Statistical Data and Metadata eXchange

SDMX CANDIDATE TECHNICAL STANDARD

SDMX-JSON DATA MESSAGE: SYNTAX AND DOCUMENTATION

Version 0.8.1

March 2015

Status of this Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document.

This is a SDMX Candidate Technical Standard for SDMX-JSON Data Message. It is made available for review by the SDMX user community and the public.

The public review period for this document extends until June 30 2015 in order to allow time for implementation. Please send your comments to the SDMX Technical Working Group (SDMX-TWG) twg@sdmx.org. Each email message should contain only one comment. All feedback is welcome.

Publication as a Candidate Technical Standard does not imply endorsement by the SDMX Sponsors. This is a draft document and may be updated, replaced or obsoleted by other documents at any time.

This document has been produced by the SDMX-TWG. The authors of this document are the members of the SDMX-TWG. Different parts of this specification have different editors.

Contents

1 Introduction

Let's first start with a brief introduction of the SDMX information model.

In order to make sense of some statistical data, we need to know the concepts associated with them. For example, on its own the figure 1.2953 is pretty meaningless, but if we know that this is an exchange rate for the US dollar against the euro on 23 November 2006, it starts making more sense.

There are two types of concepts: dimensions and attributes. Dimensions, when combined, allow to uniquely identify statistical data. Attributes on the other hand do not help identifying statistical data, but they add useful information (like the unit of measure or the number of decimals). Dimensions and attributes are known as "components".

The measurement of some phenomenon (e.g. the figure 1.2953 mentioned above) is known as an "observation" in SDMX. Observations are grouped together into a "data set". However, there can also be an intermediate grouping. For example, all exchange rates for the US dollar against the euro can be measured on a daily basis and these measures can then be grouped together, in a so-called "time series". Similarly, you can group a collection of observations made at the same point in time, in a "cross-section" (for example, the values of the US dollar, the Japanese yen and the Swiss franc against the euro at a particular date). Of course, these intermediate groupings are entirely optional and you may simply decide to have a flat list of observations in your data set.

The SDMX information model is much richer than this limited introduction, however the above should be sufficient to understand the sdmx-json format. For additional information, please refer to the SDMX documentation.

Samples, tools and other SDMX-JSON resources are available in the public Github repository https://github.com/sdmx-twg/sdmx-prototype-json/tree/master/draft-sdmx-json.

Before we start, let's clarify a few more things about this guide:

- New fields may be introduced in later versions. Therefore consuming applications should tolerate the addition of new fields with ease.
- The ordering of fields in objects is undefined. The fields may appear in any order and consuming applications should not rely on any specific ordering. It is safe to consider a nulled field and the absence of a field as the same thing.
- Not all fields appear in all contexts. For example response with error messages may not contain fields for data, dimensions and attributes.

2 Field Guide to SDMX-JSON Objects

2.1 Message

Message is the top level object and it contains the data as well as the metadata needed to interpret those data. Example:

```
{
  "header": {
      # header fields #
  },
  "structure": {
      # structure objects #
  },
  "dataSets": [
      # data set objects #
  ],
  "errors": [
      # Error messages #
  ]
}
```

2.1.1 header

Object nullable. Header contains basic technical information about the message, such as when it was prepared and how has sent it. Example:

```
"header": {
  "id": "b1804c51-1ee3-45a9-bb75-795cd4e06489",
  "prepared": "2012-05-04T03:30:00"
}
```

2.1.2 structure

Object nullable. Structure contains the information needed to interpret the data available in the message, such as the list of concepts used. Example:

```
"structure": {
    "uri": "http://sdw-ws.ecb.europa.eu/dataflow/ECB/EXR/1.0",
    "dimensions": {
        # dimensions object #
    },
    "attributes": {
        # attributes object #
    }
    "annotations": [
        # annotation objects #
    ]
}
```

2.1.3 dataSets

Array nullable. *DataSets* field is an array of *DataSet* objects. That's where the data (i.e.: the observations) will be. Example:

```
"dataSets": [
  {
    "action": "Information",
    "observations": {
        # observation objects #
    }
  }
]
```

In typical cases, the file will contain only one data set. However, in some cases, such as when retrieving, from an SDMX 2.1 web service, what has changed in the data source since in particular point in time, the web service might return more than one data set.

