

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.4473

(08/2020)

SERIES Y: GLOBAL INFORMATION
INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS,
NEXT-GENERATION NETWORKS, INTERNET OF
THINGS AND SMART CITIES

Internet of things and smart cities and communities –
Frameworks, architectures and protocols

SensorThings API – Sensing

Recommendation ITU-T Y.4473

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

GLOBAL INFORMATION INFRASTRUCTURE

General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899

INTERNET PROTOCOL ASPECTS

General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999

NEXT GENERATION NETWORKS

Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999

FUTURE NETWORKS

CLOUD COMPUTING

BIG DATA

QUANTUM KEY DISTRIBUTION NETWORKS

INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES

General	Y.4000–Y.4049
Definitions and terminologies	Y.4050–Y.4099
Requirements and use cases	Y.4100–Y.4249
Infrastructure, connectivity and networks	Y.4250–Y.4399
Frameworks, architectures and protocols	Y.4400–Y.4549
Services, applications, computation and data processing	Y.4550–Y.4699
Management, control and performance	Y.4700–Y.4799
Identification and security	Y.4800–Y.4899
Evaluation and assessment	Y.4900–Y.4999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T Y.4473

SensorThings API – Sensing

Summary

Recommendation ITU-T Y.4473 specifies the SensorThings application programming interface (API) which provides an open standard-based and geospatial-enabled framework to interconnect Internet of things (IoT) devices, data and applications over the Web.

The SensorThings API is an open standard, and that means it is non-proprietary, platform-independent. It builds on a rich set of proven-working and widely-adopted open standards, such as the web protocols and the Open Geospatial Consortium (OGC) sensor web enablement (SWE) standards, including the ISO/OGC observation and measurement data model. The SensorThings API is extensible and can be applied to not only simple but also complex use cases.

This Recommendation provides a standard way to manage and retrieve observations and metadata from heterogeneous IoT sensor systems. The SensorThings API uses representational state transfer (REST) principles, an efficient JavaScript object notation (JSON) encoding, message queuing telemetry transport (MQTT) protocol, flexible OASIS open data protocol (OData) and uniform resource locator (URL) conventions.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4473	2020-08-29	20	11.1002/1000/14375

Keywords

Internet of things, sensor web, smart cities, smart communities.

* To access the Recommendation, type the URL <http://handle.itu.int/> in the address field of your web browser, followed by the Recommendation's unique ID. For example, <http://handle.itu.int/11.1002/1000/11830-en>.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2020

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

	Page
1 Scope	1
2 References.....	1
3 Definitions	2
3.1 Terms defined elsewhere	2
3.2 Terms defined in this Recommendation.....	2
4 Abbreviations and acronyms	3
5 Conventions	3
6 The SensorThings API entities	3
6.1 Common control information	3
6.2 The sensing entities	4
7 SensorThings service interface	18
7.1 Common control information	18
7.2 Resource path	19
7.3 Requesting data	23
8 SensorThings sensing create-update-delete	31
8.1 Overview	31
8.2 Create an entity	31
8.3 Update an entity	33
8.4 Delete an entity	34
9 Batch requests	35
9.1 Introduction	35
9.2 Batch-processing request	35
9.3 Batch-processing response	38
9.4 Asynchronous batch requests	39
10 SensorThings MultiDatastream extension	40
11 SensorThings Data Array Extension	45
11.1 Retrieve a Datastream's Observation entities in dataArray	45
11.2 Create Observation entities with dataArray	48
12 SensorThings Sensing MQTT Extension	50
12.1 Create a SensorThings Observation with MQTT Publish	50
12.2 Receive updates with MQTT Subscribe	50
Bibliography.....	53

Recommendation ITU-T Y.4473

SensorThings API – Sensing

1 Scope

This Recommendation specifies a SensorThings application programming interface (API) which provides an open standard-based and geospatial-enabled framework to interconnect Internet of things devices, data, and applications over the Web.

