

# Open Geospatial Consortium

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: <yyyy-mm-dd>

External identifier of this OGC<sup>®</sup> document: <[http://www.opengis.net/doc/IS/iot\\_tasking/1.0](http://www.opengis.net/doc/IS/iot_tasking/1.0)>

Internal reference number of this OGC<sup>®</sup> document: 17-079

Version: 0.1.2

Category: OGC<sup>®</sup> Implementation Specification

Editor: Steve Liang (University of Calgary/SensorUp)  
Co-Editor: Tania Khalafbeigi (University of Calgary/SensorUp)

## OGC SensorThings API Part 2 – Tasking Core

### Copyright notice

Copyright © 2017 Open Geospatial Consortium  
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

### Warning

This document is not an OGC Standard. This document is distributed for review and comment. This document is subject to change without notice and may not be referred to as an OGC Standard.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: OGC<sup>®</sup> Standard  
Document subtype: Implementation Standard  
Document stage: Draft  
Document language: English

## License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

**Content**

- 1 Scope..... 6
- 2 Conformance..... 6
- 3 References..... 7
- 4 Terms and Definitions..... 7
- 5 Conventions ..... 8
  - 5.1 Identifiers ..... 8
- 6 Symbols (and abbreviated terms)..... 8
- 7 The SensorThings API Tasking Entities..... 9
  - 7.1 The Tasking Entities ..... 9
    - 7.1.1 TaskingCapability..... 10
    - 7.1.2 Task..... 12
    - 7.1.3 Actuator ..... 13
- 8 SensorThings Tasking Create ..... 14
  - 8.1 Link to existing entities when creating a Task entity..... 15
- 9 SensorThings Tasking MQTT Extension ..... 16
  - 9.1 Create a SensorThings Task with MQTT Publish ..... 16
    - 9.1.1 Link to existing entities when creating a Task entity..... 17
  - 9.2 Receive updates with MQTT Subscribe ..... 17
- Annex A : Conformance Class Abstract Test Suite (Normative)..... 19
- Annex B: Revision history..... 22
- Annex C: Bibliography ..... 23

## **i. Abstract**

The OGC SensorThings API provides an open, geospatial-enabled and unified way to interconnect the Internet of Things (IoT) devices, data, and applications over the Web. At a high level the OGC SensorThings API provides two main functionalities and each function is handled by a part. The two parts are the Sensing part and the Tasking part. The Sensing part provides a standard way to manage and retrieve observations and metadata from heterogeneous IoT sensor systems. The Tasking part provides a standard way for parameterizing - also called tasking - of task-able IoT devices, such as sensors, actuators, drones or even satellites. This document is the core of the SensorThings API part 2 – Tasking.

## **ii. Keywords**

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, ogcdoc, ogc documents, iot, internet of things, sensor things, sensors, swe, sensor webs, sensor web enablement, sensor networks

## **iii. Preface**

The OGC SensorThings API provides an open, geospatial-enabled and unified way to interconnect the Internet of Things devices, data, and applications over the Web. The OGC SensorThings API is an open standard, and that means it is non-proprietary, platform-independent, and perpetual royalty-free. Although it is a new standard, it builds on a rich set of proven-working and widely-adopted open standards, such as the Web protocols and the OGC Sensor Web Enablement (SWE) standards, including the ISO/OGC Observation and Measurement data model [OGC 10-004r3 and ISO 19156:2011]. That also means the OGC SensorThings API is extensible and can be applied to not only simple but also complex use cases.

At a high level the OGC SensorThings API provides two main functionalities and each function is handled by a part. The two parts are the Part I - Sensing and the Part II - Tasking. The Sensing part provides a standard way to manage and retrieve observations and metadata from heterogeneous IoT sensor systems. The Tasking part provides a standard way for parameterizing - also called tasking - of task-able IoT devices, such as sensors or actuators.

*Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.*

*Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.*

## **iv. Submitting organizations**

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

- a. University of Calgary, Canada
- b. SensorUp Inc., Canada
- c. CGI Federal, Canada
- d. Fraunhofer, Germany
- e. Compusult, Canada
- f. GeoConnections, Canada

g. Noblis, USA

**v. Submitters**

All questions regarding this submission should be directed to the editor or the submitters:

<b>Name</b>	<b>Company</b>
Steve Liang	University of Calgary, Canada / SensorUp Inc.
Tania Khalafbeigi	University of Calgary, Canada / SensorUp Inc.
Kan Luo	University of Calgary
Brian Miles	CGI Federal
Hylke van der Schaaf	Fraunhofer
Chris Clark	Compusult
Marcus Alzona	Noblis

# 1 Scope

The OGC SensorThings API provides an open standard-based and geospatial-enabled framework to interconnect the Internet of Things devices, data, and applications over the Web. The Tasking part provides a standard way for parameterizing - also called tasking - of task-able IoT devices, such as sensors, actuators, drones or even satellites. This document is the core of the SensorThings API part 2 – Tasking Core.

# 2 Conformance

Conformance with this standard shall be checked using all the relevant tests specified in Annex A (normative) of this document. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site<sup>1</sup>.

All requirements-classes and conformance-classes described in this document are owned by the standard(s) identified.

The following table list the requirements classes defined by this standard.

