Last updated: 22 July 2024

OECD Data API documentation

About the API

The OECD provides programmatic access to OECD data for OECD countries and selected non-member economies through a RESTful application programming interface (API) based on the SDMX standard. The APIs allow developers to easily query the OECD data in several ways to create innovative software applications which use dynamically updated OECD data.

Note that the RESTful SDMX API standard has different versions that are partly supported by the OECD API instance.

Terms and conditions

OECD data and the API service are offered subject to your acceptance of OECD Terms and Conditions.

What is the SDMX standard?

OECD and other international organisations have defined an ISO standard format for describing and transmitting statistical data, which can be used in internet-based electronic communications with users and third parties. The SDMX API standard supports JSON, XML and CSV formats. Please see details below.

In SDMX the measurement of a phenomenon (e.g. a population count) is known as an “observation”. Observations are described and uniquely identified by a combination of “dimension” values (e.g. a country and a year). “Attributes” allow further adding useful information but do not help identifying statistical data (e.g. observation status). Observations of a same kind – identifiable by the same dimensions – are grouped into a “dataset”. The SDMX-ML and SDMX-JSON formats also allows for an optional intermediate grouping of the observations for all values of one of the dimensions, called “series”. Specifically, the grouping of the observations for all available time periods is a so-called “time series” (e.g. the population counts for all years for a specific country). Similarly, groups can be made with any dimension. Alternatively, no grouping results in a flat list of all observations in the dataset. Descriptive information on the dataset, dimensions and attributes is called “structural metadata”. It is returned as response to structure queries and also within the “structure” part of the SDMX-JSON response to data queries.

Specificities of the SDMX-JSON format

The structural metadata for the actually returned observations are included in the SDMX-JSON response message. If possible, in order to minimise repetition, each dimension/attribute is specified

at the highest possible grouping level. Dimensions and attributes specified on dataset and series level have the same values for all observations throughout the dataset or series respectively.

To uniquely identify observations in the SDMX-JSON message, the indexes of the corresponding dimension values as defined in the "structure" part of the message at series and observation level are concatenated into the series' or observation's property name. Here, the indexes are ordered in the pre-defined order of dimensions as defined in the "structure" part of the message and separated between each other by a colon character.

The concrete values of attributes at series and observation level are also returned through their index as defined in the "structure" part of the message.

Specificities of the SDMX-CSV format

This RFC 4180 compatible format is limited to transmitting data in a flattened table.

In order to ensure the identifiability of the data contained in the message, the header row contains the column headers. After this header row, each following row contains the information (dimension values, observation value, attribute values) related to one specific observation or to one or more attribute values attached to the values of a subset of dimensions. Advanced queries for database synchronisation purposes allow also retrieving information about data that has been previously deleted. In such messages, a row can also concern several observations if dimension values are omitted.

Even if csv stands for 'comma-separated values', SDMX-CSV allows using a localised field separator, e.g., the semi-colon ';', depending on the locale of the client (as indicated in the http Accept-Language header). Note that the separator used in a message can be determined by retrieving the character that follows the fixed first column header term STRUCTURE or DATAFLOW (which may be extended by a squared bracket term).

Syntax for querying data or reference metadata

To create a data or reference metadata query, the following parameters must be supplied in an URL in the following format: an agency identifier, a dataflow identifier, a dataflow version, a list of dimension values when using a filter or and some optional additional parameters:

SDMX API version 1

https://sdmx.oecd.org/public/rest/data/<agency identifier>,<dataflow identifier>,<dataflow version>/<filter expression>[?<optional parameters>]

SDMX API version 2

https://sdmx.oecd.org/public/rest/v2/data/dataflow/<agency identifier>/<dataflow identifier>/<dataflow version>/<filter expression>[?<optional parameters>]

Notes:

- The *dataflow* in SDMX is a slice or partial view of a potentially bigger dataset or data cube. Clients can still understand it as a self-contained classical statistical dataset. It is fully identified by the triplet: agency identifier, dataflow identifier and dataflow version.
- **Reference metadata** can only be queried using the SDMX API version 2.