The scope of this Recommendation includes:

- SensorThings API entities and relevant properties, including thing, location, datastream, sensor, observations, etc.;
- SensorThings service interface for requesting data;
- SensorThings sensing operations, including reading, creating, updating and deleting;
- SensorThings batch processing requests and relevant responses, and extensions for multidatastream, data array and Message Queuing Telemetry Transport (MQTT).

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- | | |
|---------------------|--|
| [ITU-T Y.4000] | Recommendation ITU-T Y.4000/Y.2060 (2012), <i>Overview of Internet of things</i> . |
| [IETF RFC 2046] | IETF RFC 2046, <i>Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types</i> .
https://www.ietf.org/rfc/rfc2046.txt |
| [IETF RFC 2616] | IETF RFC 2616, <i>Hypertext Transfer Protocol -- HTTP/1.1</i> .
https://www.ietf.org/rfc/rfc2616.txt |
| [IETF RFC 5789] | IETF RFC 5789, <i>PATCH Method for HTTP</i> ,
https://www.ietf.org/rfc/rfc5789.txt |
| [IETF RFC 6902] | IETF RFC 6902, <i>JavaScript Object Notation (JSON) Patch</i> .
https://www.ietf.org/rfc/rfc6902.txt |
| [IETF RFC 7946] | IETF RFC 7946, <i>The GeoJSON Format</i> .
https://www.ietf.org/rfc/rfc7946.txt |
| [ISO 8601] | ISO 8601:2004, <i>Data elements and interchange formats – Information interchange – Representation of dates and times</i> . |
| [OASIS OData ABNF] | OASIS OData Version 4.0 (2017), <i>ABNF Construction Rules</i> .
http://docs.oasis-open.org/odata/odata/v4.01/cs01/abnf/odata-abnf-construction-rules.txt |
| [OASIS OData JSON] | OASIS OData Version 4.0 (2016), <i>JSON Format Version 4.0 Plus Errata 03</i> .
http://docs.oasis-open.org/odata/odata-json-format/v4.0/odata-json-format-v4.0.html |
| [OASIS OData Part1] | OASIS OData Version 4.0 (2016), <i>Part 1: Protocol Plus Errata 03</i> .
http://docs.oasis-open.org/odata/odata/v4.0/odata-v4.0-part1-protocol.html |

- [OASIS OData Part2] OASIS OData Version 4.0 (2016), *Part 2: URL Conventions Plus Errata 03*.
<http://docs.oasis-open.org/odata/odata/v4.0/odata-v4.0-part2-url-conventions.html>
- [OGC/ISO 19156] OGC/ISO 19156 (2011), *Abstract Specification Topic 20: Geographic information – Observations and Measurements*.
http://portal.opengeospatial.org/files/?artifact_id=41579

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 collection [b-IETF RFC 5023]: Sets of resources, which can be retrieved in whole or in part.

3.1.2 entity [OASIS OData Part1]: Instances of entity types.

3.1.3 entity sets [OASIS OData Part1]: Named collections of entities (e.g., Sensors is an entity set containing Sensor entities). An entity's key uniquely identifies the entity within an entity set. Entity sets provide entry points into an OGC SensorThings API service.

3.1.4 Internet of things (IoT) [ITU-T Y.4000]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – From a broader perspective, the IoT can be perceived as a vision with technological and societal implications.

3.1.5 measurement [OGC/ISO 19156]: A set of operations having the object of determining the value of a quantity.

3.1.6 observation [OGC/ISO 19156]: Act of measuring or otherwise determining the value of a property.

3.1.7 observation result [OGC/ISO 19156]: Estimate of the value of a property determined through a known observation procedure.

3.1.8 resource [IETF RFC 2616]: A network-accessible data object or service identified by an URI.

3.1.9 sensor [b-OGC 12-000]: An entity capable of observing a phenomenon and returning an observed value. Type of observation procedure that provides the estimated value of an observed property at its output.

3.1.10 thing [ITU-T Y.4000]: With regard to the Internet of things, this is an object of the physical world (physical things) or the information world (virtual things), which is capable of being identified and integrated into communication networks.