NOTE: The smaller blue text in the following table is the path fragment that appended to the following URI: [http://www.opengis.net/spec/iot\\_tasking/1.0/](http://www.opengis.net/spec/iot_tasking/1.0/), and it provides the URI that can be used to unambiguously identify the requirement and the conformance class.

Requirements class id	Requirements	Description
req/tasking-capability	<ul style="list-style-type: none"><li>req/tasking-capability/properties</li><li>req/tasking-capability/relations</li></ul>	TakingCapability entity
req/task	<ul style="list-style-type: none"><li>req/task/properties</li><li>req/task/relations</li></ul>	Task entity
req/actuator	<ul style="list-style-type: none"><li>req/actuator/properties</li><li>req/actuator/relations</li></ul>	Actuator entity
req/create-tasks	<ul style="list-style-type: none"><li>req/create-tasks/task-creation</li></ul>	creating Task entity
req/create-tasks-via-mqtt	<ul style="list-style-type: none"><li>req/create-tasks-via-mqtt/task-creation</li></ul>	creating Tasks through MQTT
req/receive-updates-via-mqtt	<ul style="list-style-type: none"><li>req/receive-updates-via-mqtt/receive-updated</li></ul>	Receiving Tasks updates through MQTT

---

<sup>1</sup> [www.opengeospatial.org/cite](http://www.opengeospatial.org/cite)

### 3 References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

OGC® SWE Common Data Model, <a href="http://www.opengis.net/doc/IS/SWE/2.0">http://www.opengis.net/doc/IS/SWE/2.0</a>
OGC® SensorThings API Part 1 – Sensing, <a href="http://www.opengis.net/doc/IS/iot_sensing/1.0">http://www.opengis.net/doc/IS/iot_sensing/1.0</a>
OGC® JSON Encoding Rules - SWE Common / SensorML, <a href="http://www.opengis.net/doc/BP/SWE-JSON/1.0">http://www.opengis.net/doc/BP/SWE-JSON/1.0</a>

### 4 Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r8], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

For the purposes of this document, the following additional terms and definitions apply.

#### Collection

Sets of Resources, which can be retrieved in whole or in part. [[RFC5023](#)]

#### Entity

Entities are instances of entity types. [OASIS OData Version 4.0 Part 1: Protocol Plus Errata 02]

*Note: Thing, Actuator, Task are some example entity types of the OGC SensorThings API.*

#### Entity sets

Entity sets are named collections of entities (e.g. Tasks is an entity set containing Task entities). An entity's key uniquely identifies the entity within an entity set. Entity sets provide entry points into an OGC SensorThings API service. [OASIS OData Version 4.0 Part 1: Protocol Plus Errata 02]

#### (Internet of) Thing

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.2060]

#### Measurement

A set of operations having the object of determining the value of a quantity [OGC 10-004r3 / ISO 19156:2011]

#### Observation

Act of measuring or otherwise determining the value of a property [OGC 10-004r3 / ISO 19156:2011]

### **Observation Result**

Estimate of the value of a property determined through a known observation procedure [OGC 10-004r3 / ISO 19156:2011]

### **Resource**

A network-accessible data object or service identified by an URI, as defined in [RFC 2616]

### **REST**

The Representational State Transfer (REST) style is an abstraction of the architectural elements within a distributed hypermedia system. REST focuses on the roles of components, the constraints upon their interaction with other components, and their interpretation of significant data elements. It encompasses the fundamental constraints upon components, connectors, and data that define the basis of the Web architecture, and thus the essence of its behavior as a network-based application. An API that conforms to the REST architectural principles/constraints is called a RESTful API.

### **Sensor**

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. [OGC 12-000]

### **Actuator**

A type of transducer that converts a signal to some real-world action or phenomenon. [OGC 12-000]

## **5 Conventions**

This section provides details and examples for any conventions used in the document. Examples of conventions are symbols, abbreviations, use of XML schema, or special notes regarding how to read the document.

### **5.1 Identifiers**

The normative provisions in this specification are denoted by the URI

[http://www.opengis.net/spec/iot\\_tasking/1.0/](http://www.opengis.net/spec/iot_tasking/1.0/)

All requirements and conformance tests that appear in this document are denoted by partial URIs which are relative to this base.

## **6 Symbols (and abbreviated terms)**

API	Application Programming Interface
CS-W	Catalog Service Web
CRUD	Create, Read, Update, and Delete



GML	Geography Markup Language
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
MQTT	Message Queuing Telemetry Transport
OData	the Open Data Protocol
OGC	Open Geospatial Consortium
OWS	OGC Web Services
O&M	Observations and Measurements
REST	REpresentational State Transfer
SensorML	Sensor Model Language
SOS	Sensor Observation Service
SPS	Sensor Planning Service
STA	SensorThings API
SWE	Sensor Web Enablement
UCUM	Unified Code for Units of Measure
UML	Unified Modeling Language
WoT	Web of Things
XML	eXtensible Markup Language

## 7 The SensorThings API Tasking Entities

### 7.1 The Tasking Entities

The SensorThings API Tasking part's Entities are depicted in Figure 1.