2.1.4 errors

Array nullable. RESTful web services indicates errors using the HTTP status codes. In addition, whenever appropriate, the error messages can also be returned using this error field. Error is an array of error messages. Example:

```
"errors": [
  {
    "code": 150,
    "message": "Invalid number of dimensions in the key parameter"
  }
]
```

2.1.4.1 code

number. Provides a code number for the error message. Code numbers are defined in the SDMX 2.1 Web Services Guidelines. Example:

```
"code": 150
```

2.1.4.2 message

string. Provides the error message. Example:

```
"message": "Invalid number of dimensions in the key parameter"
```

2.2 header

Header contains basic information about the message, such as when it was prepared and who has sent it. Example:

```
"header": {
  "id": "b1804c51-1ee3-45a9-bb75-795cd4e06489",
  "prepared": "2013-01-03T12:54:12",
  "sender: {
    "id": "SDMX"
  }
}
```

2.2.1 id

String. Unique string that identifies the message for further references. Example:

```
"id": "TEC00034"
```

2.2.2 test

Boolean nullable. Indicates whether the message is for test purposes or not. False for normal messages. Example:

```
"test": false
```

2.2.3 prepared

String. A timestamp indicating when the message was prepared. Values must follow the ISO 8601 syntax for combined dates and times, including time zone. Example:

```
"prepared": "2012-05-04T03:30:00Z"
```

2.2.4 sender

Object. Information about the party that is transmitting the message. Sender contains the following fields:

- id - *String.* A unique identifier of the party.
- name - *String nullable.* A human-readable name of the sender.
- contact - *Array nullable.* A collection of contact details.

Example:

```
"sender": {
  "id": "ECB",
  "name": "European Central Bank"
  "contact": [
    # contact details #
  ]
}
```

2.2.4.1 contact

Array nullable. Information on how the party can be contacted. Each object in the collection may contain the following field:

- name - *String.* The contact's name.
- department - *String nullable.* The organisational structure for the contact.
- role - *String nullable.* The responsibility of the contact.
- telephone - *Array nullable.* An array of telephone numbers for the contact.
- fax - *Array nullable.* An array of fax numbers for the contact person.
- uri - *Array nullable.* An array of uris. Each uri holds an information URL for the contact.
- email - *Array nullable.* An array of email addresses for the contact person.

Example:

```
"contact": [
    {
        "name": "Statistics hotline",
        "email": [ "statistics@xyz.org" ]
    }
]
```

2.2.5 receiver

Object nullable. Information about the party that is receiving the message. This can be useful if the WS requires authentication. Receiver contains the same fields as sender (see above):

Example:

```
"receiver": {
  "id": "SDMX"
}
```

2.2.6 request

Object nullable. Information about the client request.

Example:

```
"request": {
  "uri": "https://registry.sdmx.org/ws/rest/data/EXR/D.USD.EUR.SP00.A"
}
```

2.2.6.1 uri

String nullable. SDMX RESTful API URL requested by the client. Example:

```
"uri": "https://registry.sdmx.org/ws/rest/data/EXR/D.USD.EUR.SP00.A"
```

2.3 structure

Object nullable. Provides the structural metadata necessary to interpret the data contained in the message. It tells you which are the components (dimensions and attributes) used in the message and also describes to which level in the hierarchy (data set, series, observations) these components are attached.