3.2 Terms defined in this Recommendation

This Recommendation defines the following term:

3.2.1 representational state transfer (REST): An abstraction of the architectural elements within a distributed hypermedia system that focuses on the roles of components, the constraints upon their interaction with other components, and their interpretation of significant data elements.

NOTE 1 – REST encompasses the fundamental constraints upon components, connectors, and data that define the basis of the web architecture, and thus the essence of its behaviour as a network-based application.

NOTE 2 – An API that conforms to the REST architectural principles/constraints is called a RESTful API.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API	Application Programming Interface
CRUD	Create, Read, Update, and Delete
GML	Geography Markup Language
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
JSON	JavaScript Object Notation
MQTT	Message Queuing Telemetry Transport
OData	Open Data Protocol
O&M	Observations and Measurements
REST	Representational State Transfer
SensorML	Sensor Model Language
SOS	Sensor Observation Service
SWE	Sensor Web Enablement
UCUM	Unified Code for Units of Measure
UML	Unified Modelling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WiFi	Wireless-Fidelity
WKT	Well-Known Text
XML	extensible Markup Language

5 Conventions

In this Recommendation:

The word "SHALL" is used to express mandatory requirements for conformance to the Recommendation. The word "MAY" is used to refer to behaviour that is allowed, but not necessarily required. The word "SHOULD" is used to refer to behaviour that is encouraged to be followed, but is not a mandatory requirement for conformance to this Recommendation.

6 The SensorThings API entities

6.1 Common control information

In SensorThings control information is represented as annotations whose names start with `iot` followed by a dot (`iot.`). Annotations are name/value pairs that have a dot (`.`) as part of the name.

When annotating a name/value pair for which the value is represented as a JSON object, each annotation is placed within the object and represented as a single name/value pair. In SensorThings

the name always starts with the "at" sign (@), followed by the namespace *iot*, followed by a dot (.), followed by the name of the term (e.g., "@*iot.id*":1).

When annotating a name/value pair for which the value is represented as a JSON array or primitive value, each annotation that applies to this name/value pair is placed next to the annotated name/value pair and represented as a single name/value pair. The name is the same as the name of the name/value pair being annotated, followed by the "at" sign (@), followed by the namespace *iot*, followed by a dot (.), followed by the name of the term. (e.g., "Locations@*iot.navigationLink*":"http://example.org/v.1.0/Things(1)/Locations")

