SDMX 3.0 RESTful web services cheat sheet, v2.0.0

Structure queries: `https://host/structure/type/agency/id/version/item?detail&references`

		Default	Multi
type	The type of structure: `datastructure`, `metadatastructure`, `categoryscheme`, `conceptscheme`, `codelist`, `hierarchy`, `hierarchyassociation`, `valuelist`, `organisationscheme`, `agencyscheme`, `dataproviderscheme`, `dataconsumerscheme`, `organisationunitscheme`, `dataflow`, `metadataflow`, `reportingtaxonomy`, `provisionagreement`, `structuremap`, `representationmap`, `conceptschememap`, `categoryschememap`, `organisationschememap`, `process`, `categorisation`, `dataconstraint`, `metadataconstraint`, `structure`, `transformationscheme`, `rulesetscheme`, `userdefinedoperatorscheme`, `customtypescheme`, `namepersonalisationscheme`, `vtlmappingscheme`	`*`	N
agency	Agency maintaining the artefact (e.g.: `SDMX`)	`*`	Y
artefact	Artefact ID (e.g.: `CL_FREQ`)	`*`	Y
version	Artefact version (e.g.: `1.0.0`)	`~`	Y
item	ID of the item to be returned.	`*`	Y
detail	Amount of information: `allstubs`, `referencestubs`, `allcompletestubs`, `referencecompletestubs`, `referencepartial`, `raw`, `partialraw`, `full`	`full`	N
references	References to be returned: `none`, `parents`, `parentsandsiblings`, `ancestors`, `children`, `descendants`, `all`, a resource type	`none`	Y

Data queries: `https://host/data/context/agency/id/version/key?`
`c&updatedAfter&firstNObservations&lastNObservations&detail&dimensionAtObservation&includeHistory`

		Default	Multi
context	The context of data retrieval: `datastructure`, `dataflow`, `provisionagreement`	`*`	Y
agency	Agency maintaining the artefact (e.g.: `BIS`)	`*`	Y
id	Context ID (e.g.: `CBS`)	`*`	Y
version	Artefact version (e.g.: `1.0.0`)	`*`	Y
key	Key of the series to be returned (e.g: `D.NOK.EUR.SP00.A`). Wildcarding supported (e.g: `D.*.EUR.SP00.A`).	`*`	Y
c	Filter by component values (e.g. `c[REF_AREA]=CH`) with support for operators (e.g. `c[TIME_PERIOD]=ge:2000`)		Y
updatedAfter	Retrieves what changed since supplied timestamp. Must be percent-encoded (e.g.: `2009-05-15T14%3A15%3A00%2B01%3A00`)		N
firstNObservations	Maximum number of observations starting from the first observation		N
lastNObservations	Maximum number of observations counting back from the most recent observation		N
dimensionAtObservation	Id of the dimension at the observation level		N
detail	Desired amount of information: `full`, `dataonly`, `serieskeysonly`, `nodata`	`full`	N
includeHistory	Whether to return vintages	`false`	N

Legend: Mandatory path parameter / Optional path parameter / Query string parameter (all optional)

Data validity queries: `https://host/``schema``/``context``/``agency``/``id``/``version``?``dimensionAtObservation&explicitMeasure`

		Default	Multi
context	The context of data retrieval: `datastructure`, `dataflow`, `provisionagreement`		N
agency	Agency maintaining the artefact (e.g.: `ECB`)		N
artefact	Artefact ID (e.g.: `EXR`)		N
version	Artefact version (e.g.: `1.0.0`)	~	N
dimensionAtObservation	Id of the dimension at the observation level		N

Data availability queries: `https://host/``availability``/``context``/``agency``/``id``/``version``/``key``/``componentId``?``c&updatedAfter&references&mode`

		Default	Multi
context	The context of data retrieval: `datastructure`, `dataflow`, `provisionagreement`	*	N
agency	Agency maintaining the artefact (e.g.: `ECB`)	*	Y
artefact	Artefact ID (e.g.: `EXR`)	*	Y
version	Artefact version (e.g.: `1.0.0`)	*	Y
key	Key of the series to be returned (e.g: `D.NOK.EUR.SP00.A`). Wildcarding supported (e.g: `D.*.EUR.SP00.A`).	*	Y
componentId	The id of the dimension for which to obtain availability information	*	Y
c	Filter by component values (e.g. `c[REF_AREA]=CH`) with support for operators (e.g. `c[TIME_PERIOD]=ge:2000`)		Y
updatedAfter	Retrieves what changed since supplied timestamp. Must be percent-encoded (e.g.: `2009-05-15T14%3A15%3A00%2B01%3A00`)		N
references	References to be returned: `codelist, datastructure, conceptscheme, dataflow, dataproviderscheme, none, all`	none	Y
mode	The resolution mode: `exact, available`	exact	N

