FUSION REGISTRY COMMUNITY EDITION
VERSION 9

Web Services API

FUSION REGISTRY COMMUNITY EDITION WEB SERVICES API

Fusion Registry: 9.8.x
Document Last Updated: October 22nd 2019

Version: 20191022

CONTENTS

Version History

Version #	Implemented By	Revision Date	Reason
20191022	Phil Lazarou	22 Oct 2019	Updated section "Query Structures via REST API"

1 Overview

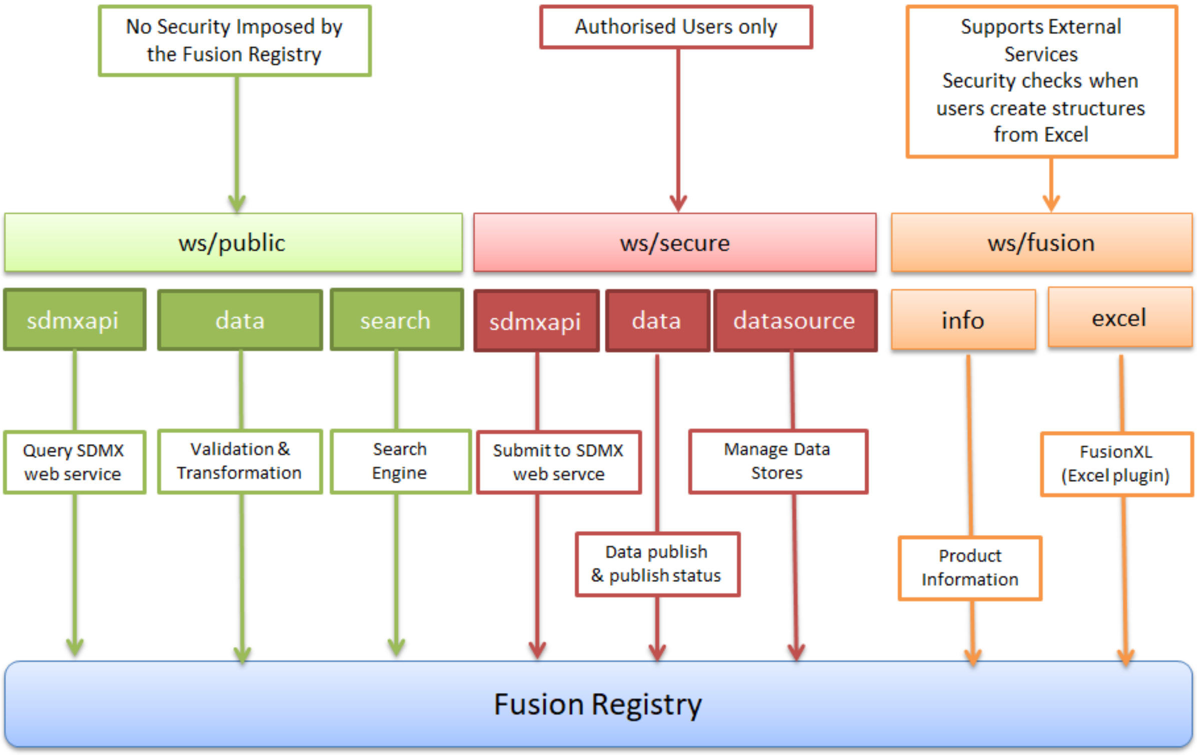
The Fusion Registry hosts a number of web services, this document does not document all the web service APIs exposed by the Fusion Registry, as most web services are made available in order to support the Web User Interface of the product. This document discusses web services which may be useful for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

1. SDMX web services for queries and submissions
2. Data Validation and Transformation web services
3. Search
4. 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 provide product information and support the 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.


