Observation JSON Format

22/08/2022

Présentation

Environmental Sensing

Table of Contents

1 Introduction **2**

1.1 Conventions used 2

1.2 Terminology 2

1.3 Rules 2

2 Structure **3**

3 Members **4**

3.1 ESId 4

3.2 ESName 4

3.3 ESIlist 4

3.4 ESInfo 5

3.5 ESParam 5

Appendix: TypeCatalog **7**

Appendix : Environmental Iindex **8**

Appendix : reserved values **9**

Appendix : Examples **10**

Appendix : CBOR format **11**

1 Introduction

ObsJSON is a text format for the ES-Observation data.

This format is an application of the JSON format (RFC 8259), GeoJSON format (RFC 7946), Date and Time format (RFC 3339).

A binary version is also defined (Appendix) with CBOR format (RFC 8949)

1.1 Conventions used

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The grammatical rules in this document are to be interpreted as described in [RFC5234].

1.2 Terminology

The terms Json-Text, Json-Value (Value), Object, Member, Array, Number, String, False, Null, True are defined in the JSON grammar.

The terms Geometry-type, Point, MultiPoint, LineString, MultiLineString, Polygon, MultiPolygon, GeometryCollection, GeoJSON-Types are defined in GeoJSON grammar.

Timestamp is defined in Date and Time format.

1.3 Rules

An ObsJSON-Text CAN contains all the ESObservation information (i.e., the ESObservation build from the ObsJSON-Text is identical to the initial ESObservation).

Values in Array are ordered and independent from the other Values.

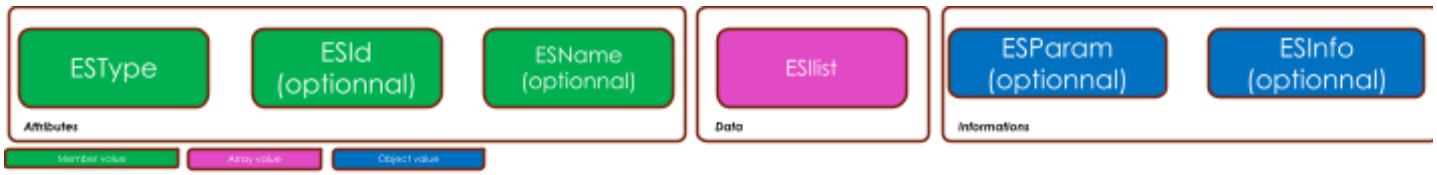
Members in Objects are not ordered.

2 STRUCTURE

An Observation is an Ilist Object with specific and complementary data (see Appendix).

An ObsJSON is a JSON-Object and represents an Observation.

The figure below shows the Json structure of ObsJSON.



ESType: "type": "obs"
ESId: "id": "ob12345"
ESName: "name": "observation initiale"

ESIlist: "data": JSON Array

ESParam: "param": JSON Object
ESInfo: "infos": JSON Array

Format

Text (JSON text), Binary (CBOR)

Examples

```
{'type': 'obs', 'data': ['Anne' 'John', 'Paul']}
```

```
{'type': 'obs', 'id': 'ob12345', 'name': 'obs init',
 'data': [
          [ 'datation', ['2021-01-04',[['2021-01-05','2021-01-05']]]],
          [ 'location', [[2.4, 48.9], [[[2.4, 48.9], [4.8, 45.8], [5.4, 43.3]]]], 0],
          [ 'property', [{'prp': 'PM10', 'unit': 'µg/m3'}, {'prp': 'Temp'}] ]
          [ 'result', [51.3, {'low temp': 2.4}, 20.8, 'high temp'], -1 ]
          ],
 'param': {'pdict': 'official', 'example':4},
 'infos': {'locbox': [2.4, 48.9], 'dim': 2}
```

The Members are:

Category	Member	Key	Value
Attributes	ESName	”name”	String
Attributes	ESType	“type”	“obs”
Attributes	ESId	“id”	String
Data	ESIlist	“data”	Array
Informations	ESParam	“param”	Object
Informations	ESInfos	“infos”	Object

An ESObservation is valid if:

- it contains at least the Type Member,
- each Member is valid.

3 MEMBERS

3.1 ESID

Description

ESId is a single key/value Member:

- Key: "id"

The value is a string (e.g., Database Id or filename)

Validity

ESId is valid if key and value are as defined.

3.2 ESNAME

Description

ESName is a single key/value Member:

- Key: "name"

The value is a string.

Validity

ESName is valid if key and value are as defined.

3.3 ESILIST

Description

ESIlist is a single key/value Member:

- Key: "data"

Value of ESIlist Member is an Array.

Validity

An ESIlist Member is valid if:

- the value is a valid IlistJSON Array

Note:

> *Observation Objects MAY use specific Iindex objects in the Ilist Object to represent Environmental Data (datation, location, property…) (see Appendix)*

3.4 ESInfo

Description

The Value of an ESInfo is an Object where Members are:

Member	Key	Value
ObservationType	« typeobs »	String
LocationType	« typeloc »	String
DatationType	« typedat »	String
PropertyType	« typeprp »	String
ResultType	« typeres »	String
nValLocation	« nvalloc »	Integer
nValDatation	« nvaldat »	Integer
nValProperty	« nvalprp »	Integer
nValResult	« nvalres »	Integer
BoundingBox	« bbox »	Array (4 Float)
IntervalBox	« tbox »	Array (2 String)
Complet	« complete »	True / false
Score	« score »	Integer
Dimension	« dimension »	Integer
Axes	« axes »	Array (1 to 3 integers)

Validity

An ESInfo is valid if the Value contains at least one Member.