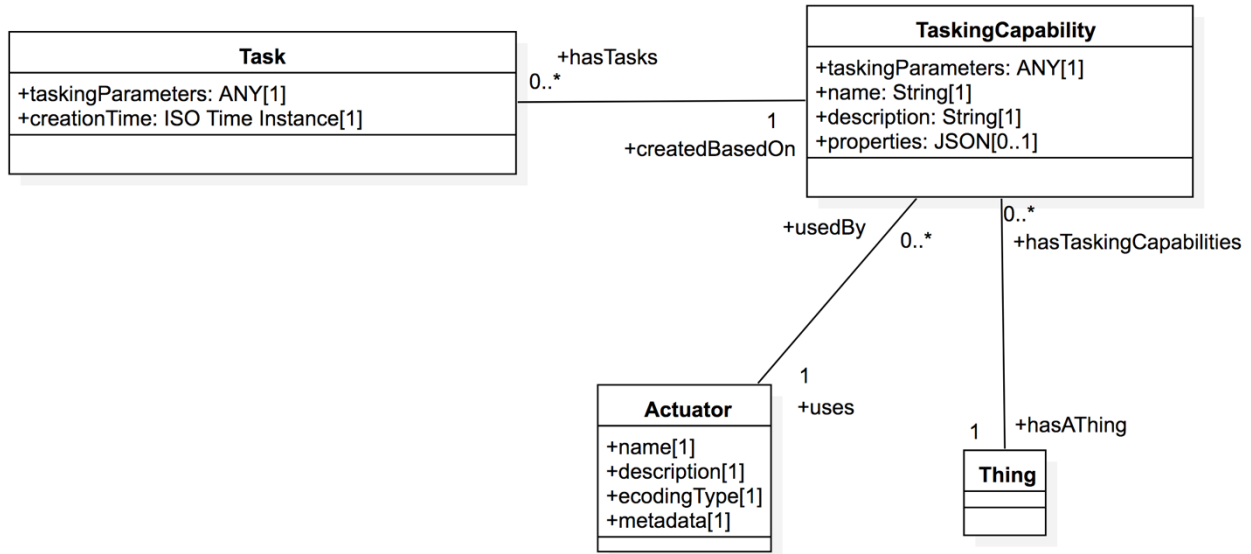


Figure 1 Tasking Entities

In this section, we explain the properties in each entity type and the direct relation to the other entity types. In addition, for each entity type, we show an example of the associated JSON encoding.

### 7.1.1 TaskingCapability

Requirements Class	
<a href="http://www.opengis.net/spec/iot_tasking/1.0/req/tasking-capability">http://www.opengis.net/spec/iot_tasking/1.0/req/tasking-capability</a>	
Target type	Web Service
Dependency	<a href="http://www.opengis.net/spec/iot_sensing/1.0/req/entity-control-information/common-control-information">http://www.opengis.net/spec/iot_sensing/1.0/req/entity-control-information/common-control-information</a>
Dependency	<a href="http://www.opengis.net/doc/BP/SWE-JSON/1.0/req">http://www.opengis.net/doc/BP/SWE-JSON/1.0/req</a>
Dependency	<a href="http://www.opengis.net/doc/IS/SWE/2.0">http://www.opengis.net/doc/IS/SWE/2.0</a>
Requirement	<a href="http://www.opengis.net/spec/iot_tasking/1.0/req/tasking-capability/properties">http://www.opengis.net/spec/iot_tasking/1.0/req/tasking-capability/properties</a> <i>Each TaskingCapability entity SHALL have the mandatory properties and MAY have the optional properties listed in Table 1.</i>
Requirement	<a href="http://www.opengis.net/spec/iot_tasking/1.0/req/tasking-capability/relations">http://www.opengis.net/spec/iot_tasking/1.0/req/tasking-capability/relations</a> <i>Each TaskingCapability entity SHALL have the direct relation between a TaskingCapability entity and other entity types listed in Table 2.</i>

Table 1 Properties of a TaskingCapability

Name	Definition	Data type	Multiplicity and use
<b>name</b>	A property provides a label for the entity, commonly a descriptive name.	CharacterString	One (mandatory)
<b>description</b>	This is a short description of the corresponding entity.	CharacterString	One (mandatory)

<b>taskingParameters</b>	The <code>taskingParameters</code> property describes optional and mandatory tasking parameters. Clients use the definition to provide corresponding tasking parameter values. To ensure common understanding between client and server, a common exchange protocol is used to express both descriptions and tasking parameter values.  SensorThings uses the JSON encoding defined in OGC 17-011 to define <code>taskingParameters</code> .	SWE Common JSON Object	One (mandatory)
<b>properties</b>	A JSON Object containing user-annotated properties as key-value pairs.	JSON Object	Zero-to-one

**Table 2 Direct relation between a `TaskCapability` and other entity types**

Entity Type	Relation	Description
<b>Task</b>	One mandatory to many optional	A <code>TaskingCapability</code> has zero-to-many <code>Tasks</code> . A <code>Task</code> has one-and-only-one <code>TaskingCapability</code> .
<b>Thing</b>	Many optional to one mandatory	A <code>TaskingCapability</code> has one-and-only-one <code>Thing</code> . A <code>Thing</code> has zero-to-many <code>TaskingCapabilities</code> .
<b>Actuator</b>	Many optional to one mandatory	A <code>TaskingCapability</code> has one-and-only-one <code>Actuator</code> . An <code>Actuator</code> has zero-to-many <code>TaskingCapabilities</code> .