No Security Imposed by the Fusion Registry
Authorised Users only
Supports External Services Security checks when users create structures from Excel
ws/public
ws/secure
ws/fusion
sdmxapi
data
search
sdmxapi
data
datasource
info
excel
Query SDMX web service
Validation & Transformation
Search Engine
Submit to SDMX web servce
Manage Data Stores
FusionXL (Excel plugin)
Data publish & publish status
Product Information
Fusion Registry

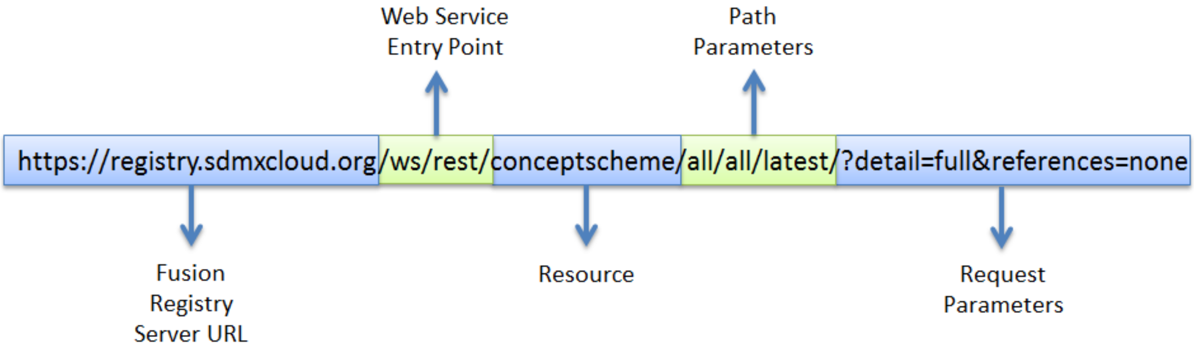
This document groups the web services into the following categories:

1. **SDMX Web Services** – these include web services to retrieve and submit structures. 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 2 - 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 3 - 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


Web Service Entry Point
Path Parameters
https://registry.sdmxcloud.org/ws/rest/conceptscheme/all/all/latest/?detail=full&references=none
Fusion Registry Server URL
Resource
Request Parameters

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 SDMX API

2.1 Submit Content (POST)

Web Service Entry Point	**/ws/secure/sdmxapi/rest**
Access	**Secure**: Admin + Agency, Data Provider Users
HTTP Method	HTTP POST
Accepts	• SDMX Structure Document (all SDMX versions) • Registry Interface Document (all SDMX versions) • SubmitStructureRequest • SubmitRegistrationRequest • QueryRegistrationRequest • Excel Document (conforming to FusionXL format)
Compression	Zipped content can be submitted
Content Type	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	Multiple formats supported. Can be specified in the *Accept* HTTP Header, or the *format* parameter of the URL request.
Error Response	SDMX-ML v2.1 Error Response Document

2.1.1 HTTP Headers

The HTTP headers can be used to specify the 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`

2.1.2 Supported Submissions

2.1.2.1 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.

2.2 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.

2.2.1 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 *Accept* HTTP Header, or the *format* parameter of the URL request.
Error Response	SDMX-ML v2.1 Error Response Document

2.2.2 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	**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`
		
		JSON Format
		`application/vnd.sdmx.json`
		
		Excel Format
		`application/vnd.xlsx`

Accept-Language	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. The corresponding locale parameter can be used to override this HTTP Header	Accept-Language : en (English) Accept-Language : fr (French) Accept-Language : * (all languages – no filter) Accept-Language : all (all languages – no filter)

2.2.3 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

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

2.2.5 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.	**full** – default. Output full response. **allstubs –** Output all the structures as stubs. **referencestubs** – Output the full query result, and any referenced structures are returned as stubs **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. *Example:* detail=allstubs detail=referencepartial
references	To define if additional structures are returned from the query. The structures can either be ones which reference, or are referenced by the structures in the query result. If the query result is for a specific item in an item scheme, then this parameter will identify the references for that item.	**none** – default. Do not output any additional structures **parents** –output structures the reference the structures matching the query **parentsandsiblings** – same as parents, but also include all the additional structures referenced by the parents **children** – the structures referenced by the structures in the query result **descendants** – children and their children (up to any level) In addition, a concrete type of resource may be used, for example: **datastructure** *Example:* references=datastructure
partial	If set to true creates partial Codelists in the response based on the Fusion Registry Content Constraints defining allowable content. The pre-requisite is that the query must be for a single constrainable structure (Provision Agreements, Dataflow, or Data Structure) and include references.	true/false **Note:** This is deprecated as of v9.2.23, use detail=referencepartial instead
format	Can be used to define the response format (as an alternative to the HTTP	sdmx (latest version) sdmx-1.0 sdmx-2.0