Example:

```
"structure": {
    "uri": "http://sdw-ws.ecb.europa.eu/dataflow/ECB/EXR/1.0",
    "dimensions": {
        # dimensions object #
    },
    "attributes": {
        # attributes object #
    },
    "annotations": {
        # annotations object #
    }
}
```

2.3.1 uri

String nullable. A link to an SDMX 2.1 web service resource where additional information regarding the structure is available (e.g. link to the dataflow). Example:

```
"uri": "http://sdw-ws.ecb.europa.eu/dataflow/ECB/EXR/1.0"
```

2.3.2 name

String nullable. Data flow name. Example:

```
"name": "Sample dataflow"
```

2.3.3 description

String nullable. Descriptio of the data flow. Example:

```
"description": "Data flow description."
```

2.3.4 dimensions

Object. Describes the dimensions used in the message as well as the levels in the hierarchy (data set, series, observations) to which these dimensions are attached. Example:

```
"dimensions": {
    "dataSet": [
        # Component objects #
    ],
    "series": [
        # Component objects #
    ],
    "observation": [
        # Component object #
    ]
}
```

2.3.5 attributes

Object. Describes the attributes used in the message as well as the levels in the hierarchy (data set, series, observations) to which these attributes are attached. Example:

```
"attributes": {
    "dataSet": [
        # Component objects #
    ],
    "series": [
        # Component objects #
    ],
    "observation": [
        # Component objects #
    ]
}
```

2.3.6 dataSet

Array nullable. Optional array to be provided if components (dimensions or attributes) are attached to the data set level.

2.3.7 series

Array nullable. Optional array to be provided if components (dimensions or attributes) are attached to the series level.

2.3.8 observation

Array nullable. Optional array to be provided if components (dimensions or attributes) are attached to the observation level.

2.3.8.1 component

A component represents a dimension or an attribute used in the message. It contains basic information about the component (such as its name and id) as well as the list of values used in the message for this particular component. Example:

```
{
  "id": "FREQ",
  "name": "Frequency",
  "keyPosition": 0,
  "values": [
    {
      # value object #
    }
  ]
}
```

Each of the components may contain the following fields

2.3.8.1.1 id *String.* Identifier for the component. Example:

```
"id": "FREQ"
```

2.3.8.1.2 name *String.* Name provides a human-readable name for the component. Example:

```
"name": "Frequency"
```

2.3.8.1.3 description *String nullable.* Provides a description for the component. Example:

```
"description": "The time interval at which observations occur over a given time period."
```

2.3.8.1.4 keyPosition *Number nullable.* Indicates the position of the dimension in the key, starting at 0. This field should not be supplied for attributes and it may also be omitted for dimensions. This field could be used to build the “key” parameter string (i.e. D.USD.EUR.SP00.A) for data queries, whenever the order of the dimensions cannot easily be derived from the structural metadata information available in the data message. Example:

```
"keyPosition": 0
```

2.3.8.1.5 role *String nullable.* Defines the component role(s), if any. Roles are represented by the id of a concept defined as SDMX cross-domain concept. Several of the concepts defined as SDMX cross-domain concepts are useful for data visualisation, such as for example, the series title, the unit of measure, the number of decimals to be displayed, the reference area (e.g. when using maps), the period of time to which the measured observation refers, etc. It is recommended to identify any component that can be useful for data visualisation purposes by using the appropriate SDMX cross-domain concept as role. Example:

```
"role": "TITLE"
```

2.3.8.1.6 default *String* or *Number nullable.* Defines a default value for the component (valid for attributes only!). If no value is provided in the data part of the message then this value applies. Example:

```
"default": "A"
```

2.3.8.1.7 values *Array.* Array of values for the component. Example:

```
"values": [
  {
    "id": "M",
    "name": "Monthly"
  }
]
```

2.3.8.1.8 Component value *Object nullable.* A particular value for a component in a message. Example:

```
{
    "id": "M",
    "name": "Monthly"
}
```

2.3.8.1.8.1 id *String.* Unique identifier for a value. Example:

```
"id": "A"
```

2.3.8.1.8.2 name *String.* Human-readable name for a value. Example:

```
"name": "Missing value; data cannot exist"
```

2.3.8.1.8.3 description

String nullable. Description provides a human-readable description of the value. The description is typically longer than the text provided for the name field. Example:

```
"description": "Description for missing value."
```

2.3.8.1.8.4 start and end fields

String nullable. Start and end are instances of time that define the actual Gregorian calendar period covered by the values for the time dimension. The algorithm for computing start and end fields for any supported reporting period is defined in the SDMX Technical Notes.