Parameter	Use
agency identifier	The identifier of the agency owning the dataflow to be queried.
dataflow identifier	The identifier of the dataflow to be queried.
dataflow version	The version of the structural definition of the dataflow to be queried. If left empty (for SDMX API version 1) or replaced by '+' (for SDMX API version 2), then the currently **latest** dataflow version is used. While this often allows getting newer data, be aware that new dataflow versions may contain non-backward-compatible structural changes.
filter expression	**SDMX API version 1** Using their identifiers, unless they are uncoded, the list of desired dimension values, except Time Period, to be included in the response. To get unfiltered content, use the "all" keyword. Dimensions should be separated by a dot (".") and for each dimension its values should be separated by the plus sign ("+"). If for a particular dimension, no dimension value identifiers are specified, then all available values of this dimension will be returned. Examples: • 'ABC+DEF..A': three dimensions without Time Period, with 'ABC' and 'DEF' as values for the first dimension, all available values for the second dimension and 'A' as value for the third dimension • 'all': all available values for all dimensions To get the list of dimensions and dimension values, use a structure query as explained below. **SDMX API version 2** Same than version 1 but: To get unfiltered content, use the star character '*'. Each dimension can only have one value. If for a particular dimension, all available values of this dimension are to be returned then wildcard the dimension using the star character '*'. Examples: • 'ABC.*.A': three dimensions without Time Period, with 'ABC' as value for the first dimension, all available values for the second dimension and 'A' as value for the third dimension • '*': all available values for all dimensions

Optional Parameter	Use
startPeriod	**Only SDMX API version 1** The start time period for which results should be supplied (inclusive). If not specified, data returned from the beginning. The value can be expressed using dateTime, Gregorian time periods or SDMX reporting periods. Examples: • '2015' • '2015-A1' • '2015-S1' • '2015-Q1' • '2015-M01' • '2015-01' • '2015-01-01' • '2015-01-01T00:00:00'
endPeriod	**Only SDMX API version 1** The end time period for which results should be supplied (inclusive). If not specified, data is returned until the currently last time period. For available value types, see *startPeriod*.
c[TIME_PERIOD]	**Only SDMX API version 2** Filter data by Time Period. A list of 2 time periods separated by the plus character '+' and the comparison operators 'ge' (greater or equal than) and 'le'' (lower or equal than) can be used with dateTime, Gregorian time periods or SDMX reporting periods to set the start and end periods. Example: • 'ge:2018+le:2024'
lastNObservations	Integer specifying the maximum number of observations to be returned for each of the matching time series, counting back from the most recent observation. This parameter expects a positive integer.
dimensionAtObservation	The identifier of the dimension to be presented at the observation level for the purpose of grouping observations into 'series', or 'AllDimensions' for a flat representation of the observations. If this parameter is not set, then the default value order is: • 'TIME_PERIOD: grouping of observations into time series • 'AllDimensions': if the data has no Time Period dimension

Optional Parameter	Use
detail	**Only SDMX API version 1** This attribute specifies the desired amount of information to be returned. Possible values: • 'Full': all data and documentation, including annotations (default) • 'DataOnly': attributes – and therefore groups – will be excluded • 'SeriesKeysOnly': only the series elements and the dimensions that make up the series keys • 'NoData': returns the groups and series, including attributes and annotations, without observations
attributes	**Only SDMX API version 2** This parameter specifies which attribute values are to be returned. Possible options are: • 'all': the values of all normal attributes (defined in the data structure) and of all reference metadata attributes (defined in the metadata structure) are returned • 'none': the values of normal attributes and of reference metadata attributes are not returned • 'dsd': only the values of all normal attributes are returned • 'msd': only the values of all reference metadata attributes are returned
measures	**Only SDMX API version 2** This parameter specifies if the observation values are to be returned. Possible options are: • 'all': the observation values are returned • 'none': the observation values are not returned
updatedAfter	**Only for SDMX-CSV v2 (see below)** If this parameter is used, the returned response only includes the observations inserted, updated or deleted since that point in time. The value can be expressed using dateTime including the client's time zone: • '2015-12-31T23:59:59.9999-01:00' We strongly recommend using this parameter to reduce the amount of data to be transferred in scenarios of frequent database synchronisations.