**Example 1 an example of a `TaskCapability` entity**

```
{
  "@iot.id": 1,
  "@iot.selfLink": "http://example.org/v1.0/TaskingCapabilities(1)",
  "Thing@iot.navigationLink": "TaskingCapabilities(1)/Thing",
  "Actuator@iot.navigationLink": "TaskingCapabilities(1)/Actuator",
  "Tasks@iot.navigationLink": "TaskingCapabilities(1)/Tasks",
  "name": "Control Light",
  "description": "Turn the light on and off, as well as specifying light color. ",
  "taskingParameters": {
    "type": "DataRecord",
    "field": [
      {
        "name": "On/Off",
        "description": "Specifies to turn the light On or Off",
        "type": "Category",
        "AllowedTokens": {
          "value": [
            "on", "off"
          ]
        }
      }
    ]
  },
  {
    "name": "Color",
    "description": "Specifies the light color in RGB HEX format. Example: #FF11A0",
    "type": "Text",
    "AllowedTokens": {
      "pattern": "^#([A-Fa-f0-9]{6}|[A-Fa-f0-9]{3})$"
    }
  }
}
```

```

}
}
]
}
}
}

```

### 7.1.2 Task

Requirements Class	
<a href="http://www.opengis.net/spec/iot_tasking/1.0/req/task">http://www.opengis.net/spec/iot_tasking/1.0/req/task</a>	
Target type	Web Service
Dependency	<a href="http://www.opengis.net/spec/iot_sensing/1.0/req/entity-control-information/common-control-information">http://www.opengis.net/spec/iot_sensing/1.0/req/entity-control-information/common-control-information</a>
Dependency	<a href="http://www.opengis.net/doc/BP/SWE-JSON/1.0/req">http://www.opengis.net/doc/BP/SWE-JSON/1.0/req</a>
Dependency	<a href="http://www.opengis.net/doc/IS/SWE/2.0">http://www.opengis.net/doc/IS/SWE/2.0</a>
Requirement	<p><a href="http://www.opengis.net/spec/iot_tasking/1.0/req/task/properties">http://www.opengis.net/spec/iot_tasking/1.0/req/task/properties</a></p> <p><i>Each Task entity SHALL have the mandatory properties and MAY have the optional properties listed in Table 3.</i></p>
Requirement	<p><a href="http://www.opengis.net/spec/iot_tasking/1.0/req/task/relations">http://www.opengis.net/spec/iot_tasking/1.0/req/task/relations</a></p> <p><i>Each Task entity SHALL have the direct relation between a Task entity and other entity types listed in Table 4</i></p>

**Table 3 Properties of a Task**

Name	Definition	Data type	Multiplicity and use
taskingParameters	<p>The taskingParameters property describes values for optional and mandatory tasking parameters. Clients use the definition to provide corresponding tasking parameter values. To ensure common understanding between client and server, a common exchange protocol is used to express both descriptions and tasking parameter values.</p> <p>SensorThings uses the JSON encoding defined in OGC 17-011 to define taskingParameters. Each field of the taskingParameters MUST have a name and a value. name specifies the parameter's name as described in TaskingCapability's taskingParameters and value is the value of that parameter for this Task.</p>	SWE Common JSON Object	One (mandatory)
creationTime	The time when the Task is created. This time SHALL only be added automatically by the service.	TM_Instant (ISO-8601 Time String)	One (mandatory)

**Table 4 Direct relation between a Task and other entity types**

Entity Type	Relation	Description
TaskingCapability	Many optional to one mandatory	A Task has one-and-only-one TaskingCapability. A TaskingCapability has zero-to-many Tasks.

### Example 2 an example of a Task entity

```
{
  "@iot.id": 2,
  "@iot.selfLink": "http://example.org/v1.0/Tasks(1)",
  "TaskingCapability@iot.navigationLink": "Tasks(2)/TaskingCapability",
  "creationTime": "2017-01-01T00:00:00.000Z",
  "taskingParameters": {
    "type": "DataRecord",
    "field": [
      {
        "name": "On/Off",
        "value": "on"
      },
      {
        "name": "Color",
        "value": "#FF0000"
      }
    ]
  }
}
```

### 7.1.3 Actuator

Requirements Class	
<a href="http://www.opengis.net/spec/iot_tasking/1.0/req/actuator">http://www.opengis.net/spec/iot_tasking/1.0/req/actuator</a>	
Target type	Web Service
Dependency	<a href="http://www.opengis.net/spec/iot_sensing/1.0/req/entity-control-information/common-control-information">http://www.opengis.net/spec/iot_sensing/1.0/req/entity-control-information/common-control-information</a>
Requirement	<a href="http://www.opengis.net/spec/iot_tasking/1.0/req/actuator/properties">http://www.opengis.net/spec/iot_tasking/1.0/req/actuator/properties</a> <i>Each Actuator entity SHALL have the mandatory properties and MAY have the optional properties listed in Table 5.</i>
Requirement	<a href="http://www.opengis.net/spec/iot_tasking/1.0/req/actuator/relations">http://www.opengis.net/spec/iot_tasking/1.0/req/actuator/relations</a> <i>Each Actuator entity SHALL have the direct relation between a Task entity and other entity types listed in Table 6.</i>