These fields should be used only when the component value represents one of the values for the time dimension.

Values are considered as inclusive both for the start field and the end field. Values must follow the ISO 8601 syntax for combined dates and times, including time zone.

Example:

```
{
    "id": "2010",
    "name": "2010",
    "start": "2010-01-01T00:00Z",
    "end": "2010-12-31T23:59:59Z"
}
```

These fields are useful for visualisation tools, when selecting the appropriate point in time for the time axis. Statistical data, can be collected, for example, at the beginning, the middle or the end of the period, or can represent the average of observations through the period. Based on this information and using the start and end fields, it is easy to get or calculate the desired point in time to be used for the time axis.

2.3.9 Annotations

Array nullable. Provides a list of annotation objects. Annotations can be attached to data sets, series and observations.

```
"annotations": [
  {
    "title": "Sample annotation",
    "uri": "http://sample.org/annotations/74747"
  }
]
```

Each annotation object contains the following optional information:

2.3.9.1 title

string nullable. Provides a title for the annotation. Example:

```
"title": "Sample annotation"
```

2.3.9.2 type

string nullable. Type is used to distinguish between annotations designed to support various uses. The types are not enumerated, and these can be freely specified by the creator of the annotations. The definitions and use of annotation types should be documented by their creator. Example:

```
"type": "reference"
```

2.3.9.3 uri

string nullable. URI - typically a URL - which points to an external resource which may contain or supplement the annotation. If a specific behavior is desired, an annotation type should be defined which specifies the use of this field more exactly.

```
"uri": "http://sample.org/annotations/74747"
```

2.3.9.4 text

string nullable. Contains the text of the annotation.

```
"text": "Sample annotation text"
```

2.3.9.5 id

string nullable. ID provides a non-standard identification of an annotation. It can be used to disambiguate annotations. Example:

```
"id": "74747"
```

2.4 dataSets

An array of data set objects. Example:

```
"dataSets": [
  {
    "action": "Information",
    "series": {
      # series object #
    }
  }
]
```

There are between 2 and 3 levels in a data set object, depending on the way the data in the message is organized.

A data set may contain a flat list of observations. In this scenario, we have 2 levels in the data part of the message: the data set level and the observation level.

A data set may also organize observations in logical groups called series. These groups can represent time series or cross-sections. In this scenario, we have 3 levels in the data part of the message: the data set level, the series level and the observation level.

Dimensions and attributes may be attached to any of these 3 levels.

In case the data set is a flat list of observations, observations will be found directly under a data set object. In case the data set represents time series or cross sections, the observations will be found under the series elements.

2.4.1 action

String nullable. Action provides a list of actions, describing the intention of the data transmission from the sender's side. `Default value is Information`

- Append - this is an incremental update for an existing data set or the provision of new data or documentation (attribute values) formerly absent. If any of the supplied data or metadata are already present, it will not replace these data.
- Replace - data are to be replaced, and may also include additional data to be appended.
- Delete - data are to be deleted.
- Information - data are being exchanged for informational purposes only, and not meant to update a system.

Example:

```
"action": "Information"
```

2.4.2 reportingBegin

String nullable. The start of the time period covered by the message. Example:

```
"reportingBegin": "2012-05-04"
```

2.4.3 reportingEnd

String nullable. The end of the time period covered by the message. Example:

```
"reportingEnd": "2012-06-01"
```

2.4.4 validFrom

String nullable. The validFrom indicates the inclusive start time indicating the validity of the information in the data.

```
"validFrom": "2012-01-01T10:00:00Z"
```

2.4.5 validTo

String nullable. The validTo indicates the inclusive end time indicating the validity of the information in the data.

```
"validTo": "2013-01-01T10:00:00Z"
```

2.4.6 publicationYear

String nullable. The publicationYear holds the ISO 8601 four-digit year.

```
"publicationYear": "2005"
```

2.4.7 publicationPeriod

String nullable. The publicationPeriod specifies the period of publication of the data in terms of whatever provisioning agreements might be in force (i.e., “2005-Q1” if that is the time of publication for a data set published on a quarterly basis).

```
"publicationPeriod": "2005-Q1"
```

