW_ID} \${DATFLOW_ACY} \${DATFLOW_VER} \${DSD_ID} \${DSD_ACY} \${DSD_VER} </pre>

		Note that dots in the version number will be replaced with the _ character, since dots are not permitted in the ID.
Dataset-Action (Since v9.8.1)	An optional parameter which allows the user to specify the value of the DataSetAction generated in the validation report. If this parameter is not specified, the default value will be used.	May be one of the following: <ul style="list-style-type: none"> • Append • Replace • Merge • FullReplace • Delete • Information
Map-Structure (Since v9.2.13)	<p>An optional parameter to inform the Fusion Registry to transform the structure of the dataset to conform to another Data Structure Definition.</p> <p>The value provided can be a URN of a Dataflow or Data Structure Definition to map the incoming data to. A Structure Map must exist in the Fusion Registry which maps between the incoming Data Structure/Dataflow and Mapped Data Structure/Dataflow.</p> <p>Alternatively the URN may be the URN of the Data Structure Map to use for the mapping (since v9.4.4)</p>	Valid SDMX URN for Dataflow or Data Structure Definition.
Inc-Unmapped (Since v9.6.5)	<p>If the Map-Structure Header is used, then the inclusion of Inc-Unmapped will output a second dataset, if there are unmapped series. The additional dataset contains the data that could not be mapped due to missing mapping rules, or ambiguous outputs.</p> <p>The format of the additional dataset is the same format as the output dataset.</p> <p>As the result may contain a separate file, the response format is either set to multipart/mixed message with a boundary per file, or if the Zip header is set to true, the output will be a single zip file. The file names are 'out' and 'unmapped' with the file extension based on the output format.</p>	Boolean (true/false)
Inc-Metrics (Since v9.6.5)	<p>Includes metrics on the transformation.</p> <p>The result will contain a separate file, either as a multipart/mixed message with a boundary per file, or if the Zip header is set to true, the output will be a single zip file.</p>	Boolean (true/false)
Fail-On-Error (Since v9.5.0)	An optional parameter to tell the transformation process to fail if an error is detected in the dataset.	Boolean (true/false)
Zip (Since v9.6.5)	Compresses the output as a zip file. This if used in conjunction with Inc-Metrics or Inc-Unmapped the zip will contain multiple	Boolean (true/false)

4.6.3 Inc-Metrics

The following JSON is an example response when Inc-Metrics header is set to true. Request Time is Epoch Time Milliseconds, and Duration is measured in the number of milliseconds taken to complete the transformation.

```
{
  "Meta": {
    "RequestTime": 1559124708568,
    "Duration": 220
  },
  "SourceData": {
    "Datasets": [{
      "Structure":
"urn:sdmx.org.sdmx.infomodel.datastructure.Dataflow=BIS:IN_FLOW(1.0)",
      "Series": 3118,
      "Observations": 3118,
      "Groups": 0
    }
  ],
  "OutputData": {
    "Datasets": [{
      "Structure":
"urn:sdmx.org.sdmx.infomodel.datastructure.Dataflow=BIS:OUT_FLOW(1.0)",
      "Series": 1753,
      "Observations": 1855,
      "Groups": 0
    }
  ],
  "UnMappedData": {
    "Datasets": [{
      "Structure":
"urn:sdmx.org.sdmx.infomodel.datastructure.Dataflow=BIS:IN_FLOW(1.0)",
      "Series": 1263,
      "Observations": 1263,
      "Groups": 0
    }
  ]
}
}
```

4.7 Data Submission

4.7.1 Overview

A Dataset can be submitted to the secure web service, authentication details need to be included in the HTTP Header of the message (Basic Authentication). The Dataset must have a reference to the Provision Agreement (or Dataflow) that the data is for. If the Dataset does not contain this information, it can be included on the HTTP Header. If the Dataset references a Dataflow, the Provision Agreement will be determined based on the user (Data Provider). If the authenticated user account is for Admin or multiple Data Providers, then the dataset must reference the Provision Agreement to use explicitly (either in the dataset header, or HTTP header).