Table 5 Properties of an Actuator

Name	Definition	Data type	Multiplicity and use
name	A property provides a label for the entity, commonly a descriptive name.	CharacterString	One (mandatory)
description	The description of the Actuator entity	CharacterString	One (mandatory)
encodingType	The encoding type of the metadata property. Its value is one of the ValueCode enumeration (see for Table 7 for the available ValueCode).	Any (depending on the value of the encodingType)	One (mandatory)

metadata	The detailed description of the Actuator. The metadata type is defined by encodingType.	Any (depending on the value of the encodingType)	One (mandatory)
----------	---	--	-----------------

**Table 6 Direct relation between an Actuator and other entity types**

Entity Type	Relation	Description
<b>TaskingCapabilities</b>	One mandatory to many optional	An Actuator has zero-to-many TaskingCapabilities. A TaskingCapability has one-and-only-one Actuator.

**Table 7 List of some code values used for identifying types for the encodingType of the Actuator entity**

Actuator encodingType	ValueCode Value
PDF	application/pdf
SensorML	http://www.opengis.net/doc/IS/SensorML/2.0

The Actuator encodingType allows clients to know how to interpret metadata's value. Currently SensorThings API defines two common Actuator metadata encodingTypes. Most sensor manufacturers provide their sensor datasheets in a PDF format. As a result, PDF is a Sensor encodingType supported by SensorThings API. The second Sensor encodingType is SensorML.

### Example 3 an example of an Actuator entity

```
{
  "@iot.id": 3,
  "@iot.selfLink": "http://example.org/v1.0/Actuators(3)",
  "TaskingCapabilities@iot.navigationLink": "Actuators(3)/TaskingCapabilities",
  "name": "Linkit Smart 7688 Duo Board with LED",
  "description": "Linkit Smart 7688 Duo Board that has an LED which can be tasked as on/off with different color.",
  "encodingType": "application/pdf",
  "metadata": "http://example.org/linkit_7688.pdf"
}
```

## 8 SensorThings Tasking Create

The SensorThings API Tasking part SHALL support creating Task entities.

Requirements Class	
<a href="http://www.opengis.net/spec/iot_tasking/1.0/req/create-tasks">http://www.opengis.net/spec/iot_tasking/1.0/req/create-tasks</a>	
Target type	Web Service
Dependency	<a href="http://docs.oasis-open.org/odata/odata/v4.0/errata02/os/complete/part1-protocol/odata-v4.0-errata02-os-part1-protocol-complete.html#_Toc406398326">http://docs.oasis-open.org/odata/odata/v4.0/errata02/os/complete/part1-protocol/odata-v4.0-errata02-os-part1-protocol-complete.html#_Toc406398326</a>
Requirement	<a href="http://www.opengis.net/spec/iot_tasking/1.0/req/create-tasks/task-creation">http://www.opengis.net/spec/iot_tasking/1.0/req/create-tasks/task-creation</a>  <i>To create a Task in a collection, the client SHALL send a HTTP POST request to Task collection's URL. The POST body SHALL contain a single valid Task representation.</i>

	<p>When clients create <b>Tasks</b> in a <b>SensorThings</b> service, they <b>SHALL</b> follow the integrity constraints listed in Table 8.</p> <p>When a <b>SensorThings</b> service receives a create <b>Task</b> request, the service <b>SHALL</b> set the <b>createdTime</b> property of the entity to current server time.</p> <p><b>SensorThings</b> API services <b>SHALL</b> support linking new <b>Task</b> entities to existing entities upon creation. To create a new <b>Task</b> with links to existing entities, the client <b>SHALL</b> include the unique identifiers of the related entities associated with the corresponding navigation properties in the request body.</p>
--	--

**Table 8 Integrity constraints when creating a **Task** entity**

Scenario	Integrity Constraints
Create a <b>Task</b> entity	<b>SHALL</b> link to a <b>TakingCapability</b> entity

## 8.1 Link to existing entities when creating a **Task** entity

**HTTP Method:** POST

**URI Pattern:** SERVICE\_ROOT\_URI/Tasks

**Header:** Content-Type: application/json

**Message Body:** A single valid **Task** entity representation.

**Example 4** create a **Task** entity, which links to an existing **Actuator** entity (whose id is 1), an existing **TakingCapability** entity (whose id is 2).

```

POST /v1.0/Tasks HTTP/1.1
Host: example.org/
Content-Type: application/json

{
  "taskingParameters": {
    "type": "DataRecord",
    "field": [
      {
        "name" : "On/Off",
        "value" : "on"
      },
      {
        "name" : "Color",
        "value" : "#FF0000"
      }
    ]
  },
  "TaskingCapability": {
    "@iot.id": 2
  }
}

```

## 9 SensorThings Tasking MQTT Extension

In addition to support HTTP protocol, a SensorThings service MAY support MQTT protocol to enhance the SensorThings service publish and subscribe capabilities. This section describes the SensorThings MQTT extension.