2.4.8 annotations

Array nullable. An optional array of annotation indices for the dataset. Indices refer back to the array of annotations in the structure field. Example:

```
"annotations": [ 3, 42 ]
```

2.4.9 attributes

Array nullable. Collection of attributes values attached to the data set level. This is typically the case when a particular attribute always has the same value for the data available in the data message. In order to avoid repetition, that value can simply be attached at the data set level. Example:

```
"attributes": [ 0, null, 0 ]
```

2.4.10 observations

Object nullable. Collection of observations directly attached to a data set. This is the case when a data set represents a flat collection of observations. In case the observations are organised into logical groups (time series or cross-sections), use the series element instead. Example:

```
"observations": {
  "0:1:0": [ 105.6, 0, 1],
  "0:1:1": [ 105,9 ]
}
```

2.4.11 series

Object nullable. A collection of series. Each series object is identified by a key (a string representing the index of each dimension value, separated by a colon) and contains the observation values and associated attributes, when the observations contained in the data set are used into logical groups (time series or cross-sections). This element must **not** be used in case the data set represents a flat list of observations. Example:

```
{
  "annotations": [],
  "attributes": [ 0, 1 ],
  "observations": {
    "0": [ 105.6, null, null ],
    "1": [ 105.9 ],
    "2": [ 106.9 ],
    "3": [ 107.3, 0 ]
  }
}
```

2.4.11.1 annotations

Array nullable. An optional array of annotation indices for the series. Indices refer back to the array of annotations in the structure field. Example:

```
"annotations": [ 3, 42 ]
```

2.4.11.2 attributes

Array nullable. Collection of attributes values. Each value is an index to the *values* array in the respective *Attribute* object. Example:

```
"attributes": [ 0, 1 ]
```

For information on how to handle the attribute values, see the section dedicated to handling component values.

2.4.11.3 observations

Object nullable. An object of observation values. Each observation value is an array of one of more values.

```
"observations": {
  "0": [ 105.6, null, null ],
  "1": [ 105.9 ],
  "2": [ 106.9 ],
  "3": [ 107.3, 0 ]
}
```

The keys in the observation object are the index values of the observation level dimensions. It's one for time series and cross-sections, but there will be more than one when the data set represents a flat list of observations.

The first value in the observation array is the observation value. The data type for observation value is *Number*. Data type for a reported missing observation value is a *null*.

Elements after the observation value are values for the observation level attributes.