4.7.2 Import Data

URL Entry Point	/ws/secure/data/publish
Access	Secure: Admin + Data Provider Users
HTTP Method	POST
Accepts	Dataset in any supported format including: <ul style="list-style-type: none"> • CSV • XLSX • SDMX



Compression Content Type	Or other custom formats added via plugin framework
	Zipped data can be submitted
	<ol style="list-style-type: none"> 1. multipart/form-data (if attaching file) – the attached file must be in field name of uploadFile 2. application/text or application/xml (if submitting data in the body of the POST) 3. application/csv if the data is in CSV format (since v9.2.7)
Response Format	application/json
Error Response	{ "Error" : "Error Message" }

4.7.3 HTTP Headers

HTTP Header	Purpose	Allowed Values
ProvisionURN	This can be used to tell the Fusion Registry which Provision Agreement to use for Data Import. This will override the Dataset Header information if it exists.	A valid Provision Agreement URN
DatasetAction	Specifies the action to perform when submitting the dataset. If this value is specified it will override any action specified in the dataset.	Append, Replace or Delete.

4.7.4 Server Response

The server will respond with a JSON object containing a token to track the progress of the import.

```
{"Id" : "unique-token-as-a-UUID" }
```

4.8 Check Import Progress

URL Entry Point	/ws/secure/data/status/find
Access	Secure: Admin + Agency, Data Provider Users
HTTP Method	GET
Response Format	application/json
Error Response	{ "Error" : "Error Message" }

4.8.1 Request Parameters

Parameter	Purpose	Allowed Values
id	The id that the server provided on data import which is used to uniquely track the dataset.	String

4.8.2 Server Response

The response status, any errors, import action, and general details about the Dataset.

```
{
  "Status": "SUCCESS",
  "Username": "user_id",
  "Start": 1464781420000,
  "Completion": 1464781422000,
  "Errors": [],
  "Keys": 960,
  "Obs": 11220,
  "Action": "Information",
  "DataflowUrn": "urn...",
  "ProvisionUrn": "urn...",
  "DatasetId": "2401cfbb-1783-4ee2-ad4f-d900332e0034"
}
```

Status returned can be one of the following:

- PENDING - the data is undergoing initial validation
- QUEUED - the data has been validated and is awaiting database import
- PROCESSING - the data is being loaded into the database
- PROCESSED - the data has been imported but not yet indexed
- INDEXING - the fusion registry is reindexing the data
- SUCCESS - the data import has completed with no errors
- ERROR - the import resulted in error

If the status is PROCESSING then the response Object will also contain PercentageComplete as a number, for example:

```
{
  "Status": "PROCESSING",
  "PercentageComplete": 72,
  ...
}
```

The DatasetId is the unique identifier for the dataset, and will match the 'id' query parameter.

NOTE: Start and End Dates is the measured in number of milliseconds since January 1st 1970.

4.9 Historical Import Status

URL Entry Point	/ws/secure/data/status/historical
Access	Secure: Admin + Agency, Data Provider Users
HTTP Method	GET
Response Format	application/json
Error Response	{ "Error" : "Error Message" }

4.9.1 Server Response

The response status for any imports the user has permissions to see, for an admin user this is all imports, for a Data Provider it is all imports for the Data Provider, for an Agency user it is all imports related to Dataflows maintained by the Agency.

The response is an Array of JSON objects.

```
[
  {
    "Status": "SUCCESS",
    "Username": "user_id",
    "Start": 1464781420000,
    "Completion": 1464781422000,
    "Errors": [],
    "Keys": 960,
    "Obs": 11220,
    "Action": "Information",
    "DataflowUrn": "urn...",
    "ProvisionUrn": "urn...",
    "DatasetId": "2401cfbb-1783-4ee2-ad4f-d900332e0034"
  },
  ...
]
```

4.10 Deleting Data

It is possible to delete data by submitting a Dataset with Action DELETE. The deletion follows the SDMX rules for delete.

In addition the Fusion Registry web services provide functions to delete data from Fusion Registry managed data stores. Data deleted via the web service will bypass the revisions logic, and therefore will not be recorded as a revision to the observation.

4.10.1 Delete Data by Datasource and Dataflow

Delete all data in a Registry Managed data source for a Dataflow (across all Provision Agreements).