### 9.1 Create a SensorThings Task with MQTT Publish

Requirements Class	
<a href="http://www.opengis.net/spec/iot_tasking/1.0/req/create-tasks-via-mqtt">http://www.opengis.net/spec/iot_tasking/1.0/req/create-tasks-via-mqtt</a>	
Target type	Web Service
Dependency	<a href="http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html">http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html</a>
Requirement	<a href="http://www.opengis.net/spec/iot_tasking/1.0/req/create-tasks-via-mqtt/tasks-creation">http://www.opengis.net/spec/iot_tasking/1.0/req/create-tasks-via-mqtt/tasks-creation</a>
<i>To allow clients to create Tasks with MQTT Publish, a service SHALL support the creation of Tasks with MQTT as defined in Section 0.</i>	

SensorThings MQTT extension provides the capability of creating Task entity using MQTT protocol. To create a Task entity in MQTT, the client sends a MQTT Publish request to the SensorThings service and the MQTT topic is the Tasks resource path. The MQTT application message contains a single valid Task entity representation. Figure 2 contains the sequence diagram for creating Task using MQTT publish as well as MQTT sending notifications for Task creation.

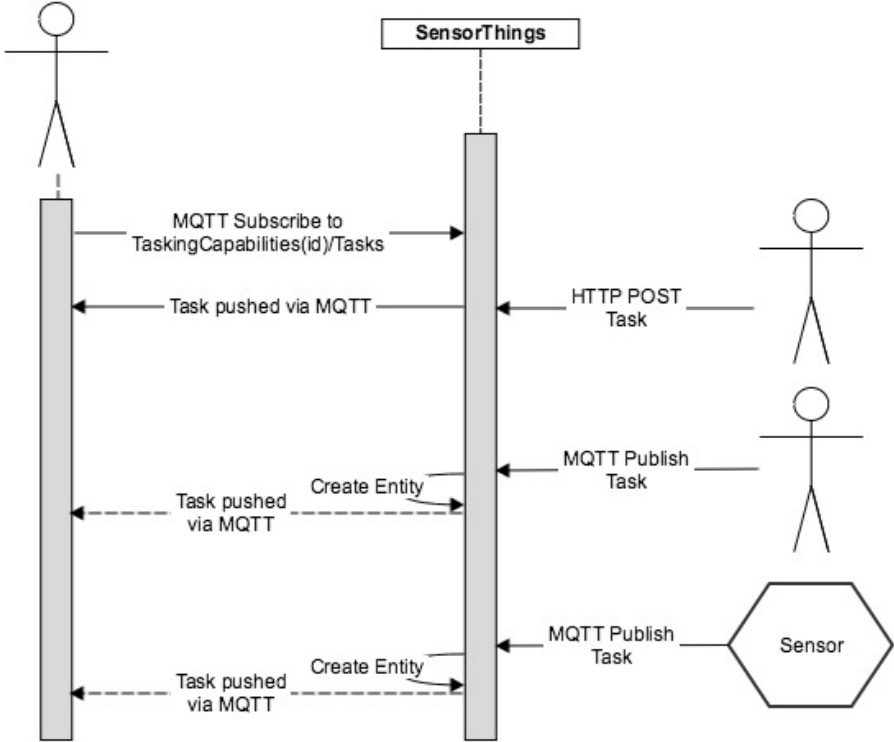


Figure 2 Creating Tasks using MQTT publish, and receive notifications for Tasks with MQTT



If the MQTT topic for the Task is a navigationLink from TaskingCapability, the new Task entity is automatically linked to that TaskingCapability respectively.

Similar to creating Tasks with HTTP POST, creating Tasks with MQTT Publish follow the integrity constraints for creating Task listed in Table 8.

**9.1.1 Link to existing entities when creating a Task entity**

To link to existing entities when creating a Task entity with MQTT, the conditions in Section 8 is applied.

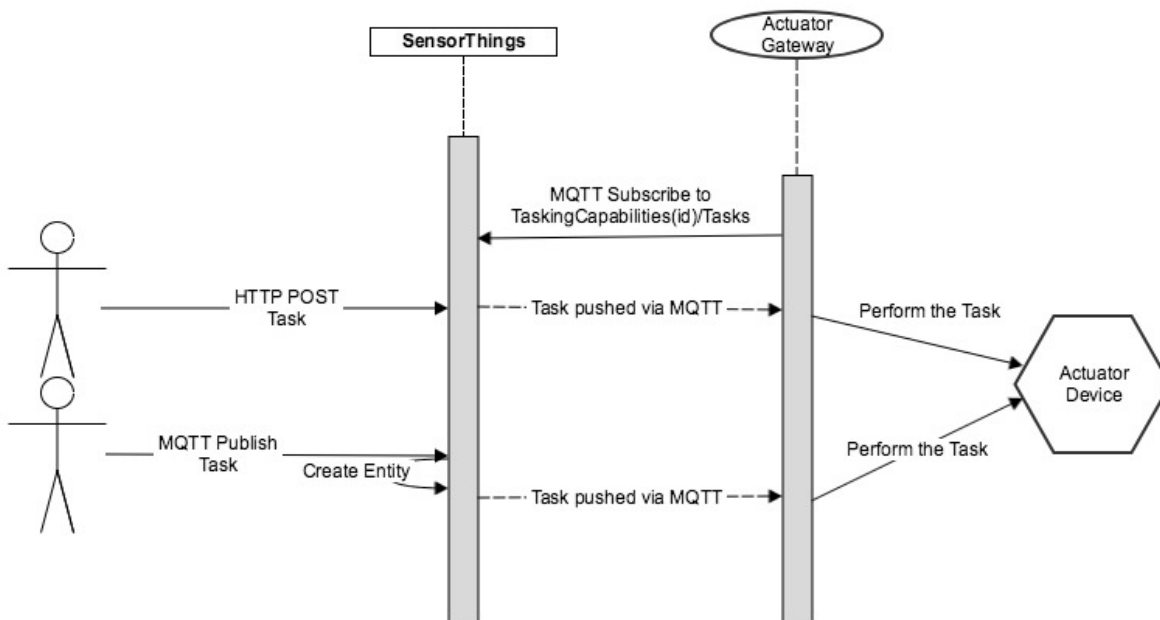
**9.2 Receive updates with MQTT Subscribe**