3 Handling component values

Let's say that the following message needs to be processed:

```
{
    "header": {
        "id": "62b5f19d-f1c9-495d-8446-a3661ed24753",
        "prepared": "2012-11-29T08:40:26Z",
        "sender": {
            "id": "ECB",
            "name": "European Central Bank"
        }
    },
    "structure": {
        "id": "ECB_EXR_WEB",
        "uri": "http://sdw-ws.ecb.europa.eu/dataflow/ECB/EXR/1.0",
        "dimensions": {
            "dataSet": [
                {
                    "id": "FREQ",
                    "name": "Frequency",
                    "keyPosition": 0,
                    "values": [
                        {
                            "id": "D",
                            "name": "Daily"
                        }
                    ]
                },
                {
                    "id": "CURRENCY_DENOM",
                    "name": "Currency denominator",
                    "keyPosition": 2,
                    "values": [
                        {
                            "id": "EUR",
                            "name": "Euro"
                        }
                    ]
                },
                {
                    "id": "EXR_TYPE",
                    "name": "Exchange rate type",
                    "keyPosition": 3,
                    "values": [
                        {
                            "id": "SP00",
                            "name": "Spot rate"
                        }
                    ]
                },
                {
                    "id": "EXR_SUFFIX",
                    "name": "Series variation - EXR context",
                    "keyPosition": 4,
                    "values": [
                        {
                            "id": "A",
                            "name": "Average or standardised measure"
                        }
                    ]
                }
            ],
            "series": [
                {
                    "id": "CURRENCY",
                    "name": "Currency",
                    "keyPosition": 1,
                    "values": [
                        {
                            "id": "NZD",
                            "name": "New Zealand dollar"
                        }, {
                            "id": "RUB",
                            "name": "Russian rouble"
                        }
                    ]
                }
            ],
            "observation": [
                {
                    "id": "TIME_PERIOD",
                    "name": "Time period or range",
                    "values": [
                        {
                            "id": "2013-01-18",
                            "name": "2013-01-18",
                            "start": "2013-01-18T00:00:00Z",
                            "end": "2013-01-18T23:59:59Z"
                        }, {
                            "id": "2013-01-21",
                            "name": "2013-01-21",
                            "start": "2013-01-21T00:00:00Z",
                            "end": "2013-01-21T23:59:59Z"
                        }
                    ]
                }
            ]},
        "attributes": {
            "dataSet": [],
            "series": [
                {
                    "id": "TITLE",
                    "name": "Series title",
                    "values": [
                        {
                            "name": "New zealand dollar (NZD)"
                        }, {
                            "name": "Russian rouble (RUB)"
                        }
                    ]
                }
            ],
            "observation": [
                {
                    "id": "OBS_STATUS",
                    "name": "Observation status",
                    "values": [
                        {
                            "id": "A",
                            "name": "Normal value"
                        }
                    ]
                }
            ]
        }
    },
    "dataSets": [
        {
            "action": "Information",
            "series": {
                "0": {
                    "attributes": [0],
                    "observations": {
                        "0": [1.5931, 0],
                        "1": [1.5925, 0]
                    }
                },
                "1": {
                    "attributes": [1],
                    "observations": {
                        "0": [40.3426, 0],
                        "1": [40.3000, 0]
                    }
                }
            }
        }
    ]
}
```

There is one data set in the message, and it contains two series.

```
"0": {
    "attributes": [0],
    "observations": {
        "0": [1.5931, 0],
        "1": [1.5925, 0]
    }
},
"1": {
    "attributes": [1],
    "observations": {
        "0": [40.3426, 0],
        "1": [40.3000, 0]
    }
}
```

The structure.dimensions field tells us that, out of the 6 dimensions, 4 have the same value for the 2 series and are therefore attached to the data set level.

We see that, for the first series, we get the value 0:

```
"0": { ... }
```

From the structure information, we know that CURRENCY is the series dimension.

```
"series": [
    {
        "id": "CURRENCY",
        "name": "Currency",
        "keyPosition": 1,
        "values": [
            {
                "id": "NZD",
                "name": "New Zealand dollar"
            }, {
                "id": "RUB",
                "name": "Russian rouble"
            }
        ]
    }
]
```

The value 0 identified previously is the index of the item in the collection of values for this component. In this case, the dimension value is therefore “New Zealand dollar”.

The same logic applies when mapping attributes.

4 Security Considerations

This document defines a response format for SDMX RESTful Web Services in JSON and it raises no new security considerations. SDMX Web Services Guidelines includes the security considerations associated with its usage.

5 References

5.1 Normative References

[iso8601] *ISO 8601:2004 Data elements and interchange formats — Information interchange — Representation of dates and times*, ISO, 2004

[json] *The JSON Data Interchange Format*, Standard ECMA-404, ECMA International, October 2013

[sdmxim21] *Section 2: Information Model: UML Conceptual Design*, SDMX Standards 2.1, SDMX, April 2011

[sdmxws21] *Section 7: Guidelines for the Use of Web Services*, SDMX Standards 2.1, SDMX, April 2011