URL Entry Point	/ws/secure/datasource/deleteDataByConnectionAndFlow
Access	Secure: Admin only
HTTP Method	POST
Content Type	application/json
Response Format	application/json
Error Response	{ "Error" : "Error Message" }
Success Response	{ "Success" : true }

4.10.1.1 POST Content

JSON Object containing the Data Source Id, and the Dataflow URN.

```
{
  "Id": "DatasourceId",
  "URN": "DataflowURN"
}
```

4.10.2 Delete Data by Datasource and Provision Agreement

Delete all data in a Registry Managed data source for a specific Provision Agreement.

URL Entry Point	/ws/secure/datasource/deleteDataByConnectionAndProvision
Access	Secure: Admin only
HTTP Method	POST
Content Type	application/json
Response Format	application/json
Error Response	{ "Error" : "Error Message" }
Success Response	{ "Success" : true }

4.10.2.1 POST Content

JSON Object containing the Data Source Id, and the Provision Agreement URN.

```
{
  "Id": "DatasourceId",
  "URN": "ProvisionURN"
}
```


4.10.3 Delete Data for Datasource

Deletes all data stored in a data source, across all Dataflows.

URL Entry Point	/ws/secure/datasource/deleteDataByConnectionId
Access	Secure : Admin only
HTTP Method	POST
Content Type	application/json
Response Format	application/json
Error Response	{ "Error" : "Error Message" }
Success Response	{ "Success" : true }

4.10.3.1 POST Content

JSON Object containing the Data Source Id.

```
{
  "Id": "DatasourceId"
}
```

4.10.4 Delete Series

Deletes one or more series for a Provision Agreement.

URL Entry Point	/ws/secure/datasource/deleteSeries
Access	Secure Admin only
HTTP Method	POST
Content Type	application/json
Response Format	application/json
Error Response	{ "Error" : "Error Message" }
Success Response	{ "Success" : true }

4.10.4.1 POST Content

JSON Array of JSON Objects containing the Provision URN and an Array of Series 'short codes'

```
[{
  "Provision": "ProvisionURN",
  "Series": ["A:UK:M:SER1", "A:FR:M:SER1", "A:UK:M:SER2"]
}]
```

4.10.5 Delete Observations

Delete one or more observations from one or more series.

URL Entry Point	/ws/secure/datasource/deleteObs
Access	Secure Admin only
HTTP Method	POST
Content Type	application/json
Response Format	application/json
Error Response	{ "Error" : "Error Message" }
Success Response	{ "Success" : true }

4.10.5.1 POST Content

JSON Object containing the Provision or Dataflow URN and an Array of Series 'short codes'.

```
{
  "URN" : "DataflowURN or ProvisionURN",
  "Series" : [{"Key": "A:UK:M:SER1", "Obs" : ["2008", "2009"]} ]
}
```

4.10.6 Delete DataSource

Deletes the connection to the data source but not the data itself.

URL Entry Point	<code>/ws/secure/datasource/deleteDataSource</code>
Access	Secure Admin only
HTTP Method	POST
Content Type	application/json
Response Format	application/json
Error Response	{ "Error" : "Error Message" }
Success Response	{ "Success" : true }

4.10.6.1 POST Content

JSON Object containing the Provision or Dataflow URN and an Array of Series 'short codes'.

```
{
  "Id": "DatasourceId"
}
```

4.11 Rolling back Data

For data stored in a Fusion Data Store, it is possible to rollback data submissions, permitting the return to a set of prior set of data. To rollback, it is necessary to know the data transaction number to rollback to. This can be obtained from the Transaction History (see section Transaction History of this document).

4.11.1 Rollback Data Submission

To rollback a data submission, specify the data transaction to rollback to. Note that rolling back of data is only permitted against data in a Fusion Data Store.

URL Entry Point	/ws/secure/datasource/rollbackDatasource
Access	Secure Admin only
HTTP Method	POST
Content Type	application/json
Response Format	application/json
Error Response	{ "Error" : ["Error Message"] }
Success Response	{ "Success" : true }

4.11.1.1 POST Content

JSON Object containing the Transaction Id of the data transaction to be rolled back. The Transaction Id ("TxId") is the "Id" value from the transaction which can be obtained by querying the Web Service for data transactions (see section 4.2). Example content:

```
{
  "TxId": 17
}
```

4.12 Data Sources

It is possible to retrieve information about the Data Sources.