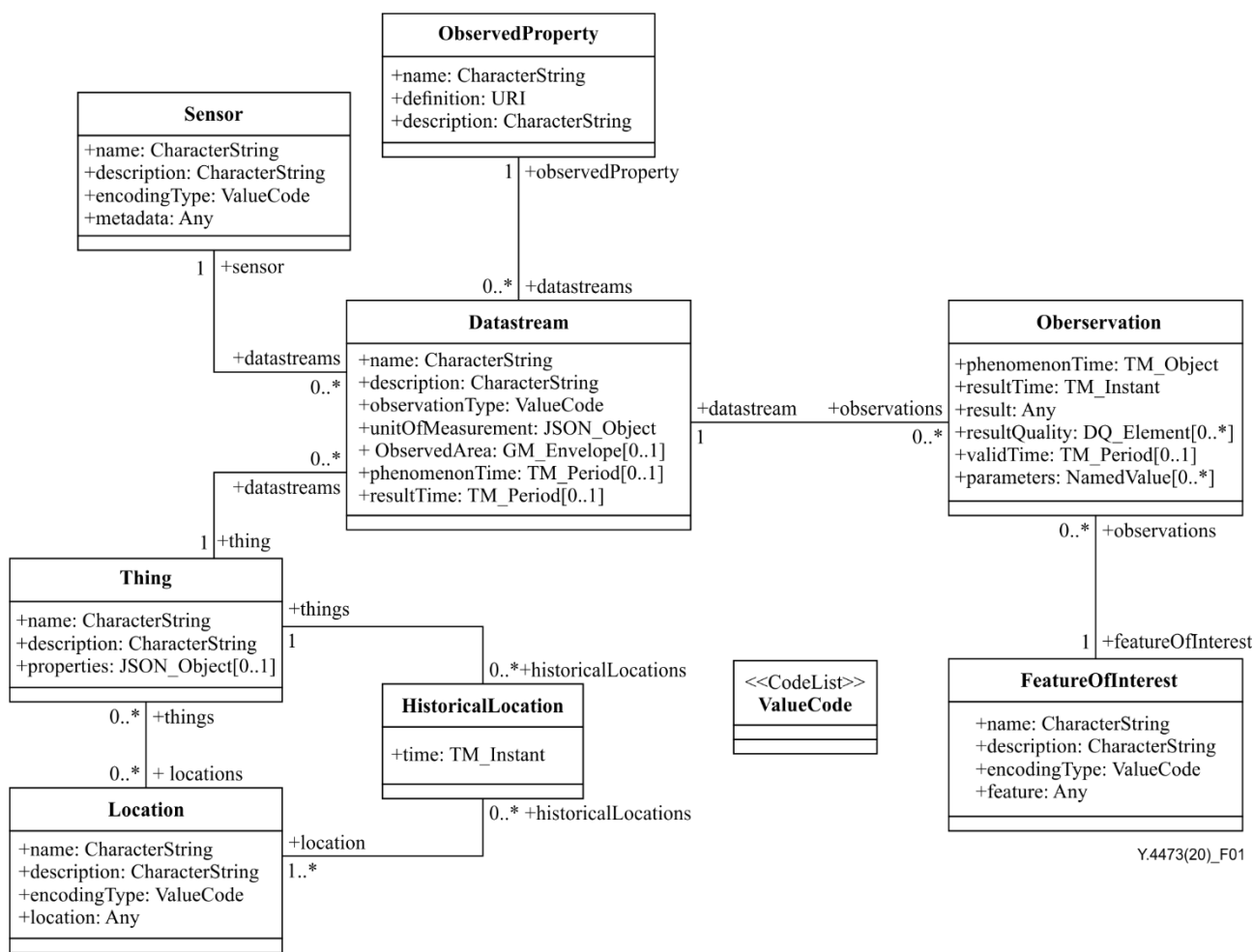
Table 1 presents common control information.

Table 1 – Common control information

Annotation	Definition	Data type	Multiplicity and use
id	id is the system-generated identifier of an entity. id is unique among the entities of the same entity type in a SensorThings service.	Any	One (mandatory)
selfLink	selfLink is the absolute URL of an entity that is unique among all other entities.	URL	One (mandatory)
navigationLink	navigationLink is the relative or absolute URL that retrieves content of related entities.	URL	One-to-many (mandatory)

6.2 The sensing entities

The SensorThings API Sensing part's entities are depicted in Figure 1. The sensing entities include Things, Location, HistoricalLocation, Datastream, Sensor, ObservedProperty, Observations, and FeatureOfInterest.



Y.4473(20)_F01

Figure 1 – Sensing entities

In this clause, the properties in each entity type and the direct relationship to the other entity types are described. In addition, for each entity type, an example of the associated JSON encoding is presented.

6.2.1 Thing

The SensorThings API follows the ITU-T definition, i.e., with regard to the Internet of things, a thing is an object of the physical world (physical things) or the information world (virtual things) that is capable of being identified and integrated into communication networks [ITU-T Y.4000].

Table 2 shows properties of a Thing entity and Table 3 shows the direct relation between a Thing entity and other entity types.