581 Appendix A. Full Example with Comments

```
{
  "header": {

    # dynamically generated GUI
    "id": "62b5f19d-f1c9-495d-8446-a3661ed24753",

    # extraction time from db (=now in SQL query), include timezone!
    "prepared": "2012-11-29T08:40:26Z",

    # optional with default false
    "test": false,

    "sender": {
      "id": "ECB",
      "name": "European Central Bank",
      "contact": [
        {
          "name": "Statistics hotline",
          "department": "Statistics Department",
          "role": "helpdesk",
          "telephone": ["+00-00-99999"],
          "fax": ["+00-00-88888"],
          "uri": ["http://www.xyz.org"],
          "email": ["statistics@xyz.org"]
        }
      ]
    },

    # receiver is optional, info from user record if authenticated
    "receiver": {
      "id": "SDMX",
      "name": "SDMX",
      "contact": [
          {
              "name": "name",
              "department": "department",
                "role": "role",
                "telephone": ["telephone"],
                "fax": ["fax"],
                "uri": ["uri"],
                "email": ["sdmx@xyz.org"]
            }
        ]
    },

    "request": {
      # include complete URL as used by the client
      "uri": "http://www.myorg.org/ws/data/ECB_ICP1/M.PT+FI.N.000000+071100.4.INX?
      startPeriod=2009-01&dimensionAtObservation=AllDimensions"
    }
  },
  "errors": [
    {
      "code": 123,
      "message": "Invalid number of dimensions in parameter key"
    }
  ],
  "structure": {
    # resolvable uri to dataflow
    "uri": "http://sdw-ws.ecb.europa.eu/dataflow/ECB/EXR/1.0",

    "name": "dataflow name",
    "description": "dataflow description",
    "dimensions": {

      # dataSet is used only if grouping of dimensions with single values
      "dataSet": [
        {
          "id": "FREQ",
          "name": "Frequency",
          "description": "Description for the dimension",

          # 0-based position of dimension in key in user request url
          "keyPosition": 0,

          # restricted list of dimension and attribute roles (time, frequency,
          # geo, unit, scalefactor, referenceperiod, ...)
          "role": "frequency",

          "values": [
        {
          "id": "D",
          "name": "Daily"
        }
      ],
    }, {
      "id": "CURRENCY_DENOM",
      "name": "Currency denominator",
      "description": "Description for the dimension",
      "keyPosition": 3,
      "values": [
        {
          "id": "EUR",
          "name": "Euro"
        }
      ]
    }, {
      "id": "EXR_TYPE",
      "name": "Exchange rate type",
      "description": "Description for the dimension",
      "keyPosition": 4,
      "values": [
        {
          "id": "SP00",
          "name": "Spot rate"
        }
      ]
    }, {
      "id": "EXR_SUFFIX",
      "name": "Series variation - EXR context",
      "description": "Description for the dimension",
      "keyPosition": 5,
      "values": [
        {
          "id": "A",
          "name": "Average or standardised measure for given frequency"
        }
      ]
    }
  ],

  # only if dimensionAtObservation <> allDimensions
  "series": [
    {
      "id": "CURRENCY",
      "name": "Currency",
      "description": "Description for the dimension",
      "keyPosition": 2,
      "role": "unit",
      "values": [
        {
          "id": "NZD",
          "name": "New Zealand dollar"
        }, {
          "id": "RUB",
          "name": "Russian rouble",
        }
      ]
    }
  ],

  # only for dimensions used at observation level
  "observation": [
    {
      "id": "TIME_PERIOD",
      "name": "Time period or range",
      "description": "Description for the dimension",
      "role": "time",
      "values": [
        {
          "id": "2013-01-18",
          "name": "2013-01-18",
          "start": "2013-01-18T00:00:00Z",
          "end": "2013-01-18T23:59:59Z"
        },
        {
          "id": "2013-01-21",
          "name": "2013-01-21",
          "start": "2013-01-21T00:00:00Z",
          "end": "2013-01-21T23:59:59Z"
        }
      ]
    }
  ]
},
"attributes": {

  # only for attributes returned at dataset level
```

```
"dataSet": [],

# only for attributes returned at series level
"series": [
  {
    "id": "ID",
    "name": "Attribute name",
    "description": "Description for the attribute",
    "role": null,
    "default": null,