4.12.1 Get all Datasources

Returns an array of data sources, each with a count of series and dataflows.

URL Entry Point	/ws/secure/datasource/getAllDataSources
Access	Secure Admin only
HTTP Method	GET
Response Format	application/json
Error Response	{ "Error" : "Error Message" }

4.12.1.1 Server Response

An array of JSON Objects, one for each data source, defining the database type, and the data stored, broken down by dataflow, then provision agreement.

```
[
  {
    "id": "IN_MEMORY",
    "registryManaged": true,
    "sqlStore": false,
    "dataflows": [
      {
        "urn": "urn:sdmx:org.sdmx.infomodel.datastructure.Dataflow=ACY1:DFID(1.0)",
        "name": "Debt service ratio",
        "provisions": [
          {
            "urn": "urn:sdmx:org.sdmx.infomodel.registry.ProvisionAgreement=ACY1:PROV1(1.0)",
            "name": "Debt Service Ratio",
            "provider":
"urn:sdmx:org.sdmx.infomodel.base.DataProvider=ACY1:DATA_PROVIDERS(1.0).DP1",
            "series": 66,
            "obs": 4674,
            "lastUpdated": 1506931845415
          }
        ]
      }
    ]
  },
]
```

A Registry Managed data source is one which can have data imported into it from the Fusion Registry, and can have data deleted from it via the Fusion Registry. When SQL Store is false, this indicates that the data store is the Fusion Store, and database queries are not executed against a SQL Store.

4.12.2 Get specific Datasource

Returns information for a specific data source.

URL Entry Point	/ws/secure/datasource/get
Access	Secure Admin only
HTTP Method	GET
Response Format	application/json
Error Response	{ "Error" : "Error Message" }

4.12.2.1 Request Parameters

Parameter	Purpose	Allowed Values
Id	The id of the data source to return information for.	String

4.12.2.2 *Server Response*

A JSON Object defining the data source connection.

```
{
  "Id": "MY_DATABASE",
  "DataSourceType" : "REGISTRY",
  "DatabaseType" : "MY_SQL",
  "ServerUrl" : "http://...",
  "Schema": "db_schema_name ",
  "Port" : 3306,
  "Username" : "user",
  "UseSso" : false
}
```



4.12.3 Get Dataflows for Datasource

Returns the data source with the given id, along with information about Dataflows that have been linked to the data source.

URL Entry Point	/ws/secure/datasource/getDataSourceFlows
Access	Secure Admin only
HTTP Method	GET
Response Format	application/json
Error Response	{ "Error" : "Error Message" }

4.12.3.1 Request Parameters

Parameter	Purpose	Allowed Values
Id	The id of the data source to return information for	String

4.12.3.2 Server Response

A JSON Object defining the data source connection along with an Array of linked Dataflows and information about the data stored.

```
{
  "Id": 12,
  "Name": "My Oracle DB",
  "DataSourceType": "REGISTRY",
  "DatabaseType": "MY_SQL",
  "ServerUrl": "http://...",
  "Schema": "my_database ",
  "Port": 3306,
  "Username": "user",
  "Dataflows": [
    {
      "URN": "urn...",
      "AgencyId": "ACY1",
      "Id": "DF_ID",
      "Version": "1.0",
      "Name": "Dataflow Name",
      "Provisions": ["URN Array"],
      "Series": 2452,
      "Providers": 3,
      "LastUpdated": "YYYY-MM-DDTHH:mm:ss",
      "DataStart": "YYYY-MM-DDTHH:mm:ss",
      "DataEnd": "YYYY-MM-DDTHH:mm:ss"
    }
  ]
}
```

4.12.4 Get Data Providers for Datasource

Returns the data source with the given id for the given Dataflow along with information about the linked Data Providers.