Table 2 – Properties of a Thing entity

Name	Definition	Data type	Multiplicity and use
name	A property provides a label for Thing entity, commonly a descriptive name.	CharacterString	One (mandatory)
description	This is a short description of the corresponding Thing entity.	CharacterString	One (mandatory)
properties	A JSON Object containing user-annotated properties as key-value pairs.	JSON Object	Zero-to-one

Table 3 – Direct relation between a Thing entity and other entity types

Entity type	Relation	Description
Location	Many optional to many optional	The Location entity locates the Thing. Multiple Things MAY be located at the same Location. A Thing MAY not have a Location. A Thing SHOULD have only one Location. However, in some complex use cases, a Thing MAY have more than one Location representations. In such case, the Thing MAY have more than one Locations. These Locations SHALL have different encodingTypes and the encodingTypes SHOULD be in different spaces (e.g., one encodingType in Geometrical space and one encodingType in Topological space).
HistoricalLocation	One mandatory to many optional	A Thing has zero-to-many HistoricalLocations. A HistoricalLocation has one-and-only-one Thing.
Datastream	One mandatory to many optional	A Thing MAY have zero-to-many Datastreams.

Example 1 – An example of a Thing entity:

```
{
  "@iot.id": 1,
  "@iot.selfLink": "http://example.org/v1.0/Things(1)",
  "Locations@iot.navigationLink": "Things(1)/Locations",
  "Datastreams@iot.navigationLink": "Things(1)/Datastreams",
  "HistoricalLocations@iot.navigationLink": "Things(1)/HistoricalLocations",
  "name": "Oven",
  "description": "This thing is an oven.",
  "properties": {
    "owner": "Noah Liang",
    "color": "Black"
  }
}
```

6.2.2 Location

The Location entity locates the Thing or the Things it is associated with. A Thing's Location entity is defined as the last known location of the Thing.

A Thing's Location may be identical to the Thing's Observations' FeatureOfInterest. In the context of the IoT, the principle location of interest is usually associated with the location of the Thing, especially for in-situ sensing applications. For example, the location of interest of a WiFi-connected thermostat should be the building or the room in which the smart thermostat is located. The FeatureOfInterest of the Observations made by the thermostat (e.g., room temperature readings) should also be the building or the room. In this case, the content of the smart thermostat's location should be the same as the content of the temperature readings' feature of interest.

However, the ultimate location of interest of a Thing is not always the location of the Thing (e.g., in the case of remote sensing). In those use cases, the content of a Thing's Location is different from the content of the FeatureOfInterest of the Thing's Observations. Section 7.1.4 of [OGC/ISO 19156] provides a detailed explanation of observation location.

Table 4 lists properties of a Location entity, Table 5 lists the direct relations between a Location entity and other entity types and Table 6 lists of some code values used for identifying types for the encodingType of the Location and FeatureOfInterest entity.

Table 4 – Properties of a Location entity

Name	Definition	Data type	Multiplicity and use
name	A property provides a label for Location entity, commonly a descriptive name.	CharacterString	One (mandatory)
description	The description about the Location.	CharacterString	One (mandatory)
encodingType	The encoding type of the Location property. Its value is one of the ValueCode enumeration.	ValueCode	One (mandatory)
location	The location type is defined by encodingType.	Any (i.e., the type is depending on the value of the encodingType)	One (mandatory)

Table 5 – Direct relations between a Location entity and other entity types

Entity type	Relation	Description
Thing	Many optional to many optional	Multiple Things MAY locate at the same Location. A Thing MAY not have a Location.
HistoricalLocation	Many mandatory to many optional	A Location can have zero-to-many HistoricalLocations. One HistoricalLocation SHALL have one or many Locations.

Example 2 – An example of a Location entity:

```
{
  "@iot.id": 1,
  "@iot.selfLink": "http://example.org/v1.0/Locations(1)",
  "Things@iot.navigationLink": "Locations(1)/Things",
  "HistoricalLocations@iot.navigationLink": "Locations(1)/HistoricalLocations",
  "encodingType": "application/vnd.geo+json",
  "name": "CCIT",
  "description": "Calgary\n Center for Innovative Technologies",
  "location": {
    "type": "Feature",
    "geometry": {
      "type": "Point",
      "coordinates": [
        -114.06,
        51.05
      ]
    }
  }
}
```