All the Value Members are optional.

Note:

That information comes from the Observation data (calculated).

A parser MAY ignore The ESInfo Member to build an Observation.

3.5 ESParam

Description

The Value of an ESParam is an Object where Members are:

Member	Key	Value
Reference	« reference »	String
ResultTime	« resulttime »	Timestamp or DateTime
PropertyDict	« pdict »	String
UniqueIndex	« unicindex »	True/false
UserData	String	JSON-value

Validity

An ESParam is valid if the Value contains at least one Member.

All the Value Members are optional.

Note:

The UserData members are without constraints.

The keys used in UserData MUST be different from those defined in the Reserved list name (see Appendix.)

A parser takes into account this data without treatment to build an Observation.

APPENDIX: TYPECATALOG

Object	Intern / extern	Type
Observation	ext	observation
DatationValue	Int / ext	datvalue
LocationValue	Int / ext	locvalue
PropertyValue	Int / ext	prpvalue
NamedValue	ext	namvalue
ExternValue	Ext	extvalue
Ilist	ext	ilist
Coordinates	ext	coordinate
datetime	ext	datetime
TimeSlot	ext	timeslot

Appendix : Environmental Iindex

To be complete

APPENDIX : RESERVED VALUES

« type »
« id »
« datation »
« location »
« property »
« result »
« information »
« parameter »
« observation »
« prp »
« unit »
« sampling »
« application »
« sensor »
« uppervalue »
« lowervalue »
« period »
« updateinterval »
« uncertainty »
« typeobs »
« typeloc »
« typedat »
« typeprp »
« typeres »
« nvalloc »
« nvaldat »
« nvalprp »
« nvalres »
« bbox »
« tbox »
« complet »
« score »
« rate »
« dimension »
« axes »
« reference »
« resulttime »
« order »
« propdict »
« unicindex »

APPENDIX : EXAMPLES

- {"type": "obs", "data": [["datation", []], ["location", []], ["property", []], ["result", []]]}

- {"type": "record", "data": ["morning", "paris", "air quality", "good"]}

- {"type": "obs", "data": [["datation", ["morning"]], ["location", ["paris"]], ["property", ["air quality"]], ["result", ["good"], -1]]}

- {"type": "obs", "data": [["datation", ["2021-01-05T22:18:26"]], ["location", [[2.4, 48.9]]], ["property", [{"prp": "PM10"}]], ["result", [51.3], -1]]}

- {"type": "obs", "data": [["datation", ["2021-01-05T22:18:26", "2021-01-05T22:18:26"], ["property", ["air quality PM10", "air quality PM2.5"]], ["result": [10.2, 21.5, 51.3, 48]]]}

- {"type": "observation", "name": "example4", "id": "example4.obs",

  "parameter": {"pdict": "official", "example":4},

  "data": [

  ["datation", ["2021-01-04T10:00:00",[["2021-01-05T08:00:00","2021-01-05T12:00:00"]]]],

  ["location", [[2.4, 48.9], [[[2.4, 48.9], [4.8, 45.8], [5.4, 43.3], [2.4, 48.9]]]], 0],

  ["property", [{"prp": "PM10", "unit": "µg/m3"}, {"prp": "Temp", "unit": "°c"}]]

  ["result", [51.3, {"low temperature": 2.4}, 20.8, "high temperature"], -1]

  ]

  }

APPENDIX : CBOR FORMAT

The Concise Binary Object Representation (CBOR – RFC8949) is a data format whose design goals include the possibility of extremely small code size, small message size, and extensibility without the need for version negotiation.

CBOR is based on the JSON data model: numbers, strings, arrays, maps (called objects in JSON), and a few values such as false, true, and null.

The CBOR format can be used with different options to minimize length:

- The precision of float values is adjustable from half precision (two bytes) to double precision (eight bytes),
- The datetime can be described by a standard text string (RFC3339) or by a numerical value (Epoch-based: six bytes).
- The TypeValue can be represented with a code value instead of string value
- The coordinates value can be described with integer instead of float (val_int = round(val_float)*10**7 : four bytes).

Example (Json format):

{"type": "observation",

"datation":["2021-01-04T10:00:00",[["2021-01-05T08:00:00","2021-01-5T12:00:00"]]],

"location":[[2.4123456, 48.9123456], [[[2.4123456, 48.9123456], [4.8123456, 45.8123456], [5.4123456, 43.3123456], [2.4123456, 48.9123456]]]],

"property": [{"prp": "PM10"}, {"prp": "Temp"}],

"result": [51.348, {"low": 2.457}, 20.88, "high"],

"coupled": {"datation": "location"}}

Example optimized (Cbor format):

{0 : [0,1,2],

1 : [[dt(2021, 1, 4, 10),[[dt(2021,1,5,8), dt(2021, 1, 5, 12]]],

[[2.4123456, 48.9123456], [[[2.4123456, 48.9123456], [4.8123456, 45.8123456], [5.4123456, 43.3123456], [2.4123456, 48.9123456]]]],

[{"prp": "PM10"}, {"prp": "Temp"}]],

2 : [51.34375, {"low": 2.45703125}, 20.875, "high"],

3 : {0: 1} }

With :

- *Observation key codification:*

    - *0: "order"*
    - *1: "features"*
    - *2: "result"*
    - *3: "coupled"*
- *Order and coupled value codification:*
    - *0: "datation"*
    - *1: "location"*
    - *2: "property"*
- *Datation value: timestamp format*
- *Location value: integer representation (four bytes)*
- *Result value: half precision (two bytes)*

Length (bytes):

- JSON: 388
- CBOR: 298
- CBOR optimized: 133