URL Entry Point	/ws/secure/datasource/getDataSourceProviders
Access	Secure Admin only
HTTP Method	GET
Response Format	application/json
Error Response	{ "Error" : "Error Message" }

4.12.4.1 Request Parameters

Parameter	Purpose	Allowed Values
id	The id of the data source to return information for	String
urn	The dataflow URN	flowUrn

4.12.4.2 Server Response

A JSON Object defining the data source connection along with an Array of linked Dataflows and information about the data stored.

```
{
  "Id" : "MY_DATASOURCE",
  "DataSourceType": "REGISTRY",
  "DatabaseType" : "MY_SQL",
  "ServerUrl" : "http://..."
  "Schema" : "matrix_plugin",
  "Port" : 3306,
  "Username" : "user",
  "DataflowName" : "Balance of Payments",
  "DataProviders" : [
    {
      "ProvisionURN: "urn... ",
      "Id" : "DATA_PROVIDER_ID",
      "Name" : "Bank of France",
      "Series" : 2452,
      "LastUpdated" : "YYYY-MM-DDTHH:mm:ss",
      "DataStart" : "YYYY-MM-DDTHH:mm:ss",
      "DataEnd" : "YYYY-MM-DDTHH:mm:ss"
    }
  ]
}
```

4.13 Structure Search Engine

Performs a text search of Structure Metadata in the Fusion Registry.

URL Entry Point	/ws/public/search
Access	Public
HTTP Method	GET
Response Format	application/json

4.13.1 Request Parameters

Parameter	Purpose	Allowed Values
query	The query string to search for	String
auto	If true, will return an array of type ahead suggestions, if false will return the search results	Boolean

4.13.2 Server Response

With auto=true

```
[ "Fish species, threatened" ]
```

With auto=false (or no auto parameter passed)

```
{
  time: 146, //measured in milliseconds
  resulttypes: [ "Codelist" ],
  results: [
    [
      "urn:sdmx:org.sdmx.infomodel.codelist.Codelist=WB:CL_SERIES_WDI(1.0)",
      "Codelist",
      "WB",
      "CL_SERIES_WDI",
      "1.0",
      "Series code list",
      1,
      [
        [
          "Fish species, threatened",
          "EN_FSH_THRD_NO",
          "Code"
        ]
      ]
    ]
  ]
}
```


4.14 Data Search Engine (Beta)

Performs a text search of Data in the Fusion Registry, the matched item is the dataflow which has a direct or indirect link to data, for example a direct link is a hit on the dataflow name, and indirect link is a match on a Code which has data for a particular (or multiple) dataflows.

URL Entry Point	<code>/ws/public/datasearch</code>
Access	Public
HTTP Method	GET
Response Format	application/json
Available from version	v9.2.23

4.14.1 Request Parameters

Parameter	Purpose	Allowed Values
query	The query string to search for	String
auto	If true, will return an array of type ahead suggestions, if false will return the search results	Boolean

4.14.2 Server Response

With `auto=true`

```
[ "Fish species, threatened" ]
```

With `auto=false` (or no `auto` parameter passed)

```
{
  time: 35, //measured in milliseconds
  results: [
    {
      "Dataflow": "urn:sdmx:org.sdmx.infomodel.datastructure.Dataflow=IMF:CBS(1.0)",
      "Matches": [
        [
          "Monetary, Central Bank, Assets, Currency and Deposits",
          "urn:sdmx:org.sdmx.infomodel.codelist.Code=IMF:CL_INDICATOR(1.0).FAAFCH_EUR"
        ],
        [
          "Monetary, Central Bank Survey, Loans, Euros",
          "urn:sdmx:org.sdmx.infomodel.codelist.Code=IMF:CL_INDICATOR(1.0).FASL_EUR"
        ],
        [
          "Latvijas Banka",
          "urn:sdmx:org.sdmx.infomodel.base.DataProvider=IMF:DATA_PROVIDERS(1.0).LV2"
        ]
      ]
    },
    {
      "Dataflow": "urn:sdmx:org.sdmx.infomodel.datastructure.Dataflow=IMF:SBS(1.0)",
      "Matches": [...]
    }
  ]
}
```

The results provide an array of Objects, each Object represents a match for a Dataflow, with a sub-array of **Matches**. Each Match is an array where the first item is the matched String, and the second item is the URN of the structure that was matched on.