	Accept Header).	sdmx-2.1 sdmx-edi sdmx-edi-lenient sdmx-json xlsx *Example:* format=sdmx-edi
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. This takes priority over the Accept-Language HTTP header	Any locale *Example:* locale=fr
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 *Example:* saveAs=myDownload
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

2.2.6 Examples

2.2.6.1 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

2.2.6.2 All structures saved to a file

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

2.2.6.3 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

2.3 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.

2.3.1 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

2.3.2 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

2.3.3 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**

2.3.4 Request Parameters

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

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

2.3.5 Examples

2.3.5.1 Delete all concept schemes

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

2.3.5.2 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

2.4 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.

2.4.1 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 *Accept* HTTP Header, or the *format* parameter of the URL request.
Error Response	SDMX-ML v2.1 Error Response Document

2.4.2 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`

2.4.3 Resource

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

Resource	Purpose
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.

2.4.4 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
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**

2.4.5 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 *Example:* 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 *Example:* saveAs=myDownload

2.4.6 Examples

2.4.6.1 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

2.4.6.2 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 Fusion Registry Web Services

3.1 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*.

3.1.1 Overview

URL Entry Point	/ws/public/data/validate
Access	**Public**
HTTP Method	POST
Accepts	SDMX, XLSX (Excel Reporting Template)
Compression	Zipped data can be submitted
Content Type	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	{ "Error" : "Error Message" }

3.1.2 HTTP Headers

HTTP Header	Purpose	Allowed Values
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 $, _, -, @, \
Structure	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.

3.1.3 Validation Output

An example output for a valid dataset.

```
{
   "Prepared": "2016-05-03T16:10:31",
   "SenderId": "FusionReg",
   "DataSetId": "012e47df-c582-400b-80fd-5e99b95f5eea",
   "FileFormat": "Structure Specific (Compact) v2.1",
   "FileName": "MyFile.xml",
   "Status" : "Complete"
   "Datasets" : [
   {
     "DSD": "urn:sdmx:org.sdmx.infomodel.datastructure.DataStructure=WB:WDI(1.0)",
     "Dataflow": "urn:sdmx:org.sdmx.infomodel.datastructure.Dataflow=WB:EDUCAT(1.0)",
     "KeysCount": 382,
     "ObsCount": 7136,
     "GroupsCount": 0,
     "Errors": false
   }
  ]
}
```

An example output for an invalid dataset.

```
{
   "Prepared": "2016-05-04T12:24:40",
   "SenderId": "MetadataTechnology",
   "DataSetId": "1aeb4cad-9262-4201-b71d-1060cd1d2aab",
   "FileFormat": "Structure Specific (Compact) v2.1",
   "FileName": "MyFile.xml",
   "Status" : "Complete"
   "Datasets" : [
      {
        "DSD": "urn:sdmx:org.sdmx.infomodel.datastructure.DataStructure=WB:WDI(1.0)",
           "Dataflow":
        "urn:sdmx:org.sdmx.infomodel.datastructure.Dataflow=WB:WDI_EDUCATION(1.0)",
           "KeysCount": 1,
           "ObsCount": 9,
           "GroupsCount": 0,
           "Errors": true,
           "ValidationReport":
           [
             {
               "Type": "Semantic",
               "Errors":          [
               ["Error  reading  dataset  '1':  Element  'StructureSpecificData/Header/Test'
        expected to
                       come after the 'ID' element"],
               ["Error  reading  dataset  '1':  Missing  Mandatory  Header  Elements  :  [ID,
        Sender]"],
               ["Error reading dataset '1': Element 'StructureSpecificData/Header/Receiver'
                       expected to come after the 'Sender' element"]
               ]
             },
             {
               "Type": "Duplicate",
               "Errors": ["Contradictory observation value reported for series
                           'A:SE_PRM_GINT_FE_ZS:AFG'.  Observation for time period '1985'
                           reports both '19.09035' and '21'  "]
             }
           ]
        }
      }
   ]
```

An example output for a server error (unable to process request):

```
{"Error": "Unrecognised file format, contents of file are: this is a bad format"}
```

3.2 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.