Table 6 – Code values used for identifying types for the encodingType of the Location and FeatureOfInterest entity

Location encodingType	ValueCode Value
GeoJSON ¹	application/vnd.geo+json

A thing can be geo-referenced in different spaces. For example, for some applications it is more suitable to use a topological space model (e.g., IndoorGML) to describe an indoor things' location rather than using a geometric space model (e.g., GeoJSON). Currently GeoJSON is the only Location encodingType of the SensorThings API. In the future it is expected to extend SensorThings API's

¹ GeoJSON: Geospatial data interchange format based on JavaScript Object Notation, defined in [IETF RF 7946].

capabilities by adding additional `encodingType` to the code values listed in the above table. For example, one potential new `Location` `encodingType` can be a JSON encoding for IndoorGML.

6.2.3 HistoricalLocation

A Thing's `HistoricalLocation` entity set provides the times of the current (i.e., last known) and previous locations of the Thing.

The `HistoricalLocation` can also be created, updated and deleted. One use case is to migrate historical observation data from an existing observation data management system to a `SensorThings` API system.

Table 7 lists properties of a `HistoricalLocation` entity and Table 8 lists direct relations between an `HistoricalLocation` entity and other entity types.

Table 7 – Properties of a `HistoricalLocation` entity

Name	Definition	Data type	Multiplicity and use
time	The time when the Thing is known at the Location.	TM_Instant (ISO-8601 Time String)	One (mandatory)

Table 8 – Direct relations between an `HistoricalLocation` entity and other entity types

Entity type	Relation	Description
Location	Many optional to many mandatory	A Location can have zero-to-many <code>HistoricalLocations</code> . One <code>HistoricalLocation</code> SHALL have one or many Locations.
Thing	Many optional to one mandatory	A <code>HistoricalLocation</code> has one-and-only-one Thing. One Thing MAY have zero-to-many <code>HistoricalLocations</code> .

Example 3 – An example of a `HistoricalLocations` entity set (e.g., `Things(1)/HistoricalLocations`):

```
{
  "value": [
    {
      "@iot.id": 1,
      "@iot.selfLink": "http://example.org/v1.0/HistoricalLocations(1)",
      "Locations@iot.navigationLink": "HistoricalLocations(1)/Locations",
      "Thing@iot.navigationLink": "HistoricalLocations(1)/Thing",
      "time": "2015-01-25T12:00:00-07:00"
    },
    {
      "@iot.id": 2,
      "@iot.selfLink": "http://example.org/v1.0/HistoricalLocations(2)",
      "Locations@iot.navigationLink": "HistoricalLocations(2)/Locations",
      "Thing@iot.navigationLink": "HistoricalLocations(2)/Thing",
      "time": "2015-01-25T13:00:00-07:00"
    }
  ],
  "@iot.nextLink": "http://example.org/v1.0/Things(1)/HistoricalLocations?$skip=2&$top=2"
}
```

6.2.4 Datastream

A Datastream groups a collection of Observations measuring the same ObservedProperty and produced by the same Sensor.

Table 9 lists properties of a Datastream entity and Table 10 lists direct relations between a Datastream entity and other entity types.

Table 9 – Properties of a Datastream entity

Name	Definition	Data type	Multiplicity and use
name	A property provides a label for Datastream entity, commonly a descriptive name.	CharacterString	One (mandatory)
description	The description of the Datastream entity.	CharacterString	One (mandatory)
unitOfMeasurement	A JSON Object containing three key-value pairs. The name property presents the full name of the unitOfMeasurement; the symbol property shows the textual form of the unit symbol; and the definition contains the URI defining the unitOfMeasurement. The values of these properties SHOULD follow the Unified Code for Unit of Measure (UCUM).	JSON Object	One (mandatory) NOTE – When a Datastream does not have a unit of measurement (e.g., an OM_TruthObservation type), the corresponding unitOfMeasurement properties SHALL have null values.
observationType	The type of Observation (with unique result type), which is used by the service to encode observations.	ValueCode see Table 14.	One (mandatory)
observedArea	The spatial bounding box of the spatial extent of all FeaturesOfInterest that belong to the Observations associated with this Datastream.	GM_Envelope (GeoJSON Polygon)	Zero-to-one (optional)
phenomenonTime	The temporal interval of the phenomenon times of all observations belonging to this Datastream.	TM_Period (ISO 8601 Time Interval)	Zero-to-one (optional)