    # inclusion of attachment level and its format to be decided
    # e.g. "attachment": [ true, true, true, true, true, true, false ],

    "values": [
      {
        # id property is optional to allow for uncoded attributes
        "id": null,
        "name": "New Zealand dollar (NZD)"
      },
      {
        "id": null,
        "name": "Russian rouble (RUB)"
      }
    ]
  }
],
"observation": [
  {
    "id": "OBS_STATUS",
    "name": "Observation status",
    "description": "Description for the attribute",
    "role": null,

    # optional
    "default": "A",

    "values": [
      # a null attribute can be used to shorten the message by
      # using 0 index later in message
      null,

      {
        "id": "A",
          "name": "Normal value",
          "description": "Normal value"
        }
      ]
    }
  ]
},
"annotations": [
  {
    "title": "AnnotationTitle provides a title for the annotation.",
    "type": "AnnotationType is used to distinguish between annotations
    designed to support various uses.",
    "uri": "http://www.myorg.org/ws/uri/for/this/annotation",
    "text": "AnnotationText holds a language-specific string containing
    the text of the annotation.",
    "id": "The id attribute provides a non-standard identification of an
    annotation. It can be used to disambiguate annotations."
  }
]
},
"dataSets": [
  {
    "action": "Information",

    # optional first time period in returned message
    "reportingBegin": "2012-05-04",

    # optional last time period in returned message
    "reportingEnd": "2012-06-01",

    # optional only for version history
    "validFrom": "2012-01-01T10:00:00Z",

    # optional only for version history
    "validTo": "2013-01-01T10:00:00Z",

    # optional only for publication release calendars
    "publicationYear": "2005",

    # optional only for publication release calendars
    "publicationPeriod": "2005-Q1",

    # optional as per annotations
    "annotations": [0],
  # optional as per attributes at dataset level
  "attributes": [0],

  # 1st alternative (only if series level
  # (dimensionAtObservation <> allDimensions))

  "series": {
    "0": {
      "annotations": [],
      "attributes": [0],
      "observations": {
        "0": [1.5931, 0],
        "1": [1.5925, 0]
      }
    },
    "1": {
      "annotations": [ 34 ],
      "attributes": [1],
      "observations": {
        "0": [40.3426, 0],
        "1": [40.3000, 0]
      }
    }
  }
},
{
  "action": "Information",

  # 2nd alternative (only if no series level
  #  (dimensionAtObservation == allDimensions))

  "observations": {
      "0:0": [1.5931, 0],
      "0:1": [1.5925, 0],
      "1:0": [40.3426, 0],
      "1:1": [40.3000, 0]
  }
},

# In case that the server does not group dimensions
# with single values at dataset level

{
  "action": "Information",

  # 1st alternative (only if series level
  # (dimensionAtObservation <> allDimensions))

  "series": {
     "0:0:0:0;0": {
        "attributes": [0],
        "observations": {
            "0": [1.5931, 0],
            "1": [1.5925, 0]
        }
     },
     "0:0:0:0;1": {
        "attributes": [1],
        "observations": {
            "0": [40.3426, 0],
            "1": [40.3000, 0]
        }
     }
  }
},
{
  "action": "Information",

  # 2nd alternative (only if no series level
  # (dimensionAtObservation == allDimensions))

  "observations": {
      "0:0:0:0:0:0": [1.5931, 0],
      "0:0:0:0:0:1": [1.5925, 0],
      "0:0:0:0:1:0": [40.3426, 0],
      "0:0:0:0:1:1": [40.3000, 0]
  }
},

# In case the client is using the detail parameter
# and the server supports it

{
  "action": "Information",

  # Detail parameter: serieskeysonly. No observation values,
  # attributes or annotations.
```

```
      "observations": {
          "0:0": [],
          "0:1": [],
          "1:0": [],
          "1:1": []
      ]
    },
    {
      "action": "Information",

      # Detail parameter: dataonly. No attributes or annotations.

      "observations": {
          "0:0": [1.5931],
          "0:1": [1.5925],
          "1:0": [40.3426],
          "1:1": [40.3000]
      ]
    },
    {
      "action": "Information",

      # Detail parameter: nodata. No observation values
      # just attributes and annotations.

      "observations": {
          "0:0": [0],
          "0:1": [0],
          "1:0": [0],
          "1:1": [0]
      ]
    }
  ]
}
```