Metadata queries:
`https://host/``metadata``/``metadataset``/``provider``/``id``/``version``?``detail`` or`
`https://host/``metadata``/``metadataflow``/``agency``/``id``/``version``/``provider``?``detail`` or`
`https://host/``metadata``/``structure``/``type``/``agency``/``id``/``version``/``provider``?``detail`

		Default	Multi
provider	The provider of metadata	*	Y
agency	Agency maintaining the artefact (e.g.: `ECB`)	*	Y
id	Artefact ID (e.g.: `EXR`)	*	Y
version	Artefact version (e.g.: `1.0.0`)	*	Y
type	The structure type (cf. Structure queries)	*	N
detail	The amount of information to be returned: `full`, `allstubs`	full	N

Useful HTTP headers

`If-Modified-Since`	Get the data only if something has changed (`If-None-Match` is an alternative)
`Accept`	Select the desired format
`Accept-Encoding`	Compress the response
`Accept-Language`	Select the desired language

Supported formats

SDMX-ML Data	`application/vnd.sdmx.data+xml;version=3.0.0`
SDMX-JSON Data	`application/vnd.sdmx.data+json;version=2.0.0`
SDMX-CSV Data	`application/vnd.sdmx.data+csv;version=1.0.0`
SDMX-ML Structure	`application/vnd.sdmx.structure+xml;version=3.0.0`
SDMX-JSON Structure	`application/vnd.sdmx.structure+json;version=2.0.0`
SDMX-ML Metadata	`application/vnd.sdmx.metadata+xml;version=2.0.0`
SDMX-JSON Metadata	`application/vnd.sdmx.metadata+json;version=2.0.0`
SDMX-CSV Metadata	`application/vnd.sdmx.metadata+csv;version=2.0.0`

Period formats

Daily/Business	`YYYY-MM-DD`
Weekly	`YYYY-W[01-53]`
Monthly	`YYYY-MM`
Quarterly	`YYYY-Q[1-4]`
Semi-annual	`YYYY-S[1-2]`
Annual	`YYYY`

Useful response codes

200 (OK)	Life is good
204 (No content)	No results. Mostly harmless, especially if you used `updatedAfter`
304 (No change)	No change since the timestamp supplied in the `If-Modified-Since` header
400 (Syntax error)	Your query checking you must
401 (Login needed)	Only available on a need to know basis
406 (Invalid format)	Not a supported format. No worries, SDMX is all you need
413 (Results too large)	You know greed is a sin, right?
414 (URI too long)	Write short sentences
422 (Semantic error)	The syntax of the query is OK but it makes no sense anyway. Tolerated only on April Fools
500 (Server error)	Someone somewhere is having a bad day
501 (Not implemented)	Feature coming soon in a web service near you
503 (Unavailable)	Try again later.

Special characters

*	All
+	Latest stable version
~	Latest version (whether stable, draft or unversioned)
,	Or
.	Subpart operator
+	Multivalues attribute

Operators for component-based queries

eq	Equal
ne	Not equal
lt	Less than
le	Less than or equal to
gt	Greater than
ge	Greater than or equal to
co	Contains
nc	Does not contain
sw	Starts with
ew	Ends with
nd	And
or	Or

Tips and tricks

- **Trust but verify!** Check the identity of the web service by connecting over https.
- **Know your mime type!** Specify the output format you want, using the `HTTP Accept header`. Else, the web service will determine the format and this choice may change over time (for example, when new versions of SDMX are released).
- **Laziness is a virtue!** Don't lose time, resources and bandwidth processing the response if nothing has changed! Use the `updatedAfter` parameter if you can handle deltas, or the `If-Modified-Since header` otherwise.
- **Travel light!** Compress the response, using the `HTTP Accept-Encoding header`.
- **The answer is 42.** Or not. Check the status code of the response. It's not always 200 (or 42).
- **Read The Fine Manual.** Check the SDMX RESTful specification (Section 7): http://sdmx.org/?page_id=10
- **Contact us!** You think a killer feature is missing? Drop us a line at twg@sdmx.org.