Table 9 – Properties of a Datastream entity

Name	Definition	Data type	Multiplicity and use
resultTime	The temporal interval of the result times of all observations belonging to this Datastream.	TM_Period (ISO 8601 Time Interval)	Zero-to-one (optional)

Table 10 – Direct relation between a Datastream entity and other entity types

Entity type	Relation	Description
Thing	Many optional to one mandatory	A Thing has zero-to-many Datastreams. A Datastream entity SHALL only link to a Thing as a collection of Observations.
Sensor	Many optional to one mandatory	The Observations in a Datastream are performed by one-and-only-one Sensor. One Sensor MAY produce zero-to-many Observations in different Datastreams.
ObservedProperty	Many optional to one mandatory	The Observations of a Datastream SHALL observe the same ObservedProperty. The Observations of different Datastreams MAY observe the same ObservedProperty.
Observation	One mandatory to many optional	A Datastream has zero-to-many Observations. One Observation SHALL occur in one-and-only-one Datastream.

Example 4 – A Datastream entity example:

```

{
  "@iot.id": 1,
  "@iot.selfLink": "http://example.org/v1.0/Datastreams(1)",
  "Thing@iot.navigationLink": "HistoricalLocations(1)/Thing",
  "Sensor@iot.navigationLink": "Datastreams(1)/Sensor",
  "ObservedProperty@iot.navigationLink": "Datastreams(1)/ObservedProperty",
  "Observations@iot.navigationLink": "Datastreams(1)/Observations",
  "name": "oven temperature",
  "description": "This is a datastream measuring the air temperature in an oven.",
  "unitOfMeasurement": {
    "name": "degree Celsius",
    "symbol": "°C",
    "definition": "http://unitsofmeasure.org/ucum.html#para-30"
  },
  "observationType": "http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement",
  "observedArea": {
    "type": "Polygon",
    "coordinates":
      [[[100,0],[101,0],[101,1],[100,1],[100,0]]]
  },
  "phenomenonTime": "2014-03-01T13:00:00Z/2015-05-11T15:30:00Z",
  "resultTime": "2014-03-01T13:00:00Z/2015-05-11T15:30:00Z"
}

```

The observationType defines the result types for specialized observations [OGC/ISO 19156]. Table 11 shows some of the valueCodes that maps the Unified modelling language (UML) classes in O&M v2.0 [OGC/ISO 19156] to observationType names and observation result types.

Table 11 – List of some code values used for identifying types defined in the O&M conceptual model [OGC/ISO 19156]

O&M 2.0	Code value (observationType names)	Content of result
OM_CategoryObservation	http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_CategoryObservation	URI
OM_CountObservation	http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_CountObservation	integer
OM_Measurement	http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement	double
OM_Observation	http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Observation	Any
OM_TruthObservation	http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_TruthObservation	boolean

6.2.5 Sensor

A Sensor is an instrument that observes a property or phenomenon with the goal of producing an estimate of the value of the property².

Table 12 lists properties of a Sensor entity, Table 13 lists direct relations between a Sensor entity and other entity types and Table 14 lists some code values used for identifying types for the encodingType of the Sensor entity.

Table 12 – Properties of a Sensor entity

Name	Definition	Data type	Multiplicity and use
name	A property provides a label for Sensor entity, commonly a descriptive name.	CharacterString	One (mandatory)
description	The description of the Sensor entity.	CharacterString	One (mandatory)

² In some cases, the Sensor in this data model can also be seen as the Procedure (method, algorithm, or instrument) defined in [OGC/ISO 19156].