Requirements Class	
<a href="http://www.opengis.net/spec/iot_tasking/1.0/req/receive-updates-via-mqtt">http://www.opengis.net/spec/iot_tasking/1.0/req/receive-updates-via-mqtt</a>	
Target type	Web Service
Dependency	<a href="http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html">http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html</a>
Requirement	<p><a href="http://www.opengis.net/spec/iot_tasking/1.0/req/receive-updates-via-mqtt/receive-updates">http://www.opengis.net/spec/iot_tasking/1.0/req/receive-updates-via-mqtt/receive-updates</a></p> <p><i>To allow clients to receive notifications for the updates of SensorThings tasking entities with MQTT, a service SHALL support the receiving updates with MQTT Subscribe as defined in Section 9.2.</i></p>

To receive notifications from a SensorThings service when some tasking entities updated, a client can send a MQTT Subscribe request to the SensorThings service. SensorThings API defined the following four MQTT subscription use cases.

Receiving notifications from a SensorThings service follows the requirement [http://www.opengis.net/spec/iot\\_sensing/1.0/req/receive-updates-via-mqtt](http://www.opengis.net/spec/iot_sensing/1.0/req/receive-updates-via-mqtt) of Sensing part, but for entities is Tasking part.

When SensorThings MQTT extension is being used for controlling an Actuator, the actuator (gateway) can subscribe to Tasks and whenever it receives a Task over MQTT, it can perform it. Figure 3 shows a sequence diagram of this process.



**Figure 3 Actuator communication to SensorThings via MQTT**

## Annex A : Conformance Class Abstract Test Suite (Normative)

### A.1 SensorThings Tasking Core Tests

#### A.1.1 Conformance class: SensorThings API TaskingCapability Entity

Conformance class id: [http://www.opengis.net/spec/iot\\_tasking/1.0/conf/tasking-capability](http://www.opengis.net/spec/iot_tasking/1.0/conf/tasking-capability)

Test: TaskingCapability Entity

Requirements	<ul style="list-style-type: none"><li>• <a href="http://www.opengis.net/spec/iot_tasking/1.0/conf/tasking-capability/properties">http://www.opengis.net/spec/iot_tasking/1.0/conf/tasking-capability/properties</a></li><li>• <a href="http://www.opengis.net/spec/iot_tasking/1.0/conf/tasking-capability/relations">http://www.opengis.net/spec/iot_tasking/1.0/conf/tasking-capability/relations</a></li></ul>
Test Purpose	Check if each TaskingCapability entity has the mandatory properties and mandatory relations as defined in this specification.
Test Methods	<p>Inspect the full JSON object of the TaskingCapability entity sets (<i>i.e.</i>, without \$select) to identify, if each entity has the mandatory properties defined in the corresponding requirement.</p> <p>Inspect the full JSON object of each TaskingCapabilities entity set (<i>i.e.</i>, without using the \$select query option) to identify, if each entity has the mandatory relations (<i>i.e.</i>, @iot.navigationLink) defined in the corresponding requirement.</p>

#### A.1.2 Conformance class: SensorThings API Task Entity

Conformance class id: [http://www.opengis.net/spec/iot\\_tasking/1.0/conf/task](http://www.opengis.net/spec/iot_tasking/1.0/conf/task)

Test: Task Entity

Requirements	<ul style="list-style-type: none"><li>• <a href="http://www.opengis.net/spec/iot_tasking/1.0/conf/task/properties">http://www.opengis.net/spec/iot_tasking/1.0/conf/task/properties</a></li><li>• <a href="http://www.opengis.net/spec/iot_tasking/1.0/conf/task/relations">http://www.opengis.net/spec/iot_tasking/1.0/conf/task/relations</a></li></ul>
Test Purpose	Check if each Task entity has the mandatory properties and mandatory relations as defined in this specification.
Test Methods	<p>Inspect the full JSON object of the Tasks entity sets (<i>i.e.</i>, without \$select) to identify, if each entity has the mandatory properties defined in the corresponding requirement.</p> <p>Inspect the full JSON object of each Tasks entity set (<i>i.e.</i>, without using the \$select query option) to identify, if each entity has the mandatory relations (<i>i.e.</i>, @iot.navigationLink) defined in the corresponding requirement.</p>

### A.1.3 Conformance class: SensorThings API Actuator Entity

Conformance class id: [http://www.opengis.net/spec/iot\\_tasking/1.0/conf/actuator](http://www.opengis.net/spec/iot_tasking/1.0/conf/actuator)