4.15 Reverse Engineer Structure Mapping

A SDMX Structure Map defines a mapping between two different Data Structure Definitions, or two Dataflows. The two structures are mapped by first defining the relationship between the Dimensions, and then the relationship between the various Dimension values. For example if the REF_AREA Dimension from the source DSD maps to the CURRENCY Dimension in the target DSD, then a mapping is required to define how the codelists map, REF_AREA.UK for example could map to CURRENCY.GBP

The reverse engineer web service expects the Structure Map to already exist in the fusion Registry. This provides the basis for which two structures are mapped and how their underlying Components (Dimensions and Attributes) relate to each other. The job of the reverse engineer web service is to update the respective Codelist Maps (or Value Maps¹). The reverse engineer web service only reverse engineers values between series keys.

The service will use the mapping information to know which Components map and therefore will be able to existing value maps or Codelist maps based on the CSV input.

URL Entry Point	<code>/ws/secure/structuremap/reverseengineermerge</code>
Access	Private
HTTP Method	POST
Response Format	application/json
Available from version	v9.3
Content Type	<ol style="list-style-type: none"> multipart/form-data (if attaching file) – the attached file must be in field name of uploadFile application/text or text/plain (if submitting CSV in the body of the POST)
Success Response	<code>{ "Success" : true }</code>
Error Response	<code>{ "Error" : "Error Message" }</code>

4.15.1 Request Parameters

Parameter	Purpose	Allowed Values
Urn	The URN of the Dataflow Map or Data Structure Map to process the CSV values against	String

4.15.2 POST Content

Comma Separated series keys of source series to target series. Each series key value is colon separated.

A:UK:IND1,A:GBP:UK:001

A:FR:IND1,A:EUR:FR:001

A:US:IND2,A:USD:US:002

¹ Each mapped Component can choose how their corresponding values are mapped. A Codelist Map can be used if both source and target Components are Coded (for example the Dimension REF_AREA may get it's representation from the Codelist CL_REF_AREA). A Value Map is an alternative mapping which not only supports the mapping of coded values, but can also be used to map values which are not maintained in a codelist. For example DOLLAR could map to the symbol \$ which is not a valid Code Id. Codelist Maps are reuseable, they can be referenced from multiple Structure Maps, Value Maps can only be used by the Structure Map which owns the mapping.

5 Annex 1: HTTP Headers

The Fusion Registry makes use of the HTTP Request Headers for compression (sending responses back in gzip format), authentication, and to determine the language to send the content back in (where applicable).

Response Headers can be used for caching, and the Fusion Registry can integrate with Varnish (a HTTP Reverse Proxy for caching responses). Varnish integration is discussed in the setup guide.

This section discusses the HTTP Headers which can be used for all requests. Where a web service makes use of other Request Headers, this will be documented with the service.

5.1 Compression

The Fusion Registry will ensure responses are sent in gzip format if the client application includes the HTTP Header 'Accept-Encoding: gzip'. Gzip compression will dramatically reduce the size query responses, so it is important to set this Header to reduce network traffic, and increase performance.

5.2 Authentication

Where authentication is required, the protocol used is Basic Authentication as documented in the HTTP/1.0 specification. This is discussed in the next section.

5.3 Language

Where multilingual responses are supported, the 'Accept-Language' HTTP Header can be used to define the preferred response language. If the content is not available in the requested language, the application will default to English.

5.4 Caching

The HTTP Response headers for data queries include the VARY HTTP Header to indicate which HTTP Headers the content of the response will vary over. For example if data is requested in JSON format, and the Fusion Registry contains structures in multiple languages, then the VARY header will include 'Accept-Language' as the response dataset can differ when the user requests the same data in a different language.

6 Annex 2: Security

6.1 X509 Certificate Authentication

The Fusion Registry supports X509 certificate (p12) authentication in order to authenticate a user. The Fusion Registry set up guide discusses this in more detail.

6.2 Basic Authentication

Authentication can be performed by using HTTP Basic Authentication as specified by the HTTP/1.0 specification. It is important to note that Basic Authentication must be used with a secure connection (HTTPS) otherwise the security credentials will not be secure across the transmission.

The HTTP Headers must include 'Authorization' where the value is the Base64 encoding of the Username:Password prefixed by the text 'Basic'.