For more information about SDMX API version 1, see here.

For more information about SDMX API version 2, see here.

Examples:

SDMX API version 1:

Observations and their normal attributes
https://sdmx.oecd.org/public/rest/data/OECD.ENV.EPI,DSD_ECH@EXT_DROUGHT,1.0/AFG+BFA.A.ED_CROP_IND.....?startPeriod=1981&endPeriod=2021&dimensionAtObservation=AllDimensions&updatedAfter=2015-01-01T00:00:00.000-01:00

SDMX API version 2:

Observations and their normal attributes
https://sdmx.oecd.org/public/rest/v2/data/dataflow/OECD.ENV.EPI/DSD_ECH@EXT_DROUGHT/1.0/AFG.A.ED_CROP_IND.*.*.*.*.*?c[TIME_PERIOD]=ge:1981+le:2021&attributes=dsd&measures=all&updatedAfter=2015-01-01T00:00:00.000-01:00

Observations and their normal attributes, latest dataflow version
https://sdmx.oecd.org/public/rest/v2/data/dataflow/OECD.ENV.EPI/DSD_ECH@EXT_DROUGHT/+/AUS.A.ED_CROP_ANOM.*.*.*.*.*?c[TIME_PERIOD]=ge:2018+le:2021&attributes=dsd&measures=all&updatedAfter=2015-01-01T00:00:00.000-01:00

Reference metadata attributes
https://sdmx.oecd.org/public/rest/v2/data/dataflow/OECD.ENV.EPI/DSD_ECH@EXT_DROUGHT/1.0/AFG.A.ED_CROP_IND.*.*.*.*.*?c[TIME_PERIOD]=ge:1981+le:2021&attributes=msd&measures=none&updatedAfter=2015-01-01T00:00:00.000-01:00

Syntax for querying data structures

To create a data structure query, a dataset identifier and agency name must be supplied in an URL in the following format:

SDMX API version 1

https://sdmx.oecd.org/public/rest/dataflow/<agency identifier>/<dataflow identifier>/<version number>**?references=all&detail=referencepartial**

SDMX API version 2

https://sdmx.oecd.org/public/rest/v2/structure/dataflow/<agency identifier>/<dataflow identifier>/<version number>**?references=all&detail=referencepartial**

For the definition of these parameters please see the above section on the syntax for querying data.

For more information about SDMX structure API version 1, see here.

For more information about SDMX structure API version 2, see here.

Examples:

SDMX API version 1:

https://sdmx.oecd.org/public/rest/dataflow/OECD.ENV.EPI/DSD_ECH@EXT_DROUGHT/1.0?references=all&detail=referencepartial

SDMX API version 2:

https://sdmx.oecd.org/public/rest/v2/structure/dataflow/OECD.ENV.EPI/DSD_ECH@EXT_DROUGHT/1.0?references=all&detail=referencepartial

API query builder

Data and structure queries can be generated using the **Developer API** feature of the OECD Data Explorer.

Requesting specific response formats: XML, JSON or CSV

The wished response format (SDMX-ML, SDMX-JSON or SDMX-CSV), content-languages and compression settings are usually provided to the API through HTTP content negotiation.