Test: **TaskingCapability Entity**

Requirements	<ul style="list-style-type: none"><li>• <a href="http://www.opengis.net/spec/iot_tasking/1.0/conf/actuator/properties">http://www.opengis.net/spec/iot_tasking/1.0/conf/actuator/properties</a></li><li>• <a href="http://www.opengis.net/spec/iot_tasking/1.0/conf/actuator/relations">http://www.opengis.net/spec/iot_tasking/1.0/conf/actuator/relations</a></li></ul>
Test Purpose	Check if each <code>Actuator</code> entity has the mandatory properties and mandatory relations as defined in this specification.
Test Methods	<p>Inspect the full JSON object of the <code>Actuator</code> entity sets (<i>i.e.</i>, without <code>\$select</code>) to identify, if each entity has the mandatory properties defined in the corresponding requirement.</p> <p>Inspect the full JSON object of each <code>Actuator</code> entity set (<i>i.e.</i>, without using the <code>\$select</code> query option) to identify, if each entity has the mandatory relations (<i>i.e.</i>, <code>@iot.navigationLink</code>) defined in the corresponding requirement.</p>

### A.1.4. Conformance class: SensorThings API Create Task

Conformance class id: [http://www.opengis.net/spec/iot\\_tasking/1.0/conf/create-entity](http://www.opengis.net/spec/iot_tasking/1.0/conf/create-entity)

Dependencies:

- [http://www.opengis.net/spec/iot\\_tasking/1.0/conf/task](http://www.opengis.net/spec/iot_tasking/1.0/conf/task)
- [http://www.opengis.net/spec/iot\\_sensing/1.0/conf/resource-path](http://www.opengis.net/spec/iot_sensing/1.0/conf/resource-path)

Requirements	<ul style="list-style-type: none"><li>• <a href="http://www.opengis.net/spec/iot_sensing/1.0/req/create-update-delete/create-entity">http://www.opengis.net/spec/iot_sensing/1.0/req/create-update-delete/create-entity</a></li></ul>
Test purpose	Check if the service supports the creation of <code>Task</code> as defined in this specification.
Test method	<p>Create a <code>Task</code> entity instance by following the integrity constraints and link to the existing related entities with a single request, check if the <code>Task</code> entity instance is successfully created and the server responds as defined in this specification.</p> <p>Issue a <code>Task</code> entity creation request that does not follow the integrity constraints, check if the service fails the request without creating the <code>Task</code> and responds the appropriate HTTP status code.</p>

## A.2. SensorThings Tasking MQTT Extension

### A.2.1 Test: SensorThings API Task Creation via MQTT

Conformance class id: [http://www.opengis.net/spec/iot\\_tasking/1.0/conf/create-tasks-via-mqtt](http://www.opengis.net/spec/iot_tasking/1.0/conf/create-tasks-via-mqtt)

**Dependencies:**

- [http://www.opengis.net/spec/iot\\_tasking/1.0/conf/task](http://www.opengis.net/spec/iot_tasking/1.0/conf/task)

Requirements	<ul style="list-style-type: none"><li>• <a href="http://www.opengis.net/spec/iot_tasking/1.0/req/create-tasks-via-mqtt/task-creation">http://www.opengis.net/spec/iot_tasking/1.0/req/create-tasks-via-mqtt/task-creation</a></li></ul>
Test purpose	Check if the service supports the creation of <code>Task</code> entity via MQTT as defined in this Specification.
Test method	Create a <code>Task</code> entity instance containing binding information for navigation properties using <code>MQTT Publish</code> , check if the server responds as defined in this specification

## Annex B: Revision history

Date	Release	Author	Paragraph modified	Description
30/11/17	0.1.0	Steve Liang		Initial draft of SensorThings: Tasking
30/11/17	0.1.1	Steve Liang		Fixed some typos, corrected some inconsistencies between UML and text
08/01/18	0.1.2	Steve Liand		Fixed some inconsistencies between this draft and SensorThings Sensing part.

## Annex C: Bibliography

ITU-T Y.2060 Overview of the Internet of Things, 2012. Available Online: <https://www.itu.int/rec/T-REC-Y.2060-201206-1>

OGC and ISO 19156:2001, OGC and ISO 19156:2011(E), OGC Abstract Specification: Geographic information — Observations and Measurements. Available Online: [http://portal.opengeospatial.org/files/?artifact\\_id=41579](http://portal.opengeospatial.org/files/?artifact_id=41579)

OGC 12-000, OGC® SensorML: Model and XML Encoding Standard. Available Online: <http://www.opengeospatial.org/standards/sensorml>

RFC 5023, The Atom Publishing Protocol. Available Online: <https://www.ietf.org/rfc/rfc5023.txt>

RFC 6902, JavaScript Object Notation (JSON) Patch. Available Online: <https://www.ietf.org/rfc/rfc6902.txt>

OGC 08-094r1, OGC® SWE Common Data Model Encoding Standard. Available Online: <http://www.opengeospatial.org/standards/swecommon>

OGC 17-011r1, JSON Encoding Rules for SWE Common and SensorML. Available Online: [https://portal.opengeospatial.org/files/?artifact\\_id=75258&version=1](https://portal.opengeospatial.org/files/?artifact_id=75258&version=1)