For example, the username is **root**, and the password is **password**, then the following is encoded

```
Base64 ("root:password")
```

This encoding results in the following string:

```
cm9vdDpwYXNzd29yZA==
```

The HTTP Authorization Header would then be:

```
Authorization:Basic cm9vdDpwYXNzd29yZA==
```

6.3 Reverse Proxy

The Fusion Registry ensures access is restricted to the web services which are marked as secure.

Whilst the Fusion Registry provides a mix of public and secure web services, it is possible to configure the deployed environment to only expose particular services to the public. One way of achieving this is by using a HTTP Reverse Proxy such as Apache or IIS.

By configuring a reverse proxy inbound request URLs can be mapped to specific web service entry points on the Fusion Registry, whilst other web service entry points can remain unmapped, and therefore kept private.

The Fusion Registry supports mapping internal web service URLs to a different public entry point. This mapping is taken into account when creating the WADL for both the public and private web services. The Fusion Registry Set-up Guide discusses this in more detail.

It is beyond the scope of this document to discuss deployment architecture, however it is worth making use of this guide when deciding which APIs to expose and which to secure. It is also worth noting the Graphical User Interface makes use of many of the SDMX web services.

7 Annex 3 – Metrics and Data Availability

The SDMX-JSON dataset includes support for reporting metrics and supplying information on what data is available based on a specific data query. This information can be obtained using the **includeMetrics** parameter. Metrics can be used in conjunction with **detail=structureOnly** to enable code-selection user-interfaces to be built with the knowledge of:

- what data exists
- how code selections change data availability
- how much data would be returned from a Data Query

Importantly all this information can be retrieved without having to first execute the data query. This annex details how both parameters are used, and what is returned from each.

7.1 Include Metrics

7.1.1 Structure Queries

When **includeMetrics** is set to true, structures that have related data available will be enriched with Annotations with AnnotationType set to **FR_METRIC**.

The following structures support metrics:

1. Registration
2. Provision Agreement
3. Dataflow
4. Data Structure
5. Concept Schemes (Concepts)
6. Data Provider Scheme (Data Provider)
7. Content Constraint (acting as a Pre-Defined Query)

Metrics returned will differ depending on the structure type. For **Registration, Provision Agreement, Dataflow, Content Constraint** the following metrics will be returned (if available).

Annotation Title	Annotation Type	Annotation Text
SERIES_COUNT	FR_METRIC	Number representing the number of available series
DATA_FROM	FR_METRIC	Number representing EPOCH time of the earliest observation
DATA_TO	FR_METRIC	Number representing EPOCH time of the latest observation
LAST_UPDATED	FR_METRIC	Number representing EPOCH time of the latest updated observation based on the sub-cube of data.

Note, last updated are available at the observation level if using the Fusion Store, but not other data stores may include this information based on the last registration date.

Data from and data to dates may not be available if using plugin stores.

For **Concepts, Data Providers, Data Structures** the following metrics will be returned (if available).

Annotation Title	Annotation Type	Annotation Text
DATAFLOW	FR_METRIC	Short URN of a dataflow (everything after the =) for example: IMF:BOP(1.0) This annotation may occur

		multiple times if multiple dataflows have data for the same structure
SERIES_COUNT	FR_METRIC	Number representing the total number of series using the concept across all datasets
DATASET_COUNT	FR_METRIC	Number of datasets available
LAST_UPDATED	FR_METRIC	Number representing EPOCH time of the latest updated dataset

7.1.2 Data Queries

When **includeMetrics** is set to true, the dataset will be enriched in the following ways:

1. A metrics section is added to the message

```
{
  + header: { ... },
  + metrics: { ... },
  + dataSets: [ ... ],
  + structure: { ... }
}
```

The message now contains four top level elements, with the inclusion of metrics.

```
- metrics: {
  lastUpdated: "2017-08-29T14:23:02",
  maxSeries: 27,
  actualSeries: 9,
  maxObs: 9,
  actualObs: 9
},
```

The metrics section contains when the dataset was last updated. In addition it calculates the number of series and observations that are included in the dataset, along with the theoretical maximum, should the cube be complete. For example if the Data Structure contains 3 dimensions, each with 3 possible codes, then the maximum number of possible series is 27 (3 x 3 x 3). The actual number of series may differ from this, as data may not be reported for all combination of codes. The maximum observations are calculated from the knowledge of the start date, end date, and frequency. For example for Annual data reported between 2000 and 2009, there are 10 possible observations per series, one for each year. If there are 20 actual series returned, then the maximum number of observations is 200, however the actual number may be lower than this, indicating holes in the cube (missing data).