3.2.1 Overview – List Templates

URL Entry Point	/ws/secure/reporttemplate/templates
Access	Secure Data Provider, Agency or Admin only
HTTP Method	GET
Response Format	application/json
Error Response	{ "Error" : "Error Message" }

3.2.1.1 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"]
   }
]
```

3.2.2 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)

3.2.3 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**

3.2.4 Query Parameters

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

<table>
<tr><th>Parameter</th><th>Purpose</th><th>Allowed Values</th></tr>
<tr><td><b>reportPeriod</b></td><td>This query parameter is used to specify the report period to generate the report for.

This is a required parameter.</td><td rowspan="2">Single time period (not a time range). Conforms to the ISO-8601 standard.

Supported time periods include
<table>
<tr><th>Period</th><th>Example</th></tr>
<tr><td><b>Annual</b></td><td>2009</td></tr>
<tr><td><b>Semester</b></td><td>2009-S1</td></tr>
<tr><td><b>Trimester</b></td><td>2009-T1</td></tr>
<tr><td><b>Quarterly</b></td><td>2009-Q1</td></tr>
<tr><td><b>Monthly</b></td><td>2009-01</td></tr>
<tr><td><b>Weekly</b></td><td>2009-W1</td></tr>
<tr><td><b>Daily</b></td><td>2009-01-31</td></tr>
<tr><td><b>Date Time</b></td><td>2009--01T20:22:00</td></tr>
</table></td></tr>
<tr><td><b>reportPeriodTo</b>
(Since v9.3)</td><td>If the template is to be created for a range of time periods this is achieved using the reportPeriodTo parameter</td></tr>
<tr><td><b>Freq</b>
(Since v9.3)</td><td>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</td><td>
<table>
<tr><th>Period</th><th>Example</th></tr>
<tr><td><b>Annual</b></td><td>A</td></tr>
<tr><td><b>Semester</b></td><td>S</td></tr>
<tr><td><b>Trimester</b></td><td>T</td></tr>
<tr><td><b>Quarterly</b></td><td>Q</td></tr>
<tr><td><b>Weekly</b></td><td>W</td></tr>
<tr><td><b>Monthly</b></td><td>M</td></tr>
<tr><td><b>Daily</b></td><td>D</td></tr>
</table></td></tr>
</table>

3.2.4.1 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

3.3 Data Transformation

3.3.1 Overview

URL Entry Point	/ws/public/data/transform
Access	**Public**
HTTP Method	POST
Accepts	SDMX XLSX (Excel Reporting Template)
Compression	Zipped data can be submitted.
Content Type	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	Defaults to SDMX Structure Specific v2.1
Error Response	{ "Error" : "Error Message" }

3.3.2 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
```

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
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 $, _, -, @, \
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	Valid SDMX URN for Provision Agreement, Dataflow, or Data Structure Definition.

	provided this value will override the value in the dataset (if present).	
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
Map-Structure	An optional parameter to inform the Fusion Registry to transform the structure of the dataset to conform to another Data Structure Definition. The value provided must 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.	Valid SDMX URN for Dataflow or Data Structure Definition.

3.4 Structure Search Engine

Permits 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

3.4.1 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

3.4.2 Server Response

With auto=true

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

With auto=false (or no auto parameter passed)

```
{
  time: 146,
  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 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.

4.1 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.

4.2 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.

4.3 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.

4.4 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.

5 Annex 2: Security

5.1 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==
```

5.2 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.