Selection of the Appropriate Representation

Use one of the following values for the 'Accept' header for **data** and **reference metadata** queries:

- SDMX-ML v2.1 generic data format (obsolete): 'application/vnd.sdmx.genericdata+xml; charset=utf-8; version=2.1'
- SDMX-ML v2.1 structure-specific data format: 'application/vnd.sdmx.structurespecificdata+xml; charset=utf-8; version=2.1'
- SDMX-ML v3 structure-specific data format (experimental): 'application/vnd.sdmx.structurespecificdata+xml; charset=utf-8; version=3.0'
- SDMX-JSON v1: 'application/vnd.sdmx.data+json; charset=utf-8; version=1.0'
- SDMX-JSON v2: 'application/vnd.sdmx.data+json; charset=utf-8; version=2'
- SDMX-CSV v1: 'application/vnd.sdmx.data+csv; charset=utf-8'
- SDMX-CSV v2: 'application/vnd.sdmx.data+csv; charset=utf-8; version=2'

For SDMX-CSV, optionally add the settings:

- '; labels=both' to include the names of objects inside the response in addition to their identifiers.
- 'timeformat= normalized' to obtain a pivotable time period format

Alternatively, it is possible to use the following non-SDMX standard 'format' URL parameter:

- SDMX-ML v2.1 generic data format (obsolete): 'genericdata'
- SDMX-ML v2.1 structure-specific data format: 'structurespecificdata'
- SDMX-JSON v2: 'jsondata'
- SDMX-CSV v1: 'csv'
- SDMX-CSV v1 as attached file: 'csvfile'

- SDMX-CSV v1 as attached file including the names of objects in addition to their identifiers: 'csvfilewithlabels'

Note: **Reference metadata** are only supported for SDMX-JSON v2 and SDMX-CSV v2.

Use one of the following values for the 'Accept' header for **structure** queries:

- SDMX-ML v2.0: 'application/vnd.sdmx.structure+xml; charset=utf-8; version=2.0'
- SDMX-ML v2.1: 'application/vnd.sdmx.structure+xml; charset=utf-8; version=2.1'
- SDMX-JSON v1: 'application/vnd.sdmx.structure+json; charset=utf-8; version=1.0'

Optionally add the setting '; urn=true' to include the URNs of each structure inside the response.

Alternatively, it is possible to use the following non-SDMX standard 'format' URL parameter:

- SDMX-ML v2.0 and v2.1: 'structure'

It is also possible to use the following non-SDMX standard 'formatVersion' URL parameter:

- SDMX-ML v2.0: '2.0'
- SDMX-ML v2.1: '2.1'

Selection of the Appropriate language

The 'Accept-Language' header is used to indicate the language preferences of the client. Multiple values, along with their respective weights, are possible. For example:

Accept-Language: ru, en-gb;q=0.8, en;q=0.7

Enabling data compression

Standard compression methods can be enabled using the appropriate 'Accept-Encoding' header. We strongly recommend using this feature systematically to minimise the usage of the internet band width and increase the download speed. For example:

Accept-Encoding: gzip, deflate, br

Additional documentation & support

The following resources describe the SDMX standard in more detail:

- .Stat SDMX RESTful web service cheat sheet
- SDMX RESTful API specifications
- SDMX-ML specifications
- SDMX-JSON specifications
- SDMX-CSV specifications

These resources provide details for querying structures, data and additional information (referential metadata), as well as for error codes and more.
Note, however, that only the syntax listed in this document is guaranteed to be supported.

For support questions, contact us at OECDdotStat@oecd.org.

To report persistent issues or propose technical enhancements based on the SDMX standard please create tickets in https://gitlab.com/sis-cc/.stat-suite/dotstatsuite-core-sdmxri-nsi-ws/-/issues/.

API showcase

The OECD Data Explorer powered by the .Stat Suite is sourced by this API.

Upgrading your queries from the legacy OECD.Stat APIs

Please see here for documentation about how to upgrade your queries from the legacy OECD.Stat APIs to the new OECD Data API.