It is important to note that in some cases the number of observations cannot be calculated. For example if the data comes from an external web service, or database, then this information is not available to the Fusion Registry. The **actualObs** will be -1 if this calculation is not possible.

As described in the next section, when **includeMetrics** is combined with **detail=structureOnly** it is possible to know what size of response will be returned without having to first get the data.

2. The structure section will also contain metrics about each dimension and code.

```

{
  id: "REF_AREA",
  name: "Reference area",
  description: "The country or geographic area",
  keyPosition: 12,
  maxObs: 27,
  role: null,
  - values: [
    - {
      actualObs: 9,
      id: "AM",
      name: "Armenia"
    },
    - {
      actualObs: 9,
      id: "DE",
      name: "Germany"
    },
  ],
}

```

Each Dimension is enriched with information about the maximum number of observations that could exist if an observation exists for every time period. And each code is enriched with the number of observations that actually exist.

3. The structure section includes codes which do not necessarily have corresponding data in the dataset.

includeMetrics returns not only metadata for data which exists, but also for metadata which remains a valid choice to add to the data query, based on the current query.

To demonstrate how this works in practice, the following data series will be used as an example:

Frequency	Reference Area	Indicator
M	UK	BIRTH_RATE
M	UK	DEATH_RATE
M	UK	LITERACY_RATE
M	FR	BIRTH_RATE
M	FR	DEATH_RATE

If the user constructs a query for UK and BIRTH_RATE and they choose to include metrics, then not only will the structure section contain the UK code, but it will also contain the FR code, even though there is no data returned for FR (the user did not ask for this data). However the presence of FR indicates to the user that this remains a valid future selection, based on their current data query. All indicators would also be in the structure section.

If the user now modifies their query to remove UK and add FR, the structure section will now change and will not contain the indicator LITERACY_RATE. This is because if the user were to modify their query to include LITERACY_RATE then there would be no data returned.

This data availability information makes it possible for a user to explore and build up a data query with the assurance that they will not end up creating a data selection that results in no data being returned.

7.2 Detail = structureOnly

This parameter, used in conjunction with metrics can be used to fully understand what data is available for a dataflow, in the context of a data query, without having to execute the data query, or bring back all the data. The response from this message includes just the header and structure, and no datasets.

```

{
  + header: { ... },
  + structure: { ... }
}

```

When adding the parameter **includeMetrics=true**, all the metrics information is included.


```

{
+ header: { ... },
+ metrics: { ... },
+ structure: { ... }
}

```

However, in addition to the code information, two additional values, **inDataset** and **parent** are added.

```

- {
  id: "SEX",
  name: "Sex",
  keyPosition: 12,
  maxObs: 27,
  role: null,
  - values: [
    - {
      actualObs: 9,
      inDataset: false,
      id: "_T",
      name: "Total"
    },
    - {
      actualObs: 9,
      inDataset: true,
      id: "F",
      name: "Female",
      parent: "_T"
    },
    - {
      actualObs: 9,
      inDataset: false,
      id: "M",
      name: "Male",
      parent: "_T"
    }
  ]
},

```

inDataset describes if the code is present in the structure because it will appear in the dataset, or because it is a future possible selection. In the above JSON only data for Female will be included in the dataset, whereas Male and Total will not appear in the dataset, but will be possible future selections.

parent is used to indicate to the user that a code is the child of another code in the same codelist, indicated by the parent field. In the above example both Male and Female are child codes of Total.

Note: If a code is the child of another code which has no data (i.e. it is only presentational), then this will be indicated by **inDataset** being set to false, and the field **actualObs** being omitted.

```

{
  inDataset: false,
  id: "EUR",
  name: "Europe"
}

```

This enables a UI to be built showing the parent code as a hierarchical node, but with no selection state, as shown in the following example:

